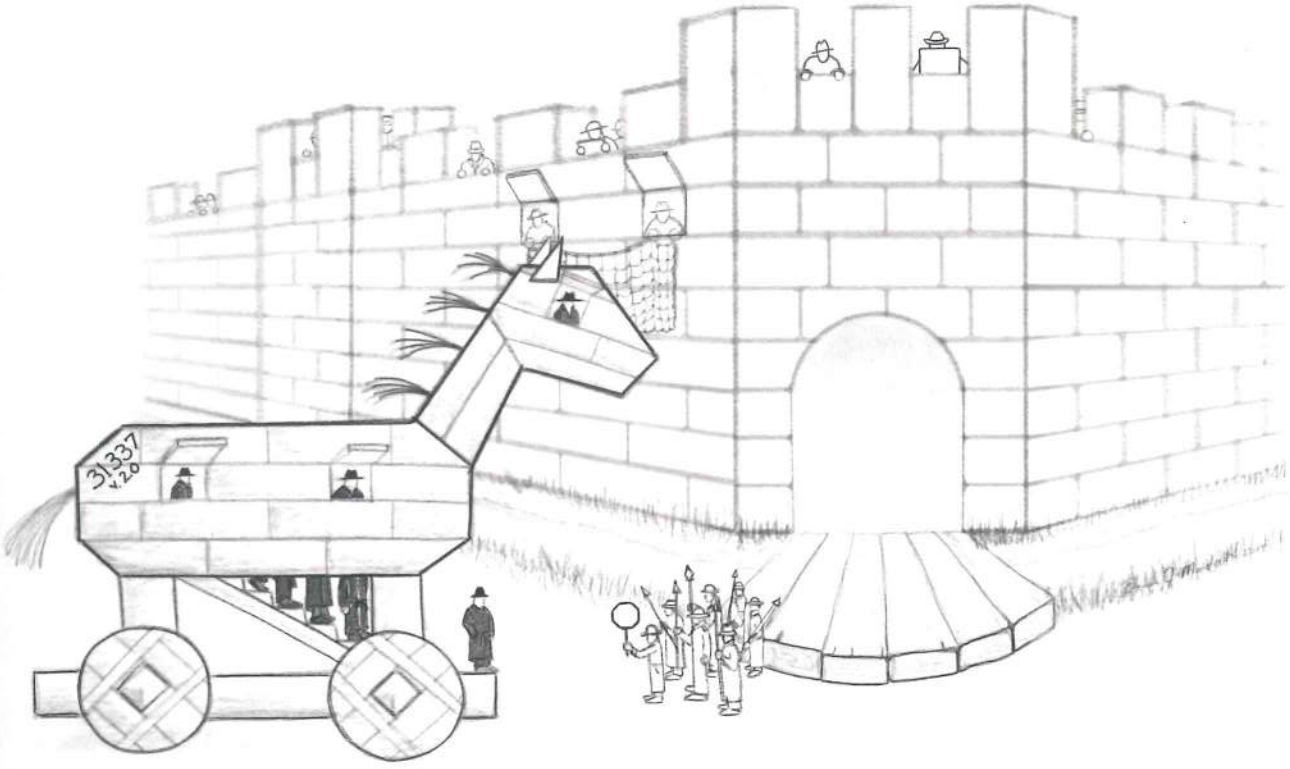
Blue Team Handbook:

SOC, SIEM, and Threat Hunting Use Cases

Notes from the Field (V1.02)

A condensed field guide for the Security Operations team.



Don Murdoch

Blue Team Handbook Vol 2: SOC, SIEM, and Threat Hunting Use Cases
Notes from the Field

A condensed field guide for the Security Operations team. (V1.02)

By Don Murdoch, GSE #99, MBA, MSISE
Illustrated by Bonnie Murdoch, BFA.

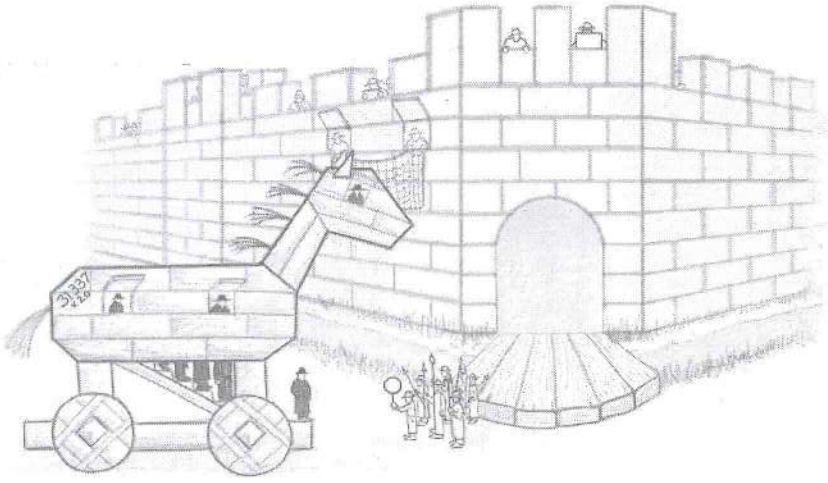


Table of Contents

Copyright © 2018 by Don Murdoch. All rights reserved. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher and author.

This book is available at special quantity discounts to use as premiums and sales promotions or for use in academic and corporate training programs. Please contact the author through www.blueteamhandbook.com.

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial Caps.

TERMS OF USE: This is a copyrighted work and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without prior consent from the author, secured via paper letter with a blue ink signature. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED "AS IS." The author does not warrant or guarantee that the functions contained in the work will meet your requirements, that its operation will be uninterrupted or error free, or that the work will qualify as an expert witness. The author shall not be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. Under no circumstances shall the author be liable for any indirect, incidental, special, punitive, consequential, or similar damages that result from the use of or inability to use the work. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

Version 1.00: Initial Printing September 2018.
Version 1.01: First Update: (v25) 11/10/18 - spelling, grammar updates.
Version 1.02: Second Update: (v27b) 3/23/19 - content updates.

ISBN-13: 978-1726273985 ISBN-10: 1726273989 V 1.0, 1.01
ISBN-13: 978-1091493896 V1.02

If you would like to get in contact with the author or illustrator, please use the contact form on the website at www.blueteamhandbook.com.

Art Notes: Art in the BTHb series is designed to be humorous and punny (yes, that IS a word!). We hope you enjoy it. If you would like to use the artist for your own books, presentations, or articles, please use the contact form on www.blueteamhandbook.com.

Table of Contents

List of Tables

Table of Figures

Preface

With the ever-advancing adversary, technology advancements, and a critical need for more skilled security operations practitioners, it is imperative for organizations to enhance their PDR cycle: Protection, Detection, and Response. This book attempts to answer that call by sharing experiences gained implementing five different SIEM technologies for more than a dozen organizations, running a MSSP division, and building several security operations centers.

Who this book is for: IT pros, cyber security pros, security operations staff, security consultants, SOC staff, SIEM designers and consultants, and line managers: those responsible for protecting information assets and teaching the next generation of security professionals.

About the Author: Don Murdoch, GSE #99, MBA, MSISE (GISF, GSEC, GCIH, GCIA, GPPA, GMON, GCFE, GCFA, GCPM, GPEN, GSNA, GPPA, GCWN, GCUX, TOGAF Enterprise Architect, SABSA Chartered Architect. CISSP, ISSAP), is a thirty-year veteran of information technology, with more than half a career devoted to information, computer, and network security. Don started his career with a boutique contracting firm located in eastern Virginia, writing COBOL and FoxPro code. For the next twelve years, Don took on role in a different aspect of IT as he grew his career: managing a small network, writing old school Perl/CGI software, developing international billing software for a startup ISP, and managing IT for an Application Service Provider right at the time of the Dot Com bubble. Don transitioned into information security where he started a DRP practice for an IT commercial spin off from a television production company. Things started cooking when Don entered his "digital combat training" phase in the "Wild, Wild, West" of academic computing for one of Virginia's largest institutions for higher learning. Don wrangled bots, tangled with well-equipped adversaries, and discovered what today are described as nation state grade attackers who were using the University network as a training ground. His University was the first in Virginia to implement a SIEM, user managed anti-spam technology, and an active countermeasures network based on Tom Listons' Labrea TarPit. After that experience, Don took on managing SIEM, conducting employee investigations, and security architecture for a Fortune 500 healthcare firm. That firm was acquired in 2012. His career took a security hiatus for a few years when he was an Enterprise Security Architect and then started running Infrastructure Strategy and Planning team for a Fortune 50 corporation. In 2016, an opportunity to develop an MSSP practice came up with a different boutique consulting firm. After two years, Don left to run the Cyber Range at Regent University, where he is today coaching the next generation of Cyber Defenders.

Don started working with the SANS Institute in 2002, taking courses, earning certifications, developing Stay Sharp courses during the mid-2000's, and currently teaches courses at the Community level. He earned the **GIAC Security Expert (GSE #99)** certification in 2014, as was later vetted as a Cyber Guardian: Blue Team in 2016 (#38).

About the Reviewers: Each of the technical content reviewers is a seasoned InfoSec pro with multiple certifications. The group represents a cross section of the community ranging from security operations and management, vendor product development and implementation, penetration testing, security engineering, and architecture. These reviewers are directly responsible for 42 pages of additional text from the original draft and collectively provided over 700 suggestions as a testament to their skills and passion for helping me to produce a well written book. I cannot thank them enough. In alphabetical order, the reviewers are:

- **Christopher Beiring:** Lead Security Operations analyst, network penetration tester, and all-around security engineer for a Virginia based consulting firm.
- **Chris Crowley,** Montance, LLC. Chris is a well-known information security consultant, a Principal Instructor with SANS, and a course author for two courses, including Management 517: Managing Security Operations.
- **John Hubbard:** John is a SANS Instructor and author, a SOC Lead for a large pharmaceutical company, and an all-around dedicated blue-teamer.
- **Seth Misenar, GSE #28:** Seth is a Cyber Security Expert who serves as a Faculty Fellow with SANS. Seth is a co-author for the bestselling SANS course SEC511: Continuous Monitoring and Security Operations. Seth provided the initial technical review for this book, when it was in its infancy.
- **Ryan O'Connor:** Ryan is an InfoSec security operation engineer for a leading security products company.
- **Phil Plantamura**: Phil is the COO for Security Onion Solutions. Phil has a 20+ year distinguished career in InfoSec working for defense, IR firms, and education.
- **Chris Sanders, GSE #64,** Applied Network Defense. Chris is a well-known security analyst, author, and educator. Chris reviewed the sections titled "A Day in the Life of SOC Analyst" and "Alarm Investigation Process."
- **Johanna Schafer, M.A.C.E.**: Johanna provided a layperson read through, checked it for readability, grammar, and punctuation for this book during the technical review process. My favorite error she found was the word "bacon" instead of "beacon", which for some strange reason, was a repeat occurrence in early drafts.
- **Peter Szczepankiewicz:** A long term colleague and 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.

Update Notes

Major updates are indicated with (V1#) in the section heading.

Version 1.01: Corrected grammar, spelling,

Version 1.02: Expanded the project plan section beginning on 20, and in particular the EDIS discussion based on request from a state InfoSec team. Reviewed and added various Windows Event IDs*, corrected a few errors. Updated the list of suspicious EXE's.

Foreword

The choirs sang and the trumpets blared as a joyous parade marched down the avenue amid the blinding confetti thrown from the high-rise windows above. Sweet smells of cotton candy and funnel cakes permeated the air. The feeling of triumph flowed through us all...at least it flowed through a much younger me. I also may have been the only one who heard the marching bands and the angelic choirs. And, even though my excitement was palatable, 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, but they didn't have a copy from which to work. At that time, our team spent countless hours thinking about our data, analyzing it, and determining ways to find evil. Our engineers built processes, tools, and dashboards so the SOC could work more effectively and empower the junior analysts. Over time, our SOC innovated, building more focused and 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 good team and a decently implemented SIEM.

But, now that I've seen more SOCs and log management implementations than I can count, both good and bad, I have since realized that maybe the choirs I heard were a little flat and that the trumpets might have been blaring something other than fanfare. If our group only had a book like this at the time, we would have used the SIEM with greater success. The most effective SOCs with the most solid SIEM implementations got there through thoughtful strategy and skill, with seasoned pros who had spent years in the fight. They answered the right questions: Which information is important? Which logs produce that information? Who needs it? Why? Encapsulated in this book is a fifteen-year career building SOCs and implementing SIEM technology for finding evil every day. Many of the thoughts in here come through in my work today.

At Security Onion Solutions, we routinely consult with organizations of all sizes which use our well-known free and open source platform as a core component of their network and enterprise monitoring solution. During many of those

deployments and in classes, we are often asked, "What should we monitor? What makes a difference? How do I find the 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, with hundreds of thousands of downloads and implementations across every industry, it keeps us pretty busy.

I met Don and many other thought leaders in our community at our annual Security Onion Conference in 2017, where Don delivered a very well-received talk on building security operations use cases. Don later asked me to be one of the technical content reviewers because of my experience as a technician, consultant, and leader. I was humbled and excited to participate. As I read through the draft, I found myself waxing nostalgic on my first SIEM deployment in the early- to mid-2000s. I later commented to Don that it felt like the words on the page were things I've been saying for a long time and were topics on many client engagements, but never written down in one place. In "BTHb: SOC, SIEM, and Threat Hunting Use Cases", Don covers how to use security focused data sources to their fullest, how to write a solid SOC focused Use Case, security metrics you can actually use, and how to build engagement plans and practices so you will be successful.

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

Phil Plantamura, COO
Security Onion Solutions LLC
phil@securityonionsolutions.com

Introduction

This idea for this book actually predates the first book in the series, BTHb Incident Response Edition. In 2011, our team needed to replace our commercial SIEM platform. We headed down a path that lead to my fourth major SIEM implementation. We needed an outline to develop use cases, document all of the attributes of a use case and SOC procedures to fully use the new platform. I wanted our chosen vendor to have the best possible chance of bidding on the work and completing our use cases on time, and on budget. After vendor selection, we engaged a major firm, and set about replacing the legacy platform. The vendor estimated they could achieve 26 to 28 use cases. After 12 weeks, we exceeded project expectations. They implemented 35 of 37 fully defined use cases that totaled 497 pages in print once the paperwork was done. The vendor liked the use case template format that they adopted it, and they still use it today. We added fourteen new right click integrations. We even had a custom UI extension that pulled in a dozen account attributes for every user account listed in an alert when we opened an alarm. Lastly, we also went from 1.5 people monitoring the prior solution to four full time analysts.

As a result of all that work, the idea for BTHb:SOCTH was born, and I started collecting notes that eventually became the book you now hold. Along the way I started a MSSP practice with a good friend working with a consulting firm. We won 78% of our POC's in year one, and had a 100% renewal rate during year two, so several of those life lessons are incorporated herein.

This book will cover many topics related to the Security Operations Team from a "Field Notes" perspective. It is based on a log career implementing multiple SIEM technologies, building SOC's, conducting all manner of cyber investigation, developing and running an MSSP. The major topics are:

1. Building a Security Operations *functional unit*, including provisioning plan, budget considerations, thought habits, analyst skills, and tiering structures.
2. Deciding how to structure your Security Operations capability and the services it will offer.
3. An extensive discussion of security focused use cases organized by their respective data source. This chapter describes what to monitor from a given data source, as succinctly as possible. Many of these use cases have a threat hunt theme to them.
4. Building Security Operations Use Cases using my own Use Case Template, followed by a complete use case to use as a model in your own work.
5. Critical SOC analyst skills and investigation processes, which my own team used while I managed a MSSP operational unit for two years.

6. A discussion on applying modern Threat Hunting to the Security Operations team.
7. And a host of other topics that relate to security operations, SOC analyst skills, and SIEM.

It is my sincere hope that this self-published 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 BTHb:INRE have shared with me over the years.

Thank you for our support,

Don Murdoch, GSE #99, MBA, MSISE

Security Operation Center Field Notes

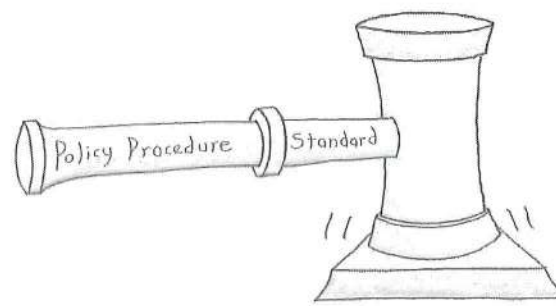
SOC Defined

A Security Operations Center (SOC) means different things to different people. Some say they "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 BTHb:SOCTH is:

> "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."

In a nutshell, the SOC is the *first line of defense*. This definition incorporates several important strategies for a successful SOC. First, a SOC must be under a single management and reporting structure so that it has a clear line of authority, funding, reporting, and accountability. Second, 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. Third, a SOC need to understand its area of operation (AO), how they will support the business, monitor business applications, and infrastructure. These criteria must be covered in the SOC charter. Fourth, SOC budget needs to be large enough to continually invest in people and support cross training instead of super sophisticated software. That concept leads to the fifth strategy: train and encourage analysts to be calm, correctly interpret alerts and their supporting data. This requires that SOC analysts are well trained.

One point deserves some elaboration. There are few different ways that the SOC team can establish its AO. The SOC can use the IT General Controls program, corporate policy/procedure, guidance from standards like the ISO 2700X series, or follow 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.



In order to achieve these various strategies, the SOC needs to know the network, the application to server relationships, what is happening on the network,

and be able to determine if that activity presents a significant enough risk to the organization that the activity needs to be effectively dealt with. SOC teams don't solve security issues with complex SIEM software. They solve it with knowledge, skill, and ability. Complex SIEM tools help – but they are not a technological panacea.

SOC Charter

Every security operations center needs a "charter". The SOC charter defines how the SOC serves the business, mandate(s), and define governance and operational rules, what its areas of operation are, and how the organization needs to respond to the alarm conditions and monitoring the SOC performs.

Note that the SOC charter is not the same as a project charter. The SOC Project Implementation Charter is the formal document that authorizes the project to develop a SOC, possibly implement a SIEM, and empowers the project manager to apply resources and create the SOC.

The SOC charter is often developed in tandem with the SOC/SIEM project implementation charter. The SOC charter should be scoped properly, whereas an implementation charter is a Project Management Institute (PMI) defined document. The term comes from the Project Management Body of Knowledge (PMBOK) as a type "project artifact". Don't get the two confused.

Business Value Chain Tie In

One concept that IT people don't often embrace is the business "value chain". The value chain is the set of activities that take inputs and convert them into an output that brings a valuable service or product to their 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, which must be monitored and fully secured following an IT General Controls Program.

In order 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 businesses' context and concept of operations[1]. Formally, a value chain should create some form of competitive advantage in the marketplace.

[1] These concepts are well defined in the Sherwood Applied Business Security Architecture (SABSA.)

Identify SOC Services

A Security Operations center can provide numerous services to the business and to IT. As you consider each of these services, be sure to incorporate them into your SOC planning process as well as the supporting skill, data sources, response patterns, and staffing to realize that service over the lifetime of the SOC. Further, as your organization considers the services it will offer the business, be careful to build out services which will be successful by only taking on a service that you can successfully deliver. The core services of a SOC *operations* team are listed out below. Your organization will certainly implement these services based on your own capabilities, funding, and staffing level.

Reactive Services	**Proactive Services**
Monitor Security Posture (Alerts)	Network Security Monitoring
Command Function (IR/Analysis)	Threat Hunting
Initiate & Manage Incident Response	Platform Health Monitoring & Support
Vulnerability Management	Cyber Threat Intel
Forensics/eDiscovery	Threat Intel Integration
Reporting	
Malware Analysis	**Other Services**
Intrusion Detection	Policy Procedure Support
Audit/Assessment	Internal Training and Support
Notification Refinement	

Monitor Security Posture: This is the primary role of the SOC: monitoring the environment for security conditions, alarms, health of the security platform, and responding through the organizations various technical solution(s).

Command Function: This may be a recurring activity, as the SOC coordinates alarm response, incident response, 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, will assist in eradication efforts, take information from the incident and use it to better implement internal systems based on newly found intelligence, and may also support pushing out updates or other fixes.

Initiate & Manage Incident Response (identification and remediation support): 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, secondary business units – a wide variety of staff outside of the SOC and IR function. In these cases, an *operational process* needs to be defined with a set of *releasable data* provided to those outside of the SOC/IR team. Don't

freelance or make up these points on the fly – plan ahead. To start planning, review the application inventory and determine if IR support can be handled internally or if a third party needs to be engaged. Once you have planned, exercise your plan at least twice a year using a tabletop exercise format. Once that is stabilized, integrate various 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.

Vulnerability Management: The SOC manager may be asked to assist, or even run, a vulnerability management program. The SOC manager should be very cautious not to take on tasking the SOC may not be able to handle: developing and deploying a round trip, full scope VA/VM program. Working through the process of safely finding, notifying, tracking, and attempting to identify the system owner and custodian, and *then gain system custodian and data owner support on remediating vulnerabilities in a timely manner 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 be oriented following 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 will also analyze the results of data collection. A key difference is that eDiscovery is focused on collecting search specific information from live, in use data and information repositories that is generated and used by people. Forensics goes deeper, examining system artifacts from the file system that show intent for users to interact with files and data, malicious software residing in memory, or data deleted from disk.

Reporting: Run reports to support compliance requirements and IT General Controls monitoring. Run reports to support alarms, incidents, and other reporting requirements. Respond to additional data requests.

Malware Analysis: If a SOC analyst can safely recover a malware sample, then they may be inclined to perform some lightweight malware analysis using services like VirusTotal, JoeSandbox, or ThreatExpert. That advice was useful in ten years ago, and is no longer considered best practice. In 2017, the better course of action is to run samples through a local malware analysis engine built on Cuckoo sandbox *to prevent informing the attacker, who is likely monitoring online services, that their malware was found*. These tools allow a user to upload a suspect binary and then advise if it is known bad and provide varying

levels of activity analysis such as registry changes, new services, file system changes, IP addresses in use, or domain names looked up. If the analysis reveals something suspicious, then the SOC analyst would take that operational intelligence and be able to better search security data. More complex reverse engineering beyond this cursory level is a very specialized skill and requires environment setup for this purpose.

Intrusion Detection: There are several detection systems can be deployed on the network or on a host. These detection systems (Snort, Suricata, Bro, PassiveDNS, etc.) all require care and feeding in order to make sure they are operating properly. Winning budget to implement a NIDS platform that doesn't maintain the ruleset isn't an optimal solution.

Notification refinement and improvement: For alarm conditions that are deemed valid, create notification with sufficient supporting information for the recipient(s).

Network Security Monitoring: NSM is the collection, detection, analysis, and escalation of indications and warnings based on network level data that indicate an intrusion.

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

Platform Health Monitoring: Monitor SIEM dashboards and alert stream, reviewing and acting on alerts following a priority basis. Monitor SIEM platform and other supporting data sources in order to detect issues and work with data custodians to ensure data survivability. Update platform definitions (assets, networks, privileged users, alarms, etc.) as the environment changes. Includes maintaining source data availability and quality by checking to make sure that events are parsed and creating new or refined alarms.

Cyber Threat Intelligence: This is the analysis of adversaries, their capabilities, motivations, and goals. Cyber threat intelligence (CTI) is the analysis of how adversaries use the cyber domain to accomplish their goals. When considering CTI, you should use multiple sources. Not all CTI sources are the same or offer the same degree of coverage. Also, CTI (in my opinion), includes understanding software vulnerabilities and ready-made attacker capabilities. For example, what are the new Metasploit exploits added this week? Metasploit makes the process of exploiting vulnerabilities significantly easier because exploits are

encapsulated into reusable code. How quickly does a new exploit appear after a vulnerability is announced in a technology you depend on? By keeping aware of attack tools, vendor announcements, and postings from major vendors from the IR community such as SANS, TrustWave, IANS, FireEye, CrowdStrike, AlienVault, and EMC/RSA, you can build a very low-cost CTI program and then make a purchase decision.

Threat Intelligence Integration: This is the process of carefully selecting and bringing in threat intelligence feeds into the system to improve alerting and better identify suspect or malicious sources, destinations, domains, and other patterns. Threat Intel sources and the information they provide should be on the detection roadmap.

Policy and Procedure Support: Many of the monitoring controls and capabilities should tie directly to established policy and procedure. As use cases are implemented, ensure that there is a tie-in to how the SOC will support PnP enforcement. More specifically, as this service area matures, ensure that SoP's are written to define how the SOC will properly engage with the user, supervisor, HR, and Legal in response to violations of PnP's.

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

SOC Project Planning Outline and Field Notes (V1.02)

"If you fail to plan, you plan to fail. "
– Commonly attributed to Benjamin Franklyn

Instead of repeating any of that content in BTHB:SOCTH, a condensed outline for planning a SOC based on the PMI PMBOK[2]. Also, *do not shy away* from using the PMI PMBOK because "project managers are annoying", "project

[2] This section was significantly updated for BTHb:SOCTH V 1.01

management is useless", or "it's just not that hard". A solid PM that understands how to drive a project to completion on time and within budget is a *tremendous ally for anyone building a SOC or implementing a SIEM.* This section provides a no frills, just 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.

Develop key business focused understanding of the organization and how the SOC can support its goals and objectives.

1. Understand the organizational need for a SOC, which means that you need to ***understand your organization's goals and objectives***. By being able to articulate how the SOC protects what the organization produces, sells, or the services provided to others, the SOC will have more credibility, be relevant to the business, and support your organization's mission statement.
2. Understand the business problem(s) the SOC needs to address and value chain resources that the SOC needs to monitor. You may need more of a "compliance" focused SOC, a tactical SOC, an Incident focused SOC, or some combination of these. The SOC will monitor several components of the value chain in addition to general IT resources. The SOC that intelligently targets the value chain for monitoring will be more successful and relevant to the business.
3. Identify the SOC sponsor. The sponsor may have an uphill struggle to initiate, build, and deploy the SOC. The SOC manager must be sure that the sponsor relationship is well maintained. The "customer" should want SOC services, and not have them dumped in their lap. The 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 to report to the SIEM platform.
4. 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 the need. Here, you should develop a formal business case. Be prepared to justify the staff, resources, access, and software needed to build a SOC.
5. Develop key "Security State" understanding (the "as is" versus the "to be" 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 the ISO 27002. See page 245 for more information.)

Build your initial business case, charter, project plan, budget request, and justification to support building the SOC.

This process will likely be two to eight months' worth of effort. Design the phases, identify the *key inputs and outputs per phase per the PMBOK,* and who will support each project phase.

1. Define the organizational ownership, responsibility, and SOC location. Attempt to locate a physical space that will accommodate twice the head count you will have in year one, so that you don't have to move in year three.
2. Identify the key roles for 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 (definitions in footnotes, more information on page 51).
3. Identify the relevant Policy, Procedure, and Governance - in place, or new PnP's that need to be written and adopted. Review existing PPG and determine if they support the SOC. Ensure that the SOC function is integrated into IT processes, particularly new application acquisition, server provisioning, and change management process. Also, the SOC will 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 they don't over react during change failures or other OS and app changes that may seem suspicious.
4. Document necessary staffing levels, training, and educational process(es) (more information on page 45). Here, concretely plan for the first year. Once that's done, develop a three-year plan and assume that you will have 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.* It is often a highly motivated person who will perform heroic acts and will eventually burn out, or a single person running a SIEM.

---

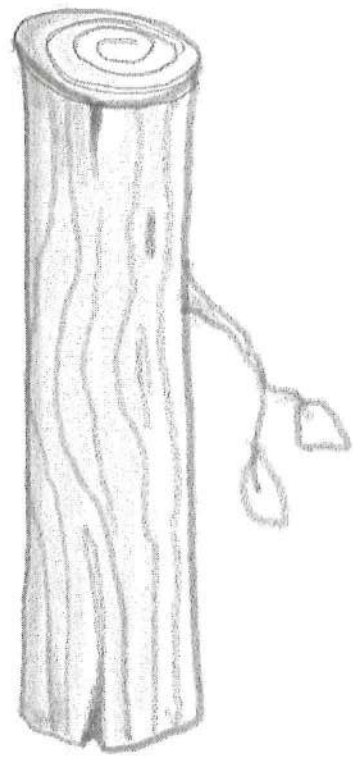
[3] Customers and users. Customers are the persons or organizations who will approve and manage the project's product, service, or result (from PMBOK V5).

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

[5] Stakeholder: an individual, group, or organization who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project (PMBOK V5).

[6] These are ITIL V3 Change Management terms.

5. Conduct a current data source survey. Identify the data sources, their logging configuration, assets, applications, application to asset mapping, data or logging suitability for the SOC. *You should not assume that every candidate data source is well instrumented and 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, you will need to inventory how each data source can provide information to a future SIEM: syslog (UDP or TCP), file write, database table, SNMP traps, etc.



Conduct an Environmental Data Inventory Survey (EDIS). (V1.02)

Not only do you need source system data, you need metadata about the network, organization, users, applications, and a mapping of the business processes that depend on the organizations applications.EDIS[7] begins with developing an inventory of major business processes along with the business process owner. From there, define the applications that enable said processes, and then the servers that support the applications - similar to BIA, BCP, and DRP planning. The difference between 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 rapidly investigate an incident Both processes collect similar data sets, and can complement one another. Data includes users and their demographics, network maps, address ranges, applications in use, app to server mappings, app to RDBMS (or other data storage), 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 to get at least "read

[7] The steps are nearly the same done in the Business Impact Analysis (BIA) phase of a traditional Business Continuity Plan (BCP), and then the Disaster Recovery Plan (DRP). If your organization as a BCP, DRP, or TOGAF[7] style EA team, then consult with them for the application and server inventory

only" credentials for the systems that house this information such as a Configuration Management Data Base (CMDB).

From a Project Management perspective, the major steps for the EDIS process are outlined below.

1. Identify and develop an inventory of major business processes and departments. 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 in order to maximize the data collection process.
3. Identify the applications which support business process, along with the data owner and system custodians, and 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. A sample list is shown below
    a. Network devices: Firewalls, IDS, VPN, DNS, DHCP, NAC, WiFi, WIDS, switch logs
    b. Technology Support systems: Mobile device tracking, Anti-Virus, Enterprise Detection and Response (alerts), Vulnerability Scanner, Password management system, web proxy, Email, virtualization platform, database systems
    c. Windows focused event logs: Application, Security, System, Sysmon Operational log. Note that as a subproject, the SOC implementation may need to spin up a separate project to implement WEC/WEF.
    d. Application logs: these usually require a database query or some other method of data collection
    e. Other relevant tools: Email security tool, Insider threat tool, System Backup logs
5. With the list of applications and security technologies, a line item for each item can be created in the project plan.
6. Estimate the number of hours to incorporate the data source for the use cases - these items will expanded in a subsequent phase, following the "progressive elaboration" model.
7. *Include a project specific line item* to develop a briefing for the SOC team that explains each data sources field set and field values.

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 SIEM Data Collection Methods and Considerations on page 207 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 get an average daily volume over at least a five-week period, which should catch any surges that naturally occur across a month boundary.
4. Determine if data can be trimmed, meaning review the data to find out if there are low to no value records provided by the data source that can be pruned or dropped either at 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.

Plan the Technology provisioning process to support the SIEM, and another identified SOC services (see p. 17). Plan for twice the data you think you will need in year one.

1. Hardware: Including disk and disk controller architecture, as influenced by logging requirements and SIEM platform.
2. 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.
3. Log storage architecture, scripting, and long-term storage requirements. For long term storage, you really need space over speed, because you rarely go back to data past a 90-day threshold. Reliable, safe, and large long-term storage is more important that blazing speeds. Blazing speed is needed for the past three days' worth of logs.
4. SIEM and supporting software. Note that most major vendors have their own predefined project plans for implementing their software, which you should leverage to the fullest.
5. Spend time on the Budget Process. A SIEM is actually a major enterprise wide application, and it deserves the same budgetary rigor as with any enterprise project. This means build a first-year model to get started, 3-year projection, and then a 5-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 over year.

6. 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 some significant custom development efforts. Each data source brings its own capabilities and will need some form of alert support.
7. In order 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 up with them.

Build your log architecture, source data collection delivery, and SIEM and logging deployment plan.

There is more information in the SIEM Deployment Checklist section beginning on page 198. Also, 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, compliance requirements, and InfoSec 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". Its critical to determine the EPS *after* the source system has had its auditing level configured!
3. Provision the hardware and storage platform and implement.
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, so you will likely need to develop a database query to pull data from the audit table, implement auditing, test, develop a method to archive current data to a historical table, and monitor to ensure that the query process has minimal system impact.

Build out Use Cases.

1. There is an entire chapter in this book on building out use cases. Review all of that material and compare it against the use cases in your chosen platform.
2. Plan how to implement the vendor-defined use cases as these should provide baseline coverage.
3. Forecast the effort required and data sources to implement your own custom use case.
4. From there, prioritize the implementation so that you will have project measurement that supports defining earned value.

Build your response processes.

Response processes are enabled by the variety of data arriving, applications, business processes, and your requirements.

1. Response processes will be driven by your security program and the applicable standard you are following.
2. This part of the planning process should answer incident resolution questions like this: "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?"
3. 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 based on security condition X, Y, or Z presenting itself?

Build your SOC Metrics, as defined in Metrics for the SOC on page 39.

Many technical platforms have reporting and measurement that supports SOC and SIEM metrics. There are many organizational metrics that need to be developed and collected. This aspect of developing a SOC and SIEM implementation plan will evolve over time.

Build, and implement your continuous training program.

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

There are several open source or very low-cost options. Consider ENISA, SecurityTube, SANS Cyber Aces, local BSides conferences, DerbyCon, and the annual Security Onion conference as more inexpensive education options.

Consider, and be Prepared, for Tough Questions

In order to fund SecOps, SIEM, and a SOC, you will undoubtedly face many questions. Here are a few of them that I have been asked over the years, condensed for publication. Determine the answer when building your funding request.

1. Nothing has happened yet. Why do we need to do this? How can you be sure that nothing has happened yet? As a possible answer, try these out: "It's not if, it's when."
2. 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?
3. Did the organization incur any costs from an incident last year? Virus outbreaks? What costs incurred from our peers and competitors?
4. How many users at what "level" were negatively affected (as in lost productivity) from an incident?
5. How are you going to measure yourselves and get on the IT Balanced Scorecard? As a possible approach, ask if you can be on the *business scorecard during the SOC charter development process.*
6. How will the team determine what alarm conditions are prioritized over something else – who wins? (Hint: asset value tied to critical business process and revenue stream protection).
7. I thought we spent X on Security last year. Why do you want more?
8. I know We don't have anything worth stealing. Why do we need to do more and more of this security "stuff"?
9. Don't those things cost millions?
10. We are doing vulnerability assessments. Isn't that enough? If you are not doing active and timely remediation, then no, it isn't.
11. Those security people keep saying no, so I'm going to say no to them this time. So there.
12. We can successfully outsource that for 1/8 the cost, right? After all, that's what the vendors say. Why do you disagree with them? Aren't they experts?
13. How will this SOC solve business problems for us?
14. What does this SOC thing look like year 1, year 3, and year 5?
15. I don't want to buy more expensive security people only have them quit. What are you going to do about staff retention? Burnout? Attrition? Internal transfer? I recently heard at a security conference when people take a SOC job they plan to quit in 18 months.
16. How will you know when you have had a success? What does success and failure look like for a SOC? (or a major security purchase?)
17. Have you been talking to the auditors again? They said something about this last year. I bought a new firewall.
18. Show me a playbook first – can you do that? Come back when that's done.
19. IT Is outsourced, it 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[8].

[8] I would encourage not to blurt out in response to this question that that is "A Guaranteed Orange Suit Acceptance Posture". It does not go over well.

20. What can you do with a third of that? Because that's all we have.
21. We spent $3.5M on SOX last year. No more!

Collect the Bread and Butter Data Sources

There are many baseline systems that need to be monitored because they represent key data sources that you need in an incident and support compliance. This is part of the EDIS process. At an absolute minimum, these information systems and data sources to collect are:

1. DNS activity, with a focus on internal to external activity first (about 8% to 10% of your networks' DNS request/response traffic).
2. Windows Domain Controller security log.
3. Most, if not all, Windows member servers.
4. *Account life cycle, process execution,* and *presence* indicators from workstations. This item is best accomplished using Windows Event Forwarding and event subscriptions *because this is a native* built in capability in Windows, and prevents the need to deploy yet another agent.
5. Perimeter firewall. At a minimum, 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 get a better data source for the communication flow. For example, if you have a proxy, you can consider not recording firewall data to/from the proxy if you can get the proxy logs. Proxy logs are superior to firewall logs as they are application aware and are user attributable whereas firewall data is not usually user attributable.
6. Database Account activity and account management.
7. For Linux, the minimum to collect are the sudo, auth, and authpriv logs.
8. Antivirus centralized console data.
9. Forward (outbound) proxy data. For the proxy, validate that the system records the user agent, referrer, the URI query string, and the allow/deny decision. If the proxy understands the site type, that is also useful.
10. Document editing "in the cloud", such as Google's GSuite or Office 365. This means who touched which file and how.
11. Shared Storage file system activity, as in who touched which file and how for user and process exposed shares.
12. VPN activity.
13. DHCP transactions.
14. Network device authentication which usually arrive through RADIUS or TACACS+. Further, network change detection, which usually comes from Syslog events.

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

Useful MBA Concepts: SWOT and PESTL

There are two business management concepts that help when designing, planning, and building a SOC: SWOT and PESTL.

SWOT Analysis

SWOT is a strategic planning technique used to help an organization identify the Strengths, Weaknesses, Opportunities, and Threats that every manager should understand, and be prepared to use in strategic planning exercises. Building a SOC is an *internal business venture*, which is affected by both internal and external pressures. SWOT analysis will improve your business case for your SOC, will also help you plan, *and if done well can help identify adversaries that will launch attacks against the organization*. Below is a very brief example to give you an idea what a SWOT analysis for a SOC project could look like.

Table 1 An Example SWOT Analysis

	Traditional Business Mgmt.	Security Operations Example
S	Characteristics of the organization that provides advantage over others	Technical controls and monitoring capabilities; strong perimeter controls; Policy/Procedure in place and followed; valuable IP in place is protected (and targeted).
W	Characteristics at a disadvantage to others (competing projects)	Moderate Funding; 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 (does not mean technical exploit!)	Improving security software; Implement awareness training; local University has Cyber program; Current VA program is semi structured.
T	Items that can cause trouble or thwart the project	Increasing stealthiness of malware; trusted insider can defeat controls (accidental or can be disgruntled); ownership; resistance to response from alerts; management expectation to "find the bad guy" is high.

PESTL Analysis

PESTL (**P**olitical, **E**conomic, **S**ocio-cultural, **T**echnological, and **L**egal) analysis is a framework of the macroeconomic and macro environmental factors pushing against the organization. It is used by strategic management, marketing, and business development teams who 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 where 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 its 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 the applications that enable the business.

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 that 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, database(s), 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 Disaster Recovery Planning / Business Continuity Planning (DRP/BCP) team can be highly valuable as they have a criticality-based view of the application stack. If you don't have a DRP/BCP team, the build the app inventory yourself with the intention that the work can be used for SecOps, as it will assist IT when there be a need for a recovery event.

For example, your organization is likely to have an eCommerce presence. There are several components on the eCommerce chain: digital storefront, order processing, messaging to the customer and supplier, the WAN link(s), back-end data storage, staff that manage all of these IT components, and protecting 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 violations in baselines 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, many reasons to fund security operations and the logging infrastructure that SOCs and SIEM platforms require.

1. Regulatory compliance (HIPAA/HITECH, Sarbanes Oxley, and others).
2. A prior incident may initiate a funding event.
3. Management Directive out of a genuine desire or a fear response.
4. Your chosen standard that defines how IT is structured, and are therefore audited against, may require SOC and a logging platform.
5. Logs within a given system are volatile. Some systems, like a Windows domain controller, only hold log data for a few hours and then the data rolls and is lost forever. Some systems hold data in memory, in a buffer, and when power is lost or the power is lost, so is the data.
6. Without Logs, you have no ability to go back in history and find issues – security, operational, or change related.
7. It is, in fact, "the right thing to do".

Security Operations Centers Cost Components

There are numerous cost components to consider when building a Security Operations Center. Below are many of the common cost components. For each of these costs, carefully analyze your current environment as you develop a "build vs. buy" analysis.

Direct Costs - There is more information on this below this list.

1. **Internal Staffing Level**. A 24/7/365 requires at least 5 whole people, using the barest bones staffing model. Target 9 staff people and one SOC manager.
2. **Vendor neutral Training**. Some examples are SANS, Security University, CompTiA CASP, and ISC2 SSCP.
3. **Product Training:** SIEM solutions have training provided by the vendor.
4. **Tools:** Desktop Infrastructure, OS, Office, SIEM license, and investigative tools.
5. **Subscriptions:** SOC's will have several subscription services, and in particular, threat intelligence services.
6. **Hardware:** Server, storage, and network Infrastructure. Be aware or the typical hardware refresh cycles – usually 4 years.
7. **Forensic Hardware:** Forensic hardware has unique requirements because these systems are usually isolated onto a small LAN in a locked room. For

example, a customized forensic workstation known as FRED can cost $5K and up, storing images on central storage can take many terabytes, and write blocker kits can easily cost $1K and up.

8. **Software licensing costs which includes annual maintenance:** Software includes SOC support, additional licenses for various management consoles, SIEM platform, ingest costs, forensic packages, PDF generation applications, BI[9] tools, and eDiscovery capabilities.
9. **Facilities:** SOC Room, furniture, shared large format monitors or projectors, and proximity card or possibly biometric door control. Also, the forensic analysis space should have its' own separate locks and proximity card control.
10. **Upgrades:** Annual upgrades – often handled on a SoW basis with the vendor.
11. **Vendor assistance:** Over time, you will need vendor assistance for new content, improved reporting, more training, and possibly upgrades.

Indirect Costs:

1. Recruiting costs such as a portion of building, recruiting, and staff pay increases.
2. SIEM Selection costs for the initial project, which includes labor expended to specific, review, and select the primary SOC toolset.
3. Developing on the job training for your SOC staff. Note, this will tie up key SME's and the time commitment cannot be underestimated.
4. Integrating new data sources, which may require customized data parsers, alerts, and reports to be created within your platform.
5. Periodic internal or external audit support.

Staff Cost Considerations: A Security Operations Center needs to be staffed by skilled Information Security Analysts. Period. Shiny SIEM solutions don't solve cases, educated and seasoned people do. ***After participating in InfoSec since 2001, I can confirm It's just that simple. Highly skilled people can produce more accurate and timely results with a moderate product set than novices with a super expensive shiny toolset.***

Base pay, benefits, and the inevitable staff turnover disrupts the cost model. The US Bureau of Labor Statistics lists Information Security Analyst base pay at $92,500 in 2016, and $95,510 in 2017. If your internal overhead load is 30% for administrative costs, a loaded position costs $124,163/year. At this rate that is $620,815 for five people per year in 2017 dollars. A 2016 study published on glassdoor.com can help to understand the hiring climate: Companies spend $4,000 to fill a position through open recruitment with a 52-day vacancy period,

[9] For example, Advizor Analytics and/or Tableau.

47% of candidates decline an initial offer. Further analysis found 50% of employees left a position due to their manager. These numbers help to define how much a temporary contractor would cost if you cannot find FTE's or need to replace an FTE. To minimize this cost, concentrate on getting SOC staff through the investment zone period and into the return zone period by onboarding them into handling specific SOC services and IR tasks as rapidly as possible.

The staffing cost is further influenced by the *coverage model*. If the team operates with 24/7/365, that requires 4.52 people, or 5 FTE's *at a bare minimum to staff the SOC with a single person staffing the facility. A lonely job indeed*. This value is based on 8,760 hours per year, two weeks paid time off and 8 holidays which yields 48.4 work weeks, or 1936 hours of coverage per person. In reality, 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 three will provide additional coverage during high activity hours, such as 7AM to 7PM and some portion of Saturday. Missing from this estimate is the percentage of "admin" time. Admin time consists of all other company required tasking that detracts from conducting actual heads down job duties.

Incident response has a very *high burn out rate* compared to other technical professions. Therefore, to compensate rotate your SOC front line through different SOC services so that they have variance in job duties. Also, look for analysts that aspire to move up and not stay doing the same thing every day. SOC managers should always evaluate opportunities to vary the job duties in order to retain people.

Facility/Space: Most organizations have a per square foot rate for office space that should be in the cost structure. As an example, the average 2016 cost per square foot in Atlanta, Georgia was $20.01 for Class A space and $16.36 for Class B space[10]. If you assume 90 square feet for shared workgroup areas with two workspaces available for a five-person team on rotation, the monthly office space cost is $2,944.80 for a two-person SOC. There are other single purchase costs, however. For example, you may want two large format monitors and PC's to run them with everything mounted on the wall. That could easily be a $4,000 single event cost item.

[10] From Offices.Net, August 2017.

Vendor Neutral Formal Education: Assuming you can find security people, you will still likely need to train them in SOC operations. As of August 2018, one of the best courses from SANS Institute is "SEC511: Continuous Monitoring and Security Operations" with its corresponding GIAC GMON certification. This course covers the practical skills needed for every SOC staff member. I can attest that there is a measurable improvement to the quality and speed of each analyst who completed this course and the corresponding verification. The August 2018 cost weighs in at $6,939 USD. Also, ensure that the travel and hotel costs are included when looking at the cost of training, and estimate $1800[11]. Stay for Day Six and compete for the SEC511 coin[12]!

Product Training: Every major SIEM vendor has a series of product training course. These are usually included in the initial proposal and implementation. There will be a cost for new staff as they come onboard. To defray the cost, I've had success by asking for a training credit with an upgrade, system enhancement activity, or adding in new product component.

Organizational specific training: On the job training will be a continual process. There are numerous studies that quantify the cost to develop robust and reusable training. Data from The Association for Talent Development[13] listed the time to develop an hour of instructional delivery (a formal class) between 43 to 185 hours for stand-up professional instruction, with numerous factors affecting the time. Don't count this lightly, and don't ignore it. For concise insight into the learning organization and how valuable it is to the company, review Chapter Seven in Paid to Think by David Goldsmith.

Desktops: Analyst hardware, monitors (the more, the merrier!), monitor arms, client-side software, analyst licenses for dozens of applications, furniture, and lighting. Think Quad 24" or 27" displays, and an Ergotron type quad arm. Nice!

Vendor Support: Dedicated vendor support for the security product suite, use case implementation, and continual upgrade processes. These support relationships are usually priced on a per hour basis with a minimum number of hours per week. If your SIEM vendor charges $225/hour and sells this support arrangement with a minimum of 4 hours, the annual support cost is $46,800.

Infrastructure: Back end servers, sensor platforms, network instrumentation such as a TAP, and multi-tiered storage. Plan for a per server hardware refresh

---

[11] I have had good luck shaving a bit by staying in the hotel next door.

[12] Coin Image provided by the SEC 511 course authors and is used with permission.

[13] https://www.td.org/newsletters/learning-circuits/time-to-develop-one-hour-of-training-2009

at 3-4 months before your server maintenance period to ensure you are not paying a premium for keeping a server online outside of its maintenance window. If you need six servers at $8,000 each, that's an initial capital outlay of $48,000 just for the hardware. Plan for 20% annual maintenance, and technology refresh at the four-year mark. Most of the SIEM implementations I've done were virtualized using VMware 5 and 6. This model can be quite successful – but you will need to add in the cost for the Hypervisor. When it comes to actual capacity, spec 2 more CPU's and 4 more GB of memory that you think you need and virtualize your platform on just that server. The long-run benefits you will gain are tremendous. These include volume snapshots, copy over to the new platform during tech refresh, and the ability to more easily mix and match drive configuration.

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

Content Development: Developing new and refining current *use cases* within the SIEM solution. Current use cases will also be updated based on improvements from the threat hunting team. The better you define the input data, content needed, analysis, rules, notification, SOC actions, and outcome desired, the more accurate a cost you will have and the higher the opportunity for success.

SIEM Software: SIEM platform licenses are most often driven by a sizing factor. Typical factors are the "ingest" rate in events per second (EPS), GB per day, or monitored device counts. You will find there is a class of non-security relevant events that arrive at the SIEM along with the data that's really needed. Depending on numerous factors (event cost, processing horsepower, log storage, event width) you may want to develop a tiered logging method. Costs can vary widely here, from $20,000 per year and on up. The better you define your environment, the better an estimate you will get from a vendor.

SIEM Software Upgrade: Some upgrades for enterprise systems can be performed through an update process, and some cannot. Experience with five different platforms leads me to advise that a complex upgrade, such as a major upgrade, may be better off outsourced to the SIEM vendor. A typical SoW will be 40 to 80 hours at the vendor's rate plus travel and expense. Ensure that you investigate this fully with your vendors and integrate at least one annual upgrade event to your platform.

Audit Evidence Support: The SOC is often asked to support reporting on security event data and incidents in direct support of an internal or external audit. Staffing this specific role is closely related to the regulatory environment and how often auditors make requests. To estimate this, determine how many audits your organization responds to per year and the reporting needed to support those audits. The SOC team should always record their time to support audits, as this is a service to the business. If you have recurring audits tied to a particular unit, then ask for accounting charge codes to document costs for the SOC to support operating units.

In House vs. Outsourced vs. Hybrid SOC

Now that you have a structured outline of the costs and most of the long-term factors involved in building a Security Operations Center, you are in a better position to consider the pros and cons of outsourcing part or all of the SOC function. Some empirically based observations on engaging outsourced Managed Security Service Provider (MSSP's) are listed here:

1. Startup time will have an impact. MSSPs in effect deploy a partial to full SIEM solution on your network. Each data source needs to be integrated into the platform, hardware will need to be deployed, and your organization will still need to define your own incident response process.
2. An MSSP will only be able to go just so far when investigating alerts. If you are fortunate, the MSSP can cover 50% to 70% of the alarm conditions well and will engage your organization on 15% of the observed alerts.
3. MSSPs will *never know your network like you do, and you cannot easily quantify this impact to their quality of service delivery.* MSSPs also are unlikely to know what changed on the network, as they rarely participate in change control.
4. MSSPs work with you through a defined SLA and reporting relationship. They cannot replace your own staff who can reach out directly to a system custodian – this is an invaluable benefit to having in-house staff functioning in a security operations role.
5. Their opinion on alarm sensitivity and configuration *is not your opinion* because they tend to look at "genuine threat" conditions, and will ignore or tune out many other conditions. Your use of SOC should include policy issues, threat hunting, audit reporting, and gleaning operational value from the mountain of data the SIEM will consume.
6. There are some tasks that should be outsourced *if they are infrequent,* like system forensic analysis. However, the battlespace of today tells us that we need a memory image more than a disk image. This is nigh impossible for a third party outsource MSSP to collect but may within their ability to analyze.

7. You cannot delegate responsibility to a third party for the security of the assets under your organizations care, no matter how much someone tries to convince you that you can. You may be able to delegate authority to operate, but not the responsibility of system security.
8. 7/24/365 monitoring by a third party will cost you less for the labor component than building your own 6-8-person team. There is no getting around that fact. If you you are being pressured to outsource, realize this argument and devise ways to respond to the argument.
9. MSSP's may also be able to perform system upgrades and very likely have done more upgrades than you. Factor in the cost of your deployment a week or two to perform an annual upgrade.
10. Lastly, you get out of a MSSP relationship what you put into an MSSP relationship. If you do not invest time, then don't assume that they will give you stellar results.

Getting into the Hunt

Historically, SOC would monitor a variety of prevention-oriented systems and respond if one, or many, of these platforms alerted the team. Then they would spend time validating the alert, communicate with the system custodian, owner, or the end user, and if the situation were an incident, they would respond.

The "reactive or detective only model or posture" from the 2000's is no longer effective today. Today's SOC teams need to change their focus, assume that there is a likely compromise, become detection oriented, and *proactively mine* the vast amounts of data coming into their systems and actively look for patterns of intrusion and misbehavior. Proactive threat hunting is an ideal career and skill development path for SOC analysts. Once they understand all of the organization's data sources, know how to handle alerts, and demonstrate that they have established research skills, capitalize on that and get them involved in threat hunting. Depending on how the SOC team operates, you could have a SOC analyst can perform hunting one day a week, one week per month, or take a particular hunt pathway. Threat hunting is further defined on page 171, with numerous use cases beginning on page 61.

SOC Directly Supports the CSIRT Function

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 regulation. The need for a CSIRT is especially true with modern malicious software running rampant, automated ransomware, industrial espionage, criminal elements, nation state grade hacking teams, and host of other aspects of digitally based asymmetric warfare. Sounds sensational,

doesn't it? Today's cybercriminal will exploit any weakness they find to extract untraceable digital crypto currency from any potential victim. Regardless of the degree of sensationalism, there are several reasons to advocate for and build a CSIRT function in your organization, which is in turn supported by the SOC function.

1. The SOC provides an active detection capability that should enable early response and limit the long-term impact from an incident. The CSIRT can then coordinate responding to an incident with the goal of identifying, containing, and reducing incident impact. Further, the CSIRT function can return Indicators of Compromise or IoC's back to the SOC so that the SOC can perform historical data analysis, such as searching for internal systems that communicated with a found IP or domain name. Thus, a more mature CSIRT and SOC team can capitalize on a Threat Hunting capability in order to seek out and find a malicious agent that was recently on the network.
2. The CSIRT influences, supports, and fully leverages the security spend. It helps to ensure that the SOC tooling is in place will support incident response and Security Operations. Or put a different way, having a single CSIRT should ensure that the best tool for the environment is in place, can be leveraged, and others are decommissioned in order to maximize the dollar investment.
3. Maintain objectivity when interacting with internal staff, classifying incidents, and prioritizing the response process. One of the challenges that a CSIRT will need to deal with is staff relationships and maintaining objectivity. Like the HR function, which is charged with enforcing company 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.
4. Lastly, regulation. There are numerous aspects of the business environment that mandate an incident response capability. These include, but are not limited to: HIPAA/HITECH, PCI DSS 3.2, and Sarbanes Oxley compliance.

Metrics for the SOC

"What cannot be measured, cannot be managed."
- W. Edwards Deming.

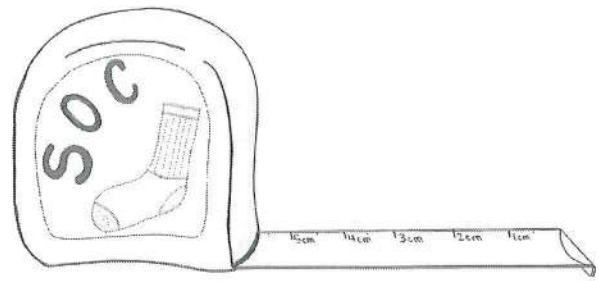
"Not everything that counts can be counted, and not everything that can be counted counts.""
- William Bruce Cameron

Mature business operating units and enterprises utilize various methods to measure the operating units effectiveness. The SOC is no exception. The question is how do you get there and avoid toxic metrics that demotivate your staff?

In the book Pragmatic Security Metrics, W. Krag Brotby and Gary Hinson make several key points about metrics. Some of their key definitions are listed here from chapter 1.6:

1. Instrument: short for "measuring instrument," that is, a device for measuring.
2. Measure: (verb) to determine one or more parameters of something.
3. Metric: a measurement in relation to one or more points of reference.

In Section 2.6, they 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."



In Section 3.2, they state "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)."

There are numerous metrics and measurements that can be developed and applied to a SOC. When it comes to designing a metric, there are several criteria.

Take these criteria into account as metrics are developed.

1. Metrics should be relevant to the goals, objectives, and the mission of your SOC as a business unit. This means you should be able to describe what you do numerically, and also explain what you don't measure.
2. As you evaluate your data, security use cases, and metrics be sure to develop a roadmap of what you will measure, how those measurements will be captured, the tie in to the IT General Controls and security program, and the business value chain where possible. Metrics can then provide guidance for decision making.
3. Be sure that what you are measuring will drive towards a a measurable outcome. Don't measure for the sake of measuring. Every metric should inform a consumer and seek to either to change behavior or demonstrate

that current behavior is operating well within established procedures. Here, you should be clear on any action that you expect a consumer of the metric to take based on the measurement.

4. A metric should support a "control", and therefore should match up to your ITGC program. If you do not have one, look to standards like the ISO 27002.
5. Bad data is marginally better than no data at all. If you are not collecting any source data for what you need to measure, start there, but do not stop. Use good data to its fullest.
6. Avoid burdening the analyst with the need to record an excessive amount of using some artificial means to track what they do like a complex spreadsheet. Instead, develop methods to mine the SIEM platform, the workflow system, 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 their native tool*. Also, following EoM principles pushes you to consistently leverage an internal capability of the SIEM platform, the less likely you are to cause mistakes.
7. Tell your story in terms of your business / organization. When telling stories, it is very important to remember the audience and use terms and definitions that they will understand.
8. Two key acronyms come to mind:
    a. Be SMART: Specific, Measurable, Achievable Relevant, Time-bound
    b. KISS, or Keep It Simple, Sam/Susan. This isn't actually meant to be cute. Rather, ensure that the name of the metric and the measurement scale is obvious. A metric the requires explanation is not likely to be an effective metric.
9. Determine how you can build a score card that measures the information and technical security posture of your organization. Whenever possible, build a tool to demonstrate how effective the technical tools work. You may not show this tool to management – but you should be ready to.
10. Work hard to avoid any toxic metrics, which are one 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

Table 2 Example General SOC Metrics

Metric[14]	Definition and Notes
# Unique Data Source Types providing SIEM data	Defines how many different information system data source types are consumed and available for analysis. This measures how many of your technical systems and applications are instrumented. There is a corresponding percentage of coverage in unique sources / total sources.

[14] In the table, MTT means "Mean Time To".

Metric[14]	Definition and Notes
MTT Detect Data Source Issue	How long does it take to detect that a data source is not functional, which is affected by the volume and velocity of data arriving from the data source?
MTT Correct for Data Source Issue	How long to resume data delivery once an error is detected in data delivery?
Time to sweep the enterprise	The Security operations function should be able to check every host in the enterprise for IoC's, or the host's security state. Ideally the average time to interrogate the enterprise should decrease over time, and the percentage of completeness should increase (# investigated / total #).
% of apps under ALCE [15]monitoring (Non-AD Integrated)	Measures how many shared applications report account life cycle events. Metric assumes that the app inventory is known. Applications that defer the central directory either through native integration or LDAP query *are counted*. Desktop apps are not normally included in this metric – only shared, server based, or SaaS applications. This metric has an *implicit assumption: you don't need to measure applications that are AD integrated for ALCE events. You will need to validate this premise for your own organization.*
% of SaaS under periodic ALCE monitoring	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 both have accounts in the central directory or can be accounted for, and that there are no disabled organizational accounts which are enabled in the SaaS application.
MTT Close an alarm by Close Category	Measures the decision-making process by the SOC analyst to close an alarm when it can be explained, processed, escalated, is non-reportable, or non-actionable (example close codes). **WARNING**: This metric and ones similar to it can easily become TOXIC to your staff. Be very careful in adopting this metric. Instead, search for alternatives to show that the staff can effectively respond to a portion of alarm conditions on a daily basis.

[15] ALCE: Account Life Cycle Event

Metric[14]	Definition and Notes
MTT Forward an alarm up Tier	Measures the lowest level analyst response time to identify that an alarm requires further action or resolution by the next level analyst or application SME.
MTT Open a formal Incident	The SOC function may open up a formal incident at any point in the alarm review process. This measures overall SOC and Incident Response team's capability to detect that an alarm or another condition is indeed a "security incident".
MTT Implement a use case	Measures how long it takes to define, document, instrument, and train on a specific SOC Use Case (see **Security Monitoring Use Cases by Data Source** beginning on page 61 for more information on SOC Use Cases).
# of Implemented Use Cases	Each organization will have a defined set of security conditions that SOC is able to handle with a supported use case or a response playbook. This metric measures SOC capability and coverage. Cautions: do not count the number of "detections" from your SIEM platform, or "alerts". Rather, take those into account and define your SOC playbook. This metric also guides how much training is required for new analysts and how well you are doing at mining your source data. Also, "# of "new" event conditions converted to Alerts can count here, but may not warrant a full use case.
# of Use Cases (rule) that never fire	While it is true that you can create, and test, a notification process for a rare condition, the SOC should keep an eye on use cases that never cause an alarm or don't prove out over a reasonable period, say one month. Try to keep use cases that never fire minimized.
# of Events Received	As a raw number, this metric isn't tremendously valuable. A better number is to measure events by severity, priority, or criticality – and don't get confused here.
# of Alerts by Severity	Based on a combination of your SIEM platforms.
# of high severity alerts not reviewed after 8 or 24 hours.	Measures how well SOC does at putting attention on all "high severity" or "high priority" alarms, per shift, and per day. If the frontline analysts are not capable of keeping up with alarms, then consider adding more staff, improving automation, and put attention to defining "close/escalate" criteria on these alarms. Once done and under control, this metric/measurement would push down the severity levels.

Metric[14]	Definition and Notes
Rules Tuned to Minimize False Positives (per week/month)	Every SOC should have at least one staff member who spends some time improving the notification rules within the platform. The better a SOC tunes the platform, the better it is demonstrating understanding of the environment.
ATT&CK Coverage by Phase	The MITRE ATT&CK matrix (page 174) is a knowledge base for understanding adversary behavior and the attack life cycle. This matrix can be used to evaluate how well the SOC and current instrumentation can identify presence of an attacker on the network.

Incident Response 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, they will turn that over to an Incident response function.

Table 3 Example Incident Response Metrics

Metric	Definition and Notes
Cost per incident	There are two dimensions to cost: One is an accumulation of non-FTE costs contributions, such as paying for credit monitoring during a data leakage event. The other is the number of FTE hours lost.
MTT to Detect a Security Incident	How long does it take for the SOC to review an alarm and determine that it is, in fact, some sort of incident?
MTT for Detect to Contain	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.
MTT to expel an intruder	Once a true intruder is identified, meaning a real adversary, how long does it take for the security team *as a whole* to push the intruder out of the network. Be careful to analyze and report this correctly for your environment to ensure that consumers understand that there are more decision makers than just the SOC involved in this metric.
Incidents opened and closed	These should be trailing numbers, meaning as incidents are opened there should also be incidents being closed. These are measured per day, week, and month.

Metric	Definition and Notes
Avoidability of an Incident	Incidents should conclude with some form of Lessons Learned function, meaning that as a result of an incident the security posture of the organization is improved to the extent possible. If it is determined that the incident was avoidable if a common security practice was in place, report it.
Thoroughness of eradication practices	Measures whether or not the original compromise event, or one that is substantively the same (like a remote exploit) is observed subsequent to the first occurrence.
MTT Notify Principle, System Owner, or Custodian (Incident metric)	The recipient of an alarm condition may be a little hard to track down. There are at least three possible recipients – a principle within the organization, such as an operational director responsible for the affected system(s), an actual designated system owner, or a custodian such as a system administrator. SOC should define its escalation points and determine how to measure how well and quickly it communicates to the designated recipient.

SOC Training, Skills, Staffing, and Roles

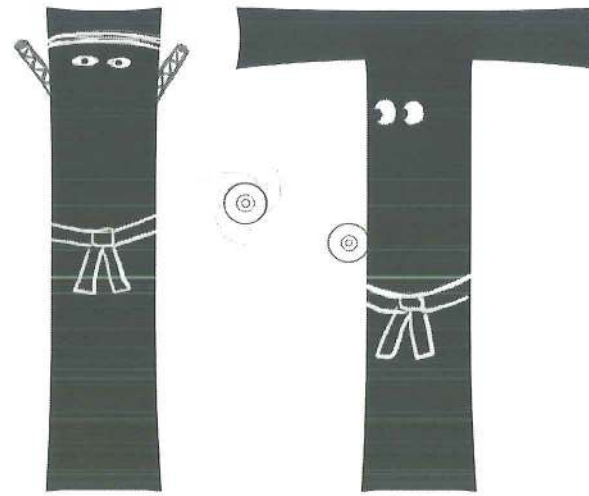
Effective security operations teams require technical skills, should possess certain personality traits, and require product training by role. Staffing our SOC team is critical to success. It is important to understand that you cannot "make" ninja grade incident handlers and SOC analysts who can synthesize 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 only comes with "time in the game". What you can do is develop people, provide them the opportunity to grow, and develop strategies to keep them in the Analyst seat. You can train people to respond to specific alarm types, handle specific cases, and work 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 Onboarding and Initial Training

Training for SOC will come in several flavors:

1. **Product skills**: These are actual vendor solution training. This is achieved through vendor offered on the SIEM platform itself or a major technology vendor such as the Cisco CCNA Cyber Ops program, because that course material targets the actual products in use.

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



3. **On the Job**: Internal training relevant to the position and team, internally developed and delivered.
4. **Success and Failure**: there's nothing quite like "getting it right" and stopping the bad guy or missing the bad guy and finding out afterwards.
5. **Cyber Range Operator training**: intensive exercises offered on a dedicated large-scale lab with a focus on hands on skill development.
6. And many more.

Your training program should include a mix of items from the items above, with the goals of ensuring that your analysts develop all of the skills listed in the next section. As new staff are onboarded onto the SOC team, you should follow a well-defined structure in order to ensure that they will have the best possible opportunity to succeed. Below is a sample onboarding model for a SOC analyst 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 particular level after four to five weeks through an onboarding program.

Week	SOC Analyst Orientation
1	• Organization onboarding • Operator Level product specific orientation (usually 2d) • Side by side observation by new person with a current analyst • New staff reviews a "good" and "bad" write up (report) of each major alarm type that they will work at their level (OTJ)
2	• Side by side alarm review and analysis – new staff works through alerts and is partnered with senior staff • By the end of the week, new staff should be able to handle several alarm types on their own
3	• New staff is introduced to more complex case types by reviewing "good/bad" reports, starts handling more of them • More advanced product training focused on system health, uptime, and data flow monitoring • Schedule time w/ each next level analyst/case lead to expand knowledge of various SOC areas
4	• Shift and responsibility rotation – work various shifts • Longitudinal "weekly" report contribution
5	• Skills acquisition testing, which should consist of scenarios that support assessing how much the analyst has learned

SOC Analyst Skills

Historically, successful IR and SOC people need to have a diverse IT background, should have a few years in the IT game, do require continual training, and have a variety of skills in order to handle the breadth of cases a SOC will face. In particular, an analyst needs to understand how to "connect the dots". Connecting the dots is a skill developed over time. Analysts can be educated on understanding a given data source, but must acquire the skill to understand one event in context of another over time.

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

Today, in order to address the skills gap, find people who are: naturally curious, can think abstractly, often took things apart and put them back to together during their formative years, have a ***strong attention to detail***, can "connect the dots", and lastly who can perform research.

Below is an Analyst Skill Development Recipe which came from surveying over 30 SOC and Security Managers during Q1/2017, and then refined as I implemented that advice. SOC Analysts need to understand:

1. **The "Attack" process and phases:** Recon, Scan, Initial compromise, Establish Persistence, Command and Control, Lateral Movement, Target Attainment, Act on Objectives, Exfiltration, Covering Tracks, Leave without Trace. The MITRE ATT&CK framework is invaluable learning tool in this space as well as the Cyber Kill Chain as described on page 184.
2. **Ethics:** Ethical behavior means that they work within their limits, ask for assistance, do not overstate conclusions, never fabricate an opinion on weak facts, keep confidential all of the data they use, and other professional responsibilities.
3. **Organization specific data familiarity:** Familiarity with your organizations source data (event types and fields) so analysts can 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 into a given system, engage the EDIS process: the senior level analyst must prepare an overview of the data for the remainder of the team.
4. **System:** Technical system access control capabilities and how a technical control can be applied.
5. **Firewall:** Firewall Principles such as log types and what the log row records; rule attributes; actions such as block, allow, reset, drop silently; interface meaning as it relates to flow; SNAT/DNAT, byte sent/received; and understanding security zones.
6. **Security zones:** In particular, security zones are specific to each organization. One can infer the meaning of a zone, but should not because a zone term may not be consistently applied by the various admins or engineers. As firewall rule sets are built, the firewall team must document what they mean by a given zone and their underlying assumption about traffic flow. Active Directory admins need to do the same thing for organizational unit definitions. In this way a firewall zone named "eComDMZ" can be connected to an ADOU named "WebSalesDMZ" when the definitions state "severs used to support web sites used for ecommerce".
7. **PCAP collection and analysis:** Network data collection and the use of tcpdump as the data collection to and then WireShark/TShark as the analysis tool.
8. **Forensics:** Forensic principles
    a. Gathering data on system following the order of volatility
    b. Establishing and maintaining Chain of Custody

   c. Documenting actions taken, data extracted, time shift, and timeline reconstruction
   d. Locard's Exchange Principle
9. **Hardware:** Hardware platform: PC, Server, Router, Switch, TAP, disk boot, MBR, and identifying volumes.
10. **Incident Handling:** Incident Handling process: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. Once someone understands these points, they need to be able to write an executive summary of a case.
11. **Investigation:** Investigative processes as described in the Alarm Investigation chapter.
12. **NAT Issues:** NAT translation and the complexity of locating the true source IP.
13. **Network Application Protocols:** DNS, SSH, HTTP/S, SMB, NTP, DHCP, FTP, SSL, SMTP, POP3, IMAP, and SIP.
14. **Network Protocol understanding:** ICMP, QUIC[16], IP, TCP, UDP, GRE, BGP, SCTP, and ARP.
15. **NIDS:** Network Intrusion Detection System (NIDS), ruleset categories and how to read events generated by an IDS using rulesets such as the Talos Snort rule set (formerly known as SourceFire VRT) and the Emerging Threats Pro feed.
16. **Application Portfolio:** Organization Application inventory, application behavior, and PoC's
17. **Policy:** Organization Policy/Procedure *and the ability to politely articulate and enforce PnP.*
18. **OS proficiency:** Windows and Linux: services, startup, log files, network connections, registry, and process identification, and how rights and permissions are applied to both Windows and Linux.
19. **Scripting:** Programming, focused on admin and data reduction scripting skills using PowerShell, Python, shell, Perl, or a similar utility focused scripting language
20. **Report writing**[17], including spelling, grammar, Word usage, and the ability to write a summary that answers "Whom, What, When, How, and Where", and make a reasonable assessment of "Why".
21. **HTTP, HTTPS, and Web Browsers:** Understanding how the most dangerous application, a browser, works is critical to analyst success. The application itself more commonly attacked than the perimeter, because it is more successful. As evidence, consider the rise in phishing tools and attention on socially engineering the end user to entice them to click. The ability, and

[16] Quick UDP Internet Connections, a protocol promoted by Google that supports multiplexed connections between endpoints and is supported by almost 1% of webservers (August 2018).
[17] Chris Sanders has the only course the author knows of on this topic.

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, the analysts need to always attempt to find the user, user agent, or system that generated the traffic.
    a. Understand user agents, HTTP status codes, URLs, and browser redirection.
    b. Research the URL data revealed through the proxy.
    c. Discretion when researching user browsing habits.
22. **OWASP Top 10:** Following on the need to understand the browser is the need to understand how an Internet facing application is attacked.
23. **All of the skills measured** in the CompTIA Security+ certification and many of the Network+ skill areas so that a SOC analyst is terminology compliant.

SOC Analyst Traits

SOC Analysts need to have specific personality traits in order to be effective at their jobs, as listed out below.

1. **Natural curiosity:** SOC Analysts will be faced 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 it worked when they were done.
2. **Organizational skills:** An ability to perform "rapid research" that allows them to separate wheat from chaff, and in particular the ability to determine if an alarm is likely to be real, based on the alarm and data surrounding the alarm. 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?
3. **Abstract thinking:** In particular, the ability to read intrusion events like alerts from a Snort/Suricata system and a Palo Alto firewall and correlate them in near real time to other data sources, visualizing activity patterns in their minds.
4. **Contextualize large data sets:** The ability to reduce a larger volume of data down into information, in context. More specifically, when faced with an alarm *right now*, determine if that alarm is relevant to a larger context.
5. **Communication:** Perform data summarization and commonality detection such that a group of original facts can produce information, and then articulate how the information proves or disproves a hypothesis.
6. **Ego:** A small ego, but not small enough that they don't take pride in their work.

SOC Roles

There are several roles that need to be staffed in a security operations center. Depending on the size, scope, and budget, a SOC may have more roles defined. Roles will also have defined interactions with other key roles, because a SIEM platform and NSM platform *actually* have user community – the SOC analysts – who are supported by the engineering, architecture, and process support side of the SOC team.

Table 4 SOC Roles and Functions

Role	Duties and Responsibilities
Analysts	As defined in SOC Layered Operating Models on page 52, this role is the primary SIEM, NSM, and log management system user. Analysts may function as incident handlers or may directly support the CSIRT function.
SOC Developer	There will be a need to write "utility" software, such as a log parser, a monitor script, an add on component for a dashboard, or a lookup tool. Many SOC's would benefit from being able to utilize development talent and skills.
Shift Lead	A senior analyst who ensures that all shift responsibilities are met and is 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	Engineers install, maintain, and upgrade the SIEM platform and its operating systems. They also implement use cases, provide troubleshooting, and configure device support. Advanced SIEM engineers can also build parsers for unsupported devices, which usually involves knowledge of regular expression parsing and SQL queries. Lastly, 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.
Security Process Engineer/ Analyst	A process engineer has a security focused system analysis role. They perform analysis to develop, define, and test use cases, train the analysts on supporting use cases, and help to develop reports.
SOC Manager	This is a day to day managerial role for the SOC team. The primary customer is the CISO. The SOC manager implements strategy and process defined by the CISO.
CISO	The CISO 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. CISO's

Role	Duties and Responsibilities
	must understand business requirements and expectations (without this understanding, they are likely to fail).

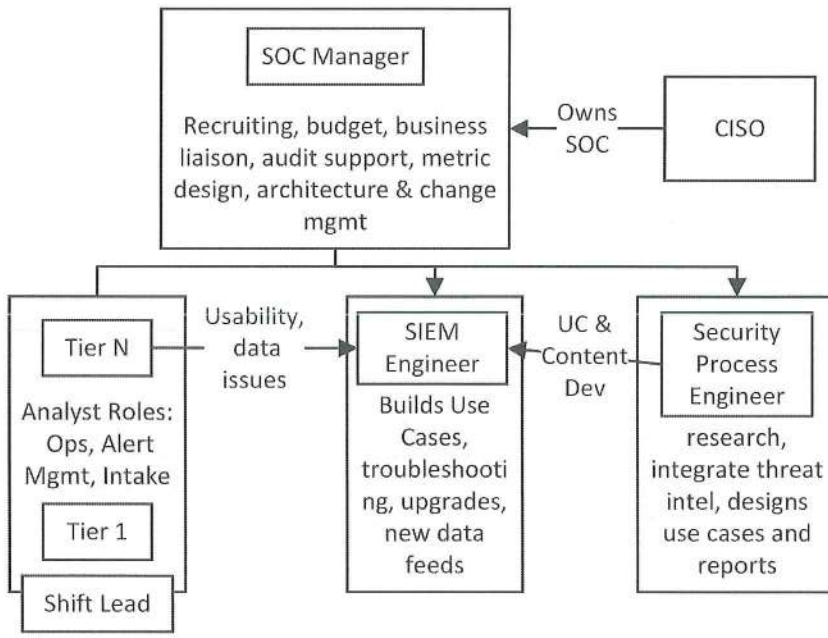


Figure 1 SOC Roles and Relationships

SOC Layered Operating Models

A small security team of just a few people rarely follows a hierarchy. The team just work each other to monitor the environment and respond to alarms. As a Security Operations Center matures and grows in size and breadth, it usually develops into some form of a tiered analyst job structure that reflects staff skill base, alarm conditions that are worked, pay level, SOC service areas they support, and training commensurate with their job level.

This section describes a two and three-layer approach, with the word "layer" being used as a placeholder. Job titles may be Tier 1 Analyst, Tier 2, and Tier 3 Senior Analyst, or titles like 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 their ability to respond to alarm and event conditions which in turn ties to pay and responsibility. Layers models also provide a way to measure progress and provide job advancement by title and pay commensurate with the skill base.

Regardless of the model and title, there is almost always a formal management layer for the Security Operations team.

What is essential to whatever stratification model your SOC uses is that each analyst must be willing to ask for help if they find themselves outside of their depth when working an alarm or handling a ticket.

Two Layer Model

In a two-layer model, the front-line staff is most often staffed with seasoned security analysts while the second layer is staffed by engineer level staff who handle more complex cases or require more complex analysis. Above these two operational layers the SOC management. Each of these levels will have end user and management interactions at some point.

Table 5 SOC Two Layer Model Roles and Responsibilities

Layer	Example Duties and Responsibilities
1	• Real time event and alarm monitoring that follows a standard operating procedure for a wide variety of alarms • Phone intake for initial case support (phone, email, webform) • Run reports • Monitor SIEM system health, data feed checks, and keep an eye on the system(s) as a whole • Gather key data, feed many case types to the ServiceDesk, handle and process more straightforward alarm conditions. and escalate more difficult or complex cases to the next layer *after* they have collected some initial data • Close some cases based on well-defined criteria. • Certain analysts may be asked to perform "longitudinal analysis".
2	• In depth analysis of alerts and events escalated by Tier 1 • Perform complex analysis and research on alerts and events, such 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 • Tendency to specialize for certain alarm types, systems, or areas of the business • Synthesize vulnerability data • Performs daily or weekly threat hunting activities

Three Layer Model

In a three-layer model, responsibilities are more stratified as the organization attempts to respond to increased need for more coverage and balance that with staff costs and skills or a need to provide more coverage like adding in overnight and weekends. Therefore, a higher separation appears to enable junior analysts to be effective for the SOC.

Table 6 SOC Three Layer Model

Layer	Example Duties and Responsibilities
1	• Real time alarm monitoring, using a well-defined SoP or other standardized Operational Guidance document • Phone intake for initial case support (phone, email, webform) • Monitor system health and data feeds • Gather key data, escalate most cases to the next layer if they can't resolve the case quickly or close some cases based on well-defined criteria.
2	• Handle escalated alarms • In depth analysis of alerts with a determination whether to forward or not to Tier 3 • Review the inventory of daily event types received by day, looking for patterns that may indicate a security issue • Take a short-term view of event patterns in support of alerts (a higher degree of alarm awareness) • Support incident management data requests • Tendency to specialize for certain alarm types, systems, or areas of the business • Perform data feed monitoring and basic system daily checks • Synthesize vulnerability data • Perform daily or weekly threat hunting
3	• In depth analysis of incidents and cases, individually, by day, or longitudinally • Manages incidents, may function as an incident coordinator/commander, or may be a second in command • Take a longitudinal view of event patterns in support of alerts and areas of the business or a client • Has operational and longitudinal responsibility for specific areas of the business, or a client area/business unit • Performs specific daily threat hunting activities

Layer	Example Duties and Responsibilities
	• May perform memory or dead disk forensics, or may 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. SOC's usually start when management hires someone either deliberately or as a 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 SOC team is formed. Muddled in these organic steps is a focus on "buying and implementing a SIEM", which gets funded as a project[18]. 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 types of 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 described on page 17 operate at different levels. The 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 the Carnegie Mellon Capability Maturity Model Integration (CMMI). There are five maturity levels across in the CMMI, with each level representing an evolutionary plateau in terms of process improvement. Normally, CMMI is applied holistically in an organization across up to twenty-four process areas, so using it needs to be adapted and focused for a Security Operations organization. *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.

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

Table 7 CMMI Five Level Maturity Model

Name	Characteristics[19]
1: Initial	Processes are ad hoc, chaotic; success depends on heroics as opposed to following a defined process. Services often work but exceed budget, time, schedule. Success is not repeatable, and heroic action frequently saves the day.
2: Managed	Workgroups (the SOC) define processes, create work plans, monitor their processes, and meet a set of "contract requirements" with its customers (the business). Configuration management is implemented with quality assurance.
3: Defined	Defined processes for managing work are used, well understood, services, procedures, tools in place. Sound project management is implemented into each process set. Difference in L2 and L3: process/procedure can be quite different in each instance of the process, whereas L3 procedures are tailored for the workgroup.
4: Quantitatively Managed	There are quantitative objectives for quality and process performance, which are used to manage processes. Measurement statistics are collected on specific subprocesses. Difference in L3 and L4 is focused on predictability.
5: Optimizing	Focused on continuous improvement.

There is also a formalized model available to assist in assessing a SOC's maturity level. The SOC-CMM was created by Rob van Os, MSc. as part of his Master's thesis work[20]. Rob's methodology, tooling, and spreadsheet is setup to evaluate the SOC across five dimensions and twenty-five aspects. The five domains are Business, People, and Process, which are evaluated for maturity. The other two are Technology and Services, are evaluated for both maturity and capability. You can use this tool to evaluate your current state and develop a road map of

[19] 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. 1.3 URL: https://resources.sei.cmu.edu/asset_files/Webinar/2010_018_101_22253.pdf (8/16/18)

[20] The SOC-CMM is available at Rob's website: https://www.SOC-cmm.com/

how to mature your SOC. Rob's site is https://www.SOC-cmm.com/. Rob's tools are gaining quite a bit of traction.

Measuring Data Source Integration Maturity Levels

To apply the CM CMMI, the SOC can consider how it integrates a data source into the SIEM and its operational process as a process that must be well managed and repeatable. As a primary requirement, in order 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 (L1) is characterized by:

- Getting data into the SIEM can be very ad hoc. A well-meaning system custodian sets up a syslog feed and lets SOC know that new data is arriving.
- Someone in SOC works with system custodian to make sure that the systems data can be gathered into the SIEM.
- Someone else in SOC works with the SIEM vendor to make sure that data is parsed.
- 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 (L2) is characterized by:

- Data input goes through a consistent process. Source system instrumentation is well defined, if there is a need for custom by the vendor it is planned, and a synthetic transaction is setup (see p. 193).
- The source system is fully exercised so that all of the security and operationally relevant events are logged.
- Once the data arrives, 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 utilize all of what the source system can provide.

Data Input as a Defined Process Characteristics:

- Organizational policy and process artifices require that data source input is integrated into the organization. For example, policy requires logging to the SIEM and/or log management platform is required, and a check to see this is setup properly is integrated into the Configuration Management process.

- The SOC manager ensure that all users have completed onboarding for each data source and confirms that there is equivalent understanding how to use the data.

Measuring Alarm Processing Management Maturity Levels

To apply the CM CMMI, 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 as it is the result of processing data input into the SIEM and then acted on by the SOC analyst. In order to realize this service, all aspects of the SIEM and SOC are part of the service delivery: data input, parsing, health monitoring, alerts raised in response to events, analysis of alerts to validate that they are true positive, severity is responded to, based on the impact to the organization, incident response is activated to the degree needed, and the resulting incident is tracked through to resolution.

That is quite a mouthful! Below 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 read, review, and respond to alerts as rapidly and completely as possible.

Alarm Processing as an Initial (L1) Process has these characteristics:

- SOC Analysts review alerts as they arrive, with some notion of priority and impact. High priority alerts which are likely to be true positives receive attention such as reporting for resolution, data owner and system custodian notification, or routed to Tier 2 for further investigation.
- SOC analysts may or may not be consistent with each other in the decision-making process on 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 such that critical alerts always receive some form of treatment as a "stop gap".

Alarm Processing as a Managed (L2) Process Characteristics:

- Event data is reviewed in order to find other alarm conditions in order to improve detection capabilities.
- Alerts are tuned so that false positives can be minimized. This part of the process should influence the source system in a feedback loop.
- Alerts are treated consistently by all members of the SOC, with most of them mapped into incident playbooks or other supporting processes.

- Alerts are resolved, routed, or investigated as they arrive within a certain defined time frame – say 30 minutes for critical, one hour for medium, and within 4 hours for low.
- 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 improves the alarm management process.

Alarm Processing as a Defined (L3) Process 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 is fully trained on the data source and the alarm conditions it brings to light.
- The SIEM platform is augmented with orchestration and automation in order 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 Check List

Start with this list for your SOC turnover. Cover turnover at the beginning of each shift. Each shift needs to summarize running alerts, incidents, and follow up items for the following shift as these are critical for situational awareness. Review this list and define what analysts do per shift, develop an organization specific model, and then update the end of shift turn over. 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 point in your environment.

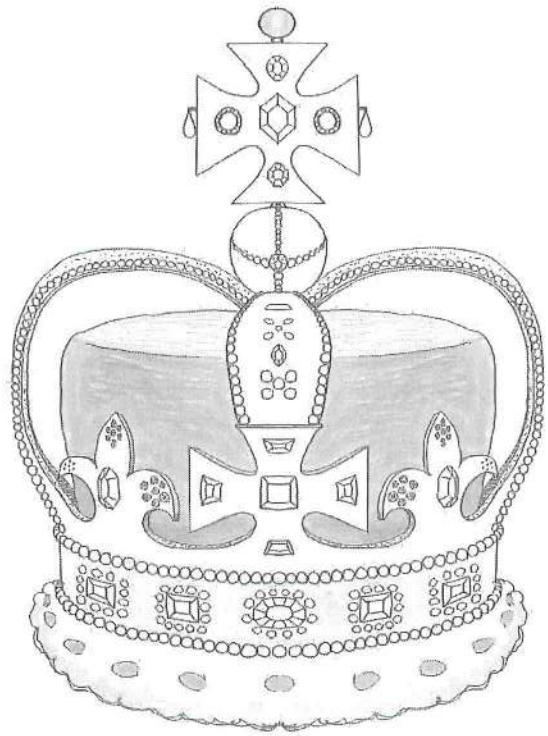
1. Turnover from prior shift, which should include:
    a. Key events
    b. Status for ongoing incidents
    c. Staff outages
    d. Major data issues
    e. Any system stability issues
    f. Relevant communication topics
2. On the normal maintenance day, review the scheduled changes so that the SOC won't over react for alarm conditions that can be explained by a change running through change management.

3. Quick review of yesterday's alerts so that any recurrence or repeat conditions which may reveal a repeat event are easily identified.
4. Review the daily briefing, which should cover topics like:
    a. New alerts instrumented in the system
    b. Data sources leaving the system
    c. New data sources coming into the system
    d. OTJ training gaps

Security Monitoring Use Cases by Data Source

As you read through this chapter you should:

1. Think about common scenarios for attacks and adversaries who target your organization. To aide in this process, this chapter introduces a scenario, an attacker plan, and a defense plan.
2. Ensure you know how to get the data to support the use case points.
3. Gather vendor and product documentation so analysts can easily look up event detail as they use event data when researching an alarm.
4. Determine the risk that the use case is designed to address for the organization.
5. Decide and document how the analyst will respond to the use case.
6. Review the enterprise data inventory survey in the SOC planning chapter.



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).

The Scenario

Every organization has a wide variety of types of data that they can collect. As the SOC considers collecting data, evaluate that data against the actual needs of the organization, how that data will be used to detect an adverse security condition, how close the data is to the end user focused application, and how user attributable that data is. To illustrate these concepts, this section will walk through an example of systematic intellectual property theft conducted against VictimCo.

The Setup

First and foremost, VictimCo is in possession of valuable data. After surveying its business environment, value chain, and identifying its valuable data, VictimCo makes several key determinations:

1. There are well-known users with a public presence, ripe for targeting. Of this user population, a small number of them frequently travel and speak at conferences. There is also an active traveling business development group.
2. Critical Intellectual Property is spread out on the network, meaning it is not consolidated. Further analysis here indicates that there is a mix of storage platforms. The list includes SAN, NAS, OS based shares, and web-based collaboration sites. A significant portion of IP is "Trade Secret" and not necessarily well marked.
3. Even though there is a web proxy, it is not configured with an overly restrictive policy.
4. More than 30% of the user population have elevated rights on their workstations, which is implemented by adding the user's primary login account to the "Administrators" group.
5. The firewall has a default by deny stance and logging is pretty good.
6. There are easy methods of data egress because outbound FTP is not restricted from the desktop and websites that allow users to paste data and then get a URL to that pasted data are not blocked.

The Attackers Plan to Find Data and Exfiltrate

VictimCo's team develops an outline how an attack would progress. An attacker who is after valuable intellectual property will need to build and execute a plan similar to the one outlined below.

1. Reconnaissance: Website scanning, performing Internet based "doxing" type research, and gathering candidate email addresses for a targeted phishing campaign.
2. Weaponization: An attacker would use a variety of tools to craft malicious payloads, such as PDF documents with droppers, macro enabled Office documents, or a website with malicious content.
3. Delivery: The most likely chance for success for an attacker is to send email with a convincing pretext. The pretext is written so the message appears to come from a professional colleague or someone with interest in working with them. For example, a phish campaign would indicate that they read a book or article by the recipient, express an interest for some collaboration, or suggest that the recipient may be interested in other articles, books, or sites. The links for these sites would include malicious sites. Often malicious sites would contain a link to a login site that looks like it belongs to the recipient's organization – but in fact it didn't – with the goal of capturing real credentials.

4. Exploitation, Part One: Once credentials were captured, an attacker would use them on any exposed site or may even attempt to login to a common VPN service.
5. Exploitation, Part Two: While gathering the username pattern from email addresses, an attacker may attempt to brute force or "password spray" any potential account against any exposed web interface that VictomCo has.
6. Installation (Remote Access): Once the attackers had a minimal foothold they were capable of establishing persistence on several users' workstations, gather higher level credentials, move laterally within the network, and gain access to shares and sites with trade secret data.
7. Command and Control: With a foothold, the ability to run services and enable persistence, proxy service aware command and control agents that communicate out to C2 nodes can be installed on target systems.
8. Act on Objectives: Once sensitive data is located, it can be exfiltrated in numerous ways. Examples include uploading to file share sites, FTP on non-standard ports, email, direct transfer to using netcat are but a few.

The Defense Plan

The primary underpinning of the defense plan is to prioritize capturing user attributable data that matches likely attack vectors. Gone are the days of collecting millions of firewall records in hopes of analyzing that data. Modern data collection should be user attributable, be as close the application as possible, and provide execution context *because the modern attacker must live off the land,* which means they need to change the OS and use scripting languages present on the system. To quote Alissa Torres[21] from SANS, "Malware Can Hide, But It Must Run." Prevention technologies are certainly valuable tools and they do protect the network. However, we must assume that they will fail in the digital arms race so we must instrument for post exploitation detection. Furthermore, no advanced preventative technology can counteract a user who knowingly, willingly, clicks on a link in an email and ignores all the warnings.

Instrumentation	SIEM and Security Architecture
Use the dnstwist algorithm to develop an inventory of domains that look like VictimCo. These domains can be checked historically for site visits, email communication, and blocked in the proxy with a redirect. Twist domains can also be DNS sinkholed.	A search can be run to see if any system that uses a FQDN from the dnstwist list is hit. Examples – user made a DNS request, attempted to visit a website, sent or received email to one of these domains.

[21] https://digital-forensics.sans.org/blog/2016/10/29/malware-can-hide-but-it-must-run

Instrumentation	SIEM and Security Architecture
Confirm that email activity is logged and recorded at the SMTP conversational level, and configures 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	Should a user succumb to a phish mail, the SIEM/email system can be queried to see if other users also received the same mail and then the email can be removed from their inboxes.
Implement sysmon on high value user's workstations in order to collect process invocation (parent, child, path, command line). Registry Change – Sysmon Event ID 13	Create alerting that will fire if an office application or the email application opens up a scripting tool or a command prompt, which is a high value alarm and indicator that a user clicked through and opened a malicious email payload.
Update Windows domain auditing to collect command line path from 4688 events (detailed Tracking). Collect same 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 which is provided by the 4688 event. The threat Hunt 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 then monitor by exception.
Update all Internet facing site to minimally log access attempts (success and failure) to the SIEM. (custom development in many cases).	Access attempts can enable several account management use cases – a new account successfully logged on, a brute force was successful, brute forcing is being attempted, and a spray attack (try one password for all possible accounts) can also be attempted. If the source machine is on the inside, it can be thoroughly investigated for C2.
Always on VPN (enhance security architecture)	Traveling users can be redirected back into the company network so that security controls and monitoring is actually effective.

Instrumentation	SIEM and Security Architecture
Update the Windows Workstation Presence Indicators as a second phase approach, once it is proven to work and the issues are resolved. New Service: Event ID 7045 Scheduled Task: Event ID 4698 Local Group Changes: 4731,4732,4733,4734	Windows has a native capability to centrally collect audit logs. At a minimum, several event types need to be collected which are known as "presence indicators": login, screen lock, reboot, screen unlock. After that: local group management. Finally, service state changes. These event types can be used to detect when workstations are used outside of normal business hours and for unauthorized changes, new accounts, and service installation – all underpinnings of persistence and lateral traversal.
Persistence detection: Autoruns (daily, for all workstations and servers).	An advanced detection technique is to consume the output of "autorunsc" into the SIEM, sort the data using Long Tail Analysis (or stacking) in order to detect any new persistence entries.

Once operating system data is collected, then focus on valuable network level trace data. Network level instrumentation should focus 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 to jointly define use cases. This is a process that, if done well, will pay huge dividends down the road.

First, let's level set on the phrase "use case"[22]. 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 book on page 133.

[22] Adapted from the Wikipedia article on Use Cases

For a SIEM and a SOC team system, building a use case has several requirements based on the definition from above:

1. You must be able to describe the *observed* condition that is relevant to your *security posture and realizes the use case* (the **objective**).
2. The system providing data to the SIEM must be capable of *actually auditing* the desired behavior (observe the action by person or system **interaction**).
3. The system must provide the event record *with sufficient fidelity* to the SIEM (as defined under Log Record Data on page 223.) (define the **action**).
4. The SIEM must be able to process and present the event at the necessary level of granularity for the Sec Ops function (to measure or observe the **interaction** for the **actor**).

Briefly, the security focused use case development process is:

1. Understand how the use case maps to or supports a *Business Capability or a Requirement*: Uptime, brand protection, dozens of compliance requirements, fraud prevention/detection, IP theft, or minimize disruptions.
2. Design the question that the use case should answer. How would the attacker gain needed access, cause damage, exfiltrate data, or what accounts would they need to use?
3. Determine and test the data sources and the data elements that provide the visibility needed to answer the question.
4. Evaluate the data by establishing normal baselines and other analysis dimensions. Characteristics to understand include volume, peaks/lulls, outliers, averages, frequencies of types of data or specific elements, duration of normal behavior, and how do you find something "new".
5. Establish the SOC guidance and processes that will be used to filter out false positives from the baseline data. Ensure that guidance is developed to support identifying malicious use or operational issues.
6. Various techniques exist to visualize data such as bar chart analysis, graph analysis, simple timeline presentation, and other summarizations.
7. Many data sources naturally lend to building correlation rules where one data source compliments 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 the use case. Again, as in the rest of the content of BTHb:SOCTH, most of these use cases are based on experience and were implemented to one degree or another. This inventory of use cases is not exhaustive, and should not be considered "the definitive list". Hopefully as you look through these they will

assist you in implementing your own use cases based on your data sources and value chain.

Example: Web Presence Attack

Many organizations operate a web presence. According to the January 2018 Netcraft survey, there were 213,053,157 unique domain names, 7,228,005 web-facing computers on the Internet[23], all hosting 1,805,260,010 sites.

These systems can be as basic as an externally hosted public information sharing site to a full-fledged eCommerce and customer engagement platform. The illustration below 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. At the business conceptual and contextual layer, 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 a component. For example, a site may have a product catalog, a blog site, and a contact page, each of which can use its own subcomponents. On the private, or 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 some hybrid applications, such as a forum which requires some form of registration that may be separate from the customer engagement account database.

Cutting across all of these technologies is an exploitable attack surface emerges. Some of the best guidance to understand this attack surface is the consensus driven OWASP Top 10 Most Critical Web Application Security Risks list, which is updated every few years. SOC and IT can leverage this list to determine what type of monitoring may reveal an exploit, what type of security monitoring capability needs to be deployed to protect the web presence. From a Cyber Kill Chain perspective, the attacker needs to perform reconnaissance against the site and its numerous components. Not shown is the weaponization step, as attackers develop attack capabilities using 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 their own persistence mechanism such as a backdoor shell. As mentioned elsewhere, this scenario shows why it is necessary to keep informed about updates to Metasploit.

[23] https://news.netcraft.com/archives/2018/01/19/january-2018-web-server-survey.html

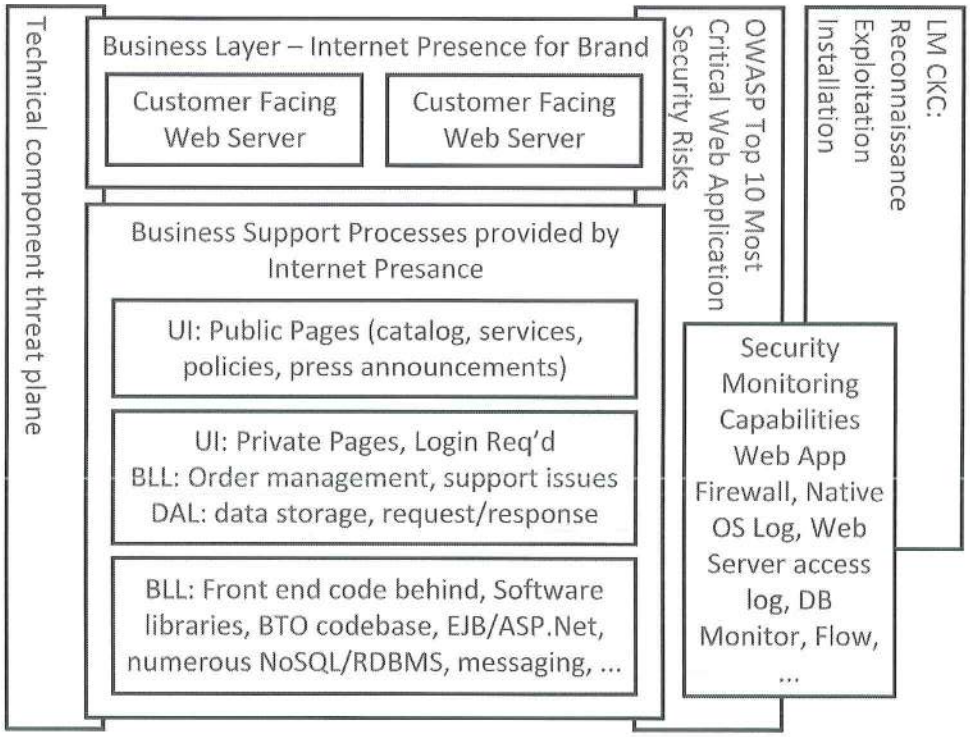


Figure 2 Web Presence Attack Components and Attack Surface

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 visit a site, download a file, or click a link in an email and then ignore security warnings. Collectively, these attacks all fall under the umbrella term "social engineering" which are leveraged through email, email attachments, watering hole attacks, and manipulating text stored in forum posts.

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 that could be a malicious PDF file or an office application with a macro that has malicious script code in it. Once the payload is opened, or successfully delivered *and the user interacts with the payload*, the attacker can begin to leverage a wide variety of techniques found in the MITRE ATT&CK matrix.

The elements illustrated in the next figure along with references to the Cyber Kill Chain and the MITRE ATT&CK Framework.

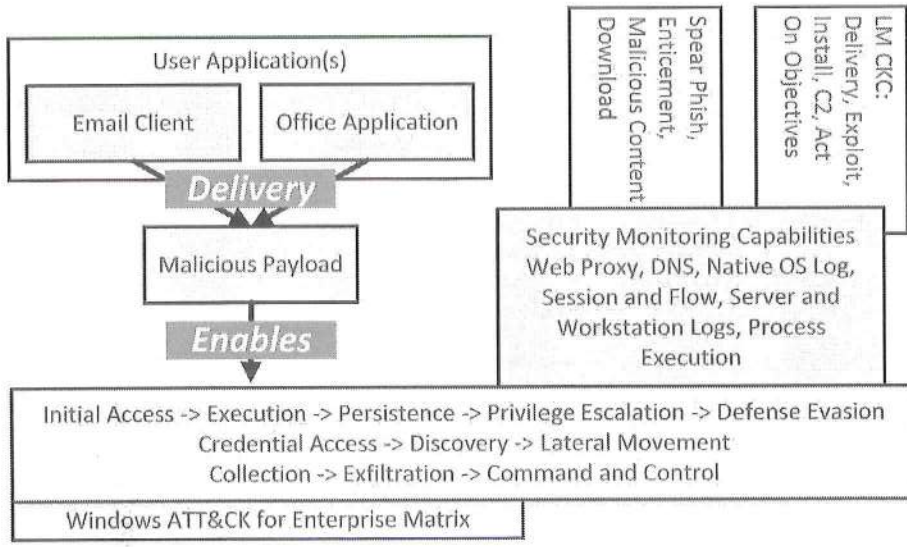


Figure 3 Example: End User Payload Focused Attack

One of the first actions is to establish persistence so that the attacker can return. After that, attackers vary their actions. They may attempt to crack passwords, gather hashes from the SAM database or from an inbound authentication token on a connection, search the network for an opportunity, or if they are lucky, pillage the system they compromised because it was just the right one. In any event, an attacker has numerous opportunities that can be used to take advantage of Windows and Microsoft networks. The older the operating system and the less out of date it is where they establish the first foothold, the better (for them, not the blue team).

Organizational Considerations for Use Case Development

Before we dig into dozens of data source specific technical use cases, security operations really need to understand several key factors and answer several questions. This chapter will lay out several models of the modern attack landscape so that security operations can determine where to engage, what logging sources will protect the business, and how to instrument data collection.

Questions to Answer:

1. Is there a security minded business analyst available to help develop security focused use cases?
2. What is the business operating environment?
3. What is the application and data resources the resources the business depends on to achieve its mission objectives?

4. Is there logging enabled for the applications and the platforms they depend on?
5. Who is the application and platform owners and custodians SOC will need to engage with?
6. What will the SOC do in response to an alarm?
7. How will you maintain your use case library?
8. At what point does a use case produce change control items?

"Top Ten" Security Operations Use Cases

This section is here to provide a starting place for your SIEM and SOC efforts. When formulating your organizations top ten use cases, consult standards like the Australian Signals Directorate Strategies to Mitigate Cyber Security Incidents[24]. The ASDSMCSI is designed to assist organizations to help security professionals to mitigate incidents. There are various other sources that can be consulted. Below is a list of the top ten security use cases that a SOC team should implement as early as possible.

1. Privileged Entity Monitoring.
2. Brute force Authentication failures.
3. Authentication Anomalies
    a. Service Accounts used for interactive logon.
    b. Service Accounts used from non-authorized source systems.
    c. User logon locally (on LAN) within a short window of a VPN logon.
    d. User logon more than an hour before or after normal work periods.
    e. Interactive User authentication from multiple source systems.
    f. Shared account usage (which should not be confused with accepting the idea that using shared accounts is acceptable).
    g. Default account usage (same caution).
4. Session Anomalies. There are numerous examples in this area.
    a. The typical end user should have a session beginning and ending with ten (or less) hours from each other.
    b. Significant profile change in web browsing habits.
    c. Spike in outbound firewall denies.
    d. Workstation network to workstation network communication.
    e. What is a reasonable clipping level for sessions?
5. Account Anomalies
    a. Accounts used before the user's start date.
    b. Accounts used after the user's end date.
6. Data Exfiltration indicators

[24] https://acsc.gov.au/infosec/mitigationstrategies.htm (8/18/2018)

   a. HTTP(S) Send/Receive mismatch. Data received from a site is often many times data sent to a site, by byte volume, as most of the time the browser is downloading a file and rendering it for the user.
   b. 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.
   c. Use of file storage sites (Dropbox, Box, Microsoft OneDrive, GoogleDrive, SugarSync, Leapfile, etc).
   d. Use of websites that allow for 'easy information sharing or text storage', where users can cut/paste information in an unregulated manner.
7. Signature Matches to known Vulnerability Scan Results.
8. Any excessive 'service failures', such as A/V agents that repeatedly fail or backups that fail. Note that outage detection also provides operational value as well as security value.
9. Insider Threat Indications -
   a. Accessing "security research" sites.
   b. Use of USB drives.
   c. Authentication baseline violations.
   d. Authentication failures against file shares, applications, servers, internal SharePoint sites, etc.
10. Security Log Data failure conditions.

AntiSpam and Email Messaging

If there is one system people use *every day,* it must be their email system. Typical email is person to person or person to a small group, with a low ratio of email with attachments to those without. This is a good starting place to define "normal" for your organization.

Use cases based on email are easier to implement for locally hosted email systems than cloud email systems, because it is easier to get logs from on-premises systems. Free email systems are highly unlikely to provide meaningful data export and SIEM integration. Paid cloud email systems may not support SIEM integration or may only provide user activity through a downloadable report.

1. **Email with attachments that have spaces or multiple periods**: Attackers like to obscure their malicious content. Two methods are to add several spaces and include multiple extensions like ".docx.exe".
2. **Email burst or flood**: A rash of inbound email can easily be a phish, or some other campaign you may not want.

3. **Infected sent mail**: Internal users *sending* virus-laden messages internally or outbound. This condition indicates a failure in the local A/V client or malicious software on the system.
4. **Spam sent mail**: Users who sent email that is identified as spam is not normal. If the organization has an upline anti-spam system, the user's messages were likely blocked. It may be advisable to let the user know, *assuming* there isn't a further negative condition.
5. **Non-authorized systems sending mail**: The only systems that should be communicating to any one of the messaging TCP ports should be well known and understood (such as the internal messaging systems). Below are common messaging ports for email systems. For continuous monitoring, the system should create an alarm for traffic *outbound* on these ports from a *non-authorized source*. For Threat Hunting, a report that includes these ports and the senders should be developed and periodically reviewed.
    a. 25/TCP - Simple Message Transfer Protocol (SMTP)
    b. 110/TCP – Post Office Protocol (POP version 3). POP2 is on 109; but the likelihood of seeing this is minimal.
    c. 143/TCP - Internet Message Access Protocol (IMAP)
    d. 209/TCP & UDP - Quick Mail Transfer Protocol (QMTP)
    e. 220/TCP & UDP - Internet Message Access Protocol (IMAP), V 3
    f. 465/TCP - Authenticated SMTP over TLS/SSL (SMTPS)
    g. 587/TCP - e-mail message submission (SMTP)
    h. 993/TCP - Internet Message Access Protocol over TLS/SSL (IMAPS) – Apple systems use this port.
    i. 995/TCP - Post Office Protocol 3 over TLS/SSL (POP3S) – Apple systems use this port.
6. **Messaging IoCs**: Intelligence sources do list known email addresses as an IoC. Messages to and from these email addresses are suspicious. You cannot necessarily prevent a malicious user from *sending email inbound, but* you can monitor and communicate to a user who received it. Any traffic to an email address identified based on a feed from a threat intelligence source should definitely be investigational, whether it was blocked or not.
7. **Significant volume changes:** From a threat hunting perspective, the team should look for volume-based changes, such as a user who rarely sends attachments suddenly sends a large number of attachments to a competitor may indicate intellectual property theft or industrial espionage.
8. **Autoforwarding:** Users who send large amounts of data to their home email addresses may expose the organization to an unacceptable risk.
9. **Email with competitors:** Most, but not all, organizations do not routinely send a large portion of email with a competitor. This pattern is also subjective, but it may reveal an insider threat.

10. **Users generating numerous Non-Delivery Reports:** This condition may indicate their account is being used to probe for valid email addresses at a particular domain or some operational issue.
11. **Constant email transmission**: Users sending email every hour of the day, which may indicate something on their system is attempting to use an email capability for covert communications.

Email and Web: Interactions with Look a Like or Doppelganger Domains

Phishing scams sometimes use domains that look very close to your domain and are changed by single letter replacement, fuzzing, omitting a character that the mind will naturally fill in, or transposing two letters. For example, blueteamhandbook.com can easily be changed to b1ueteamhandbook.com, bluetaemhandbook.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 like "dnstwist" by Marcin Ulikowski[25]. 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 like dnstwist is to generate likely look-a-like or domains that a phishing campaign is likely to use against your organization. The dnstwist tool can also be used to check and see if a twisted domain is registered.

DNSTwist Domain Enabled Use Cases:

1. **Investigate Domain Name:** Pull out the domain name portion of email sent to/from the organization and compare it to a twist domain. If you have a match, then drop the email. Even better, do not accept email from a twist domain.
2. **Check twist against browsing:** Pull the FQDN portion of a domain name from web server logs, and alarm if a user visits a twisted domain name.
3. **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 that if a user attempts to visit a suspect site it will be directed to 127.0.0.1 (there is no place like home).

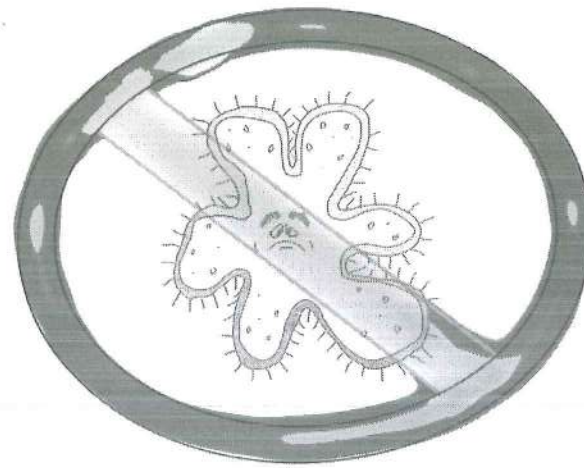
[25] https://github.com/elceef/dnstwist

Implementing this type of a DNS sinkhole not only protects your users, it also allows for an alarm condition.

Antivirus (A/V) Systems

Antivirus systems are bread and butter security component, even though traditional A/V has a tough time keeping up in the digital arms race. Nonetheless, maintaining a desktop security suite that includes A/V and other HIPS components should be part of your defense in depth architecture. There are several conditions should be monitored.

1. **A/V detection**: Finding and cleaning a piece of malware on a system, a USB drive, a CD, an email attachment, download from the web, etc. should be investigated. Just because the A/V system did its job doesn't mean that the PC is "clean." The file name provides significant clues to the issue. For example, if the A/V system removes a file with "drop" in the name it may mean that a "stage two" tool successfully got onto the system. The immediate web, email, and USB device history right before malware observance should be checked. A ".scr" file is most often a screen saver, which are often malicious. Realize that a screensaver, by its very nature, captures the username and password. *Remember, even though the A/V protected the user the malicious file got on the system somehow*, so work to answer that question and examining quarantined files in context.



2. **A/V Potentially Unwanted Program** (PUP) detection. The names of executables that are classified as PUP's may be early warning indicators, may reveal insider threats, uncover unauthorized software installation, or may provide a clue that an attacker tried one tool and then moved onto another when the first tool failed.
3. **Password Dumpers**: In particular, if a password dumper was seen and subsequently quarantined means that 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 names are PWDUMP#, cain.exe, wce.exe, mimikatz.exe, mimidogs.exe, and gsecdump.exe.

4. **A/V *agent status and update failure* conditions**. A/V agents should run successfully, should receive their updates, and not crash. Example events can range from an inability to scan a drive, inability to quarantine a file, an agent that will not start, the agent starts and crashes, it cannot retrieve updates, or is shutdown unexpectedly. A truly 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 updates to A/V. 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). If not, then you could have a machine out of compliance.
5. **Repeat A/V offenders or reinfections**: Users who routinely get an infection notice need more attention. This use case is longitudinal in nature, meaning that you want to know if a machine is being re-infected *over time, say 14 to 60* days. You can further differentiate this by end user logged on and virus name or family. Same user means the user is causing higher risk, different users may mean a kiosk or loaner is the victim PC, and the wider variety of malware means the greater likelihood that the compromise is more severe. Here, company policy should address what happens when users cause repeat 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, particular websites, or maybe even a particular network share.
6. **Multiple Infections in a short time frame**: A host that has multiple *different infections based on the binary and/or the virus name* may warrant more rapid attention than an asset with a single virus.
7. **Escalation based on Asset Value**: Assuming that your SOC team has valid intelligence on the value of an asset, an infection on a "critical" asset such as one that falls under the Payment Card Industry Data Security Standard (PCI DSS) warrants more rapid attention than an asset with a "medium" value.
8. **Anti-Virus Infection notifications in close proximity to spyware, malicious site, email attachment file open events**: The A/V system may be the indicator that gets your attention. Here is where an EDR application can be very informative and assist in a rapid MTTD. An analyst should be able to locate the file name and then determine how the user triggered the malware. Common methods are opening an attachment, inserting an infected USB drive, or visiting an infected website. If the source is email, then make an immediate search for other users who received the same email by subject line and sender, then seriously consider forcibly removing this email. In the case of a particular site, block access.

9. **A user with Elevated Access logging into 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 login to an infected machine, then at a minimum an automated notification advising them to change their password is in order.

Windows Defender has its own set of Event ID's. In many cases, Windows Defender also includes a suspicious file name, a unique threat ID, a severity rating, file path, error codes, and other useful details. These details are representative attributes to use when building dashboards, alarms, and reports.

Table 8 Windows Defender Application and Services Logs\Microsoft\Windows\Windows Defender\Operational and System Log

Event ID	Name
1000	An antimalware scan started
1001	An 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
3002	Real time protection failed

Application Whitelisting

The security posture of the endpoint is more important than ever before. These systems are designed to monitor executables running on a system when running in detection mode. When running in prevention mode, they will stop running an executable that doesn't match an approved policy. Application whitelisting can identify several adverse conditions. Further, since this capability records end user activity, it can be very useful in an employee investigation case.

1. **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 installed. The installation process

can be checked against an authorized change in the change management system to determine if it was authorized. Also, application whitelisting can help detect if the system's integrity was violated.

2. **Unauthorized drivers**: When a user inserts a USB device into a system, the OS will respond to that notification event and attempt to install the appropriate driver software. Removable storage devices can be an avenue for data exfiltration or can provide an avenue for malicious software entering the environment.
3. **First Observed Binary**: Introduction of a new binary in the environment may indicate an adverse condition. A user running several new binaries on their system may also be of note. An application whitelisting suite may not cleanly detect this specific item, unlike an EDR platform. Alternately sysmon can be deployed because it will generate a file hash as each process is invoked. Note that first observed binary analysis needs to be done by file hash, not file name because while the name may change, the hash will not. You may need to instrument some other method, such as manual review to realize this condition.

Windows has two facilities of note: AppLocker for Windows 7 and above, and Software Restriction Policies for Windows Vista and below. Both technologies record control binary usage. AppLocker events in the AppLocker event log and can be enabled using group policy. There are at least sixteen different events recorded. EventID's 8020 to 8027 are focused on package deployment issues, so they are not listed here.

Table 9 Windows AppLocker: Application and Services Logs\ Microsoft\ Windows\ AppLocker

Event ID	Level	Name
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. (EXE's).
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/MSI's).
8007	Error	FileName was not allowed to run (by policy).

Command and Control

There are several methods to detect command and control, aside from using an IoC list, an IDS rule, or a domain block list. Rather than duplicate information here, please refer to the Threat Hunting chapter for a discussion on command and control as described on page 178.

Data Loss Prevention (DLP)

DLP systems come in two broad types. Data in Motion systems are deployed to monitor traffic as it moves through a system and the network. Data movement can be email, FTP, copied to a USB or CD, saved off to cloud storage, copied to a network share, and therefore DLP needs to be integrated into an OS level service. For example, DLP software that analyzes email is plugged in to the messaging pipeline as a Message Transfer Agent (MTA). Data at Rest systems (or agents) find files of interest-based searching, such as a a file share or a web repository like SharePoint.

Once alerts from a DLP system reach a certain threshold they may need to be investigated by the proper *internal team*. Realize that 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 ten spreadsheets and the DLP system intercepts them and encrypts them for delivery to the recipient, the user may be aware this is normal and wanted that action to occur because they are trusting the DLP system.

Before going too far down the road with DLP, the SOC team will need to determine if there is a more applicable internal team who are a better business fit than the SOC (there should be...). For example, an insurance company likely has a Member Privacy team, or HR may perform investigations using the DLP system. One argument in favor is that by sending in alarm data to the SIEM, an end user activity report will have a more complete picture to present during an employee investigation. Further, since the DLP system identifies potential IP loss, an analyst can incorporate these alerts to gain a better picture of end user activity and notify the appropriate internal team if warranted.

Regardless, in motion DLP systems identify data exfiltration, whether intended or not. At rest DLP systems identify where valuable data resides. Today, attackers are interested in the data, because that's where the money is.

Domain Name Services (DNS)

Gathering DNS data presents a few data collection and data reduction challenges that you will need to work through. DNS detection requires detecting

name queries that are outside of the norm *and being able to detect the true source IP address if at all possible.* One issue that will prove to be difficult is a lack of internal reverse DNS lookups and stale DNS entries. If you can't reliably lookup an IP to a name, there will be a small impact on alarm processing time. The situation can be a bit worse when an IP comes back to multiple systems.

Collecting DNS: Collecting DNS from a DNS server can be problematic. For example, Windows DNS requires that you enable "debug logging", and then fully parse that data through a either a local or remote file reader process. Another problem with DNS is that most (90%+) of the traffic on the network are local queries. Local queries are normal. When considering how to collect DNS, focus on collecting internal to external queries, find where those queries are resolved, and collect data at that point using network extraction as the collection method. If there is a mirror port available at the perimeter, DNS query and responses will be logged *from the internal DNS server(s)*, as they are forwarding queries on behalf of the end user. If you collect DNS traffic via a mirror port on the same switch as the DNS server, you will collect a significant amount of normal query traffic for the internal network that will have low to no value for identifying attackers. There are at least 30 defined record types available for use, with the more common being A, CNAME, PTR, SPF, AAAA, NS, and MX. TXT records are seen, but in low volume. There are at least two well-known tools to collect DNS: PassiveDNS and Bro IDS.

DNS Monitoring Use Cases and Detection Patterns:

1. **Young (< 7d old) or recently registered domains (and thus, websites):** Malware is increasingly using sophisticated DNS lookups and query types to *signal* their command and control network. Attackers, and in particular Phishers, are using recently registered domains as spreader points. Techniques vary in exactly how recently created domains are used for an attack. 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, look very closely at the domain registration details. As an example, on 11/05/17, a check of domainpunch.com found 85,794 dot-com domains registered on the prior day. There are also several sites that provide lists of newly registered or expired domains, every day, usually for a charge. Examples include whoxy.com, whoisxmlapi.com, domainlists.io, domains-index.com, etc.
2. **Names not in the Top 1 Million List:** As described on page 112.
3. **Long, misshapen, or weird second level domain names**: Most second level names should be less than 24 characters. DNS names have a maximum of 255 characters in total. In practice, some analysis should be performed on DNS names that are 72 characters or longer. Really long names (>128

characters total) and continued query/response is most likely DNS tunneling or a DGA. You will need to establish these two thresholds for your environment.

4. **Hexadecimal Domain Names**: Domain names should be readable by people; after all, they are designed to help people locate resources. Hex is not usually human-readable[26]. Malware uses Hex values as beacons, may have Base32 encoded commands disguised as a name component, and usually require specific query and answer resource records set to specific values. Examples include FrameworkPOS, FeederBot, Morto, etc. Based32 encoding is used because the characters in a DNS name are effectively limited to 37 possible unique characters.
5. **TXT Records/Lookups**: DNS can provide freeform lookup information from a domain. Historically, the most common uses for TXT records are to help validate email delivery with Sender Policy Framework (SPF). Other normal uses are DomainKeys (DK) and DomainKeys Identified E-mail (DKIM). Query/response outside of these purposes is not normal, and further, illegible data in a TXT query or response is suspect. In contrast to names, the data returned from a TXT response can be Base64 encoded.
6. **SRV Records**: Server Resource Records are used to define a network location for a server that provides a specific service. They are actively queried by internal Windows systems within AD for many resource types. From the Internet, they are commonly used for communication-oriented services like SIP, email, some games, Session Traversal for NAT (which, in turn, support real-time audio/video/messaging), among other services. Again, you would want to establish a "normal" baseline and then be advised of "new" services queried. Also, a high volume of different queries to a particular DNS site where the request/response types are *not the same* type of lookups would not be normal.
7. **Private IP addresses returned**: Name server queries to Internet sites should rarely return private (RFC 1918) IP addresses. NetGear's "routerlogin.com" is one of the few examples of a private IP returned from your local DNS.
8. **TXT without A Records**: A direct query for a TXT record without a preceding A record lookup is not normal. Further, domain names that don't have A records that support their TXT and SRV records is also not normal.
9. **Long TXT record queries**: Assuming that you can monitor for query types, excessive queries or long queries returned from an Internet server may be used for command and control. Look for Base64 encoded data. TXT records are used for SPF, so they do occur. Tools known to use TXT records include dns2tcp or DNScapy.

---

[26] 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.

10. **Look-a-Like or fuzzed domains:** Review the section Email and Web: Interactions with Look a Like or Doppelganger Domains on page 73 when working through DNS use case development.
11. **DNS queries *not from* authorized servers**: An enterprise should only have a small number of internal DNS servers that can forward queries to servers on the Internet. *Any* DNS query outside of this boundary should be investigated, if for no other reason that ensuring the sender is properly configured in order to provide operational assurance.
12. **Volume and volume profile changes**: Establish a baseline profile for DNS traffic. These indicators can become alarm conditions once baselines are established. Examples are:
    a. Average queries per hour during working hours/off hours.
    b. First time use domain queries (new domain name seen).
    c. Volume of SRV RR, TXT, and MX queries.
    d. Internal failures – lookup for domain fails.
13. **Name analysis**: High volume queries with hostnames that are random for the same 2nd level domain *and* the same length indicate a DNS tunneling tool is sending data to the attacker's site, because the DNS server is consuming the host name as encoded data.
14. **Foreign countries**: You should study your organization's communication and operating model to determine how much communication occurs to countries outside of your own country. For example, a University with a varied foreign student population would consider this normal, but an insurance company that operates in a few states in the US would consider several queries to foreign countries abnormal. Note that if you are reading this book and you are in a foreign country, queries to name servers in the US and several European countries may be very common and may make this analysis more difficult.
15. **Queries to Dynamic DNS providers**: There are several dozen dynamic DNS providers operating today[27] who provide nearly free or inexpensive name to IP DNS resolution. A common model is for a home user to register their IP address and allow certain services through, with a name unique to them that their ISP would not provide. For example, a VPN client. Attackers can easily use these services as an avenue for hosting malicious services such as C2 DNS service because DDNS providers allow for rapid changes of a name to an IP address and can be used at nearly no cost.
16. **Abused Top Level Domains (TLD's)**: Spamhaus maintains an ever changing, evidence-based inventory of the top ten most abused domains names[28], which is expresses as an aptly named "badness index". Integrating this

---

[27] Lists include : http://dnslookup.me/dynamic-dns/, GitHub: Nate Guagenti / neu5ron, and http://mirror1.malwaredomains.com/files/dynamic_dns.txt (3/26/18)

[28] The interactive list is available here: https://www.spamhaus.org/statistics/tlds/ (8/18/18)

functionally into the SIEM may not be practical, but integrating a check of the domain TLD into the incident response process and the analyst checklist certainly is. As of June 28, 2018, there are 1,503 TLD's.

17. **Traffic to external IP without DNS query**: Direct HTTP, HTTPS, FTP, SSH, and likely other protocols directly to an IP address is suspicious. It is not common for an end user to type in https://#.#.#.#/. With whatever method you have, review which end systems are communicating outbound directly to an IP without a name. 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, like a call to any site that offers a NSlookup function. (Root DNS servers don't count!)
18. **Use of non-authorized DNS**: There are several free DNS services available on the Internet other than the DNS that the sites ISP provides. 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.

End Point Detection and Response

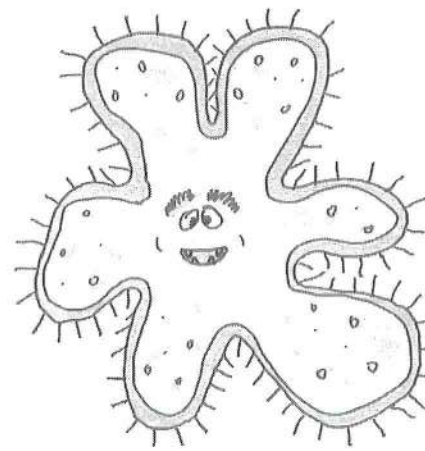
Entering the desktop protection field are highly capable software platforms focused on threat hunting for the endpoint. Vendors include FireEye Endpoint Security, Carbon Black Cb Response, Guidance Software EnCase Endpoint Security, Cybereason Total Enterprise Protection, Tanium, CrowdStrike Falcon Insight, and CounterTack Endpoint Threat. The Gartner 2018 magic quadrant for this space lists more than twenty vendors.

Here, alerts from an endpoint protection system can have a high "signal to noise" ratio because those alerts are pre-validated. These events occur because a binary matched an entry on a known feed list or matched a detection criterion. By consuming and analyzing endpoint protection system alerts *only (and not all endpoint activity)*, you can actually *perform threat intelligence and assessment* on the user population. If users are predominantly becoming infected and then bringing their notebook back into work, or if users insert USB drives in their computers and there is infectionware on the USB, then SOC has a better-defined path for security awareness training and remediation.

Microsoft's EMET is a free tool from Microsoft. It can perform a similar detection and mitigation, but it logs locally and doesn't have a comprehensive console like the vendors listed above. According to the EMET 5.5.1 User Guide, EMET reports to the local event log, so if there isn't a method of consuming that log then these events would stay on the system. The application whitelisting policies, which are also configured with Group Policy are excellent candidates for forwarding to the SIEM.

In order to get data into the SIEM platform for SaaS EDR, some form of encrypted syslog service or some sort of REST API needs to be configured to send data into the SIEM. *Do not send all endpoint data that the EDR platform collects to your SIEM.* Rather, send the "condition detection" data to the SIEM.

End Point Detection Use Cases:



1. **IoC hit:** IoC hits 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 by hash value.
2. **Binary first observed:** A "first occurrence" of a binary, 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.
3. **Hash Checks:** A locally configured alerting list, such as a hash value of a particular binary. These don't have to be malicious. For example, you could have a deception system or practice in play to detect if a user opens or copies a "Top Secret look alike" file (this is an example of a HoneyToken).
4. **ASEP Registry Key:** Modification of a specific registry key used to establish persistence, such as the Run, Run Once, or RunOnceEx.
5. **Printer:** Also, the printer key can be configured to load an arbitrary DLL: (HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors).
6. **Specific directories:** Modification of directory or file within a file system.

Data Islands or System Snowflakes

For Windows, there are at least four security contexts that need to be considered for Microsoft Windows systems. Domain, domain member server, domain member workstation, and standalone workgroup systems. Each of these systems need to provide data for the SIEM. For Linux, most systems are their own source data island, which usually provide data via syslog. For an application its own localized and application specific audit system is yet another data island. those systems represent a challenge – little to no standards, they may not even generate enough logs, and they will require greater customization. However, this last group contains the item the attackers want most – your Crown Jewels.

Windows Account Life Cycle Events (ALCE)

These events record new accounts, account modifications, deletions, disable, and enable changes. These events all have discrete event IDs, record the account that was changed, and the user who made the change. Account management should be done 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 therefore used. This is a common requirement for most IT General Controls programs (ITGC).

Table 10 Security Log: Account Management Events

Event ID	Name
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 accounts 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 Lifecycle use cases:

1. **Short cycle account create and account delete events:** This use case catches accounts that are created and removed within a very short time window. As a bonus, the severity would be raised if the account was used, such as a logon event between the create and delete event.
2. **Short Cycle elevated group add and group remove events:** This use catches accounts added to highly privileged groups like "Domain Admins" and then quickly removed from the group. Extensive damage can be done in a short time.
3. **Accounts created/modified/disabled by staff *other than* designated account managers:** This condition helps identify policy violation, rogue admins, or attackers who gain access to a domain admin level credential. In more mature organizations there will be a few IT that 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.
4. **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, and 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 is used to manage accounts and any ALCE outside of that realm is unauthorized.

5. **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 if 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.
6. **Account deletion:** Some environments may choose not to delete account. Rather, they change the password to something very complex, remove *all groups (including Domain Users),* and disable the account permanently. This method allows for Security IDs in the NTFS file system to resolve to an account name, but effectively prevents account usage. Also, this method allows for a re-enabled account for an account held by a user who terminated can be more easily identified when these accounts are added to a tracking list.
7. **Accounts created and disabled or deleted this week for new users and users terminate employment:** This use case can be satisfied with a daily and a weekly report for all systems where user accounts are created, like an active directory domain or a constituent application.
8. **Accounts used prior to the authorized use date**: It is common for organizations to create accounts ahead of a new users' arrival. These accounts should not be used before the user arrives. This use case implies that the organization has a method of connecting account creation, account names, and the relevant dates in order to conduct this analysis.
9. **Accounts created outside of the domain context**: There are two cases here. One case is when accounts are created on a workstation – this should be a very rare occurrence. The other use case is when accounts created on member servers. Local accounts may be needed for a specific application. These use cases look for ALCE's *not* from the set of domain controllers.
10. **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. There

are several websites with lists of default accounts and passwords[29]. Default account usage has been in the OWASP Top 10 for several years.

11. **Local account creation and elevated access**. One of the key tenants in an Active Directory domain is *centralized authentication.* In reality, 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 a domain level account and then add that account to the local administrators' group. The SOC should understand how elevated access is applied and be able to detect elevated account usage.

Advanced Monitoring Rules and Alerts Which May Require External Scripting

1. **Accounts created in a *constituent system* which do not match an account in the *primary directory*.** This type of rule requires that systems managed with local accounts need to be cross indexed with the primary directory. In order to have this rule function, some sort of lookup in AD would be required and a constituent system will need to push an ALCE events to the SIEM. This use case will mature 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.
2. **Post ALCE events to an account tracking database**. For example, an Identity Management system which has more metadata about an account than a directory holds. Since the SIEM gets the actual event as they occur, it would be a better use of system resources to push that event to the IdM rather than have the IdM deploy yet another monitoring agent to the domain controller.
3. **One over One[30] manager notification on creation, modification, or disable**. This notification would normally come from an IdM, because while AD user accounts do have a manager field, that field may not be populated. If and IdM is not in place, *and the manager attribute is populated at the time of account creation in the directory*, then a notification would allow for the user's manager to know when the account was actually created.

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

---

[29] For example, the aptly named www.defaultpassword.com, or defaultpasswords.in.

[30] 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.

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[31]. Further, there are several groups within Active Directory that provide elevated access. Even worse than that, 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 which 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 out a report 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 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, the IDs are summarized below in a table rather than listing them out once per line.

Table 11 Windows Events: Group Changes (Security Log) (V1.02)

	Security			Distribution		
	Local	**Global**	**Universal**	**Local**	**Global**	**Universal**
Created	4731	4727	4754	4744	4749	4759
Changed	4735	4737	4755	4745	4750	4760
Deleted	4734	4730	4758	4748	4753	4763
Member Added	4732	4728	4756	4746	4751	4761
Member Removed	4733	4729	4757	4747	4752	47620

[31] In the United States, this requirement is derived from Sarbanes Oxley controls.

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 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 that* the message explains what the group controls access to - meaning 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 us that instead. for example, "Shared Drive, Monthly Financial Summaries, path name \\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 who is in a set of defined groups logs into the network. The event ID is 4964. In effect, this event ID records when accounts that are members of any *administrator defined* group logs 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 security focused valid reason. In order to implement this, assign the Security Group ID field to the registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit as a semi colon 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, then there is an audit record when *any user who is a member of a group that provides some form of elevated access* logs in.

Account Usage Events

Standalone System: When a user logs into a *standalone* system, there are a distinct 4624 Type 2 and 4776 events written to the security event log

(assuming that the policy is set 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 that the SAM database used to authenticate the user is one resident on the physical system. By default, a *standalone* system has the default value "WORKGROUP" in the Account Domain field unless the NetBIOS workgroup name is changed. For a standalone system where someone logs in with a Microsoft account (a cloud account used to connect the system to the Windows Store and other identity), such as a home user's system, the process is similar to a standalone login.

Domain System: This process is different when a user logs onto a workstation and the user is authenticated from the domain. There is a 4768-event written to the DC that authenticates the user, 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, there will be a 4647 followed by a 4634 event. As users authenticate to Windows file shares, there are 4624 Type 3 events record on the *serving* system. As users authenticate to other Kerberos integrated services, there are 4769 events registered on the DC.

Table 12 4624 Logon Types

Event ID	Level	Name
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 or Terminal Services or Remote Assistance 11: Cached credentials (off domain)

Table 13 Other Logon Events

Event ID	Level	Name
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[32]	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 14 Account Logon Failures Status Codes for Event ID 4625.
4648	Informational	Logon attempted using explicit credentials (RunAs, scheduled task runs as a specific user, uses alternate credentials, and a user runs a program requiring admin rights and User Account Control enabled.)

Account Lockouts (Security: 4740, 4625/0xC0000234): The applicability of this alarm will depend on the time of day and the account time. Numerous lockouts Monday morning are "normal", while account lockouts significantly outside your organizations normal working hours are suspicious. Also, monitoring account lockouts conditions are an example where SOC can provide operational value. For example, if a service account is locked out, then the application itself is very likely down, degraded, the service capability is under attack, or a script is misconfigured.

Account Logon Use Cases:

1. **Concurrent console logons (4624, type 2) from multiple sources within a short timeframe**: This condition indicates an account is being used from multiple systems. For *most* users, any count above two is out of the ordinary. For example, an instructor in a classroom may make a change to all the classroom PCs, but an accounting staff member is unlikely to logon to three PCs at once.
2. **Logons from internal and external, within a short window, not over RDP (4624, type 10)**: This condition may indicate account misuse, credential theft, account sharing, or a behavioral issue. Note that to make this use case effective, you will need to correlate *specific* account types that indicate *user*

[32] 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 ☹.

presence such as a PC console logon and a VPN login, not an RDP login and a VPN login.

3. **Geographically improbable VPN Logins**: Modern VPN systems can provide country and city of the source IP for a connecting IP, or the data can be enriched with geo lookup data as it arrives at the log collection point. More sophisticated platforms like Splunk actually have built-in functionality to detect geographically improbable access. This means that a user logged in from one location and subsequently logged in soon after from another location that they could not travel to in the time allotted. For example, from France at 10 AM and then Canada at 10:15 AM, same day. If your SIEM doesn't have this functionality, then as a simple check, check the country code to make sure it matches the country for your user population. Depending on the user population, this check may reveal a compromised account. For systems that have a tracking list functionality, check the current login state or province and country with the prior login. If they are different and an insufficient time has elapsed from the prior login to allow for travel, there may be a problem.
4. **Network Switches/Routers/Devices**: These devices represent the network infrastructure beneath the network operating system and application stack. Thus, they must be kept secure. Even in large corporations, there are often a small group of well-known users that 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.

Account Lockout use cases:

1. **Lockouts that originate from an external source:** Once lockouts reach a certain level from an externally facing login point, such as a Citrix remote desktop server or a VPN server there is external password guessing in process.
2. **Rhythmic lockouts:** Multiple periodic or rhythmic account lockouts can indicate password guessing, a service with outdated credentials, or a script attempting to logon with outdated credentials.
3. **Multiple lockouts *from different sources:*** These events will occur when the Workstation Name and/or the Source Network Address are different. If the count of unique sources is greater than 2, investigate why. The various reasons behind an account logon failure and the status codes. Review the table below to create more specific alerts. For example, a few events with a code of 0xC0000064 or 0xC000006A simply indicate user error. However, varying unique user names from the same source workstation name and/or the source network address indicate that there is account reconnaissance in progress.

Table 14 Account Logon Failures Status Codes for Event ID 4625

Status/Sub Status:	Name
0xC0000064	User name does not exist
0xC000006A	User name is correct, but the password is wrong
0xC0000234	User is currently locked out
0xC0000072	The users account is currently disabled
0xC000006F	The user tried to logon outside time of day restrictions
0xC0000070	Workstation restriction, or Authentication Policy Silo violation, which needs to be correlated with Event ID 4820 from a DC
0xC0000193	Account expired
0xC0000071	Account has an expired password
0xC0000133	System clocks too far out of sync (DC to PC)
0xC0000224	User is required to change password at next logon
0xc000015b	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 login (RDP or Interactive) should not be required. If a service account is used for Interactive or RDP logon and the designated account holder cannot explain its use right away, the account is compromised.

Microsoft Routing and Remote Access

Microsoft has had dial in and VPN functionality since at least NT 4.0. There are dozens of RRAS events which provide quite a bit of logging. Today, chances are that remote access is provided with a different VPN technology. If you are using RRAS, be aware that the tracing level logs need to be enabled using the RRAS console, and they are written to %windir%\tracing.

Normal logging needs to be enabled in the RRAS console. Choose "Log all events". Normal events are written to an actual log file: %windir%\system32\LogFiles

Monitoring Jump Boxes

One relatively inexpensive technique that can provide a high degree of security by limiting access into the server farm using RDP or SSH so it only comes from a jump box farm. The concept is that RDP, SSH, and other direct logon capabilities are blocked into and out of server segments *unless* they originate from a specific

set of jump box resources, those resources are located on their own routable segment, and only designated accounts can login to the jump boxes. Once the jump box farm is setup, deny inbound access into all other server segments for port 3389 (RDP) for Windows and 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 example where an EDR applications can really shine because they provide highly granular awareness of EXEs launched, network connections, registry activity, and so forth. All jump boxes should have WEC/WEF enabled.

Jump Box uses cases:

1. **Remote connections:** Any remote access management attempt into the server segment(s) not from a jump box or the jump box network segment is, at best, suspicious and disallowed if at all possible.
2. **Limited exe's:** A limited set of executables should run on the jump boxes. Any executable outside of what's needed to permit management to and from the server segment needs to be run down. Not only that, the inventory of executables running on these systems should be very stable.
3. **Limited user access population:** Only a specific set of users should be logging into them, so login attempts outside of that group should cause an alarm. If these user accounts are in a specific AD group then that group can feed a list within the SIEM. If the user accessing using RDP or SSH isn't in this list, raise an alarm.
4. **Service accounts:** Service Accounts should not be logging into a jump box.

Table 15 RDP Events from Applications and Services Logs -> Microsoft -> Windows -> TerminalServices-LocalSessionManager

Event ID	Name
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.
1105	Remote Desktop Services: Session reconnection succeeded: Records user, session ID, and source address

Table 16 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.

Network Hardware Devices and Appliances

Network hardware such as switches, routers, access points, storage systems, IP cameras, door controllers, acceleration servers, load balancers, and all sorts of appliance-oriented systems are initially configured with default local accounts like "admin" or "root". *All of these devices* must be configured to centrally log, must be configured 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. There may be some systems that have a characteristic that can justify a local account database, but it is unlikely that there is a solid reason why those systems should not centrally log.

Network Hardware Use Cases:

1. **Identify Network Hardware:** You can achieve this objective (or at least make progress towards achieving it) by scanning the network with nmap, looking for a response from ports 443/TCP, 80/TCP, and possibly 22/TCP. If systems are responding and not generating a log record, or at a minimum not seen as a source client IP for authentication on an AD DC, then an appliance of some sort is identified. Next step is to identify the system and determine if it can log, and should be configured to log.
2. **Collect authentication and change activity:** Depending on the device, you may want logging from them. At a minimum, you want change activity, user logins (success and failure), and system reboots.
3. **Monitor for default account attempts:** Logins to network hardware should be monitored so that default accounts like admin, root, and supervisor are not used.

4. **Monitor for outbound traffic:** Most network devices should generate very little outbound traffic outside of a few specific sites, which are most often for content or system updates. Alarm conditions will vary. For example, tuned alerts if a piece of hardware makes DNS requests or communicates to sites outside of a small list. Alternatively, review outbound traffic from a piece of hardware periodically to detect anomalies. Items of note include NTP requests, DNS requests not to the local DNS, and software updates from vendor network names.

Printing

Print servers can be configured to record when a user prints a document, the document name, and document size. You are more likely to need to enable print job monitoring through event log forwarding to support long term employee investigation. In order to support any assertion other than "User X printed a job to printer Y at time Z", you will need to enable supplemental auditing[33] to capture the print job name and conduct a forensic examination of the workstation or have an EDR application in place in order to fully support this degree of attribution.

Based on empirical observation, the Windows Print event order is: 800 -> 801 -> 311 -> 842 -> 804 -> **307** -> 802. Of these, the most relevant are 311 and 307.

Table 17 Windows > PrintService > Operational

Event ID	Level	Name (based on empirical observations)
307	Informational	Print Document owned by user (identifies user, print server name, printer, and if auditing is enabled, the file name).
800, 801	Informational	Print Job Diagnostics (spooling)
311	Informational	Printing a Document (Identifies the user and printer name)
824	Informational	Print Job Sandbox / Isolating print job
802	Informational	Print Job deletion
842	Informational	Print job isolation and print process tracking

[33] The GPO path is: Computer configuration >> Administrative Templates >> Printers>> allow job name in event logs.

Operating System Security, Change, and Stability

There are several conditions that affect system security and stability. By being able to monitor these conditions, the SOC can support helping to identify system stability issues and help to identify operational issues.

Operating system stability Use Cases:

1. **Adverse events by population**: The same adverse stability event occurring across N% of your environment. Think 2%, so if you have 1,000 servers that means an error occurs across 20 systems within a 24-hour period. As you find and remediate systemic issues, you would set this higher. Rather than focusing on "100 systems", though, this metric is better related to a single digit percentage of servers, because that metric has operational and security value. This particular condition is where the event taxonomy will come in handy, because identifying all of the source events by specific type would be exhausting.
2. **Security service failures:** The use case relates to security focused services failing, because that can indicate the environment cannot be properly monitored or active tampering is occurring.
3. **(Un)Installs outside of the change or maintenance window:** For changes (1022, 1033, 903-908), SOC should be able to perform long tail analysis of the installed application on a daily basis. For centrally deployed applications, the count of successful installation should be the size of the target population.
4. **Clearing the event log**: When configuring an alarm for this condition, make sure the alarm is for the security service and not the ADFS service. The ADFS service logs events to the Security log with event ID 1102. Clearing the event log should rarely, if ever, occur on the network. There are multiple configuration changes that can be done to compensate for reasons someone would cite to clear a log. For example: if the event log is too large, then its size can be reduced through group policy by increment, like dropping the size by 10% every six hours until the log reaches 128MB[34]. 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 reason for legitimate clearing of the log is if it is truly corrupted and a reboot didn't fix the log.
5. **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.

[34] This number is based on using Server 2008 and 2012 in a highly virtualized environment. The Windows admins found that was a size that provided several days to months of record keeping and still allowed the Event Viewer to be responsive. YMMV.

6. **New Scheduled Task**: Windows records this event using ID 4698 in the security log a very common and almost 100% reliable indicator of lateral movement when there are local logons (4624), new service (4697) and new task (4698) within two minutes of each other.
7. **Windows kernel errors (Blue Screen):** BSOD's are an issue for operational reasons as well as security reasons. Implementing a Windows rootkit is actually more difficult than one might think. If BSOD's occur on any recurring basis, it very may well be worth investigating why from a security perspective as well as an operational perspective.

Table 18 Windows OS Stability Events

Log	Event ID	Level	Name
System	104	Informational	Event Log was Cleared
Security	1102	Informational	Audit Log was Cleared
System	7000, 7023, 7024, 7026, 7031, 7032, 7034, 7013, 1069	Error	A service failed to start – these events can relate to login failures as well as general startup error and are varied.
Application	1000	Error	Application Error
Application	1001	Error	Application Hang
System	1001	Error	Blue Screen (system faults), also called a BugCheck
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 Initiate 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.)
System	41	Error	The system has rebooted without cleanly shutting down first. (Win10)

Log	Event ID	Level	Name
			These are Blue Screen of Death (BSOD) messages
Security	8222	Success Audit	Volume Shadow copy created.

Table 19 Microsoft-Windows-Kernel-Power

EventID	Name
41	The system has rebooted without cleanly shutting down first. Was there an adverse indicator right before this event – malware attempted install, NIDS alarms, excessive failed logins, failed hardware, failed service?
6008	The previous system shutdown at on was unexpected. Registered when the system actually restarts. Time delay between a shutdown and a restart should be established to make sure that the hardware recovers. Target no more than 5 to 8 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. In particular, security focused services (DLP agent, anti-virus, software deployment, etc.) should never fail to start.

Data Leakage (USB Insertion)

Windows has improved in its ability to track USB insertion events as new versions are released.

Table 20 USB-USBHUB3 Events

Log	Event ID	Name
USB-USBHUB3	43	New Device Information (limited observation)
USB-USBHUB3	400, 410	New Mass Storage Installation However, you will need to research storage volumes to use this properly.

Also, if auditing is enabled on this log area – "Application and Services Logs > Microsoft > Windows > DriverFrameworks-UserMode > Operational", Windows will log up to 18 events in this log when a USB drive is inserted into a system.

Table 21 Windows > DriverFrameworks-UserMode > Operational (USB, Win10)

Event ID	Level	Name

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, if "Audit PNP Activity" is enabled, the system will log these events relating to USB usage (and some others):

Table 22 Audit PNP Activity USB events

Event ID	Level	Name
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.

Brute Force Failed Authentication Attempts

This particular use case does have a nice pattern to it. Some consistent percentage of your user population will forget their account credentials routinely, every Monday morning, and will repeatedly try to login will lock their accounts, wait a little bit, and eventually call the Service Desk for assistance. Outside of that window, repeated account lockout conditions that repeatedly occur indicate one of a few things:

- A completely misconfigured system or application
- A user with a device that has an old credential that needs to be updated
- The account that is repeatedly locked out or failed to logon and is under a password guessing attack.

Misconfigured systems also have a detectable pattern: a rapid number of events from the same source address which repeats, and then either a well-defined pause or the events stop altogether as the application usually throws an error to the user. Note that the use cases below mention lockouts (Event ID 4740). Your organization may, or may not, have enabled that setting in AD.

Brute Force authentication use cases:

1. **Continual Authentication Failures and Account Lockouts**: Once failed logons reach a certain threshold or clipping level, there is a reason to monitor the account and investigate. For failed logons, consider starting at 30 failures within a 10-minute period and adjust from there.

2. **Repeated Account Lockouts**: Once an account logon failure occurs often enough, more sophisticated systems like Active Directory can be configured to lock the account. For monitoring account lockout, start with 5 consecutive lockouts before you trigger a brute force alarm, assuming the account lockout policy is set to 5 failed attempts and then the account locks for a short period of time like 1 minute.
3. **Low and Slow Authentication Failures:** Attackers need to determine how account authentications failures are handled by various applications. More specifically, web-based applications frequently have observable differences if 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 results and how long it takes the site to respond, on average. Therefore, an attacker may attempt a lower volume of authentication attempts in order to discern the pattern.
4. **Password Spray:** This attack attempts to use a single likely to be true password during the authentication attempt against all accounts in the domain, one time per account. The goal is to attempt to discover an account reset to an organization's default like "Spring2018", or to see if accounts use one of the most common passwords[35]. This type of attack can be found by searching for failed logons for multiple unique user accounts from the same source machine (name or IP address).

DHCP and Data Link Layer Analysis

DHCP and data link layer (L2) analysis can reveal significant issues on the network. In order to perform these checks, your organization needs some supporting controls and practices in place to provide system awareness at the MAC level. This discussion is focused on automating the analysis processes for rogue device detection. The lists described have a specific purpose. The system in place may achieve these lists by having an attribute for the entry purpose that may work as well as separate lists.

First, you will need to build and maintain a list of known MAC addresses for "quiet" stand-alone devices. There are all kinds of devices that don't supply a host name during the lease process, don't participate in a Windows domain, and can exist on the network. If you plan on identifying devices that are not authorized, you will need to 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.

[35] There is a Wikipedia article with the top 25 most common passwords that can be used. https://en.wikipedia.org/wiki/List_of_the_most_common_passwords (8/18/18)

Second, you need some sort of a naming convention so that you can disregard known systems, instead identifying a system that is "not like the others" that can be implemented during the DHCP lease negotiation process.

Third, end user systems and VOIP phones should go through a pre-provision 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 pre-provisioning, 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 the most difficult part, the SOC will need to know where the device actually is by switch port, floor, and building so that a technician will know about where to go to identify a rogue device. This will require naming of the DHCP scope with location information like "Bldg 16 3rd Floor East" and importing that into the SIEM as a network description. From there, the SOC can log into the switch and get the exact port information. Assuming that switch ports have the corresponding jack identifier and floor location, the end systems location would then be known. Yes, there are organizations that actually go to this level, I have seen it, they do exist.

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 the segment usage will further sharpen alarm development.

DHCP and Layer Two Use Cases:

1. **Non-Phone 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 if devices other than phones are on that network. Examples: 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, or if a non-phone appears on a phone network it can be identified. Or your call manager may be able to export a list of known MAC addresses of enrolled phones, and if a device that is a non-enrolled phone shows up on the VOIP networks 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 organizations VOIP phones can help improve rogue detection.

2. **Non-known 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 of your desktops something like "DSK####", portables "NB#####", and developer virtual machines "DEVVM#####", then you can monitor for two conditions of note. A system that registers with DHCP and doesn't supply a known host name in the pattern should alert. Alternatively, if there is full end workstation MAC registration, then the alarm can be raised when the "MAC address not in Workstation List" (which, of course, will need constant maintenance...)
3. **Security ISO:** There are known bootable ISO's that supply default names which can indicate malicious intent. This list will change over time though. As an example, if a host named 'kali' show up it means someone has booted a known penetration testing distribution.
4. **Rogue systems:** If a system appears that has no name, not in the auto provisioned MAC list, not in the manually build approved MAC list, doesn't supply a host name, is not on a guest network, and doesn't authenticate to the domain within a reasonable time (say 5 minutes), you have a ... Rogue One.

Next Generation Layer 7 Firewalls

Today's firewall is not simply a stateful inspection engine as they were in the 1990's. Today there are complete open source stateful inspection firewalls such as PFSense which are more advanced. As the security market has matured, next generation firewalls (NGFW's) such as the Palo Alto, WatchGuard, Sophos, FortiNet, and others are capable of acting as multilayer application gateways complete with TLS/SSL Interception with full "layer 7" analysis capabilities. Thus, they are capable of more sophisticated decision making and alerting that a stateful firewall won't provide. NGFW's provide numerous detections and alerts into a SIEM and can be placed in the ideal location, as a perimeter point control gateway.

For example, the Palo Alto NGFW (when subscriptions are purchased and enabled), can check URLs against an authorized list, site categorization, check files against a known threat database, apply an access policy by username, source address, or time, enforce protocol usage by TCP or UDP port, and numerous other application level controls as well as being the perimeter firewall. PanOS7 and 8 have eight different log record types, with some records having over 50 fields, based on the analysis performed and the result provided from its analysis engine.

One challenge that the SOC team faces is understanding what NGFWs do, how they process data and how they report conditions to the SIEM. In order to make best use of these systems by the SOC, a "mini manual" should be developed that explains the NGFWs capabilities and how to properly read the log data generated by the NGFW.

NGFW's can function like web proxies, may be able to perform file checking, and may have NIDS functionality. Review the use cases in other sections to determine how applicable they are to the NGFW. Many of the other use cases in this chapter can be applied to a NGFW.

TOR Overlay Networks

TOR is an overlay networks that require specific software to access. For some organizations, traffic using TOR may be complexly normal, and for others TOR usage may indicate a significant security risk.

TOR network use cases:

1. **Software analysis:** Looking for applications by name that are used to access TOR networks like the TOR browser and TOR messenger.
2. **TOR exit note IP addresses:** Accessing published IP addresses of TOR sites, which can be done by comparing outbound firewall traffic to a known list or by building NIDS rules to detect SYN packets to IPs in the list.
3. **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. Also NIDS rule sets can detect DNS traffic attempting to resolve .onion domains.
4. **TOR SSL:** SSL analysis can be done against .onion certificates. The Emerging Threats ruleset has NIDS rules to generate these alerts.

Tor Exit Nodes:

- TOR: https://check.torproject.org/cgi-bin/TorBulkExitList.py?ip=1.1.1.1 or https://torstatus.blutmagie.de/
- If you cannot access the TOR site, there are others who provide the same information. For example: https://www.dan.me.uk/tornodes or https://udger.com/, who has a subscription service and maintains list of known attackers, fake search engines, TOR exit nodes, and other CTI type sources.

DarkNet Unused Network Monitoring

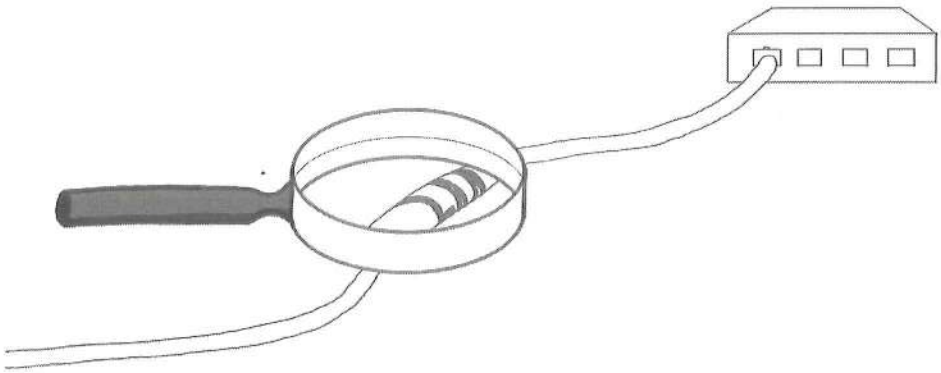
As defined in BTHb:SOCTH, a DarkNet network is an unused internal network range. For example, if your organization uses the 10.0.0.0/8 exclusively, then the entire 192.168.0.0 range and 172.16.0.0 to 172.31.255.255 range are "dark". Traffic directed towards those IP ranges or originating from those IP ranges is inherently suspect. As a recommendation on internal DarkNet monitoring, gather the list of all *known and in use* IP ranges, and then build a list of network segments *not in that list*. You would then build a report, query, or a script to analyze log data once per day to determine if there is internal traffic in the exclusion list. When you find IPs in use on networks that are not defined, then, at a minimum, you will have awareness that your network model within your platform needs to be updated. If you detect a system attempting communication to most of the IPs in a given segment, and then multiple segments, you have found an active scanning host. Common data sources to check for internal DarkNet monitoring include DNS requests, DHCP activity, authentication events to Windows domain controllers, and outbound firewall activity. There are numerous other sources – but these are the ones most likely to get a hit if a DarkNet is in use.

Network Intrusion Detection / Prevention

NIDS/NIPS systems monitor the data on the wire itself as it traverses a TAP, mirror or SPAN port, or are placed inline. These platforms search for patterns such as network scanning, protocol/port mismatches, command and control behaviors, certificate exchange issue detection, tunnel decode, and matches against a wide variety of signatures. Even the Bro network analysis system can extract files from network flows and pass them off to an analysis engine or just store them for later analysis.

Today, the challenge to a NIDS/NIPS engine's effectiveness is that more and more sites are using encryption, so when a normal legitimate website is compromised and used for nefarious purposes, the NIDS/NIPS will not be able to inspect the application flow or stream contents after the initial certificate exchange and session key is built out. Common deployments of NIDS/NIPS are at a chokepoint such as the perimeter or a DMZ, so positioning *inherently* affects their effectiveness. Also, NIDS/NIPS must be configured to know untrusted and trusted segments. NIDS/NIPS rulesets are configured in of two ways – to detect an attack from any source to any destination, or in a specific direction flow (trusted to untrusted, for example). The net effect here is that

the ruleset may catch *inbound traffic*, but may not catch someone inside using



an attack against an outsider. As a compensation, multiple NIDS/NIPS instances can be deployed with tuned rule sets based on *directionality*.

NIDS/NIPS signature detection varies when it comes to identifying true or false positives or negatives. Some rules, such as those that detect connections to the DShield.org top attackers list is very good, while rules focused on current events occurring "right now" will need tuning in order to improve over time. Further, NIDS rule effectiveness will diminish over time as the security issue they identify is dealt with. A rule that detects cleartext telnet logins against network hardware becomes less valuable once no more telnet access enabled. A rule for a spam campaign from four years ago is another example.

NIDS/NIPS Use Cases:

1. **Same alert, high volume, single target:** Repeated alerts directed towards the same "target" need to be either a) tuned because they are most likely a false positive or b) should be investigated because the rule is firing on a serious condition.
2. **Same alert, multiple targets:** When an alarm arrives for multiple targets, the same general rule applies – determine if the rule can be tuned based on an understandable condition, and if not, investigate.
3. **Multiple alarms, same system:** There are several rule conditions which can "stack" on 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 destination, then it is a sure sign of compromise.
4. **Vulnerability Correlation**: There is one area where ID/PS detection can really payoff: 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 ID/PS alarm against a known

vulnerable application, regardless of the TCP or UDP port. Any alarm that can leverage a known vulnerability should be investigated.

5. **Known command and control**: Botnet or command and control triggers based on IP addresses, domain names, pulse patterns, user agents, and other conditions. 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 conform C2 and other systems on the network involved in the same activity.
6. **TLS/SSL Blacklists**[36]: There are known certificates which are used by malicious software and botnets. This type of alarm can be applied to HTTPS traffic by matching the certificate necessary to setup the TLS connection. 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 look into the open source Emerging Threats Botnet Command and Control Server Rules section, what you see is that there is 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, 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.
7. **High alarm counts and singleton events:** Alerts at *either end* of the spectrum need specific attention. The top 3 to 5% of ID/PS alerts should be checked daily in order 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.
8. **Internal vulnerability scanning:** *Internal* to *internal* port and vulnerability scanning should only come from a few authorized sources, like the Vulnerability Assessment platform or the security engineering team. Outside of these few known IPs and users, *internal* scanning is suspect.
9. ***Internal* to *external* scanning:** While it is certainly permissible to conduct vulnerability assessment or scans against your own systems, any other scans can be considered an intrusion attempt, may indicate an attacker has control of an internal system, or a user up to no good. For example, someone could be moonlighting and running their own pen test business from the organizations network.

---

[36] https://sslbl.abuse.ch/blacklist/

10. **Include Eric Conrad's Whitecap rules:** These rules are designed to detect malicious use of ICMP traffic. The ruleset is implemented to ignore (passes) known good ICMP, and alerts on the rest, so it takes advantage of how a Snort ruleset is constructed. The rule set, in effect, whitelists good traffic and then assumes that any other ICMP traffic should be investigated, so they take advantage of rule processing order and the pass rule option. ICMP can be used for tunneling, and it can also be used for communications.

ID/PS Signature Updates are a *significant* part of maintaining their effectiveness. Most commercial vendors provide some form of automated update process, or a simple process to download the current content update, review, and then commit it to the system. Open source NIDS systems like Snort and Suricata have a variety of scripts to pull down rule sets and use an Oinkcode key value with the ruleset provider.

Perimeter Security Focused Access

Every organization has a perimeter point. This is the demarc router and the primary perimeter firewall. Realize that with the advent of highly portable and powerful computing devices the 1990's view of the perimeter is effectively dissolving, so traditional prevention first thinking needs to migrate to assume compromise, improve detection, and hunt for the adversary.

Some Interesting statistics help characterize data that flows on the Internet, and thus will affect your perimeter[37] in varying degrees.

1. 51.8% of all Internet traffic is from bots, 48.2% is from humans.
2. Over 50,000 websites are hacked every day.
3. As at June 2018, there are approximately 1.89 billion websites.
4. 338 million domain names were registered as of first quarter of 2018.
5. There are 3.7 billion global mobile Internet users as at January 2018.

For small organizations or sites with just a few IP addresses, the Internet access line may actually plug straight into the external firewall interface. Larger organizations, particularly those with fault tolerant firewall configurations, will have a gateway on the router and routable IPs on the firewalls.

[37] Statistics are from https://hostingfacts.com/internet-facts-stats/ as of August 22, 2018.

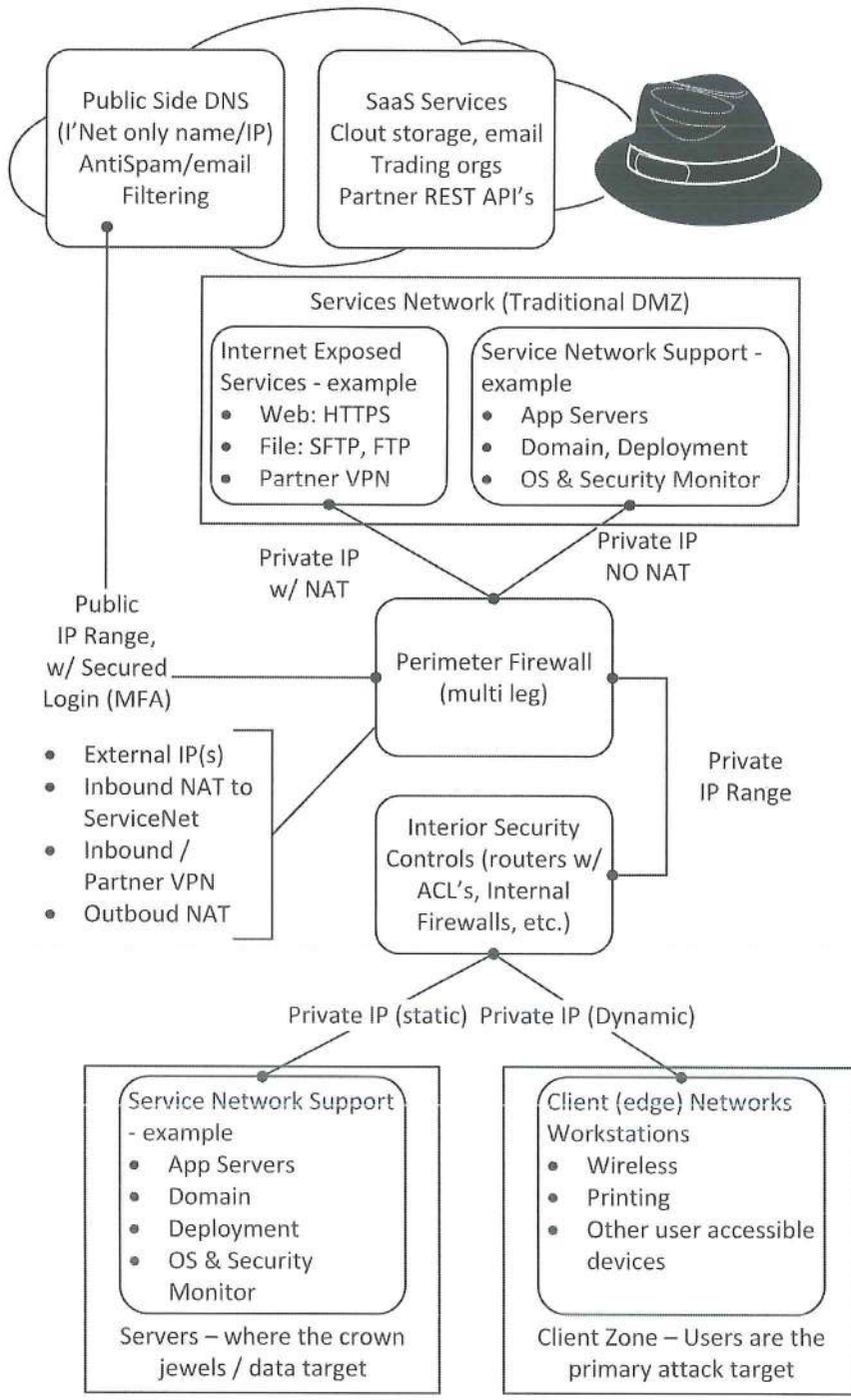


Figure 4 Perimeter Use Case Illustration

Very few company resources, if any at all, should ever be "outside" the firewall. The perimeter router is the last control point for outbound traffic and therefore

it is the first control point inbound. You should know where NAT translation occurs - on a perimeter router or in the firewall.

Organizations take very different approaches to applying security controls on the perimeter router and logging activity. In the next figure, several security zones are represented that may have different logging levels. Each one of the intersection points is an opportunity to collect session data with tools like a firewall, NetFlow, or Bro IDS. Each of the systems represented are also examples of application and operating system data sources.

One challenge you may have is resolving NAT translation because the SOC wants the true source IP address whenever possible. For example, if a user accesses the web interface on the DNS hosting site from the corporate firewall, you won't know the true client IP *from the DNS hosting site's perspective.* The best you could do is match access logs by adjusted time, source IP and port to the site perimeter firewall with its state logs of which internal IP visited the DNS site at the same time through the same outbound TCP port.

Regardless of how the perimeter devices are managed, *any and all changes or direct logins* to the perimeter router and firewall should generate a log record into the SIEM.

Architecture and configuration details provide considerations that may allow for data reduction and filtering. Examples are:

1. **Inbound accept logging:** Data will be duplicative, as most inbound traffic will be protected by the perimeter firewall and possibly a local firewall for an internet facing resource.
2. **Inbound deny logging:** Your organization should have a default inbound deny policy in the firewall. Occasionally there is a desire to send tHypberhe perimeter router logs into the SIEM. This is suboptimal because this posture will log a large amount of traffic that will provide very low value. The commodity Internet is loaded with viri, scanners, and attackers just looking for any opportunity to get in – all of which occurs 7/24/365.

Perimeter Traffic Use Cases and Detection Rules: There are thousands of application level protocols in use today. Many of these are service protocols. As you can see in Figure 4 Perimeter Use Case Illustration on page 108, several services are depicted in use on the service network. From the figure you can see a few application protocols in use on the service network. Rather than write dozens of pages that say why you should or should not allow a given network layer protocol or application layer protocol, this section will define a model for you to evaluate *if* a protocol should be allowed or not, and if not, how you can monitor for protocol usage.

IP Network Layer Protocols[38]: The next layer network protocol is defined in byte 9 of the IP header. Common next layer protocols that are likely to traverse your perimeter are listed in Table 23 IP Next Layer Protocol Numbers (IPv4) on page 110. Many of these network layer protocols are not in active use today. Common protocols your organization needs include TCP, UDP, ICMP, some form of routing, and likely use a VPN that is based on TLS, L2TP, or IPSec.

Table 23 IP Next Layer Protocol 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 determine if the IP protocol is actually needed. If it can be demonstrated with a communication flow and that is in in active use to support a business process, it is needed. If not, ensure that you can monitor for it or deny the IP layer protocol.

An example: You may never have heard of Stream Control Transmission Protocol (SCTP). Are you aware that this protocol has innovative features beyond TCP, such as multihoming and multistreaming in a single SCTP association? It is becoming more popular, particularly for web servers? Can your NIDS decode SCTP data? As of August 2018, Suricata can parse SCTP, but there are very few rules available in the Emerging Threats pro feed. Coverage is limited to some very specific areas such as Denial of Service and a few buffer over flow conditions. Also, the manual lists that Snort won't handle multiple encapsulations[39]. Would your firewall admin even think to mention enabling a new IP layer protocol to the security team? Here is an example protocol level advisory. ForcePoint[40] advises "A flaw was also found in the [at least Red Hat] Linux kernels implementation of SCTP protocol in which a remote attacker can

[38] IANA's protocol list: https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

[39] http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node10.html

[40] https://support.forcepoint.com/KBArticle?id=CVE-2016-7117-and-CVE-2016-9555-Linux-Security-Vulnerabilities

trigger an out-of-bounds read (CVE-2016-9555)." Therefore, understanding and having the ability to *monitor for protocol usage* outside of what the NIDS system can process or what you expect to see is necessary.

Protocol Detection with NIDS: In order to detect protocols that are not known to be in use, you can use a Snort rule like this for IGMP (IP protocol #2).

```
alarm ip any any -> any any (ip_proto:igmp; msg: "IGMP
Observed";)
```

Or this rule for SCTP (protocol 132, Snort doesn't have a label for SCTP):

```
alarm ip any any -> any any (ip_proto:132; msg: "SCTP
Observed";)
```

Protocols by name are listed in a typical Linux system in the /etc/protocols file.

Application layer Service Protocols: Many application protocols run on a standardized UDP or TCP service port. For example, several just won't work without extensive hoop jumping unless they run on a standard port: DNS, NTP, DHCP, BootP, SNMP, SMB, NFS, and AFP are but a few. As you can see in Figure 4 Perimeter Use Case Illustration on page 108, several services are depicted in use on the service or DMZ network. For the SOC team, it is critical to understand what services are allowed into the service network segment (or DMZ) for the organization.

Perimeter use case requirements:

1. **NAT:** The SOC *must* know all of the external DNS entries and NAT translations in order to have the best possible awareness and ability to correlate internal private IP addresses to external IPs. A usage mapping is also essential.
2. **Top 1M:** Determine your daily baseline, such as top 10,000 external IPs and whether they are in the a top 1M domain / IP list. See p. 112.
3. **Protocols in use and protocol volume:** From there, you can compare each day against this baseline to locate potential deviations, abnormal flows, or systems communicating to potential suspects using an unusual protocol.
4. **Persistence:** Also, the ability to detect persistence, such as a connection lasting more than 24 hours is a *key capability*.
5. **Forgery and Private IP:** Forged IPs, private IPs attempting to egress, and other traffic anomalies.

Top One Million Site Checks

Several data sources can be checked against a top one million site list. The premise is that well know sites which are in the list have been around for quite some time, are well maintained, and if a security issue appears it will be responded to in a timely manner. Please do not infer that a site is "completely safe" or acceptable if it is in a top one million list. The primary goal of eliminating sites in this list is to more readily identify potential suspects.

There are a few lists and sources for the Top One Million sites, such as Majestic and Cisco Umbrella 1 Million[41]. Both are still offering the list for no charge. Alexa was the first to provide this list, and they stopped in late 2016. Cisco's list is based on DNS queries, while Majestic's list is based on websites names, so there are differences at the bottom of the list, approximately 990,000 and higher.

The recommendation to monitor visited sites or DNS queries outside of a top one million list should be taken to mean that the site is acceptable in the workplace, as many adult content sites are high in the list. The intention of monitoring DNS, proxied sites, or IP addresses queries outside of these inventories is to identify newly suspect domains, sites that may have a higher probability of hosting malware or are newly registered. In effect, this analysis is a significant data reduction exercise and a great example of long tail analysis.

Remember, attackers have compromised many top sites over the years, there is malware that uses Twitter, Facebook, and a Gmail based C2 library.

Several data sources provide information that can be checked against these lists.

Using DNS name top 1M ranking. Mark Baggett, GSE, wrote a domain_stats tool[42] that can return several values from the whois record of a domain and also a ranking score for the domain based on its position in the list. If your SIEM can make a web API call, then an analyst can quickly use the result. This functionality can be integrated as a right click operation. Better yet, integrate the check to an event record with a custom attribute as data flows through log collection.

Examples below are from Mark's GitHub page:

> Alternatively, you can query individual entries in the whois record by including field names in the path.

[41] https://umbrella.cisco.com/blog/2016/12/14/cisco-umbrella-1-million/
[42] https://github.com/MarkBaggett/domain_stats

```
student@SEC573:~$ 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.

```
student@SEC573:~$ curl http://127.0.0.1:8000/alexa/sans.org
25646
```

Top One Million Use Cases:

1. **Review rank ordered domains outside of top 1M list:** Pull the DNS queries over the past 24 hours, remove any local queries such as a query without a period or a Windows resource record lookup, and scan through a rank ordered list of the DNS names that are not in the reference list. The point of this exercise isn't that the top site list are sites that are appropriate for your organization – rather that you need to look for odd names, new names, or first use names so this data reduction exercise focuses your attention. Further refine the result set from the step above, and add the creation date to the list. Queries against young domains are inherently suspect.
2. **Review for DGA names:** Malware is increasing its use of domain generating algorithms (DGA) so it can figure out how to communicate to its control network through names that change daily. DGA based names tend to be long – greater than 32 characters. DGA names are programmatically generated and most often relate to the date, so that these names are new every day. These names should naturally fall out of this analysis.
3. **Firewall data:** Consider running the suspect IPs recently visited list against an inventory of top sites, and if there are site/IP isn't in that list, investigate then.

Top Ten IP Address Use Cases

This section isn't combined with another section because the "top ten IPs" list can come from a wide variety of sources, with the data source influencing the interpretation of an address belonging in the "top ten". Also – the idea of responding to the "Top Ten" completely negates looking at the "Bottom Few", which is *often where the real action is*. Dr. Eric Cole has often advised that when an IP address that is in all three of these categories: top talker, highest bandwidth, and encrypted, it's malicious. The discussion below takes those points further.

Top Ten IP Address use cases:

1. **Top Ten "outbound connections" + Top Ten "data flow" + Top Ten "Workstations" or "Servers":** This condition may indicate data exfiltration, and can be used to profile the overall network activity. When an IP appears in all three lists, spend time confirming if the machine is compromised.
2. **Top ten outbound with a connection lasting more than 24 hours:** This is another example of a possible data exfiltration, particularly if it comes from the workstation side. From the server side, SOC can build an inventory of recurring patterns of "Server A to Site B", and then eliminate them from the daily check in order to identify new long running connections. Once understood, the time will likely change based on the source network.
3. **Top Ten (or more) DNS requests for newly registered domains**: If you have a DNS logging capability or have implemented PassiveDNS, then you can take "yesterdays" DNS queries and compare it to newly registered DNS names (see the DNS section for more detail and use domain_stats for bulk checks). This result set would prompt a pivot to other data, such as NIDS. An analyst should review the result set to detect anomalies or security issues. The benefit of this particular use case is that it can be fully scripted, and once done analysis can be done 'first thing' or by the overnight SOC team.
4. **Top Ten outbound denies by source and/or destination port**: Aside from the smallest of networks, servers and workstations should be deployed in their own segments (broadcast domains) 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 a threat hunting activity can also help identify operational issues, like systems not using proper DNS, SMTP, proxy, or NTP servers.

Web Application Firewalls (WAF)

A Web Application Firewall is an application level protective system that resides in front of a web server or, in the best case, fully integrated into a web server process. WAFs function at the application level because they understand the HTTP/S protocol, monitor traffic, can load balance, monitor the execution flow, may validate document content such as validating an XML document. and can make filtering decisions based on their rule set. To be truly effective, WAFs need to see all traffic unencrypted which complicates deployment. Systems like Palo Alto NGFW, Citrix NetScaler, and F5 Local Traffic Manager perform a TLS/SSL front end interception function so they can analyze traffic. Some WAFs can be built in to the web server process like the Apache mod_security. However,

single server module deployments may not scale well when compared with a load balancer that has WAF capabilities.

For the SOC team, the primary use case for WAF is that it functions like an IDS because it can inspect HTTP traffic and can therefore identify web application attack traffic. When the WAF finds an issue, such as a "select * from TABLE", SQL injection statements like "; 1=1", or Base64 encoded fields in an XML document where there should not be one, the WAF should provide an alarm and a detailed log record. It can provide a very useful data source for network forensic analysis.

Refer to Webserver and Application Server Activity on page 117 for further information.

Web Proxy and URL Activity (V1.02)

A web proxy is one of the more valuable data sources for the security operations team. Web proxies record which source IP went to what site, the user, date, time, should provide HTTP return codes, hopefully the user agent may record if the user is redirected, and other details about web browsing. More sophisticated proxies will provide a classification value[43] or description for the URL such as "Advertising", "Dating", or "Folklore". A web proxy will also report the access control decision. Examples of these decisions attributes are allowed, allow by specific rule, block, bypass by rule, bypass by user request, or bypass on first observed. Note, here that "who" can be a user, a server, an application.

Web proxy use cases:

1. **End user workstations *not using the proxy:*** This condition should be rare – you have a proxy and users use the proxy, right? Systems *not* using the proxy are negating the security value of the proxy, and therefore do not generate any application level log records. Users not using the proxy server(s) should be blocked, and that log record enable to log, so that this condition can be corrected. Further, by employing a proxy and blocking outbound use of common webserver ports like 80/TCP, 443/TCP, 8080/TCP, and 8443/TCP, you can detect rogue devices on the network like a disposable Raspberry PI plugged in an using OpenVPN over 443/TCP.
2. **Servers using, or not using, the proxy:** If your organization allows servers to have Internet access, then you should maintain a list of authorized sites. When a server accesses a site outside that list, the site should be checked to determine if it is a risk or should be added to the known server sites list.

[43] For example, Fortinet provides a list of categories: https://fortiguard.com/webfilter/categories

3. **Suspicious user agents**[44]:. End user desktop user agent strings should identify the browser, operating system, and may identify some key features. For example, on Windows 10, Internet Explorer 11 identifies itself with this user agent[45]: "Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko". The list of user agents should be reviewed using long tail analysis. You are looking for user agents that are not used by your installed browser base, such as a web spider, a scanner, misspellings of browser user agents, the Python programming language, or applications self-identifying by the user agent string. Systems making routine use of the web and *not* supplying a UA string should also be investigated, because this isn't normal.
4. **Name to IP mismatches *over a short amount of time:*** Sites do not normally change their IP address frequently. Site to IP relationships that change more than a few times within a day are suspicious, aside from sites that are actually hosted from multiple IPs. Larger sites will be load balanced with multiple IPs so make sure to check the DNS results before going too far with this condition because sites can be load or geo balanced.
5. **First time use sites:** Users within an organization will display a habit of using the same set of sites, so the ability to detect a new site can be very useful. The very first time a site is seen may be a result of a user clicking on a spammy link, a banner add that takes a user to a malicious site, or some other condition. A quick check of new sites once per day can reveal an issue.
6. **Consistent, repeatable browser pings or beacons:** Small sets of data going to the same site (meaning DNS name) over time indicate that a persistence mechanism or something nefarious is in place on the sending system. You will likely find a variety of SaaS applications in use, or user installed "information push" applications like a stock ticker. Connections not in this inventory need to be investigated. A beacon will have specific characteristics: they will repeat on an interval, connection duration will be short, and most of the beacons will be about or exactly the same size. One powerful tool to perform beacon analysis is RITA from Black Hills InfoSec.
7. **HTTP/S traffic *without a preceding DNS lookup***. HTTP traffic to a site by IP without a preceding lookup is rare. To realize this condition, begin by reviewing HTTP/s requests directly to IPs, determine the ASN and country code. and then check the ASN and country code to determine the usage pattern. Avoid a DNS reverse lookup from the beginning, because that may tip off an attacker. Use a DNS lookup Web service instead[46]. It is possible that a site or application may redirect a user directly to an IP address, but with more and more content delivery networks in use and IPv4 overcrowding these conditions may indicate suspicious behavior.

---

[44] https://udger.com/resources/ua-list maintains a list of user agents.

[45] https://msdn.microsoft.com/en-us/library/hh869301(v=vs.85).aspx

[46] One example (no endorsement) https://www.whoisxmlapi.com/dns-api.php

8. **WebShells**: As discussed in Webserver and Application Server Activity, presence of a webshell would mean that either an internal user is actively compromising an external system, or the system has been compromised and the shell has been there for "a while".
9. **Web Access Including an Executable or script suffix (V1.02):** A increasingly common practice for an adversary is to install a minimal downloader or dropper, and then to pull down an EXE, PS1 file, VBS, or some other form of executable content. For example, the Sysinternals tool psexec can be downloaded directly from https://live.sysinternals.com/psexec.exe. Based on the list of common EXEx described on p. 120, monitoring for the sysinternals.com domain and also monitoring URL's that contain executables or script suffixes may indicate an intrusion event.

Webserver and Application Server Activity

Much of this data can be tremendously valuable for consumer analytics, so the organization may already have an existing capability when it comes to analyzing this data source. This is another case where the majority of the data is normal activity, and before data is sent to the SIEM, the data can safely be reduced.

Before you contemplate pulling in webserver data into the SIEM, investigate how web server log data is currently used. The web admins likely have profiles and software to analyze their logs. By reviewing these reports, a baseline can be established which can then drive how to detect threats to web servers. Remember that the SOC is looking for security issues, not marketing intelligence.

For an internal webserver, ensure that you collect authentication logs, the connecting user agent, and the source IP. If there is a way to record a logout, that's a bonus (users don't often click on the logout link). After login and logout activity, you would want a "critical transaction", which should show up as a submit on a specific page as a POST.

For an external webserver, you may have difficulty getting the true connecting IP. Many security architectures define a perimeter firewall with a NAT address in place, so the web server logs will not contain the true source, as the webserver is deployed in a DMZ using an RFC1918 address. More sophisticated perimeter security models employ load balancers or reverse proxies can use the X-Forwarded-For (XFF) HTTP header field. This field records the originating source address. The next step is ensuring that this field is recorded in the webserver logs, and then that the SIEM can parse the field.

Regardless of the decisions made about consuming web server logs, there are several conditions to monitor for web servers are described below. 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. Here are several considerations:

1. Produce an analysis of the number of unique pages or URL requests from a single source IP address. If 4/5 of your visitors interact with up to 125 URLs most of the time, then that represents the "normal usage pattern." You would confirm this by comparing the URL list against a stable DEV or QA version of the website – *not the production version*. There are two reasons for this. First, validate that QA mimics production. Second, if an attacker were capable of dropping off a malicious file like a webshell or the system, you would not want that file to be excluded. Now determine what remains in the last 1/5 of the time to detect spiders, vulnerability "testers", and other indicators.
2. The 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.
3. Most log records should result in a 100 (information), 200 (success), or 300 (redirect) status code. These are all "normal" results. 400 and 500 level status codes, or a subset of them, can have real security value. For example, a 400 Bad Request, 401 Unauthorized, or 405 Method not Allowed are codes that can be a result of scanning or malicious server exploration.
4. 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 and consume these log records.
5. If possible, then determine the "average visit time" for this period.
6. By understanding what is normal, your SOC team has a way of detecting when visiting IPs or DNS names are acting outside of this boundary. 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:

1. **Webshells**: For server-side deployments, and in particular any system that is internet exposed and has a web server, configure file blocking for known webshells. There are several reputable lists of webshells, and there are numerous PHP shells on Github[47] (and likely many other places.)

[47] https://github.com/bartblaze/PHP-backdoors, https://github.com/JohnTroony/php-webshells, https://github.com/BDLeet/public-shell, https://github.com/tennc/webshell

2. **Attack Type Traffic:** Page requests that are *not part of the application URL Inventory.* If a user can successfully get a webshell on the server. To determine this, you would need a list of the URL's that the application has, 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 webshell 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 URL, 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.
3. **HTTP return status codes:** These should be in the 100, 200, and 300 range most of the time. 400 and 500 error status codes 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.
4. **Injection Strings:** Useful strings in an IIS log that relate to an attacker attempting to penetrate a connected SQL server include:
    a. XP_CMDShell – attempt to invoke something at the command line
    b. Select * - attempt to get all data from a table
    c. Or 1=1 – an attempt to perform SQL injection
    d. SQL keywords – update (change table), cast (manipulate data type), union (merge multiple result sets)
    e. Pages that normally receive POST requests are also receiving GET requests
    f. Windows process names like "cmd.exe" and Linux shells like /bin/sh.
5. **Sessions**: Long running sessions. Consider looking into sessions that are 3 times the typical session length to start. This condition *may* indicate account compromise, because an attacker may be constantly interacting with the site.
6. **Spidering IPs**: IPs that visit most, if not all, of the possible URL's, visit URL's rapidly (say, thirty within a second or two), or generate numerous page failures indicate some form scanner, vulnerability analysis, or web content spider. Users who interact with a web server don't click on thirty links or buttons within a few seconds, unlike a web spider program or a scanner. Having your website indexed in Google isn't a technical attack, but it may be a data leakage issue.
7. **Application server conditions**: The application server provides back end access to the supporting database on behalf of a front-end web application. It is frequently a target of the attacker. There are security issues identified by the Open Web Application Security Project (OWASP) that 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.

Windows Firewall (V1.02)

Windows advanced firewall, when enabled, can write a variety of events to the Security log. Process ID's in WFAS events can be correlated to 4688 events by process ID. Protocols are listed by their numeric value, in decimal (TCP = 6, UDP = 17).

Event ID	Name
5146	The Windows Filtering Platform has blocked a packet.
5146	The Windows Filtering Platform 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.
5149	The DoS attack has subsided and normal processing is being resumed.
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 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.
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.
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.

Windows Process (Sysmon and EventID 4688) (V1.02)

When evaluating what processes to monitor in the environment, consider a few key questions.

1. How is remote administration and what are the remote execution tools used to perform remote management by system custodians? Common methods include direct RDP, WMIC, WinRM, psexec, ssh access, Group Policy, and package build and deployment.
2. 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?
3. What are the network flows for remote access? For example, are jump boxes used (see p. 92), specific segments for IT management, are end users granted RDP or SSH access, and what are the user accounts.

If you do not have an Endpoint Detection and Response agent, then the next best tool is to instrument Windows to collect process invocation and command line data. Windows event 4688, which can be instrumented on Server 2008 and forward to collect detailed process execution. With Server 2012 the full command line[48] can be captured by enabling the "Audit Process Creation" audit policy and the "Include command line in process creation events" GPO settings. Note that the time for the 4688 event is not contained within the event itself, it comes along with the event.

Table 24 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
```

[48] On a standalone system, 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 value to 1. Since a local system does not process a domain applied GPO, the registry value needs to be manually created.

```
        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 Sysinternals sysmon application. Sysmon can be configured to record more detailed information about processes and network connections. Sysmon records key details about processes, network connections, registry changes, hash values for a binary, parent process ID 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 4688 event from above is:

Table 25 Example Sysmon Event

```
Process Create:
UtcTime: 2018-09-12 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=C9A28DC8004C3E043CBF8E3A194FDA2B756CE90740DF217548833728
1B485F69,IMPHASH=3A8297483C1777054C7BAFEA5E6A8853
ParentProcessGuid: {1834af5b-f063-5b98-0000-001064510100}
ParentProcessId: 824
ParentImage: C:\Windows\System32\services.exe
ParentCommandLine: C:\WINDOWS\system32\services.exe
```

Sysmon is configured using an XML file, of which there are several well-known

examples[49]. 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 found. Here, use long tail analysis in order to arrange the results from most to least so you can find the high volume binaries, and the singletons. From within a PowerShell ISE or PowerShell command shell, run with administrative rights, and run code below to show the ImageName (4th property). During an incident, you would want the command line (8th property), particularly for PowerShell. Also, 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, nor can it provide duration.

Table 26 Powershell code to list Sysmon EXE's in Long Tail Analysis order

```
$Hash = @{}
$entries = Get-WinEvent -filterhashtable
@{logname="Microsoft-Windows-Sysmon/Operational";id=1} |
%{$_.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 above code yields the following chart, as an example of long tail analysis:

LTA based charts group data together on a like basis. The intention of an LTA chart is to observe what happens at the singleton layer because single events are rarities, odd ducks, and may reveal security issues.

[49] GitHub sources – MotiBa, SwiftOnSecurity, ion-storm, MHaggis, Malware Archeology.

LTA can be used for a wide variety of data sources. If you took NIDS alerts then you can look at the high-volume alerts and better tune them, with the goal of decreasing the alarm count where possible. LTA is a powerful technique to understand your data.

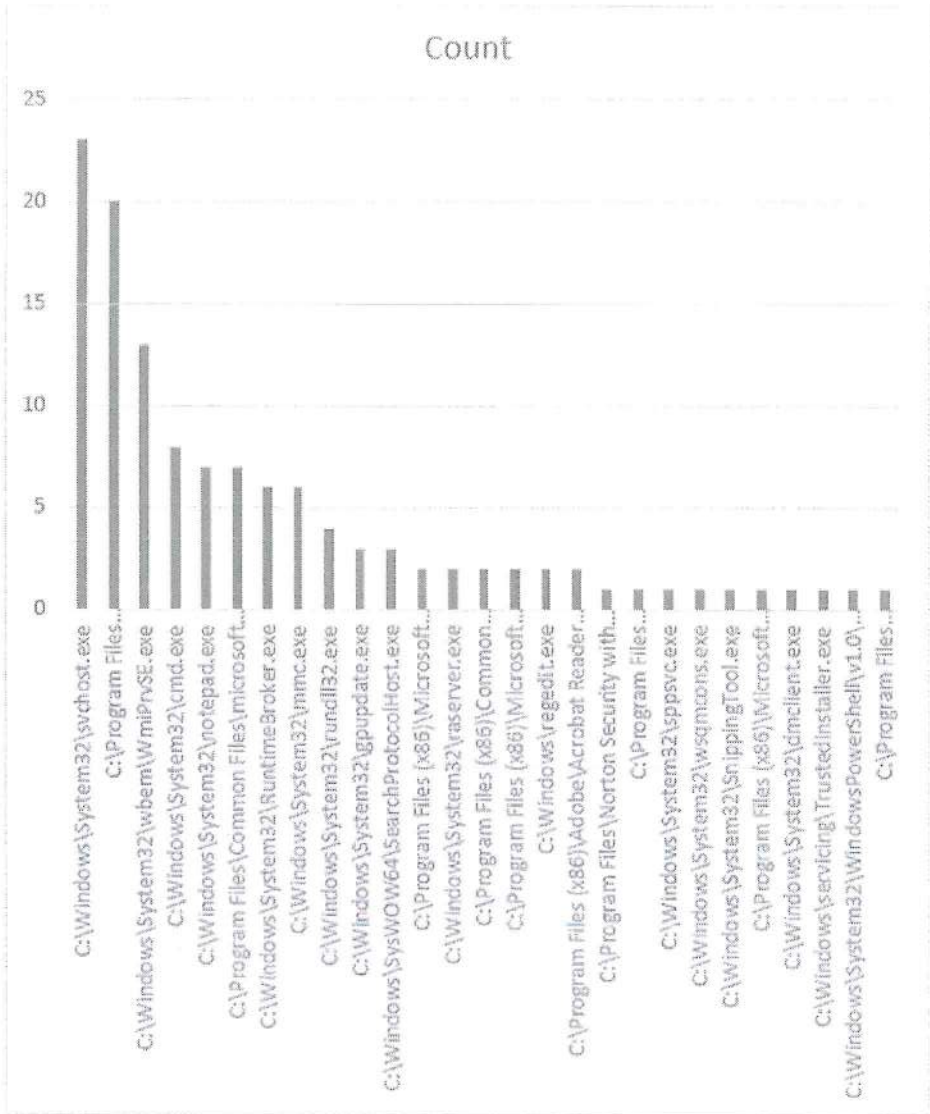


Figure 5 Windows Sysmon Process Long Tail Analysis

Table 27 Microsoft-Windows-Sysmon/Operational (v 7.01 as of March 2018)

Event ID	Name
1	Process creation

Event ID	Name
2	A process changed a file creation time
3	Network connection
4	Sysmon service state changed
5	Process terminated
6	Driver loaded
7	Image loaded
8	CreateRemoteThread
9	RawAccessRead
10	ProcessAccess
11	FileCreate
12	RegistryEvent (Object create and delete)
13	RegistryEvent (Registry value set)
14	RegistryEvent (Registry object renamed)
15	FileCreateStreamHash (File stream created)
16	Sysmon configuration change (cannot be filtered)
17	PipeEvent (Named pipe created)
18	PipeEvent (Named pipe connected)

Windows Process Execution Patterns and IoC's (V1.02)

Various Windows applications and binaries can be misused, or used to investigate a system, by an attacker. Most of these commands are executed early through an attack with a parent process name of "cmd.exe" or "powershell.exe". For the sake of organization, they are listed in alphabetic order by binary or topic, not frequency of usage.

For Version 1.02, this list was expanded. *Nonnative windows executables are indicated with a double asterisk after the exe name* - pwdump?.exe** is an example.

- **at.exe**: Used to schedule a job.
- **attrib.exe:** Can be used to hide files and directories.
- **cmd.exe**: There are a number of oddities that you can detect when cmd.exe is the parent process. Of note – cscript.exe. wscript.exe, and powershell.exe can be used by attackers. Further, when a productivity application such as Word, Adobe or Excel launch cmd.exe, the source file is most likely up to no good.
- **cscript.exe/wscript.exe**: These are older scripting tools, predating PowerShell, and are still viable today.

- **csvde.exe/ldifde.exe**: Can be used to extract Active Directory information into CSV files (bonus if you happen to find them, and the Admin Pak isn't installed).
- **dsquery.exe**: Used to extract a wide variety of information from Active Directory. Dsquery is more often used to extract user and group information.
- **dsget.exe / nltest.exe:** Used to determine the domain controller and its IP for the local logon session.
- **fsinfo.exe**: Used to get the list of connected drives
- **ipconfig.exe**: Get the NIC and DNS configuration.
- **mimikatz.exe**:** Used to extract plain text passwords, Kerberos tickets, hashes, and PIN codes from a running Windows system. The tools author also indicates it may be known as kdll, kdllpipe, and katz[50].
- **net commands**: There are numerous net commands – like "net localgroup administrators" to find out who is in the local Admin group.
- **netsh advfirewall**: Used to review and/or change the local firewall configuration.
- **netstat.exe**: Get list of listening ports.
- **ntdsutil.exe**: This is an Active Directory admin tool, and is used by adversaries for AD recon and configuration data. In particular, it is possible to extract the NTDS.DIT file.
- **ping.exe**: Test connectivity using ICMP. If the site permits ICMP to exit the network, the 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.
- **psexec.exe:** This Sysinternals tool can be used to execute remote commands on a Windows system, which it does by temporarily installing a service on the target. There should be a 5145 event in the Security log against the *\ADMIN$ share name. 7045 event in the system log when the service was remotely installed. Sysmon events 1 and 2 also provide traces.
- **pwdump?.exe****: Over the years, pwdump has appeared in many forms with increasing numbers in the file name. It is used to dump hashes and passwords.
- **reg.exe**: Query the registry, export and import sections, modify, or add keys to the registry.
- **rundll32.exe**: Rundll can be used to execute a script or invoke a DLL itself. Note that to invoke a DLL, the DLL name and the entry point for the DLL are specified on the command line.
- **qprocess.exe**: Displays process information.

[50] It may also appear with the name "mimidogs.exe".

- **sc.exe**: Command line service query and configuration tool.
- **schtasks.exe**: 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 a 200 event when executed. A 4648 is also registered.
- **sdelete.exe****: This Sysinternals tool is used to securely wipe the contents of a file.
- **shutdown.exe**: used to halt or reboot a system.
- **systeminfo.exe**: provides an in-depth inventory of a system.
- **tasklist.exe**: Used to see what processes are running.
- **tree.exe**: Produces a nice diagram of the file system directory structure.
- **ver.exe**: Retrieve current Windows version.
- **vssadmin.exe:** The volume shadow service administration tool. Adversaries use this tool to create, disable and/or delete volume shadow copies. When a shadow copy is created, Event ID 8222 is posted to the security log.
- **wce.exe****: The Windows Credential Editor a security tool to list logon sessions and add, change, list and delete associated credentials.
- **wevtutil.exe**: Used to retrieve information about the event logs, run queries, and clear the logs – look for the "clear-log" command line option.
- **wmic.exe**: Oldie and a goodie, has hundreds of query capabilities about a system and can also interact with remote systems. There are 4688 events recorded when WMIC is used. There will most likely be a a 4703, 4674, and possibly a 4656 event recorded, depending on the access level required.
- **vssadmin.exe**: can be used to remove shadow copies with a command like this: "vssasmin.exe Delete Shadows /ALL /Quiet".

Ransomware Extensions and file patterns: A growing area of protection is monitoring a local system and centrally access 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 anti-virus enabled. SOC should check for updated lists of ransomware extensions and search for them. In particular, there are extension lists to feed the FSRM service on Windows.

Office Applications: Office applications should not the parent process for command line tools either. Almost any combination of Adobe Acrobat (usually AcroRd32.exe), 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.

Note that attackers are working hard to develop and deploy obfuscation techniques that thwart string bases searches that can be instrumented in a SIEM. The analyst will need to compensate by understanding what is normal, or

expected, and then being able to detect when not normal and then determine the investigation path[51].

Windows Presence Indicators

Without having a solid set of presence indicators from Windows a significant side of the equation is missing. From workstations, realize that the user and their applications are the current attack space. From servers, the SOC must be able to monitor a privileged user. Modern attackers are pressing attacks against end users through malicious email, infected websites, or intercepting software updates when a user connects their system to a masquerading wireless access point. System admins are, by definition, users, and are also susceptible to these forms of attack. Therefore, presence data collection is essential.

Several goals are accomplished by centrally collecting Windows event and process activity data. It is essential for the modern digital battlefield, as the target of the attack is now on the end user. For servers, the normal collection process is to use a SIEM agent (proprietary or otherwise). From the workstation side, use Windows Event Collection and Forwarding. The reasons for collecting data from all Windows systems are:

- First, investigations involving end users are more complete, because data from centrally managed systems can be reviewed along with centrally managed systems.
- Second, attacker trace activity can be analyzed network or domain-wide, instead of analyzing a single system in isolation.
- Third, by collecting local authentication data (in particular 4624 events), lateral movement can be detected.
- Lastly, Long Tail Analysis can be performed of executed processes, network connections, event ID's, hash values, and any other collected attribute across the enterprise and analyzed for outliers/anomalies.

Microsoft has a built-in feature called Windows Event Forwarding that allows you to centrally collect event ID's from windows workstations. When setting this up, avoid collecting everything. Instead enable data collection waves as listed in the next chart. These Waves are organized by the likelihood that the data will assist in an incident, only, and are somewhat subjective. Review the list and make your own determination for your workstation collection.

[51] One of the better research papers on this is from FireEye: https://www.fireeye.com/content/dam/fireeye-www/blog/pdfs/dosfuscation-report.pdf (8/18/18)

Table 28 Windows Presence and Process Indicators (Workstation focus)

Source	ID	Wave	Description
Application (Errors)	*	3	Windows Error Reporting for application crashes
Security	*	1	Anti-malware events (Microsoft Antimalware, Windows Defender)
Security	1100	2	The 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 – as these are Windows being Windows)
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' o 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
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 Failure
Security	4698	1	A network share object was deleted.
Security	4699	1	A scheduled task was deleted
Security	4700	1	A scheduled task was enabled

Source	ID	Wave	Description
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 Discussed in Table 10 Security Log: Account Management Events on page 84.
Security	4731 4735 4734 4732 4733	1	Local Group Change events – users added/removed from local groups. Discussed in Table 11 Windows Events: Group Changes (Security Log) on page 87.
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 screen saver was invoked The screen saver 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
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

X-Forwarded For, NAT, and the True Source IP Topics

Many of the systems described act on behalf of an end user or system and change the source IP address in the process. For example, when users connect to a proxy, their browser connects to the proxy first, the proxy engine will evaluate the protocol request, and if permitted, it will then generate *a new connection out to the resource.* As the transactions occur the proxy engine will monitor and return permitted traffic back to the client. This process isolates the true source IP address of the client, and instead the resource connected to logs the IP of the proxy (or the firewall). When users connect through a stateful inspection firewall, their true source IP is changed to the IP on the "other side" of the firewall when Network Address Translation is in use – this would usually be the external or WAN IP in most cases.

Reverse Proxies or Load Balancers: When you are using a proxy in reverse order form the Internet to the web server in a service network, enable the X-Forwarded-For option so that the source IP is recorded in the log. Not all systems enable this by default, as it does impose some degree of overhead to track and record the connection.

Firewalls: Depending on the logging capability of the firewall, you may or may not be able to capture the NAT relationship. More sophisticated firewalls like the Palo Alto separately records the post NAT Source and and post NAT Destination IPs in a specific field for the TRAFFIC log.

SOC and SIEM Use Case Template

Even trivial use cases implemented within a SIEM or in support of the SOC team need to be well documented. A well-defined use case provides the actionable documented process component that supports the SOC team and therefore is a key building block of the Standard Operating Procedure (SOP) manual. This use case template and development process is in active use by one major SIEM vendor and is what I use today in many client engagements.

If you need additional background in use case development, you should consult a software engineering and architecture book and read the chapters that define use cases and explain how to perform requirements analysis. Examples include:

- Managing Software Requirements: A Use Case Approach, Second Edition, by Don Widrig, Dean Leffingwell (2004, available in Safari as of January 2017).
- Use Case Modeling 1st Edition, by Kurt Bittner and Ian Spence (2001, available in Safari as of January 2017).

SIEM/SOC Use Case Development Process

First - 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, in order 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 do help to articulate a use case so consider creating them to explain more complex use cases.

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 a Business Issue. Business issues can include supporting an IT General Control, supporting compliance/policy/procedures, verifying system uptime, providing brand protection, being able to detect fraud, detect, and if possible, thwart intellectual property theft, or minimize disruptions.
2. Design the question that the use case should answer. How would the attacker gain needed access, cause damage, exfiltrate data, or what accounts would they need to use? In other words, you must be able to describe the observed condition that is relevant to your security posture (the objective).
3. Determine and test the data sources and the data elements that provide the visibility needed to answer the question. The system providing data to the

SIEM must be capable of actually auditing the desired behavior (observe the action by a person or system interaction). For commercial systems, this may be a matter of enabling a logging function. For bespoke systems, there may be a need to enhance the system itself.

4. Evaluate the data by establishing normal baselines and other analysis dimensions. Characteristics to understand include volume, peaks/lulls, outliers, averages, frequencies of types of data or specific elements, duration of normal behavior, and how you find something "new". The system must provide the event record with sufficient fidelity to the SIEM, as defined under Log Record Data Elements on page 223. It is very useful to include screenshots and sample log records in the use case document itself. The SIEM must be able to process and present the event at the necessary level of granularity for the Sec Ops function (to measure or observe the interaction).
5. Build the necessary SIEM content (rules, dashboard, alert, reports) that realize the use case. Practically, this means matching up the input data and its fields with SIEM processing rules, internal lists, timing, staged rules, and other SIEM capabilities.
6. Establish the SOC guidance and processes that will be used to filter out false positives from the baseline data to support identifying malicious use or operational issues. Various techniques exist: bar chart analysis, graph analysis, simple timeline presentation, supporting correlation data.
7. Test the use case by causing the condition(s) necessary for the use case and SIEM content to function. If the SIEM doesn't respond as desired, then "rinse and repeat" to determine what was missing.

Template Instructions

Each section of the use case template is below in a heading style with the instructions on populating that section between square brackets "[]". General advice on implementing the use case is given in angle brackets "<>". After this use case is a fully defined sample for your reference. To implement this model, build a document with these headings, 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 how the Security Operations Center 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 here. The problems statement should clearly define and identify the issue, and provide direction to achieve a solution. Ideally, it expresses a solvable problem.]

<First, write out the issue without regard to getting it right. Make sure that the initial draft identifies the issue at hand and needs to be solved by the SOC Developer. Refine the problem statement and make sure that 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 – alert, email, record, post to list, etc. 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 characteristics.

1. Correct or accurate, in user terms, and unambiguous.
2. Can be implemented, or feasible.
3. Be necessary to support the use case. Keep requirements on point.
4. Ideally, requirements communicate priority. Practically, requirements can be satisfied during implementation phases for the use case.
5. Measurable or verifiable in some way which will manifest through the source data and actions that 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 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 speaking, 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 for the SOC that the system needs to include.]

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, content components such as internal lists, dashboards, output reports, etc. Each named item needs to be listed here so that the Use Case can easily be maintained as the system is updated. This section is concretely answered in terms of the specific platform tool itself. It may be very useful to diagram how the components interact.]

Use Case Data Source Description

[List of data sources and the system types that support the use case. Indicate if the data source is currently available, or steps 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 involved component of the data stream, there will be a Use Case Data Stream Analysis section. Each section needs to match the Use Case Component Name and Use Case Data Source Description sections above. 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 while not 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 trimmed and the reasons why trimming the data is valid and does not represent a degradation of the security monitoring capability of the organization.]

<Discuss what are the characteristics of an event. Include screen shot and/or plain text as necessary or if they would be helpful. Always indicate how the data was produced – when, where, from whom.>

Cyber Kill Chain Analysis and Support

[Indicate how this Use Case supports the Lockheed Martin Cyber Kill Chain. The Kill Chain is described beginning on page 180.]

CKC Phase	Use Case Support
Reconnaissance	
Weaponization	
Delivery	
Exploitation	
Installation	
C2: Command and Control	
Actions on Objectives	

Assumptions and Limitations

[In order to implement any use case there are often assumptions made about the data sources, delivery, and IT operations. This section is where you will document them. For example, if the use case relates to Active Directory monitoring, then implementing the use case assumes that new servers will be configured to report and that if an Active Directory Domain Controller(s) fail to report for some period of time that condition can be detected and resolved.]

Alternative Solutions and Discussion

[If there are any alternatives to implementing the use case as described, list them here in this section. 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 to be sufficient on first review, but may prove out not to satisfy the desired level of monitoring since a real time alarm is needed.]

<This section should demonstrate that the analysis and design team actually thought about 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, which 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 company policy/procedure that the use case supports.]

Profile	Description
File Name	Deliverable_Project_Title_DATE.doc
Process Owner	Lists the single person within the organization who are responsible for the most relevant process that the use case addresses.
Original Author .	State who wrote the original use case.
Policy/Procedure:	Title or reference to company policy/procedure that the use case directly supports.
Industry Reference	This line lists applicable standard references. For example, a reference to an item in NIST SP800-53, 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 a word auto update field).
Approval	Record who and when the use case was approved.

Version History

Version	Revision
1.0	This section provides history of use case changes, enhancements, and revisions.

Complete SOC and SIEM Use Case Example

Monitoring Elevated Access Group Membership

This Use Case was developed and successfully used at several Fortune 500 companies with three different SIEM platforms. It is adapted based on those experiences for general publication.

Name: Monitoring Elevated Group Membership

Purpose: The purpose of this design document is to fully discuss the requirements to monitor changes in an elevated access group within the Active Directory Domain through active monitoring by the SIEM system. The SIEM will provide notification to the designated recipient when membership in an elevated access control group changes (addition/deletion from) 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 the group grants administrative or other privileges. Users should only be added or removed from groups in response to a support ticket, but the "owner" may not be aware of the change, or may not have approved the change. Any user who has the ability to change group membership *may* change one of these groups, 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 would enhance elevated access management by only granting access through a "system administrator" account. For example, if a regular users account was "BSmith04", then their administrative account would be "BSmith04_SA" or "DA_BSmith04" (SA: System administrator, DA: Domain 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 the change occurs.

Assurance metrics:

- Achieve 100% rate of group change notifications to the group "owner".
- Ensure that additions/removal from monitored groups occurs within a reasonable time window (usually within two minutes of the change).

- Assurance that only authorized users change membership in elevated access groups.
- Further assurance that only a "secondary" account is added to a monitored group.

Requirement Statement(s)

- Ensure that the reporting systems provide discrete data that shows:
  - Who made the add/remove occur to which monitored group
  - When the add/remove occurred to the monitored group
  - Bonus: the account managers workstation name (changes should only occur from an authorized pool of systems)
- A success notification should go to the group owner
- A failure notification should go to InfoSec for investigation
- Changes should occur from a group of "authorized users"; changes not made by the account managers are suspect
- Data will arrive from a variety of DCs and Windows domains
- As illustrated in Figure 6 Maintaining Inventory of Elevated Access Groups simple table should be built and maintained that stores:
  - Group Name: This is the minimum set of groups that can make a change to AD itself. Note that there are several Certificate services groups not listed, so if there is a CA, the group list will grow
  - Staff member who is the designated "Group Owner"
  - Notification Recipients
  - Notes to explain what the access controls or grants

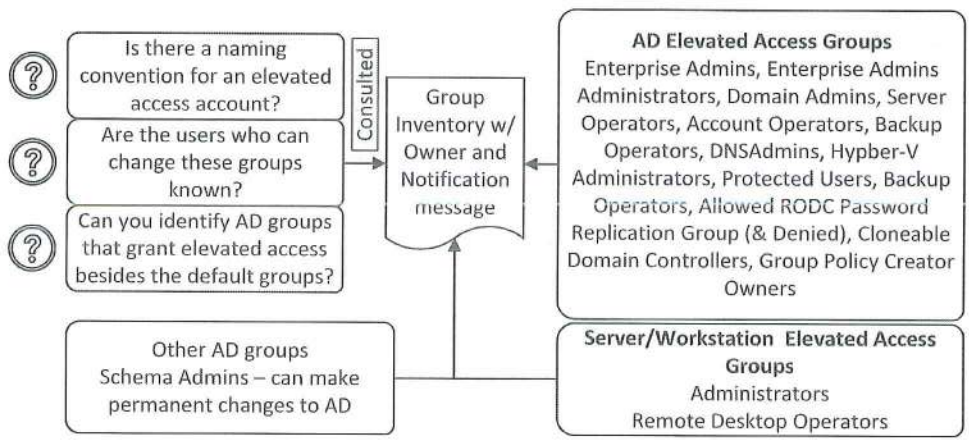


Figure 6 Maintaining Inventory of Elevated Access Groups

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 that group changes made by the service account are authorized, but changes made by others may not be and would trigger a notification to the security monitoring team.

Design Specifications and Discrete Objectives

Configure DCs for forwarding of local data with either a local agent, for a supplemental syslog feed tool like NXlog / Snare, the Splunk forwarder, etc. Also, you could configure a DC to accept a remote WMI call for remote collection.

Ensure that all DCs report into the SIEM and that there is periodic review of the number of DC's per domain so that SOC can ensure all DC's are reporting. A supplemental alarm to detect **Failure** to report within a short period should be setup with a notification to the operations team.

Build an easy to update process, such as an updateable spreadsheet tool and an update process for SOC to maintain the group/owner/purpose relationship. (implies access and a check at new group creation time if 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 / failure** (more fine-grained auditing is preferred with the Advanced Audit Policy). Once configured, also enable account management auditing on member servers and workstations.

Create a standardized communication template for the email-based notification. English will be fine. (spell check!)

Determine the account names of authorized users who can make a change, so you can detect if a non-account manager changes membership in a group. This behavior may change the notification template.

- A "change" that is implemented by a non-authorized user should go to the security team for investigation.
- The tuned notification message will be sent to the "group owner".

A long-term record of the notifications delivered should be maintained, **based on *the*** *owner who should have gotten the notification at that time.* Several options exist, such as having email CCd to a security-controlled mailbox or writing a record to a log file.

Security Operations Center Notification

The options implemented depend on the specific platform's capabilities.

1. Dashboard: Show the most recent group changes.
2. Report: Produce a report each day for the event ID's that correspond to the monitored groups (see Windows Group Life Cycle Events on page 86).
3. Notification Monitoring: SOC needs to monitor the sender account's email inbox for Non-Delivery Report Failure (NDR) messages. These would occur if the user's email account is deleted, or temporarily disabled, so that a secondary recipient can be found.

Use Case Component Name(s)

1. List of DCs which should be listed under "Domain Controllers" in Active Directory Users and Computers.
2. Notification rule for changes made in the domain, which will identify the domain controllers.
3. Notification rule for *local group* changes that do *not* come from the domain controllers; by default, this rule will be used.
4. Inventory List of monitored groups (the list should have the group name, the recipient who receives a notification, and a supporting comment).
5. Windows Event ID's for group changes: Group types – Universal, Domain, Global, Local – Create / Delete / Add / Remove

Use Case Data Source Description

1. Active Directory centralized audit events
2. Windows Member Server centralized audit events
3. Windows workstation forwarded events
4. Notification table (group name, recipient list, notification-reason)

Use Case Data Stream Analysis

The pseudo code for data steam analysis is listed here:

```
For each Domain Controller
      For each Group Membership 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
```

Cyber Kill Chain Analysis and Support

Cyber Kill Chain Phase	Notes
Reconnaissance	
Weaponization	
Delivery	
Exploitation	Membership in an elevated group can grant access to an outsider or unauthorized insider.
Installation	
C2: Command and Control	
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.
- As new elevated groups are created and used within an information system, the notification table will be kept current.
- Elevated group membership is "clean" at the point of use case implementation.
- A simple reporting function can be developed for sanity checking (DS query, for example).
- Email template will be clear enough to the recipient community.
- A group of authorized account managers can be built and maintained.

Alternative Solutions

- Monthly reporting was considered as an alternative solution. This was deemed ineffective because a monthly report allows for membership and access for an extended period of time.

Partial SOC Use Cases

Partial Use Case: Windows Network User Presence

Employee Investigation and Desktop Presence: One case type that 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 that relate to this use case: central logs from the domain, and the individual user's workstation event log. There are also likely to be dozens of other logs that can provide insight to answer this question.

Domain Controller: DC's will record initial logon and initial Kerberos ticket requests, but they do not record local logon/logoff events. Local events are recorded in the local security log with Event ID 4624.

User System: The user's security log on their system is where you will find the necessary events. When a user logs on to a Windows workstation they are assigned a session ID which must be used to track activity across a number of events.

To support working this use case, the SOC needs to run a report that identifies all access that an individual user name. Therefore, the SOC needs to understand how a user is identified in each system.

Partial Use Case: System Not Logging/Reporting

This condition can be detected in several ways. The easiest way is to review a daily report of how many events a particular system generated, and if it consistent to +/- 10% or a similar threshold, assume the device is "working". Once that process is in place, the next step is to implement an hourly report or dashboard view.

The more sophisticated version of this use case is to build a system monitor that looks for the "last event seen". If the time between the last event and the most recently observed event is past a threshold, raise an alarm. Or if there is a significant amount of time between data from a critical system.

For further definition of this use case review the Implement Synthetic Transactions section beginning on page 193.

Partial Use Case: External (VPN) and Internal (Desktop/Server) Access

A user should only login to a "console" if they are physically present in a building where the system resides. If a user logs in remotely over a VPN first, and then logs in to a console *within a certain time frame*, it may indicate that their account is in use by an unauthorized party.

There are accounts and some very specific systems which should be exempt from this relationship, such as a designated system administration account that logs into a data center KVM system which is actually plugged into the KVM on a server itself. In this case, it would be difficult to develop rules to detect this condition.

Partial Use Case: IDS Stacked Events

Phase One

1) A single source triggering multiple targets across a single event/alarm type.

2) A single source triggering a high number of repeats events across a single alarm type.

3) A single source triggering multiple events/alarm types for a single target.

Phase Two

1) A single source that triggers any alarm against a target, and then the target triggering an alarm within X minute(s), meaning that one system is successfully attacked and then becomes an attacker.

SOC Notification and Actions: The SOC team will review these alarm conditions, search for additional reinforcing indicators around the most critical events. If a pattern emerges that indicates that

Partial Use Case: Policy Violation Issues

Many IDS systems and Snort/Suricata rule sets can detect software usage that represents a *policy violation* but may not necessarily represent an intrusion that would warrant the SOC team investigating past verifying the accuracy of the alarm. From an *environmental awareness* perspective, these types of alerts provide additional color and awareness of how the users are consuming bandwidth. This particular use case is a great example of how the SOC function

can improve the operational posture of the organization by identifying high bandwidth consumers.

Also, don't relegate this type of analysis from the "IDS/IPS" point of view, or go to extra levels of instrumentation to get these answers through your SIEM. If your organization has a NGFW such as a PaloAlto system, then you can get a fantastic level of application 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 their NGFW that can be configured in "TAP" mode and give tremendous visibility to the security team without giving them access to the primary perimeter security control system.

In order to realize these uses cases, you would need to find the legitimate use of the application, adjust the rule set, and then enter a monitoring phase. For example, several software publishers such as RedHat makes software updates available via BitTorrent. Your company may have an Xbox or PS4 in an employee lounge.

In these cases, the SOC team would seek to validate the alarm as a true positive, write the summary, and provide it to the staff members management layer and also to HR in some cases. The point of note for these types of use cases is that they should be as highly automated as possible and not divert from true incident investigation. They can be handled in numerous ways. For example:

1. Go through the IDS ruleset, gather up the signature IDs that represent a policy violation for your organization, and get a daily report organized by source user or IP address. Run the report daily or weekly, look for a volume of activity that warrants follow up, and assign them as learning exercises to junior team members to develop the report.
2. Be careful with this one: automate an email notification to the subject's manager, once a threshold is reached.
3. Export these low-level application detections from your NGFW, review the identification by user, bandwidth consumption, and pattern of activity.

A Day in the Life of a SOC Analyst

This section provides a framework for activities performed and the 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 in order to resolve an alarm. If the first layer of analysts can quickly resolve a large percentage of the 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 related data to resolve an alarm. Both can be a never-ending exercise. 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, which are expanded on in this chapter.

Use the list below to provide a repeatable structure the duties for the SOC analyst to ensure that major areas receive some attention each shift or each day (depending on the task).

1. Perform A**larm Triage** Overview. The analyst should follow a priority model as alerts are raised. If the alarm is valid, the analyst may work the alert, collect some initial data, investigate, start a ticket, or escalate.
2. Perform a **Dashboard** review in order to maintain situational awareness.
3. Review **Security State** Data. This activity is focused on ensuring the proper data is coming into the platform, every day.
4. **SIEM System component** health review (daily).
5. **Identify and Report Operational Issues,** which puts the SOC in the role of being a good team player.
6. Perform **active threat hunting** by reviewing specific security data (daily).
7. **Review** security intelligence data, bulletins, postings, and other sources of current information and instrument into NSM and/or SIEM platforms.

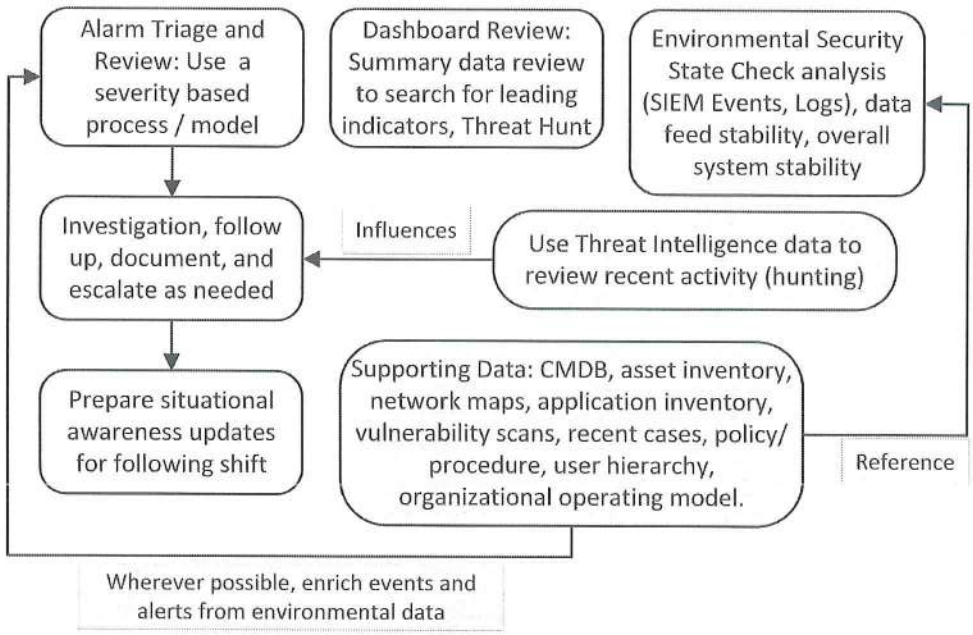


Figure 7 Daily Analysis Overview

Alarm Triage Overview

Analysts review and process the various alarm panes through a triage process, from the highest severity rating to lowest severity rating. The severity rating methodology, and thus the relative urgency rating and presentation, is SIEM specific. If severity is organized by the Cyber Kill Chain, then "System Compromise" would be at the highest level as this classification represents a successful breach to system security. As early as possible, an alarm needs to be categorized so that the right attention is applied to the alarm.

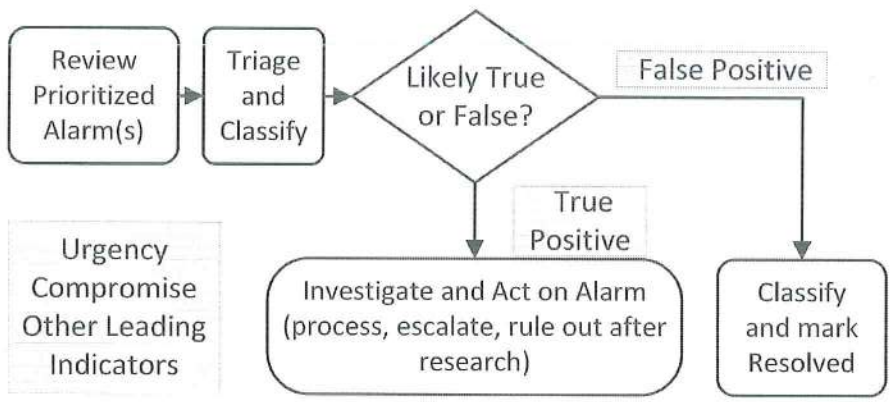


Figure 8 Alarm Triage Overview

Other SIEM's may use a composite scoring method that drives a color scheme where Red is the highest criticality and green is the lowest. The objective is that the highest priority alerts are reviewed and remediated first, and that a *reliable method* must be in place for the analyst to recognize severity and thus prioritize the alerts they work. See Severity, Priority, Urgency, and Reliability Criteria on page 195. The table below provides some examples of actions and examples that analysts can take in response to alarms. This is by no means an exhaustive list.

Table 29 Analyst Action Examples

Action	Example
Assess and close alerts that are *non-actionable 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 a *false positive*	A QuickTime alert from a version 8 years old, after confirming the target doesn't even have QuickTime installed or is current.
Mark alarm for Investigation (this may lead to any other classification)	NIDS alerts that need to be researched and reviewed against other data
Escalate the alarm when it is beyond skill level to assess	Tier 1 - PowerShell exploit code observed in conjunction with numerous authentication failures for the host
Process the alert, based on current skill level	Potentially unwanted programs or browser toolbars are observed and a cleanup activity will resolve the issue; generate a service desk ticket and mark the alarm as "under remediation".

Each SOC will need an outline to determine which alarm gets the most attention, what issues are higher priority than others, and also keep a technology inventory on hand to confirm the validity of an alarm.

As an analyst takes an alarm for review, there is an underlying process support reason why the analyst should immediately mark the alarm as "under review" and "by whom". SIEM platforms constantly have to serve many requests – accepting data, managing it, making decisions, building alarm displays, answering queries for reports, and performing real time analysis. A system 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 determine a method to internally coordinate who is processing what alarm in order to minimize potential system impact and to let others know an alarm has eyes on it.

Dashboard or Summary Data Review

1. Monitor the alarm data, system, and environment using a "Top X" to a "Bottom Y" point of view, and then reverse the order. When data is evaluated using the seldom few or the tail of a volume-based curve, it's called Long Tail Analysis (LTA). There is tremendous value in the singleton events that exist in your environment. The objective at this stage is "big picture" awareness to make sure that single events are not lost.
2. If it's available in your system, check for Threat Intel activity on the summary dashboard. Threat Intel activity can be classified on multiple dimensions: 1) IP or name reputation, and 2) validated threat data, such as a known malicious domain or file hash of a malware sample, or 3) brand issues. The objective is to quickly focus attention on current, *known, validated* threats that appear in the environment, investigate, and remediate if the alarm bears out. In other words, threat intelligence data sources are useful indicator, but the analyst should ensure it is current.
3. Check the current period SIEM and Logger event groupings, or "event types", for new event types and a lack of event types that should be seen. As any environment changes, so will the security event data. For example, a new event type may appear if a system module is enabled, a problem is corrected, or an upgrade occurs.
4. Inspect assets for vulnerabilities that may have appeared.

Security State Data Review

Numerous systems report events which never raise to the level of an "alarm condition" based on the tuning and parameters of the SIEM system, the SIEM may not have rules for the event, or with knowledge of multiple sources a group of disparate events may be an incident. By reviewing the event stream the security operation function can detect potential incidents, can validate that reporting systems are functioning, detect anomalies based on significant volumetric changes, find errors, and – perhaps most importantly – validate that reporting systems are functioning.

Critical Device Review: Review outbound traffic, internal scan or alerts that were directed against or related to these systems, log volume, event variety,

account management, and vulnerability status. Search NetFlow to determine if any new port/system combinations appear.

Validate data health: The SOC team must have a method to ensure that all data sources that *should* report to the SIEM platform *are actually* reporting, reporting in a timely manner, and reporting accurate data. There is nothing worse than working an alarm or an incident, looking for data, only to find out that it is not available. It is worse when the data has been missing several for days. This breakdown condition can occur when the source systems' reporting profile changed, a source system was reinitialized, the collection system failed, the system was taken out of service, the data communication path is broken, or the source system had an upgrade and the data format changed so it is not currently parsed by the SIEM.

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

1. Automate an alarm condition so if the typical reporting profile changes significantly, the SOC is alerted. For a host defined in the SIEM, there should be an "is data expected" attribute that indicates that the asset should provide data not less than once per day. Thresholds will be variable – not every data source report at the same volume.
2. Rules or external scripts that monitors for existence of the data at the log collection point. For example, syslog sources write to a file, so it would be easy enough to script out a "if file exists" and "if file is growing" check.
3. Perform periodic manual review, once per day, *of the variety of data arriving and parsed in the SIEM from the source system*. This is more than a volumetric check. It means that an analyst looks to see that the source is providing the expected variety of data based on its specific profile. For example, if a perimeter security appliance reported VPN access activity for the past month and that activity suddenly disappears but other data is arriving at the same volume and velocity, the analyst should flag that condition for follow up.
4. Create an artificial method to ensure a data source is operational, as described in Implement Synthetic Transactions on page 193.
5. SIEM and Logger event throughput and volume should follow a "regular pattern" for the environment without huge swings in rates. For example, most environments would experience a spike for 30 minutes around the beginning of the work day, and perhaps 40% of typical event flow during holidays from low staff levels. Remember, volumetric checks are *indicators*, not a sign of intrusion.

SOC Support System(s) Component Health Review

When the SOC team is not responding directly to alerts, they should take on other support and maintenance tasks. There should be a system health check conducted at least one per day, as described below.

1. Review daily reports, script output, or dashboards for general indications and system component issues. Examples of these checks include:
    a. Trend of disk space consumed in the aggregate, over time, and in specific directories on the SIEM systems.
    b. Disk space consumed during the day. Plain text logs can easily grow very large before periodic compression which usually occurs overnight. If this metric routinely reaches, say, 80% of a particular volume's free space, there is risk that a high-volume event storm can have a severe negative impact and bring the SIEM to its knees.
    c. Long running queries. If your SIEM solution, or other SOC support systems use a relational database management system (RDBMDS), these are called Data Manipulation Language (DML) queries. Start with checking for queries that run for more than 120 seconds, and work to improve. The goal is to keep query time such that the system itself does not impair analysis.
    d. OS based virtual memory swapping that causes an impact. Occasional swapping isn't likely to be an issue. Consistent swapping indicates there isn't enough memory.
    e. Memory leaks, which show up as the memory devoted to a given system or process increasing over time.
    f. Report execution times increasing over time.
2. High Volume Data Drop. There are numerous high-volume data sources, and numerous event types from those sources. By comparing the volume and event distribution on a day by day basis, SOC is providing assurance that the components are working and there hasn't been a significant change in the source data profile. Environment issues can also be detected and should be checked against "Known Changes". For example, if the perimeter firewall normally reports 12 million accept or deny events for the web servers in the DMZ and that suddenly drops to 1.5M, or grows to 24M, then something obviously occurred. These significant shifts in event volume could occur when a new web server was stood up, or the primary webserver was moved "to the cloud" and SOC had no foreknowledge of the change. To continue with this example, if the perimeter firewall stopped reporting accept and denies for the primary web server because it was "moved to the cloud" and

there is no compensating monitoring capability enabled *before the change,* then SOC must inform IT management that a detrimental condition has occurred in the team's ability to monitor and provide assurance services to the business.

3. "Low Volume Source Testing". There are several devices and conditions that report infrequently. An example is a malware analysis or detonation system like a FireEye AX. These low volume devices should be well known and there should be a simple test to ensure that they do report an observed condition in a timely manner. Another example may be an audit condition, such as granting a user a sensitive role, in a specific application. By ensuring that these "low volume" conditions are known and can be periodically tested. SOC is actually providing a valuable business service by ensuring the protection and detection mechanisms are functioning, compliance objectives are met, and that last uncoordinated or unannounced "low impact" system upgrade or change did not actually break the security platform's ability to monitor.
4. Check threat feed activity to ensure that your SIEM has current data, and that the number of indicators is what you expect. For example, the Open Threat Exchange (OTX) categorizes major threats as "pulses". The count varies each day and is usually within about 2% to 3% from the prior day.
5. Report shift summary data and turn over for next shift.

Identify and Report IT Operational Issues

Event data review can also provide *operational awareness*, point out issues, and be used to keep systems running well. Here are several issues that the SOC observed and reported, based on real life experience, which when diagnosed and reported improved daily operations.

1. Detecting active directory replication issues due to a rise in Kerberos error condition events from one particular DC. The DC was failing so authentication requests started timing out across the WAN.
2. A 1200% increase in VPN authentication failures, which identified numerous configuration errors from a failed update due to human error.
3. Misconfigured whole disk encryption software due to excessive 'administrator' logon failures at a rate of 4.5M 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 domain controller to go to high utilization. High utilization affected end users resetting passwords because the DC in question happened to have the PDC Emulator role.

4. MTU network mismatch sizes from excessive ICMP error messages that were returned, which lead to understanding why email wasn't flowing to a location in Europe on a network that had an artificially small MTU size.
5. Being able to track down devices using cached credentials which were causing their accounts to be locked out when the 60-day password cycle time hit and they updated their account. This is the case where "My 'device' knocked me off the network".
6. Systems which reappear on the network after an extended period of time and need to be updated or reenrolled in the domain.
7. Services failing because the developer *used their own account and didn't create an authorized service account* (this is more common than one might think) and the developer quit. Note that this particular condition is often tied to a critical business process, so when the SOC finds problems of this nature, the SOC actually helps to restore a business process.
8. Occasional bluescreens or unplanned reboots (Windows System Log, event ID 6008).
9. Wireless AP configuration errors, which show up as a significant rise in TACACS+ or RADIUS failures, so much so that the devices had trouble authenticating real users.
10. Software deployment errors, which may appear as a rise in errors form the application log, a webserver log, or a Java application server log.
11. Failed backups, system wide, which could be catastrophic if an issue occurred because the one backup / recovery person was on vacation (they were.)

Active Threat Hunting

SOC Analysts can also perform threat hunting, which is a great way to vary their work load and keep them interested in the job role. See Applying Threat Hunting to the SOC on page 171 for a discussion on this topic.

Review Security Intelligence Data

There are numerous sources of security intelligence such as vendor bulletins, the various SANS newsletters, AlienVault OTX bulletins, and security focused websites. Senior staff should identify what intelligence sources are useful for the SOC, rather than every analyst and subscribe a centralized mailbox shared among the SOC, rather than each analyst subscribing. This method provides a centralized source to search for keywords that can provide significantly improved search results than just hitting Google. Even Twitter feeds from well-

known security professionals can be a solid source of threat, exploit, and vulnerably data.

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 (see p. 184).

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 record keeping process to mark them as false positive or 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 the analyst gather the most recent 24 hours of security event data, checked 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 it may be very easy to close some tickets and there is nothing wrong with closing a "high value" alarm if the analyst can classify the ticket as something other than an "incident".

Alarm Investigation and Processing: As alerts are reviewed against the SoP, some can be easily handled and some will require investigation. 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.) Several investigation activities are listed here for common data sources that feed a SOC and a SIEM.

NOTE: As data is reviewed, it needs to be recorded to support the *incident timeline*.

Techniques and Analysis Methods by Data Source

1. Process information (4688) and sysmon: Process information is highly useful in reviewing an incident.
    a. What processes were executed for the hour perform the alarm?
    b. What command lines appeared?
    c. What network connections did a process make?
    d. The other items below can also be adapted for process data.
2. Endpoint Detection and Response (EDR): These systems 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:
    a. "First Run Binaries" – an executable that has not been seen in the organization.
    b. Active and recent network connections, particularly connections outbound to the Internet and the source application.
    c. Watchlist hits and submissions. EDR packages can compare real time data against an inventory of known IOC's, effectively automating this function. Any watch list submission against a threat source or threat catalog, such as communication to a poor reputation IP, should be investigated.
    d. Files executing from 'temp' directors.
    e. Files executed from a browser or an office application.
    f. 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 a cmd.exe process.
    g. Connections from a document type or executables from a document type.
3. Recent DNS queries and Responses[52]:
    a. Did the suspect system generate more than a few NXDOMAIN responses? Users do make typos, so a few are OK, but they should be followed with the right domain spelled properly and name type-o's should be obvious.
    b. Consistent DNS communication to a specific domain, *most often not in the top 1M domain lists,* or a newly created domain (< 30d old). A flood of outbound random A record queries that come back with random CNAME response records is a telltale case of DNS tunneling.

[52] One of the better SANS RR papers on this topic is "Detecting DNS Tunneling" by Greg Farnham. https://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152

   c. DNS queries where the hostname domain name (not the TLD) score low for the entropy score, as assessed by Mark Baggett's freq.py[53] tool.

4. Network Intrusion Detection/Protection System (NIDS/NIPS):
   a. 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"?
   b. What was the composition of the alarm pattern? Do multiple events stack, relate, or reveal a pattern?
   c. Are other systems on the same segment generating the alarm as the suspect?
   d. Did the suspect generate NIDS alerts *after* the current alert? In other words, was it compromised, and then used to compromised 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.
5. Perimeter Firewall and other session-based sources:
   a. Which IPs on the Internet has the system communicated with? How many of them have reverse DNS entries? Do any of them have *recent* poor reputation and appear on a threat intel feed?
   b. Has the inbound or outbound profile for the suspect changed day over day or, if possible week over week? For example, an uptick in events, new ports observed, or a change in the baseline of denied outbound traffic? A new protocol?
   c. Has the activity profile (events per hour) significantly changed?
   d. Are there outbound ports or protocols (like SCTP) in use that the suspect IP doesn't normally use?
   e. Which systems external to the suspect's network segment communicated to the suspect in the past hour? Day?
6. Proxy (Web filter):
   a. 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.
   b. What sites were denied, observed for first use, user override click through allowed, 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 user to confirm going to a site. A daily check of these sites may reveal a security threat.

---

[53] Review the SANS Blue Team WIKI for usage: https://wiki.sans.blue/Tools/pdfs/freq.py.pdf as well as Marks Github site.

   c. Perform top one million site checks as described on page 112.
   d. Is the balance of proxy traffic on par with your sites profile? When in doubt, think 20 to 1, meaning that 20x the data coming back in from a web request than goes out *based on the data payload*. This occurs because users send small amounts of data, while server respond with large amounts.
7. Authentication sources (AD, database, application, email. ...):
   a. What user accounts authenticated *from* the suspect IP in the last hour? Day? As a corollary, is the suspect a "shared asset" like a RDP jump box, a Citrix desktop server, a database, or some other system that authenticates users?
   b. How man success and/or failures have come from the suspect in the last hour? Day? (For Windows, these are Event ID 4624 and 4625)
   c. As a point of reference[54], the typical organization has between 120 to 570 AD authentication events registered per user, per day, with most of the requests ranging between 300 and 450 (middle region of the bell curve). You should establish similar metrics for your own organization.
8. HIDS (such as OSSEC, sysmon[55], 4688 events, and OsQuery):
   a. Have there been registry key or file system changes that cannot be explained?
   b. Are there process command lines that are suspicious?
   c. Are there shell or scripting processes being executed from office productivity applications (Word running CMD.exe which then starts a PowerShell script)?
9. Asset History:
   a. What are the types of events and alerts for the source and/or destination asset?
   b. Is this a first observance for an asset – on a non-DHCP assigned (fixed IP) network space?
   c. Does the asset by name have the same IP address? This is particularly relevant for networks identified as dynamic, or DHCP assigned.
   d. Is the asset (host) under recurring attack?

---

[54] 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.

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

Alarm Classification: Alarm research should result in several actions, such as:

1. 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 of time until the underlying rule can be improved.
2. Mark the alarm as under investigation, keep open for a period of time, in order to research an issue and keep the alarm visible to the SOC.
3. Process through the ticketing system for remediation as soon as possible.
4. Temporarily suppress the alarm, such as when an alarm storm occurs, while an issue is being investigated.
5. Close the alarm is a false positive *with sufficient notes to explain why the analyst classified it as a false positive*.

Performing Well Rounded Alarm Analysis

There are several threads that can be pulled based on events from the network and operating system ecosystem when investigating an alarm and draw conclusions. The primary objective of alarm analyst is to shorten the "Mean Time to Disposition" for an alarm: is it a True Positive or a False Positive? Other actions can result such as tuning. But when it gets right down to it, a SOC analyst wants to validate an alarm as false first, so that they can focus on the ones that may be true. As an investigator, the analyst must deal with several factors that compete for attention. Analysts should make every effort to avoid spending too much attention down one investigative path at the exclusion of others, particularly when another path has better source data to determine if the alarm is true or false. Further, the analyst needs to pull out key supporting details that prompt them to pivot from one data source to another in order to validate a true or false hypothesis for each alarm.

At the Security Onion conference in 2016, Chris Sanders[56] who runs Applied Network Defense held a presentation titled "The Investigation Labyrinth" where he made some excellent points. Chris provided quantitative analysis of how the initial decision, or the opening move, during the analysis process affect the close rate and time of an alarm. Sanders' research used several different groups. He provided the analyst groups with data sources that most SOC teams would have on hand[57], that is integrated into this section.

---

[56] http://chrissanders.org/2016/09/effects-of-opening-move-investigation-speed/

[57] Many of his statistics are listed below based on his research are in this discussion, with his permission.

Alarm Investigation Process

The first stage in response to an alarm is illustrated here. The first stage is the opening move, or the immediate triage phase where quick decision is made whether or not to investigate the alarm and if so, what are the primary data sources that aid in responding the alarm. Note here that the analyst needs to prioritize which alarm they will process – usually an urgency value drives that decision.

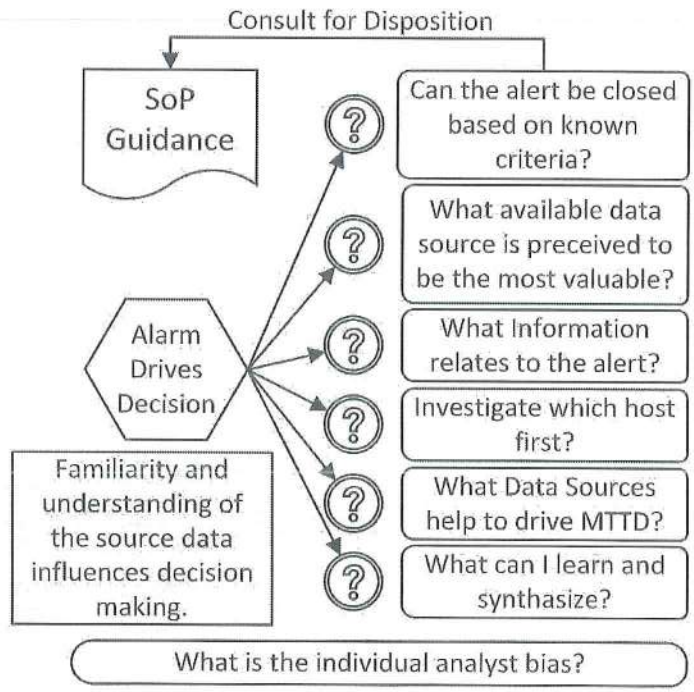


Figure 9 Decisions Driving the Opening Move

Assuming the alarm will be investigated, then there are decisions on what data to retrieve and how to go about getting that information. If the analyst is disciplined about the investigation process, they will start retrieving data from multiple sources as the retrieval process can take several minutes. There are dozens of data sources to go after, each with their own level of context. For example, firewall logs for the source and/or destination, supplemental alarm data, NetFlow, packet capture (if available), intrusion detection system data, anti-virus data, proxy server data, and launching an investigative tool on the interior host itself. While data is coming to them, the analyst should research the alarm name. Analysts needs to determine what conditions must be present for the alarm to be a "true positive", which may be answered by proving the converse – some fact data present shows the alarm is a false positive. Remember that each decision to gather or review a data source informs following decisions and the various investigation paths.

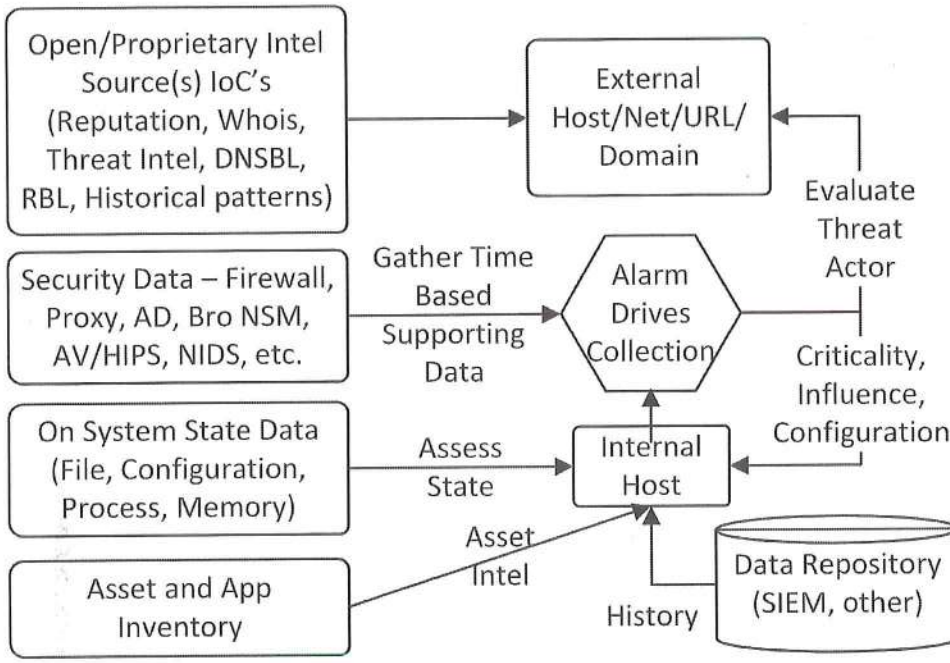


Figure 10 Review Data Sources

Once data arrives then the analyst needs to synthesize that data, which really means they may need to review dozens of disparate information sources, mash it all together in their head, pull out the common threads, and either close the alert, continue working it, or escalate. This is no easy task!

Once data source synthesis occurs, then some action usually takes place as illustrated here. Actions can range from trigger a network packet capture, push an A/V update to the host, put in a service desk ticket to remediate the host, begin an incident report and start preserving data, notify the shift supervisor of a possible HR/Legal incident – there are dozens of possible actions that can take place.

Sanders' research found that if an analyst attempts to prove the alarm is valid, they take two thirds times more on an alarm than the analyst who seeks to disprove the alarm is valid. This significantly affects MTTD, and explains that proving the "negative case" is more efficient use of time.

Ensure that if an analyst is going to work with one data source, use is the one most likely to aid in providing a resolution, with the intention to push towards a false positive conclusion. Most seasoned analysts (including myself) do prefer the highest context data possible, and we may overlook or deprioritize lower context data that may have led to a faster alarm resolution or conclusion thus improving the Mean Time to Decision (MTTD). Note that immediately working

the PCAP data first is a tendency of many analysts. Sanders has found that starting with PCAP data will significantly increase the time to close the alarm. Further, Sanders found though his research that 72% of analysts prefer to review high context, but rather unorganized, PCAP data early in the process. While packet capture data may provide very high context, flow data and intelligence data will take less time to synthesize and aid to resolution, like those provided by the Bro IDS system. When PCAP data is replaced by reviewing Bro logs instead of high context PCAP data, the Mean Time to Dispassion (MTTD), or the average time to close an alarm improved 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 actually makes the case that workstation and server data provided by detailed process auditing through detailed process tracking for Event ID 4688 and sysmon can decrease alarm closure time and improve accuracy.

Any effort expended to organize data and to script routine analysis steps will pay off time and time again. Automation tools like Kansa (written in PowerShell), Windows Forensic Tool chest, 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 can be more authoritative than the network. It's filesystem, configuration directory or registry, memory contents, vulnerability state, and process inventory reflect the on system operating state.

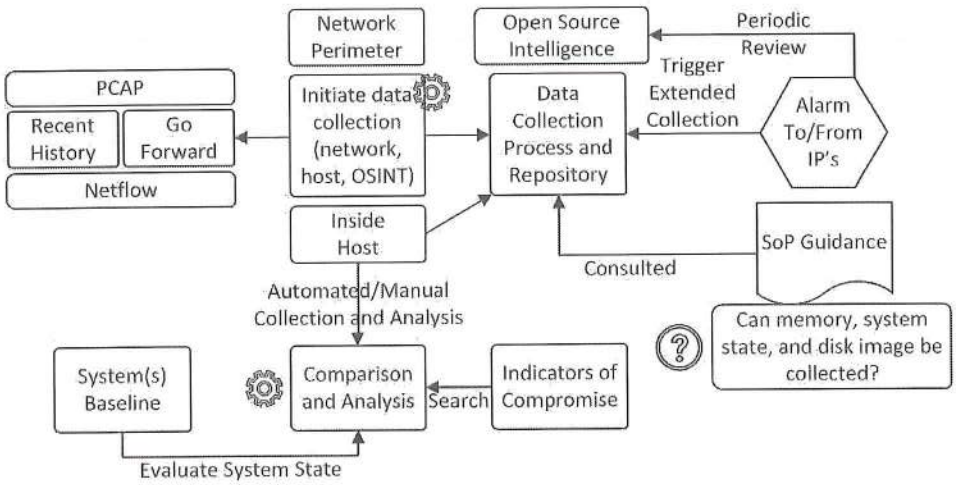


Figure 11 Data Analysis Processes

In Effect: The more steps taken during the analysis process, *and the order* of taking those steps, will significantly affect the time to close *or* the time to declare a serious incident. Frequently, the analyst will need to collect a variety of other data sources to better answer questions. These actions will add to the time to make a disposition.

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

Skill Development Moment: Cognitive Bias Awareness

One particular 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, this "is a systematic pattern of deviation from norm or rationality in judgment[58]." 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[59]."

The analyst has a key takeaway from these two definitions: the analysis process can be adversely affected by one's own perceptions, thoughts, and experiences such that a perceived notion or thought will remain intact *even when the fact*

[58] Definition is from "The Evolution of Cognitive Bias" by Haselton, M. G.; Nettle, D. & Andrews, P. W. (2005).
[59] Definition is from "Building an Intelligence-Led Security Program" by Allan Liska, published by Syngress (2014).

data presented surrounding a case change. Cognitive bias lead to perceptual distortion, interpreting data incorrectly, and faulty judgments.

Analysts counter their own bias by gathering as much fact data as possible in as timely a manner as possible about a given case. Make sure that each of those pieces of data is on the table to draw a true positive or false positive conclusion. As a compensator for cognitive bias, analysts should place discovered fact data in a normalized timeline. This process helps to lay out the facts of a case in order. If necessary, actually test the hypothesis in a sandbox or some other isolation environment.

Skill Development Moment: Graph Theory vs. 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 12 Graph Theory Illustrated.

Several papers exist that explain how to apply graph theory to cyber security. In order to be successful, both attackers and red teams must build graph-based relationships as they empirically make discoveries, in order find a path through the network so they can act on their objectives. Once an attacker has *any* sort of a 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[60] by John Lambert of Microsoft explains this concept (paraphrased here):

"As defenders, we tend to think in a list: the list 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 have to go about a process of learning how one asset or user is connected to another, often by network connections or security relationships as they look for the next island".

[60] Ref: https://blogs.technet.microsoft.com/johnla/2015/04/26/defenders-think-in-lists-attackers-think-in-graphs-as-long-as-this-is-true-attackers-win/

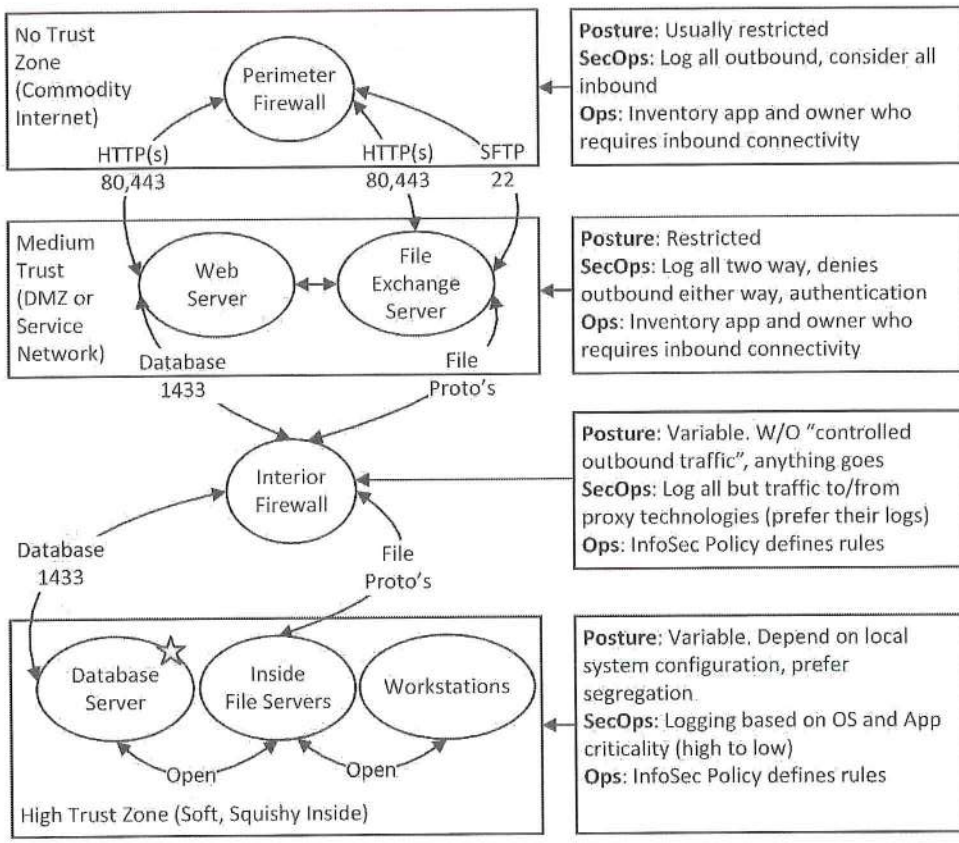


Figure 12 Graph Theory Illustrated

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 in order 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 from pivot to pivot.

In order to apply graph theory to threat hunting, the defense team should think through and answer the question "where the attacker can go next from the affected or identified system" and "how they can get to the affected system". These two points focus attention on the trace data from the network. In essence, SOC should build its own graph as they work through a case.

Alarm Statistics

There are several studies that provide some revealing statistics when it comes to alarm management. You can use this information to benchmark your own

SOC capability, and also to understand some of the measures developed to describe alarm management effectiveness.

Alarm Statistics from the Cisco 2017 Security Capabilities Benchmark Studies

Cisco's 2017 78-page report[61] has a wealth of statistics relating to alarm volume (and several other areas). This report is well worth the read as it presents quite a bit of data about the size and composition of security organizations, and it can help you to measure your own SOC. 2900 respondents in thirteen countries were included.

1. 28% of all *investigated* alerts were legitimate, with 56% of all alerts were investigated on a daily basis across the population
2. 44% of security managers seeing 5,000 or more alerts per day.

Alarm Statistics from FireEye 2017 Report

FireEye commissioned IDC research to collect data on alarm processing in large enterprises. The report[62], titled "The Numbers Game", provides some key statistics on alarm volume. This report is well worth the read, not only to help you understand the volume of alarm data but also to measure your own SOC.

1. 37% of respondents indicated they face over 10k alerts/month.
2. More than 35% of companies say they spend 500 hours per month responding to alerts.
3. 48% were actual malicious events, while 52% were false positives, with 64% of all alerts were redundant.
4. 75% of all "critical" alerts were responded to in five hours.
5. 62% of the largest organizations review security product configurations to reduce the alarm volume on a monthly basis.

[61] https://engage2demand.cisco.com/LP5681_ty
[62] https://www.fireeye.com/StopTheNoise-IDC-Numbers-Game-Special-Report.html

Applying Threat Hunting Practices to the SOC

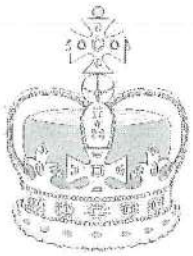
Threat hunting, for the purposes of this book, is defined as "leveraging information to proactively search out and identify if an attacker was successful in compromising your network, applications, data sources, or systems on an iterative basis". In effect, threat hunting seeks to *proactively* leverage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

The term "threat hunting" became popular in around 2011. In 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.

Threat hunting is a matter of discipline. It begins with establishing a hypothesis or a testable condition that is used to find a compromise. When it comes to making use of data sources, use as many data sources as you can, and work to determine the current network and system state so analysis against that known baseline can effectively locate threats. For example, establish your baseline of typical daily account lockouts. When this value is *significantly different* from the norm, the condition should be investigated.

Threat hunting takes a longitudinal approach to alarm, event, and activity data, as opposed to alarm monitoring which is focused on reviewing and coming to a disposition on the truth or falseness of

the alarm itself. Hunting can be performed several different ways. One analyst can hunt one day per week, or one person can hunt for an entire week once a month and rotate to another analyst, or some combination thereof.



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).

The SANS Institute published a paper in August 2016 titled Generating Hypotheses for Successful Threat Hunting by Robert M. Lee and David Bianco. This paper makes several key points. A summary of Lee and Bianco's key points from this particular paper are listed below, slightly adapted and combined for presentation here in BTHB:SOCTH.

1. An analyst's ability to generate a hypothesis is based on observations. A hypothesis is derived from threat intelligence, situational awareness, or domain (environment) experience.
2. Hypotheses must be testable, grounded in reality, are reusable, and need to be updated over time. Guard against personal bias in developing a hypothesis.
3. The hunter must know the data and technologies at their disposal.
4. The use of IoC's and Tools, Tactics, and Procedures (TTP's) of an adversary have entered mainstream cyber security[63]. Using an IoC may not lead to a formal hypothesis but may lead to alerts and further investigations.
5. IoC's are not a panacea. They should be used in context as a tool. Context is important to properly using an IoC. Many IoC's, such as domain names and IP addresses, have a shelf life to them. Further, what is an IoC for you may not be an IoC for another organizations network.
6. The paper presents a Crown Jewel Analysis process. In essence, this process means you identify the most important data sources and applications, build attack graphs and attack pathways in order to inform monitoring capabilities and hunts.
7. People, process, and the business environment are critical to the organizations threat landscape.

Once you understand your baselines and the data at your disposal, you can develop models, tests, and other hypothesizes about the detection measures that can be used to initiate a threat hunting effort. After the hunt activities are defined, various platform tools can be used to search for threat indicators. For example, there are several highly capable Endpoint Detection and Response systems available that can be leveraged to hunt for security issues on systems.

[63] AlienVault's Open Threat Exchange is an example.

These tools allow for broad analysis of the operating environment in use on the Windows platform: registry changes, file system changes, IP address communication patterns, first observed binaries, binary comparison by hash to known malware databases or not found in the corporate image, and a host of other process analysis practices. If the hunt team finds something, they would trigger an incident, desktop clean up, or desktop reimaging process as appropriate.

Threat hunting benefits the organization in several ways. First, hunting *maximizes* all of the security spend through data mining, analysis, reporting, and improved alerting. Second, hunting can also detect deviations against normal system operations and error conditions can be detected through summary data review. Third, through the practice of keeping a close eye on the environment, adversaries can be detected earlier while damage control can be more effective, with the specific objective to reduce 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 automations and improve alerts. Fifth, standing up a hunt team is a great way to provide rotation for the SOC analyst so that they do not spend their life reviewing alarms.

Here are some examples I have implemented since 2004.

Leverage understanding of what is normal user activity: People are, more or less, creatures of habit. Therefore, their system usage patterns will follow. For each of your data sources, develop a profile of the "average user" (you may actually have several) based on the data you will see from that data source. For example, how many authentication events occur on the DC's and over what timeframe does the "average user" generate. Essentially, this part of a Threat Hunting capability needs to have a set of norms to compare against, based on the user population. Note that a "server" is a special purpose user – and severs tend to be much more well 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, on a daily basis, look and see which internal IP and what port or service are being denied outbound. This simple technique allowed me to catch users with attack and recon tools, IRC based chat bots on portables that came back onto the network from an extended absence, a misbehaving user or two, and misconfigured systems. In particular, we would find systems with invalid NTP settings, users who manually configure their own DNS servers, unauthorized SMTP servers, use of a SSH encrypted relay, among other things. Once your

organization has confidence in this practice, make up a dashboard in your SIEM for more continuous or real-time monitoring. Each of these examples can uncover something of note.

System Stability is a great source for threat hunting: We even identified a financially significant system that had a disk failure because it consistently BSOD's at 2 AM during the A/V scan. The system as so old it wasn't enrolled in the monitoring platform, yet it ran a critical business process. True this particular is not a pure play infosec issue; rather, it is a great example of how the security team was operationally relevant to the organization.

Outbound Threat Intel Contacts: If any system reaches out to known IP, domain, or URL based on your threat intelligence feed, investigate it.

Anomalous Device Communications: Devices normally communicate in well-known patterns, such as ephemeral TCP to 80/443 for web traffic, instant messaging (MSN uses 1863/443, while Skype defaults 23399), and another explainable user/server traffic. Once the SOC teams understand what's normal, then they can effectively find 'not normal' by reviewing reports out of the system itself. This is an example of exception reporting, as in "everything except the 20 or 30 known TCP/UDP ports", the SOC team should check this daily.

Web Proxy Block traffic: In one environment, the list of users and sites blocked by the proxy is normally low (>10 users, > 500 blocks). When either of these dimensions were three times larger, investigate the condition because it is more than 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.

Leverage the MITRE ATT&CK Framework

To directly quote the site[64]: "MITRE's Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™) is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary's lifecycle and the platforms they are known to target. ATT&CK is useful for understanding security risk against known adversary behavior, for planning security improvements, and verifying defenses work as expected." Or put differently, the ATT&CK framework provides a basis for you to understand the attacker can successfully enter a network, establish persistence. Communicate and fly under the radar, and eventually act on their specific objectives. As you navigate the matrix you need to review the technique, determine your level of

[64] https://attack.mitre.org/wiki/Main_Page

logging or defensive overage, determine the likelihood of a successful attack, and guild in a mitigation.

As an example: there are several different spearfishing attack methods that can provide an attacker with initial access into the network. By understanding this attack vector and the organizations susceptibility to it, there is a clearer need for email focused security awareness education such as an anti-phishing campaign. After that, improve technical detection capability such as configuring sysmon on Windows systems to detect scripting languages invoked from a productivity application. Third, justify technical protective systems such as a web filtering proxy, and anti-spam solutions. If those systems fail and that particular vector is suspected, then IR staff should understand how to investigate email content and browser history. If your organization has completed all of these steps, then ensure that these systems are operating properly, logging centrally, and SOC understands how these data sources work.

Example Threat Hunt Check List

1. Prepare and execute threat hunting
    a. Search for signs of Command and Control
        i. Look for beacons using a tool like Real Intelligence Threat Analytics (RITA) by Black Hills Information Security, with its patented analysis engine.
        ii. If you don't have RITA in place, then review the top 20 IPs with the greatest number of connections, the longest connection time, and the most about of data moved. Ensure that any system in all three lists has a well understood communication pattern.
        iii. Look for long running transactions (> 8 hrs).
        iv. DNS responses w/ high entropy domain names.
        v. Unknown user agents observed.
        vi. SSL interactions w/ known-malicious / self-signed sites.
        vii. Dynamic DNS queries to D-DNS providers.
        viii. Long DNS queries, DNS txt queries, excessive DNS failed queries.
2. Observe a potential adversary as they would go after your "crown jewels"
    a. An adversary is after sensitive / valuable data (operate on objectives), 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 moves laterally, so search out 4624 authentication

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

c. Today will "live off the land", meaning PowerShell and leveraging built in commands. Review the output of sysmon and 4688 data for the invoking process, the invoked process, and the command line used for PowerShell and cmd.exe processes.

3. Leverage strong "egress detection":
    a. Document which systems *should* be used for specific services and look for systems that violate those rules such as DNS, FTP, email (SMTP, IMAP, etc.)
    b. Monitor all DMZ assets for initial outbound attempts – they should normally respond to inbound, if there are outbound it should be very well understood.
4. Monitor privileged accounts, meaning that you get the current membership of elevated groups and then review actions taken by these users (in the aggregate). Activities like scheduling tasks should clearly relate to system management.
5. Ensure that account life cycle events to elevated groups are fully monitored and supported by job roles.
6. Newly Registered Domains – there are several sources such as whoisxmlapi. Cost for these services runs about $100/mo. Conceptually: pull the list at 1 AM and run the prior day's queried domains against this list from your proxy or URL filter to determine if a user successfully connected to one of these.

Hunting Historical Data Based on Current Intel and Alarms

Various sensor systems like a NIDS are kept current through rulebase 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 rulebase (formerly SourceFire VRT) or Suricata running the Emerging Threats Pro can detect a wide variety of network conditions in addition to malicious software, shortly after its discovery. Once a script like PulledPork runs and updates the rule base, the NIDS will have the new pattern.

Take conditions that are updated in the daily rule update feed, and then reprocess the last 3 to 7 days of PCAP data, or another appropriate data source. For example, if a new malicious IP is identified, then compare that IP to recent firewall log or Bro connection logs. Get DNS name, and, if you have DNS logs, go through and find systems that queried for that domain name, or compare newly indefinite DNS names with recent proxy server logs.

An Example: When working a case about fifteen years ago, our team found a group of 20 infected systems that had software on them that "clicked" the URL's for hundreds of foreign website's banner ads. This was back in the day where advertisers readily paid for clicks on banner ads. We found the most relevant IoC, and then analyzed the prior months data. We found several groups of 20+ systems that each spent 3 to 4 days "clicking", and then the attacker stopped using those systems and moved onto the next group. Net effect: this was a revenue generation scheme where the attacker was "hopping" from group to group in order 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 setup, 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 their primary workstation, maybe a training computer, a secondary computer, or tablet/smartphone device.

For example: a few IPs visible externally to the Citrix site – maybe two, within a 30-minute period, but certainly not 25. A few internal addresses, such as their primary workstation, the training room, and maybe 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, load it up in Excel, and create a pivot table. Then sort in descending order to see if anyone logs in from more than a few addresses.

Lateral Traversal: Lateral traversal can be detected using this this technique as described in Lateral Movement or Lateral Traversal on page 180.

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 1:10 to 1:20, depending on the site. For example, a user entering and interacting with a SaaS application enters data, runs queries, maybe uploads some data, and often downloads reports or other output. The typical pattern here is a small amount up, a large amount back. When systems violate this pattern, *especially if they violate it outside of normal business hours*, you have something like data exfiltration. We've found people sending data wholesale up to file sharing sites and other examples of data exfiltration.

Command and Control Detection

There are several methods to detect C&C. In particular, users browsing habits do not generally have regular, definable "heartbeat" access patterns and have a significantly higher ratio of data received from a web server as opposed to the amount sent by a browser. Users click, read a little, click some more, and then often go onto another website. In contrast, C&C communications patterns have some rhythm to them – they pulse, beacon, or communicate following a regular pattern. C&C botnets have used messaging systems such as Internet Relay Chat and short strings to send/receive messages in plain text, which eventually became encrypted text. Then other methods evolved such as 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 API's, such as Twitter and FaceBook have created C&C networks by reading profiles, posting comments, "liking" articles, enticing users to grant access to their pages – just about anything a user can do with a site can be automated.

One of the more sophisticated techniques that 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 of the HTTP header itself[65]. This condition may cause the CDN to route traffic to the domain inside of the HTTP header, not the TLS header.

Past that level of C2 is an attacker using a neutral space C2 capability[66]. For example, posting to Twitter feeds, Facebook pages, or some other location that the victim systems can access as well as the attacker group. It is a common technique to direct users to these sites via a spear phish email.

---

[65] Adapted from https://attack.mitre.org/wiki/Technique/T1172 (8/18/18)

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

Table 30 Network Based C&C Detection

Criteria	Explanation
Well known IP to Site Relationships	You can filter out from evaluation much of the top 1M list as described on page 112.
IP Reputation	New "site to IP" relationship: for example, a site registered within the last 7 days. A newly observed IP for your site. Site to IP change's. Most sites don't change their IPs frequently (depends on DNS detection with Bro or PassiveDNS).
IP Validated Poor 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	Historically, a twenty-four-hour DNS TTL value was quite common. Today, TTLs may be arbitrarily set low for sites in order 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[67] list or the Cisco Umbrella[68] list. See page 112 for further details.
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 31 Application Content Based C&C Detection

Criteria	Explanation
File Transmissions	FTP and HTTP/S upload type file transfers are a normal occurrence for some segment of the population – both user workstations and servers. When an internal IP *sends* data significantly above its threshold, then 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

[67] https://majestic.com/reports/majestic-million (6/10/18)
[68] http://s3-us-west-1.amazonaws.com/umbrella-static/index.html (6/10/18

Criteria	Explanation
	working hours, while batch processes are more frequent overnight (YMMV). 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.
Known Malicious URL	If a user actually visits a known malicious URL, then of course it should be investigated! There are several open source URL lists available.
Protocol violation or mismatch	Protocols (network, system, application, etc.) are well defined by RFCs, which occasionally have wiggle room. IANA has defined well-known ports which use well-known protocols. For example, HTTP is normally delivered in 80/TCP, whereas HTTPS is normally delivered in 443/TCP and the traffic begins with a TLS exchange. If outbound HTTP/S traffic is observed on nonstandard web server ports and a technical component can detect this "protocol mismatch" condition, it should be investigated to determine if the traffic supports a real organizational requirement. While these are not perfect or account for every possibility, RFCs do provide a basis for applications and network services to communicate. Violations to these rules may indicate C&C usage, or other issues. For example, numerous protocols have had tunneling capabilities built that can use a data field or a normal communication capability for hidden communication. Or outbound traffic carried over 443/TCP which is not HTTP and which did not begin with a TLS exchange.

Lateral Movement or Lateral Traversal

Lateral movement or 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 usually achieved based on a process like the one below:

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 are a susceptible user to establish their toehold.
2. The minimal component reaches out to gather a secondary tool, such as a backdoor, a more sophisticated trojan, C2 agent, a keylogger, etc. This reach out process is known as the second stage install process.
3. Some form of persistence mechanism is created, which can be exposed by reviewing the system configuration using autoruns analysis. Check out the article in the InfoSec Handlers Diary Blog[69] for July 6, 2018 for a great discussion on how to get autoruns data into Splunk.
4. Some form of beaconing occurs, which can be observed by analyzing network traffic. Beaconing[70] hast these characteristics:
    a. Recurring connections on an interval – think regular patterns.
    b. Connections will persist and show up again after a reboot.
    c. Small outgoing/incoming packet sizes, for command and control, because it doesn't take much to tell an agent what to do.
    d. Traffic will usually be permitted through corporate defenses and carried over HTTP (port 80), HTTPS (port 443), DNS (port 53), and in some cases, ICMP.
5. The attacker will then perform several distinct actions upon gaining access to a system:
    a. Escalate privileges in order to dump, and then crack, the local password database, either the Windows SAM or the 'shadow' file on a UNIX/Linux based system.
    b. Gather an incoming credential, such as capturing and then reusing a Windows authentication hash.
    c. Establish a scheduled task or install a service that can be used to provide a return path, or act as an agent which connects to a C2 network.

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

1. Users authenticate *as themselves from* their assigned workstation to a common set of resources a few times per day, and use those resources for the work day. An example pattern could be once at the beginning of their day, and perhaps again if they logout at their assigned lunch time. After that, authentication from user workstations to central resources is lower,

[69] https://isc.sans.edu/diary/rss/23840

[70] One tool to detect beaconing is RITA. It performs sophisticated statistical analysis on Bro logs.

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 they use new network resources.

2. 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.

3. Proxied authentication where a user's credentials are authenticated but the caller is not part of the domain. Examples are LDAP based authentication), VPN sources, and RADIUS.

4. Rarely (and I do mean rarely) there may be user accounts used to a system directly, or with a local account on a system. These are recorded with a 4624 event (logon type 3) recorded in *that systems security event log*. Collecting and counting these in the aggregate can reveal lateral traversal. Local accounts can be detected with the Workstation Name field is the reporting system name or not the domain. There is also an absence of an entry on a DC when a local account is used.

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

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

Pass the Hash

Pass the Hash is an attack technique where 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 (like the local administrator):

1. Security Log: Event 4624, LogonType of 3, because this example requires NTLM authentication as the Authentication Package.
2. The session key length will be 0.
3. The Logon Process will be NtLmSsP.

4. The account referenced will not be a domain logon, it will be a local logon, and not the ANONYMOUS LOGON account.
5. The same account being used to connect to multiple systems from the same source system (source network address field).

Other Windows System Traces

Process execution: There are numerous tools available to attackers which 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 a of course, wce.exe. In order to view these events, collect 4688 and make sure that Detailed Tracking[71] is enabled. Review sysmon data and Event ID 4688 data. Use of PowerShell, when scripts are run from a nonstandard location, have odd names, long command lines, or make Internet connections.

WMIC calls are not common, especially from workstation to workstation.

Profiles: When a user interactively logs on they will have a profile created for them, and a set of directories is created.

Persistence Mechanisms: Windows Event ID's listed can be found by reviewing:

1. **RunAs events**: 552 or 4648.
2. **Scheduled Task creation**: 602 and 4698
3. **Service Creation/Installed**: Event 601 and 4697 with odd names, long names, misspelled names, or random names.
4. **Admin Rights:** Assignment of administrative rights after login show up as a 4672 event, which may be granted to a new locally created account.
5. **New Local Accounts:** Local accounts that cannot be explained, *especially accounts ending in a dollar sign*, because these accounts are an attempt to look like a computer account.
6. **Remote Logins:** TerminalServices-LocalSessionManager events with ID 21.
7. **Administrator account usage:** Use of accounts named "administrator". regardless of location. Users should always be performing any elevated action with a specific authorized account.

Compromise and Recon Related Events on Windows 10/Server 2016: Recent enhancements were implemented by Microsoft with improved auditing, specifically to identify if a compromise is in process.

---

[71] 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 in order to get the command line in a 4688 event.

1. 4799: security-enabled local group membership was enumerated
2. 4798: user's local group membership was enumerated
3. 4627: Group membership information
4. 6416: A new external device was recognized by the system
5. 4624's new Linked Logon ID, Elevated Token, Virtual Account, and Restricted Admin Mode fields
6. 4688's new information on Process start events.

Special Groups: This feature logs a particular event (4694) when members of a monitored group login to a system. To use this, enable "Audit Special Logon" under Logon/Logoff in the Account management section of group policy. Then you need to collect up the Security Identifiers for the accounts you want to monitor, which is visible on the "attribute editor" tab for an account in the Active Directory Users and Computers application (ADUC). Also, well known SIDs are defined in KB article 243330.

Network Traces

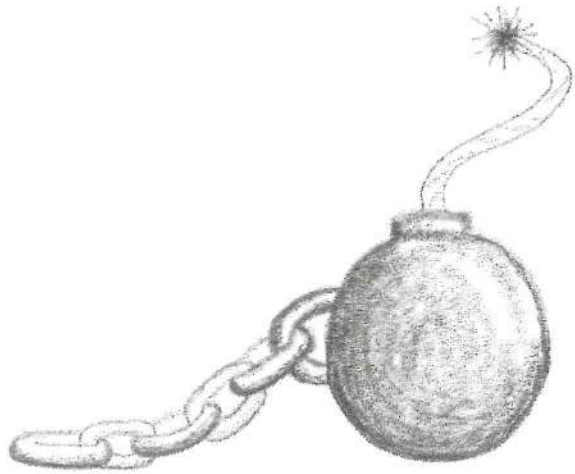
System to System communication that doesn't support the target systems usage pattern. Examples include:

1. Workstation to Workstation using RPC over port 445/TCP and WSMAN 5985/TCP. Windows 10 does frustrate this trace with P2P based operating system updates, however.
2. Server to server communication may also follow this pattern, but research will be needed to narrow down what is suspicions.
3. ICMP traffic between workstation networks.

Using the Lockheed Martin Cyber Kill Chain

The concept of cyber kill chain, or the steps involved in responding to the stages of an attack process, can be very useful in how SOC leverages every piece of fact data within their realm. This model is illustrated in Figure 13 Lockheed Martin Cyber Kill Chain and Security Controls on page 186. It was originally designed as

a reference mechanism. Today, its use has expanded to include an organizational method for security event data.



When the SOC team detects activity at that matches one of the steps in the chain, they can immediately pivot based on the IoC or alarm and look backwards in current event and alarm data while the issue is resolved. For example, if persistent malware is detected in a system "Run" keys which corresponds to the "Installation" step, then retrieve the last modified time from the key and start looking back in time for connections to machine or system activity that can identify how the system was "Exploited".

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 necessary decision making to destroy the target. This term was adapted by Lockheed Martin back in 2011 as a reliable model or framework in order to describe attack stages, understand how attackers operate as they progress through the stages, and then to ensure that a protective and/or detective control is applied at each stage. By understanding the steps that an attacker needs to go through, superimposing that on the "observe-orient-decide-act" or OODA loop. 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. The SOC team can use these principles to build out *proactive* monitoring controls, reporting, and alerting. Furthermore, by being able to provide a model for gap detection and analysis against a well-defined attack progression pattern, the technical and operational environment can be better instrumented to repel the borders.

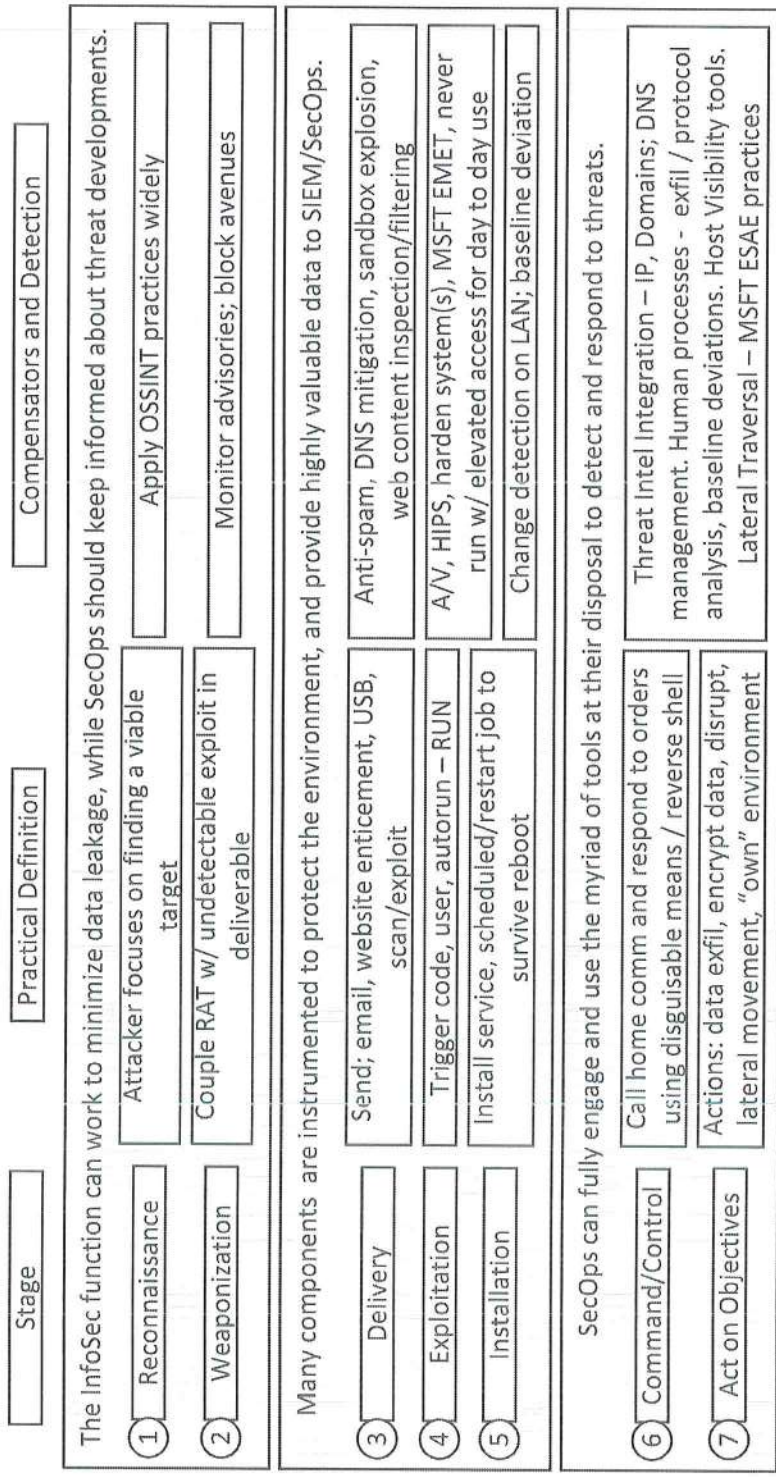


Figure 13 Lockheed Martin Cyber Kill Chain and Security Controls

Indicators of Compromise and Attack Data Dependencies

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

"An IoC is a piece of forensic data observed on the network, in a log file, a persistence facility, or the operating system that are likely to indicate malicious activity which can aid security operations or incident responders to detect breaches, malicious activity, misuse, or some other form of attack."

Hand in hand with IoC's is another term you may come across, "Indicators of Attack[72] (IoA)." An IoA differs from an IoC because IoAs focus on what an attacker is attempting to accomplish, not just the malware or observed behavior. IoC's depend on some form if signature or pattern match, like a firewall record with a source IP matching a threat feed. In contrast, IoA's can be a directed spear phish email, a password spray condition, outbound traffic for tftp (69/UDP), 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 are not part of their job responsibility.

SOC and threat hunting teams can use these indicators to evaluate their environments. Review the data dependencies in order to ensure the reporting system is configured to generate the 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 attention of the SOC.

Table 32 Indicators of Compromise Forensic Data Dependencies

Indicator	Data Dependency
Unusual outbound network traffic	Firewall Logging, including a "default deny" policy Web proxy logging Protocol mismatch detection from Bro logs Unusual IP protocols which can be detected by a simple NIDS rule
Account management anomalies	Central directory, designated account managers, knowledge of privileged accounts, service accounts, consistent account naming,

[72] CrowdStrike has one of the better articles that explain IoA, IoC, and the differences. URL: https://www.crowdstrike.com/blog/indicators-attack-vs-indicators-compromise/

Indicator	Data Dependency
	and account correlation across various systems. Cross system correlation may require a consistent attribute be added to all systems such as an employee unique identifier.
Privilege account misuse/anomalies	Inventory of privileged accounts for the directory and systems which reside in "account islands". Account islands are often application specific, and contain very valuable data.
Geographical account usage patterns or improbabilities	VPN/Citrix logging, rules to detect "inside in use" vs. "outside detected", 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 Login activity from non-AD integrated systems
Confirmed Threat Intel "hit"	Intelligence feed(s) and the ability to match an attribute against data arriving in the system, such as an IP address, DNS name, email address, malware hash, etc.
First Seen Binary	File system integration or a HIDS tool and an inventory of known/observed binaries, 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, DB activity logging, standard deviation detection, and encryption keys to allow for protocol inspection
HTML or website query size and ratio mismatches (producer to consumer)	Web server and proxy server logging which has enough granularity; depends on ability to get packet size
URL Hits above baseline	Web server and proxy server logging which has enough granularity to provide the full URL (UDP based syslog may cut this off)

Indicator	Data Dependency
Protocol abuse; mismatched protocol to well-known ports	IDS or NGFW capable of analyzing network traffic at the *application level and analyzing port to application usage*
Registry (or /etc)/Filesystem changes outside of the Change Window	File integrity monitoring such as OSSEC, Tripwire
DNS mismatches, fail spikes, DGA queries, excessive TXT queries, rapid name to IP changes (Fastflux)	High quality DNS audit log *with the true source*, request/response, and an analysis engine (can be performed on or near the DNS server with a tool like Bro IDS or PassiveDNS)
OS Changes of significance outside of the change window	Patch detection, "system" activity logging such as new services, share creation, application install/modification/removal, and other OS state change data
Tablet/Mobile/Phablet or other IoT device profile changes	A MDM solution (may be pricy...) that can be applied to the IoT device
Unexplained Disk usage / volume changes	Operating system integration with a performance monitor 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, NIDS

SIEM Field Notes

After having researched, selected, and implemented SIEM solutions in dozens of environments over 15 years and running a MSSP, I've found that there are a number of sound strategies to effectively run a SIEM. This chapter attempts to capture those lessons and offer advice.

General Principles to Run a Successful SIEM

Running a successful SIEM requires that you leverage and apply *knowledge of your environment* to identify your assets, networks, unused networks, applications, and privileged accounts. After that, the SOC must understand what assets support which business processes and applications, implement monitoring to defend the assets, and understand how the attacker thinks and build instrumentation to see them. More advanced systems can integrate vulnerability management into the system team.

SIEM deployment is a *program, not a one-time project*: While it is true that the initial deployment of baseline SIEM technology can be treated as a project, and the vendor can be critical to initial success, the long-term lifeblood of SIEM is to ensure its care and feeding is part of the Information Security Program and thus the IT General Controls program. SIEM is the primary enabling and supporting tool for SecOps. As such the platform needs to be maintained through budget support, training, new use cases need to be developed, changes to any system that integrates with or supplies data to the SIEM are *well coordinated,* and new data feeds are integrated as the technology landscape changes. To help ensure that the program doesn't fail, make sure that you understand these points, budget for them, and then create subsequent processes to keep these issues under control.

Write, and then Implement, succinct Use Cases: Often technology types like to turn the wrench and make things work. Avoid the temptation to going to the console and creating a monitor, alarm automation, or an event specific dashboard without taking the time to document the idea and determine how a SOC analyst will act. By taking the time to write out and validate with the security team what the actual monitoring use case is, make sure that it can be implemented, and going the extra step to match up a use case to the security program and various "standards", you will have a much more effective capability. This is why there's a whole chapter beginning on page 133 focused on developing a SOC and SIEM focused use case.

SIEM *enables* system and network monitoring security state as the primary, bot not only, supporting tool: As a tool, it requires people to manage the

system. It must be maintained. The SOC needs to aggressively defend the information feeds, monitor overall system health, and always improve the alarm conditions or rule base after every lesson learned event. Practically, this means that use cases and content are updated over time.

Realize that event data changes over time: Applications, systems, technologies, and OS's are upgraded. Data sources may be reinitialized by their respective administrations and not coordinated with the SOC. The SIEM must keep pace with data changes and event format changes. If dependency on the SIEM holds up a major system upgrade then the SIEM may just may be left out in the cold, so keep informed about major changes in IT.

SIEM, and thus SecOps, require organization specific content, automation, and environmental awareness: Just like no two fingerprints are alike, no two SIEM systems or the networks they monitor are identical. Further, networks change over time. For example, metadata about a user can be highly valuable to enrich alarm review. If the solution can query AD and get current data about a user as an analyst works with an event or an alarm, the analyst will always spell the user name correctly, and will not need to open another tool and navigate to the user's metadata which has a measurable positive decrease in click operations. Enhancing, or making access to supplemental data sources from the console *significantly* improves usability.

Be outcome focused: Monitoring and logging for its own sake has some value, but likely has little value to executives. To achieve value, be sure that you know the components that support your *value chain* and instrument your monitoring to provide assurance that the value chain is functional and is as secure as possible.

Use DNS names, not IPs for data transmission: When configuring a system to report to the SIEM, use *logical* DNS names, and use a variety of them. For example, build out a DNS name set for the logging infrastructure that can support your deployment model, site model, and security zone structure. By using logical names, the servers themselves can be replaced, capacity can be introduced to solve for a resource issue, and through load balancing one node can take over for another during a maintenance period.

Plan for at least 40 days for online immediate retention: There will be a natural threshold for how long investigations and pattern analysis process need rapid data access. To support that time box, plan to keep about six weeks of data on hand on fast storage without having to go to the long-term log store. It is likely that the SOC will need to answer questions like what happened last month or perform other reporting and analysis around a monthly boundary, so

six weeks should cover that time period. Also, employee investigation usually takes some time before HR or Legal come to the SOC analysts.

Implement one fully exercised data source at a time: When a data source is added to the SIEM, be sure to fully exercise its monitoring capability to ensure you are getting the auditing you need, you locate events you can drop, and get the most out of it you can. You want to make sure that all of its options are enabled, that it generates the maximum possible log or audit data, unnecessary data can be trimmed, and that the parsing mechanism handles all of the possible event types and fields.

Run and archive complete reports that map your compliance, regulatory, and IT General Controls program(s): You may lose access to your long-term data for a variety of reasons. Or it may be impractical to go and get a set of log files from many months ago, import them back into your SIEM architecture, and then run a custom configured report for that time frame to answer a specific question. Instead, design and develop useful reports that highlight data and activity relevant to your environment. For example, account life cycle transactions. The report would show when an account was changed, by whom (user or service account), what change occurred, if one account was modeled after another, and any other relevant detail. Then if you need to investigate what happened to a particular user, it would be a matter of searching the reports. There is an inherent assumption that reports can be written to a searchable archive, like a web server that can parse and index PDF files, of course!

In real life, I have found that many SIEM platforms summarize data in their default reports. As an analyst, I really want the time of an event as well as the time zone. Be on the lookout for a default report that can be tuned to provide time-based detail, because when you need to reconstruct an event time line a summary pie or bar chart report will provide little to no value. In contrast, a time-based set of events that make up the bar chart can be tremendously valuable *because the chart can be recreated* with desktop applications.

Implement Synthetic Transactions

For each and every monitored data source, your implementation should be able to generate measurable synthetic transaction that proves the data source is functional *and* is capable of generating an applicable alarm condition within a time frame acceptable to your organization (aim for 2 minutes or less). Each day the system should report on these synthetic transections (not alarm on them). Synthetic transactions are an incredibly valuable technique to confirm or ensure all of the data sources are healthy. There are several times involved in

measuring a synthetic transaction, so it's important to understand what to measure.

- Source event generation time by class of data source, such as an OS, firewall logs, cloud logs, or host telemetry. There are at least three classes of data sources here. First, a system that is capable of generating the log record as it is written, so it can be quickly consumed. Second, cloud systems send log data through a pipeline from the source to a pick-up point, and they can take several minutes to an hour or more to make a log record available through a cloud API. Third, there are several periodic log sources that assess the environment. As an example, pulling the local autoruns registry key, pushing that to the event log, and then having those records pushed to the SIEM.
- Delivery and transform time. These are the times it takes to read the record and make it ready for consumption in the SIEM.
- Presentation time, which is how long it takes for the move from the pick-up time to the analyst console (this time should be short!)

On Windows, this could be implemented by running an "eventcreate" command from a scheduled task that writes a particular event to the Application log at 11PM with a text string specific to the server. For example, the command below can be setup as a scheduled task. It can then provide the basis for a daily report to confirm each and every system reported into the SIEM in a timely manner.

```
"EVENTCREATE /T SUCCESS /ID 999 /L APPLICATION /SO SOC /D "SOC
Check Transaction".
```

You would then query the SIEM to find out how many of these you received, how long they took to arrive, and what percentage of your Windows systems currently active in the directory produced the event to measure what percentage of overall environment are properly instrumented.

On Linux, this would be a "logger" command run could from a cron job during at 10 PM. Each day, a report can be run next morning to get the count of systems that produced these messages.

Another example of a synthetic transaction is to attempt a download of the EICAR antivirus test file, which should trigger an event through the A/V system.

This capability allows the SOC to "auto assess" the environment. There are numerous other synthetic transaction and analysis processes that can be implemented with some creativity thinking. For a Windows domain, you could script out a process that polls the directory, gathers the list of systems, forward and reverse resolve the domain name, and attempts to map in the C$ drive of

every server defined in the domain with a user account to generate a failed logon event. This process would allow you to detect dormant servers, newly installed servers, and servers that are or are not configured to report to the SIEM.

For non-Windows systems, attempt to telnet or SSH to the system, and determine if it reports to the SIEM. Next, each day, pull out the IP addresses of every internal system that made an outbound connection and determine if it has a reverse DNS entry, is defined in the AD domain, and listed in whatever asset tracking system is in place.

What's important here is that you take an information assurance perspective for your environment, the types of data you have on hand, and how that data can be leveraged to keep the SIEM system and the environment as close to one another as possible.

Severity, Priority, Urgency, and Reliability Criteria

A *significant* part of how a SIEM processes data and determines the level of "urgency" is based on a severity or a priority rating of an asset or event data as it goes through its decision-making process to raise an alarm. Some systems increase the likelihood of an event becoming an alarm when multiple related or duplicated events are observed, and use that a reliability rating. You will need to learn these terms for your platform, and be able to explain to others how they influence the event stream. Of the three, *asset priority* is mostly standardized because most SIEM platforms have some method to record the importance of an asset to the organization on a repeatable scale such as 1 to 5.

There are numerous factors for these criteria, just as many opinions on what they mean, and how to measure or assign them. What makes a difference here is feeding the SIEM team's ability to have as much criteria as possible that will properly influence the evaluation pipeline. For example, a SIEM platform may be able to decrease the calculated value of an event if the source network is Guest WiFi. Alternately, if the asset is the primary finance server, the asset priority may be higher than other servers and make it more likely for an event to become an alarm.

The SIEM management team should seek every opportunity to *derive* asset priority based on known asset data, such as data from the CMDB or other asset management system, even if its spreadsheets. One of the more reliable teams that also use these criteria is the DRP/BCP team, so reach out to them as they likely have a criticality assessment for applications and the servers that depend on them. The intention here is to leverage an Authoritative System of Record (ASOR) outside of the SIEM whenever possible and encourage adoption of that

system which in turn maximizes the overall technology spend and helps to make the SIEM consistent with other IT processes such as Disaster Recovery.

Figure 14 SIEM Urgency Score Influencers

For example, a compliance risk factor may be discerned from a CMDB export if an asset is marked with a compliance requirement. A hospital may assign a "HIPAA Data" attribute to the Electronic Medical Records (EMR) system. That attribute can be extracted from the CMDB and then used to influence a "compliance risk factor" that somehow becomes visible in the SIEM, expressed as a higher than average asset value.

As much reliable fact data as possible should be leveraged and brought forth through the data import process, manual asset configuration, and event parsers in order to enrich alerts, messages, and reports produced by the system. These enrichments should affect the overall severity score from a base event.

Reliability: It is often desirable to specifically influence the severity score based on a time threshold, an event count threshold, or a combination of two or more events that indicate a specific outcome.

Event Generators Influence Severity

The source of an event such as the firewall, an operating system, an application, a NIDS, a HIDS console will carry forward event attributes and decisions that the generator made about the event itself that influence the severity score. In some cases, generators will simply record an event. For example, a highly intelligent NGFW system or a security console for a HIDS or EDR platform can generate events that have gone through analysis, correlation, and a scoring method before they reach the SIEM and report that event with high confidence. In turn, the SIEM would generate a high value alarm as it trusts the source system.

Asset Have Multiple Values: Understand Why

Assets have values that can be expressed in a SIEM. However, if there is only one single "value", you will need to make a choice in how the platform will use the value. Two common examples are:

Operational Value: Assets and network segments have *value* to the organization. The higher the value, the greater the risk to the organization if that asset is adversely affected (a target) or is adversely affecting others (a source).

Compliance Value: Assets and in some cases network segments may be governed by organizational policy. It is very important to be clear and consistent when applying a "compliance value". You are likely to be measuring risk of being out of compliance if an asset is adversely affected. Compliance or regulatory standards include PCI DSS, HIPAA, GLBA, and SOX.

Asset Lifecyle: Assets can be classified as production, quality assurance, and development. This is yet another value that has meaning to the SOC.

Vulnerability Data

A vulnerability assessment can generate a composite score, which is usually derived from a formula based on the individual vulnerabilities on a system. Regardless of the method, a more vulnerable system should contribute to an alarm severity score when the event data clearly relates to the vulnerability. Wherever possible, correlate a CVE based on the event data to the CVE of the vulnerability item as the CVE provides a well-established model to discuss vulnerabilities and their remediation steps.

IP Address or Device History

Some more sophisticated SIEM systems can track sources and destinations with some sort of timeout value when a system is the source or target of an attack pattern. For example, if a source routinely receives "firewall.block" events, its score as an adversary should influence the severity.

IoC Contributions and Threat Intelligence Feeds

Several SIEM vendors operate, or can consume, threat intelligence feeds as an Indicators of Compromise. These contributions to the severity should be blatantly obvious, such as changing the color of the alarm to something that doesn't match the normal scheme or adding a specific icon to enhance the alarm.

IoCs can be very beneficial when bringing issues to the attention of a SOC analyst when that analyst knows how to properly use or read them. An IoC hit should be a fact that influences an investigation like any other data source. However, they are not an "end all, be all" data source. In particular, domain names and IP addresses may be nefarious last week, cleaned up this week, be fine for a few months, and then fall from grace.

NIDS Deployment and Data Collection

A key data source for SIEM is the NIDS system because they can extract information right off the network. When a NIDS system is placed at the perimeter on the "inside" interface of the firewall, it will only capture and alarm of activity destined for the Commodity Internet. In contrast, if you can deploy NIDS in the interior of the network between your servers and your workstations and tune the ruleset based on the likely direction of attack, you have a much better chance of catching an intruder. Realize that once an attacker gets inside the network, many of the attacks that will not work from the perimeter are likely to work on the interior, so the rulebase may need to be adjusted.

Often, IT hardens systems which face the Internet. If not, their systems would be owned within ~~hours~~ minutes[73]. In contrast, interior systems are in the "trusted" zone, and more susceptible to an attack. Further, even open source NIDS systems (Snort or Suricata) have rule sets with a small degree of tuning can be very effective at catching an internal intruder.

SIEM Deployment Checklist

There are numerous items that should go onto your SIEM deployment checklist, and if deploying a SIEM or building a SOC, integrated into the project plan.

1. Ensure you know if you are building a compliance SIEM, a tactical SIEM, or some hybrid of both. This decision will affect how much event data you will initially log, reporting, and data retention for the long haul.
2. Understand the components of the "traditional perimeter" (even though that is dissolving every day).
    a. Does the firewall use a default deny policy?
    b. Can you review the rule set in order to understand the purpose of permitted flows and authorized systems for specific data types?
    c. Are you auditing enough? Are you trimming enough?
    d. Where is the NIDS and the Bro system placed?
    e. How can you detect long running transactions (IP/TCP)?

[73] One of the reviewers asked if the cross out was intentional: Yes. The intention is to represent an advance in attacker capability and the weaponization of malicious software commonly available.

   f. Where and how can you detect Internet Protocols that are flowing out of the network?
3. Plan for a significant increase in data input every year (think 20% to 50%). This may present itself as several new data sources, improved auditing, or new servers coming into the network. The point is that the amount of data and types of data are ever increasing.
4. Make every effort to gather both interior session/NIDS data and internet connection point session/NIDS data, as described Perimeter Security Focused Access in on p.107. Being able to capture workstation to workstation network traffic and server to server traffic is highly valuable when searching for sings of lateral movement.
5. Identify, mine, and maintain key data inventories: There are a *minimum set of inventories you will need*. Along with each inventory, you will need a reliable method to understand how these change over time to prevent data from getting stale.
   a. Server inventory: Domain Controllers, DNS, application, Prod/QA/Dev, security support systems, storage, network appliances.
   b. Asset Criticality: As discussed above.
   c. App to Server to Storage mapping relationship. Consult the application portfolio, the DRP/BCP team recovery plan, and Enterprise Architecture teams to learn these relationships.
   d. Network Device inventory: switches, routers, acceleration servers, load balancers, firewalls, access points.
   e. Identity map: elevated access accounts, privileged groups, and authorized account managers.
   f. Identify systems that not use the centralized directory for user account authentication and roles.
   g. Naming conventions: servers, workstations, network hardware, accounts, service accounts, etc.
   h. Internal network ranges and purposes: ICS systems, HVAC, DHCP, wireless internal, wireless guest, server, storage, cold build, jump boxes, Citrix published desktops, VDI, and any other purpose assigned network segment. From this inventory, you will develop an inventory of "darknets". DMZ network ranges.
   i. External network ranges, NAT translations, and DNS names.
6. Determine your email gateway, as the SIEM will likely email people reports and some sort of internal transaction.
7. Decide what time zone you will operate in, *and ensure that you have time zone shift data so the SOC can consistently map event times to UTC.*
8. Staff Training: platform, SOC skills, IT skills, incident report writing, and how to read and interpret each and every data source.

9. Document your use cases, which will then extend to the data sources and components in your SIEM platform that support these use cases. Here, you should develop and maintain a naming convention for your SIEM platform so that instrumented content can be connected to a use case, or some other reasonable reference.
10. Don't just accept default SIEM content and rules blindly. Ensure that you understand what the vendor has defined and what is relevant for your network and operating environment.
11. Hunting:
    a. PowerShell is weaponized, so you need to be instrumented to detect it through the 4688 Event ID and enabling detailed tracking for workstations and servers through group policy. Note that stand alone systems will require supplemental configuration.
    b. Once you have detailed tracking setup, next step is to deploy sysmon and an XML setup file, which means a deployment package.
    c. Two amazing FOSS tools are Security Onion with Bro IDS and RITA from Black Hills Information Security. Review these tools, and develop a deployment plan if you cannot afford a commercial alternative.
    d. Recurring analysis consumption: Autoruns output from all of your systems so that you can perform long tail analysis on what is configured in system ASEP's.

Understand Why SIEM Deployments Fail so It Won't Happen to You

Over the past fifteen years I've read a variety of articles where someone says "half of all SIEM implementations fail", or somehow asserts that this technology is somehow substandard. This section recounts some of those reasons and wherever possible, some compensators for those reasons.

SIEM is implemented as a "one-time project". This is a common failing. The trap is that you can implement a solution as a "twelve-week project" when the reality is that SIEM and the SOC processes it supports is long-term foundational investment.

Compensators: *Very Clearly define your use cases, monitoring, and data feeds in phases or logical groups.* Avoid the temptation to "get everything in the SIEM" without having a defined monitoring use case for each data feed. Take all of the topics in the Security Monitoring by Data Source Use Cases chapter, determine what you need the most, and then place them on a schedule for

implementation. In other words, enable one data source, bring the use cases for it to completion, and then move on to the next data source.

Spending too early: Implementing a SOC should not start with SIEM product selection. A myriad of untuned and partially implemented tools leads to alarm fatigue, blind spots, a poorly running system, a bad reputation, staff burnout, and that can lead to techno-atrophy.

Compensators: Create and foster a team of motivated people who are skilled in the art of intrusion detection and incident handling *before you spend the first dollar* on a SIEM product. Begin the log analysis process in order to build up a skill base and assess your data quality. Ensure that your organization actually *needs* a SIEM, that there are relevant and useful data sources to feed into the platform, and that you understand the environment sufficiently to detect, identify, and respond to events and alerts. For example, one of the primary data sources is the perimeter firewall. Have you investigated the amount of logging on your perimeter firewall? Are there useful or useless rules? Do you have an outbound default deny posture? Furthermore, local Windows high fidelity auditing is a must to detect todays attacker. Do you have Windows event collection and forwarding enabled? Do you have command line recording for processes as recorded in a 4688 event? These are critical for long term success.

Scripting and analysis of actual source system log data, using a data reduction approach, long tail analysis, and pattern analysis can determine if the current environment can satisfy your use cases. You would be surprised just how effective a SIEM implementation will be if you can show how to produce the results from the source system itself.

Attention and Administration: SIEMs require care, feeding, monitoring, and they can become overwhelmed. At some point, there will be a ginormous spike in data rates. Collectors can fail, other IT staff will upgrade systems reporting to the SIEM without advising the SIEM management team, resulting in data collection failure. Someone will create a report that pummels the datastore into brief unconsciousness, and run that report every five minutes so the SIEM enters a coma. Someone will configure the system to discard all data or not create an alarm because they checked the wrong box in a policy setting. Someone else will configure the built-in Vulnerability scanner to scan a Class B (/16) address space daily, and the system will dutifully create 255 scan jobs for 255 addresses. Which is 2000+ processes on the host OS, or a recipe for an internal denial of service attack.

Troubleshooting all of these conditions actually interferes with using the platform for its intended purpose – alarm management. Monitoring platform health is one of the *best uses* of a vendor on a support contract: have them

performing day to day maintenance on the platform while the SOC team uses the solution for continuous monitoring, threat hunting, and incident response. One solid hour per day will pay off.

Compensators: Create an alert/report that informs if a data source system and all monitored operating systems have not provided data based on a reasonable threshold (start with 24 hours.). Over time, this function will change based on the pace of data coming into the system. For example, if the perimeter firewall and the domain controllers haven't reported in 5 minutes, there is likely a serious problem somewhere. Create a "synthetic transaction" wherever possible from each of your source systems, as described on page 193.

Inadequate staff to handle and respond to alarms: In nearly environment large enough to afford a SIEM, there are a myriad of data sources that cause alerts. Tuning these alerts *takes time*. And once alerts are tuned, the environment is likely large enough that there will be more alerts than staff to triage, prioritize, and handle them.

Compensators: Prioritize alarm processing based on the likelihood of an identified threat *compromising* a system, working from highest to lowest asset value. Determine what threshold you need, and use that to drive your *minimum* staffing level. Review other alerts that can't be handled in the aggregate.

Improper alignment: As discussed elsewhere, SIEM *must* be aligned to, and be instrumented to monitor the *value chain*. Senior management, and in turn an extended family of stakeholders rarely care about the number of antivirus events or the volume of intrusion attempts stopped by the firewall. They passionately care about protecting and expanding the business, which in turns means keeping the value chain operational and supporting the business.

Compensators: When the SIEM and the SOC business cases are built out, there should be a well-defined rationale for the technology stack and the staff. If the team does a solid job articulating what the SOC will do to protect the business and how the technology platform aligns to key *IT support systems and will be monitored by* the SOC, then these functions will have executive management support. After that is done, ensure that you build a quarterly report that explains how you support the business case.

Useless data, too much data, data that doesn't support Use Cases: Many organizations implement a solution and send all possible data to it, only to find out that performance lags, alarm presentation is significantly delayed, and running reports renders the user interface ineffective. This is not a "*tactical* SIEM" – it's a log collector.

Compensators: The first thing to do is *choose your data wisely based on what you need to monitor*. The second thing to do is *drop useless information*. Yes, that's right – don't take everything you can, and don't be afraid to drop data that has little to no value. The third thing to do is determine what the best way is to support a use case or answer a query, and then have at most two data sources that can answer that question. If you find that there are three data sources, then you have room to prune and likely should. When making a data decision, chose the best "user attributable data" whenever possible. Here are some examples:

1. From the perimeter firewall, *drop data from the proxy server outbound and back* in exchange for gathering data directly from the proxy server. Here, the firewall is telling you "the proxy was allowed out and it got a response". The proxy advises the same thing with less events (think 4:1 or 6:1 ratio). The proxy understands the application, more readily qualifies the URL and action, *and in most cases,* proxies identify the end user. Based on implementing this rule with a few clients, the net savings can be between 27% to 78% reduction in raw data for the firewall itself – not to mention the improvement in CPU overhead on the firewall.

2. From Windows domain controllers, very carefully consider dropping "machine authentication" data. Instead, gather focused user presence, group changes, and process activity events from the workstations using Windows Event Forwarding. Here, you would need a very granular pattern of machine names that are supported by the organizational naming convention (you have one, and people follow it, right?). Instead, use WEC/WEF from workstations to gather user presence indicators, task creation, account life cycle events, and service start/stop, and reboots. Windows workstations can generate as much as 40% of the events written to the domain controller event log. They are easy enough to identify, because a machine identifier ends with a dollar sign. You can then drop several event types where the user ends with a dollar sign. I have seen cases where attackers create account names that end in a dollar sign in order to look like a machine account.

3. From the perimeter firewall, discard outbound DNS request records. Instead, use PassiveDNS or Bro IDS at the perimeter to gather outbound DNS queries. Here, you are trading a *known action (DNS to/from)* for a protocol aware action that advises what was queried and the record type. You may not experience as much of a net savings, but what you will certainly gain is more intelligence and give yourself a detection capability by performing long tail analysis on DNS names.

Compute Power and Performance Issues: Depending on the system's architecture, various parts of the processing chain may have a problem weathering an "event storm".

Compensators: The better SIEM architectures incorporate a "store and forward" approach, where a processing node can do its part and wait on the next part. Modern SIEM solutions have various technical solutions to meet this goal. Fortunately, you can spend up or spend outward (meaning scale horizontally) to deal with this failure point, to one degree or another. When provisioning hardware if you want to use physical systems, purchase multiple socket systems and populate half of the sockets with the biggest, fastest CPU you can. Also, buy the highest possible density memory that populates half of the system's memory slots. These two techniques allow you to easily expand a single host system simply by increasing compute power without needing to "replatform" – just by another dual set of CPU's and more memory, and move some hardware.

The single pane of glass story: SIEM vendors really like to tell the story that the SIEM is the "single pane of glass" when it comes to all of our security data. Realize that while this is a *really good idea*, and it looks awesome in a demo, this capability comes with a price.

Compensators: Avoid the perceived need to put all of your candidate data into the SIEM. Consider how the SOC can leverage reporting or a source system's native UI to support realizing a use case.

Solve the right SIEM problems, not all of them: A SIEM can solve a wide variety of data correlation issues, but in many cases it should not. Some orgs have pushed a vast amount of data to the SIEM so it doesn't work well because its overwhelmed, and there were better solutions external to the SIEM.

Compensators: Avoid solving a security problem or building incident detection capability in the SIEM when there is a better, more efficient, and purpose-built tool that does a better job. Instead, automation notification, reporting, or integrate a check in the purpose-built system for the SOC team and then communicate out any actionable alarm conditions. Also, by making *better use* of existing systems, and *not spending* on SIEM, you are materially helping the budget and demonstrating that you are a responsible participant in IT's budget. As an example, an email burst or users setting up auto-forwarding to home email accounts should not be solved in the SIEM.

You are on the wrong battlefield: Many SIEM's are instrumented with non-user attributable high-volume data such as the perimeter firewall and NetFlow. While those data sources are useful, they are less valuable than user authentication on the domain controller end user workstation presence and

process data that can come through detailed process auditing provided by sysmon, detailed tracking (Windows 4688 event), and EDR platforms.

Compensators: The victim of today is the end user workstation who is attacked through phishing, browser exploits, watering hole attacks, web browser-based attacks, and susceptible end user software. Actively seek to respond to this change in attacker behavior and active targeting by collecting workstation process data. Whenever possible, prefer data from workstations and domain controllers, then member servers, and lastly non-attributable sources.

SIEM Event Categorization and Taxonomy

Every SIEM vendor describes events somewhat differently, with different levels of hierarchy in their taxonomy. Vendor solutions may agree in some areas, but none of them agree completely with each other. What makes a difference is that you understand how your site-specific data sources map into the categorization or taxonomy model. For example, Palo Alto firewall "deny" event means that the firewall did not permit action by policy and may arrive as a different event name than an IP tables "drop" log event, and these should map to a "firewall.deny" *categorized* event, as should every other firewall technology that behaves in a similar manner (denying traffic by policy with logging). There is an inherent question *not* answered by this level of mapping. IPtables can either "block" a packet and returns an ICMP error message or "silently drop" a packet, meaning that the event is never forwarded through the chains and the sender does not receive an ICMP message. Palo Alto calls this "drop ICMP". Neither of these conditions map nicely to "firewall.deny", although many platforms may map this specific condition to a "firewall.deny" or equivalent in their taxonomy. These conditions are fundamentally different. Returning an ICMP packet includes outbound traffic whereas a silent discard does not. How is this distinction reported based on the taxonomy?

Being able to properly read the taxonomy is a skill that all users of the SIEM must quickly establish.

Networks, Assets, and SIEM Automation

When considering what types of automation will enhance your use of a SIEM platform, make note of what you need to do in order to enrich and sharpen the alerts that the system brings to your attention. Also, when it comes to automation, asset, CMDB, and network data should be automated and fed into the SIEM to keep it current as possible. Most often this process involves identifying the target fields the SIEM needs, determining the best source system and field in your enterprise, extracting those fields using some automation, and then pushing them into the SIEM.

Active Directory User Lookup. Any user attributable data source will usually provide the *account name* as the username field. This action often turns to an analyst pivoting to a tool such as Active Directory Users and Computers to look up the user, perform a search, and then *transcribe* information into a report. Instead, to make this more efficient, script out an AD lookup to collect user and user group data, which is commonly used during an icndent. Two PowerShell commands are shown below. The first one pulls all of the *defined user attributes* set on the account, and the second pulls the groups assigned to the user.

- Get-ADUser don.murdoch -properties *
- Get-ADPrincipalGroupMembership don.murdoch | select name

Active Directory Groups, Servers, Workstations, and Domain Controllers: There are built-in elevated access groups in Active Directory that need to be polled, have their user accounts extracted, and loaded into the SIEM so that supplemental searches, alerts, and reporting will have more accurate data for alerting. For example, you should query the membership of the various "Admin" groups so that privileged users can be monitored for suspicious logon activity, like these users being used for NTLM authentication across more than 5% of the domain within a few minutes. Second, query AD for servers where the last logon time stamp is more than 24 hours ago. You would use this result set to remove hosts from monitoring by agents and clean up asset definitions if these systems are truly no longer part of the domain. Third, you should *always be receiving large amounts of data from your domain controllers*. If you query the domain and get 11 DCs and there are only 9 DCs defined for collection in the SIEM, then agents need to be deployed to the missing two DCs.

CMDB Data: Any SIEM solution worth its salt will have a rich asset attribute set such as IP, FQDN, Data Owner, System Custodian, Primary/Secondary Application, Operating system, Business Unit, Criticality, Sensitive Data indicators such as HIPAA / PCI, change window time, and a host of other data elements that should (under ideal circumstances), come from a CMDB system. As a recommendation, avoid attempting to maintain your own "SIEM CMDB". There are many others within IT that have a reason to maintain this data. Build an asset import model to routinely consume that data, and work to improve other systems rather than recreate your own "data fiefdom" – it's not your core business!

Asset detection: This function is realized by network scanning and/or passive asset discovery. Here, SOC wants to know about, and maintain data for, newly discovered assets from *most* fixed IP address ranges (workstations reside on DHCP networks, and they are comparatively volatile). There are at least four reliable methods for asset detection:

- Physical Inspection of the network, wire closets, and data center. This is a time-consuming and difficult in large environments.
- Traffic Analysis through network extraction and packet capture and then passive asset detection, service usage, and protocol usage.
- Configuration database or file analysis, such as pulling systems defined in the virtualization console, a CMDB, and from Active directory. This method will not find everything, and it may also return recently deactivated systems.
- Active network scanning using nmap or vulnerability scanners. This method is quick and more likely to find standalone assets, but may disrupt some systems such as an ICS PLC.

SIEM Data Collection Methods and Considerations

There are a variety of methods to get data into the SIEM platform. This section will provide notes on how these can be used for SIEM platforms, with some practical considerations for each.

1. Syslog UDP (most systems use RFC3164 syslog/UDP with a limit of 1024 bytes total length – timestamp, facility/priority, and message inclusive).
2. Syslog TCP
3. Syslog TCP + TLS (rsyslog, syslog-ng)
4. SNMP trap, and for some solutions, SNMP polling
5. Local log reader and syslog (UDP/TCP/TLS)
6. Windows Event Log Polling is an example of a local binary or non-text data source reader, usually enabled via an agent utilizing an OS native API call. Event log polling can be combined with Windows Event forwarding and event collection for the best of both worlds.
7. Database polling (reach out to the database server)
8. Remote file monitoring which is usually enabled through CIFS or NFS
9. IoC integrations, such as STIX and TAXII
10. Standardized Log Formats which support automatic field extraction such as ArcSight CEF, LEEF, and JSON attribute pair formats
11. Automated or Manual upload of data, usually for asset and network definitions from a CMDB (CSV, TSV, XML)
12. And, of course, manually loading data into the system through a CSV or JSON import process.

Syslog UDP: This is least reliable and at the same time most common method of collecting log data. UDP is a "fire and forget" protocol and depends on the application itself to enforce data reliability, *when the application thinks reliability is necessary*. Syslog/UDP packet size is specified in RFC 5426 and 5424, with a minimum datagram of 480 bytes and a suggested maximum of 2048 bytes. However, there are numerous legacy systems that implement the older

BSD based syslog systems with a limit of 1024 bytes (RFC 3164). These limits have the effect that longer records are truncated. For example, proxy and webserver log records delivered over syslog/UDP can occasionally be truncated. In addition to all of the other attributes, URL's for some parts of an application can be long and exceed the packet length. This has the practical effect that before you consider fully relying on syslog/UDP, review the breadth of event data to validate that all of the necessary log message attributes for the data source fit in the maximum syslog packet size. In contrast, syslog/UDP is very light weight, built into nearly everything, is very easy to configure, accommodates *most* of the data you want per record, and also very easy to manage on the receiver side because syslog software is text file based. Lastly, senders can often have multiple receivers. Plus, syslog data is easy for a person to read.

Syslog TCP: Syslog over TCP wasn't formerly standardized until RFC 6587 in April 2012[74]. Long before that mainstream syslog receivers such as rsyslog and syslog-ng would accept data over TCP. Syslog/TCP solves two problems and introduces several *potential* problems. TCP delivery does ensure that the packet gets there because the native protocol itself is reliable, and packets are not arbitrarily truncated. In contrast, since TCP is a reliable protocol, any logging agent that is configured to use TCP may fail if the receiver is not available or is restarted. For example, if an agent cannot establish a connection at start up, it may never log until it is restarted. An agent may also freeze if the syslog server stops responding. These behaviors should be tested as syslog/TCP agents are deployed.

Syslog TCP + TLS: This capability extends syslog transports to ensure that data is encrypted *and* both servers and clients can be strongly authenticated. There can be several hours of overhead in order to get TLS setup and working properly. If you plan on using this capability, *ensure that you actually monitor the traffic* with tcpdump so that you know that the data is actually encrypted.

SNMP Traps: SNMP comes in three different versions. You should prefer SNMPv3 because it provides for access control, supports authentication, and TLS encryption. Remember that SNMP wasn't originally designed as a security tool. It was designed for system monitoring and remote configuration. It is also inherently stateless and most commonly implemented over UDP. You can gain some specific security benefits from gathering data from devices. For example, a device can report reboots. Port scans can be detected by a significant change in the tcpOutRsts value. Lastly many agents can inform about storage related events, like inserting a USB drive by changes made to the hrStorageTable value.

---

[74] But not on 1 April. Those are very different RFC's. 😊

Windows Event Log Polling: There are three main methods to poll a Windows event log.

1. Install an agent on the system that monitors the event log and pushes events to the SIEM as they are written to the log. Examples include OSSEC, Winlogbeat, NXlog, Snare, or a proprietary agent from a SIEM vendor.
2. Run a remote query to pull the with WMI, or read the event logs using the native Windows API over the wire.
3. Setup and use Windows Event Collection and Forwarding and then install an agent on the WEC/WEF server(s).

Regardless of the method used, there is one aspect of Windows events that really affect SIEM platforms. Windows events can be very "wide". In most cases, the event itself includes a wordy explanation. For SIEM platforms that are byte consumption based this will affect your end cost. Furthermore, computer accounts are, as far as Windows authentication is concerned, equivalent to user accounts so they generate the same event ID's.

Field Note: When a system has a local physical hard disk, it is unlikely that polling the logs will ever have a noticeable system impact. However, if you have several dozen virtual machines that are running from the same underlying storage unit that maps to a single disk spindle, LUN, or disk shelf, you will create disk contention at some point. Contention also becomes noticeable when a large number of virtualized hosts use the same storage. When a collector happens to request data from many systems at the same time that point to the same storage, the system is effectively hitting the same disk.

There are several strategies to deal with this: avoid requesting logs every 30 seconds. Instead prefer a two to five-minute interval. If possible attempt to vary the poll rates by distributing multiple collection services which remotely poll at different rates. Use push agents on high volume systems. Leverage collecting log data from a system that is already collecting logs like Microsoft SCOMs Audit Collection Services, or use Windows Event Forwarding.

Pulling data with WMI is normally only allowed for an admin user in older versions of Windows. In more recent versions you can achieve this configuration by granting access permissions to a user level account by adding it to "Event Log Readers" on Windows 2008 and above.

Local Database Polling: There are several application systems that write their audit records to a set of tables, which can then be remotely polled by a SELECT statement from a collector which will then forwarded data to the SIEM. There are some issues that can cause problems with this method.

The audit data must be written with a unique identifier, such as an ID value or a date. This value needs to order the table, meaning that the value must be part of the primary key. The polling agent will need to keep track of the most recently polled value. When a source system that uses a unique identifier is reset, the audit ID value itself is often reset back to zero. Therefore the agent will no longer get data from a select statement because its next value, the one it is tracking for subsequent queries, is far beyond zero. Therefore, the collection agent will need to be reset to 0, or the lowest value for the unique ID field in the database.

For audit tables that use a time value, the database collector must keep track of the most recent time that it successfully retrieved data. In the case of a datetime value as the primary indicator, you *do not want* to reread the database table and thus re-query and pull in old data that was previously consumed, processed, and then stored.

Field Note: Regardless of the ID value, there is a catastrophic issue that only becomes apparent when the data collector is running repeat queries in production at some close interval, say every 5 to 10 minutes: query contention. Production does not always match pre-prod, so this issue is not likely to surface. Realize that since the polling agent is reading several from a table, meanwhile the system is writing one row at a time as an auditable transaction occurs. Databases need to manage access to their tables, so it is possible that the polling process may wait while a write transaction completes. The situation can occur when write transactions is blocked the SELECT (read) operation as the collector pulls audit data for a period of time. If the polling process ran twice a day and blocked transactions from completing for only thirty seconds, it is unlikely that this condition would interfere with normal operations or even ever be discovered. If the read operation occurs every five minutes and takes thirty seconds minutes to execute, that means than ten percent of the time the system cannot write to the table so the auditable transaction will be blocked. To minimize the impact from the polling process, the audit tables primary key needs to include the unique query value (ID or date/time). Further, the ID should not be string data. The SELECT statement should be written to use the index to minimize blocking.

Remote File Monitoring: This data collection is used to compensate for systems that write log data but cannot post or send data to a SIEM. There are two main methods: a) install an agent that can read, and keep track of a pointer within the log file or b) create a share from the system where the log is written, and configure a remote reader to read that share. If you need to monitor a file, start by looking into NXlog. NXlog has a capable architecture for reading and forwarding data written to a local log file using the im_file input module.

For example, Windows DHCP service writes logs for each day such that the most recent six days' worth of logs are on the system (on Next Monday, the DHCP service will over write the prior Monday log). To consume this log data, the SIEM needs to be able to reach out and consume the log which requires a share, a user account with read access, and the ability to track a file position using its collection agent.

Indicators of Compromise (IoC's) integration: Many SIEM platforms can accept threat feeds which in turn provide IoC's such a known malicious IP address, domain name, or email address, user account. The SIEM's rulebase, in turn, takes these values into account as events arrive. Analysts should not take threat feed IoC's as absolute truth – rather as values that influence the alarm assessment.

STIX and TAXII: These two standards relate to cyber threat intelligence. STIX is focused on modeling or representing CTI, whereas TAXII is a protocol for exchange CTI data between systems. Systems are starting to support these standards, so they should be part of your SIEM architecture.

ArcSight CEF, LEEF, and JSON formats: These formats can arrive in a several ways. They are different than flat formats which depend on a regular expression parser. Instead, these formats are key value pairs, which significantly improves data consumption process.

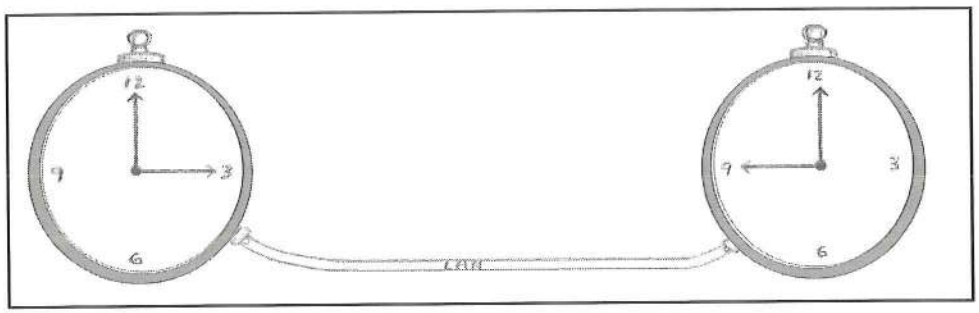
Summary

To sum this all up, running a successful SIEM requires that you leverage and apply *knowledge of your environment* to defend the assets and gather data in a survivable manner from your assets. Once that's in place, *understand* how the attacker thinks and build instrumentation to detect them. Keep up with the system security state by integrating vulnerability awareness into the SOC team. Above all else keep looking for "evil" because it is looking for your crown jewels.

Timekeeping and Event Times

There are several possible "times" and timekeeping issues that SOC analysts, incident responders, and SIEM engineers need to deal with when it comes to collecting and storing data, discussing alarms, and analyzing event data.

Event Time: This is the time when an event actually occurred, or when something happened in the environment. Several factors affect what we think of as the event time, and the SOC analyst needs to understand them and how they can affect the analysis process and timeline event construction. When building timelines, it may become necessary to record the event time as reported and the adjusted time for the observation so that the analyst can get a true picture of event sequencing.



Device Time: This is the time as far as the reporting device is concerned – its own view based on its internal system clock. Most of the time the event time and device time will be the same *when the event was reported*. The system device time should be set to the correct time as defined by the organizations time source, as well as the time zone where the system resides. Further, the device should be synchronized with the organizations central time source. If the device time is off by X minutes, then the "event time" will be off by the same amount and in the same direction so the analyst will have to report an updated event time. Also, when the time on a device is shifted, it brings into question *how long that condition has existed*, and also *what the drift pattern looks like*. For example, 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?

Alarm Time: This is the time that the SIEM or other *primary security component* raised an "alarm condition". This time can measure how effective the log consumption and analysis processes are for the overall system for a single event that causes an alarm. Or in the case of multiple events that become an alarm, it usually records when the condition reached the alarm threshold, but *usually does not record when the condition started*. For single events, most SIEM platforms should provide a 35 second to 5-minute delay in the time that a

source system detected something to an alarm raised on the central alarm panel. You may also have a system that has its own internal SIEM like capability. For example, the Palo Alto NGFW analyzes the prior day's data, overnight, and then can report a "correlation" event after the analysis.

Adjusted Event Offset: Some SIEM platforms can adjust an event time, usually for a transient condition. While this idea may seem desirable at first, try to avoid it. What would happen if the system admin or custodian fixed the source systems error, and SOC didn't find out about it? 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 or in their timeline analysis, and not by modifying the source record. If the source record is *modified* then there is an inherent data integrity concern, and the offset must be adjusted in response to the source system time changing and this adjustment must be constantly explained. You are much better off recording the event time and adjusting your timeline presentation, and then explaining why the time is adjusted.

UTC: UTC stands for "Coordinated Universal Time", which is an international standard that keeps the time accurate to within 0.9 seconds of the earth's rotation. UTC occurs at 0 degrees longitude, which goes through London, England and then through France and Spain. UTC is kept accurate through a series of atomic clocks. Most organization should standardize on UTC for all of the network devices with automatic adjustment for daylight savings time, which would make the time consistent with major operating systems such as Microsoft Windows.

Time zone: These are regional offsets from UTC, which exist for social, conventional, legal, and commercial purposes. Time zone offsets are either positive or negative. Practically, across the world, time zones tend to follow a country, state, county, or natural boundary and run roughly north/south. For example, in the US, the commonwealth of Kentucky is in two time zones with the dividing line following county boundaries. Not all time zones are in even hour offsets. For example, Australia uses eight named time zones and three of them include a half hour offset from UTC.

While you read the examples below, note that they are written with an EST point of view. Therefore, the one-hour shift in standard time vs. Daylight Savings time *also* needs to be incorporated into constructing the true event time. To illustrate this point, use a website like https://www.worldtimebuddy.com/.

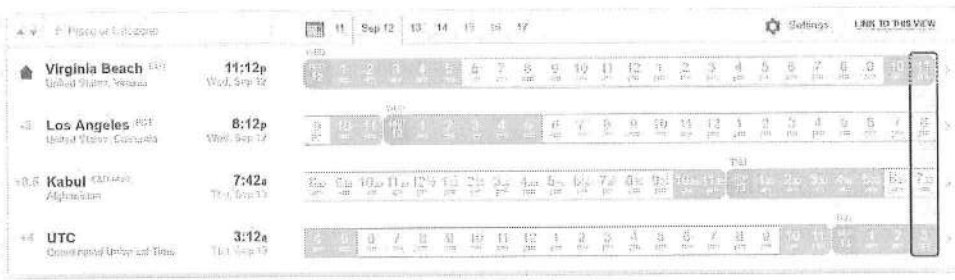


Figure 15 Time differences by time zones

- Eastern US and eastern Canada up to the eastern edge of the Ontario Province is normally UTC-5, unless it is daylight savings time, so 11:12 PM EDT is 3:06 AM UTC.
- Pacific US and Canada are normally UTC-8, so 8:06 PM PST is 3:06 AM the next day in UTC.
- Afghanistan follows UT + 4:30 and Pakistan follows UTC +5, even though the two countries share a roughly north / south border with Iran. So 7:42 AM in Kabul is 3:12 in UTC.

Epoch: Historically, many systems store time from "their beginning" reference time, and keep an integer count as an offset of their epoch. Most Unix systems started counting at Jan 1, 1970, UTC. If you find yourself working with time on these systems, and the particular tool reports time as a long integer, you will need to carefully perform date math and conversions to and from epoch timef.

Daylight Saving Time

In the early 1970's modern daylight-saving time was generally adopted. The change was made to maximize the amount of daylight hours so into the evening. Therefore, twice a year in many countries and most states in the US (but not all) the clock time moves forward one hour in the spring and moves back one hour late in autumn. Realize, though, that daylight is relative to the earth's hemisphere. Spring and autumn are opposite for the northern and southern hemispheres. For the SOC, event data and reporting sources must be configured to accommodate DST. Different countries implement DST differently, and not necessarily along the exact schedule, so if the SOC receives data from different countries event times can be more difficult to track. If your system clocks and various supporting processes can't accommodate this change automatically, there will be a serious adverse effect on your ability to analyze event times. Two helpful websites are:

- https://www.timeanddate.com/ is a complete site with time zone maps and converters.

- http://www.thetime zoneconverter.com/ is a simple page to show you time in another city, with hundreds in the target list.
- https://www.worldtimebuddy.com/ is a very nice site that allows comparison by different time zones and color codes daylight vs. night time.

Network Time Protocol (NTP)

With respect to time values, an organization should install and configure time servers, push time server settings out to any device that can accept a time server setting, and then configure the device to understand its own time zone. This approach will ensure the devices themselves are consistent with each other, and that "the time" is presented to system users that they will understand and is consistent with their watches (or these days, their cell phones)!

The reference implementation for network time is the Network Time Protocol (NTP). Network devices must be configured to use the same time and keep their clocks in sync with the organizations central time source. Adjust the presentation of time based on a time zone offset to the end user if it is actually necessary. SOC analysts should be trained to think in UTC since data sources commonly use UTC. NTP.org maintains a pool of network time servers that use round robin DNS, 0.pool to 3.pool.ntp.org. There are also continent specific time servers. Sites must implement NTP in order to create their own network time infrastructure. Most major network and operating system products can be configured to use an NTP server.

The ntp.org site also maintains a list of NTP Device Vendors for hardware clock vendors. Local network-based hardware clocks have a few advantages. First, systems don't need to connect to the Internet for time synch, such as an air gapped network. Second, hardware clocks are generally more accurate than a PCs 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 do have the corresponding disadvantages – cost and installation.

NTP Device Configuration

There are a few different ways of configuring systems so that they will use a centralized time service. Systems that are active participants in a Windows Active Directory domain, by default. The Windows PDC emulator can also be configured to be an NTP server and announce itself as a reliable time source (see KB 816042).

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/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

The DC that has the "PDC Emulator FSMO" role is the reliable time source for the AD forest. The DC with this role should point to the site's reliable time server, such as a hardware clock or an ntp.org time server. Normally, this role is assumed by the *first DC* installed in the forest. This system should *not* be set to "time.windows.com", because that's the default NTP server name for *all Windows systems and* is historically overloaded.

Manual Log Analysis for IR and the SOC

This section was written to provide a checklist and structured process for Incident Responders and SOC when the need arises to perform manual log analysis, when a sophisticated SIEM or log management solution isn't in place. This condition can occur 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.

Some example situations may be before the SIEM is setup, they system is significantly degraded, you are offsite at a field office, you may be a consultant working with clients, or any number of situations where there is no centralized solution.

Step	Action
1.	**Understand the Case**: Get a handle on the case, the situation, or the issue that needs to be investigated. This step is instrumental to pointing you in the right direction. Avoid being myopic though; use the parameters of the case to help prioritize activity.
2.	**Identify the Relevant Log Sources**: For the supporting data you need, determine which log sources will inform the investigative process.
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 some time/dated notes with a screen shot to explain how the log data is collected.* Also ask for a screen shot of the source system's time and UTC offset if you don't have direct access. You will need to confirm these settings in order to organize log data in a timeline and may want to manipulate the time in the *analysis output step* so the data is in time order.
4.	**Self-Document Collected Data:** Admins won't often produce log files using self-documenting file names or hashes. As log data arrives, name the log file in a self-documenting manner and get an MD5 or SHA hash of the file. Which log file name is better: 20170304.1705.PerimiterFirewall.last4days.csv or FWinfo.csv?
5.	**Data Reduction**: 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 log file, and analyze that data. Here, the 'reduced' file would mimic the original file name but add something like "incident_data", or "relevant_info" to the file name.

Step	Action
6.	**Time Adjust:** 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 in order to normalize the data for your timeline. This action should result in set of files specifically for this purpose in a separate directory. Your methods need to be are reputable, 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.
7.	**Find the Clues:** There are numerous clues in log data. Once you've collected information, start to dig through the clues. Examples are: 1. Volume changes, such as a significant uptick or drop in data. Lack of data may in fact be a major clue. For example, no firewall data from the perimeter firewall for a seven minute period when you think the incident happened would point to disabled logging during this period or 100% CPU load where logging did not occur. 2. System change activity, such as file system, directory, process, or actions taken by staff to make changes to a system. 3. 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. 4. Begin timeline reconstruction and visualization. Walk through your reduced data set 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.
8.	**Correlation and Pivoting:** 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 fact 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:** As you are building up the case, remember what you are after: finding 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 on page 163, it

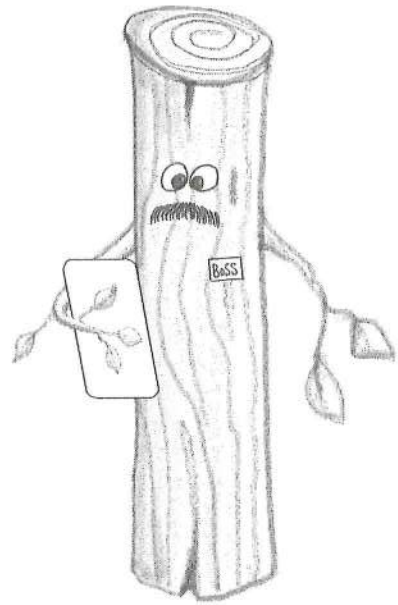
Step	Action
	is can be more expedient to disprove a theory and then further instrument the system to detect if the event happens again.
10	**Summarize findings:** Since you have significant amounts of time to investigate, gather, and perform an analysis, cross communicate your process to at least one peer in order to check your method, validate the conclusion, and keep others who need situational awareness in the loop.

A note on Peer Review: Taking some time to go over a case with a peer may reveal a clue or an avenue of investigation you missed.

Peer review has several benefits. Some examples are:

1. Case review will also provide cross training opportunities.
2. Work product can be self-monitored.
3. Case work from one event may shed light on another case.
4. Mistakes can be identified and corrected *before* the report or analysis is finalized.
5. Work product quality should become more consistent across the board.

Log Management



There are numerous reasons to capture logs. First is their operational value so system admins can review activity to confirm services are operating properly and/or detect system issues. Second is the obvious security value of an audit record. Third to support a compliance standard or legislative/regulatory requirement. Depending on your point of view, you will undoubtedly change the order of these three points.

Reliable and informative logs are the lifeblood of the SOC, threat hunting, and continuous monitoring programs. In order for these processes to function, millions of pieces of original fact data go through a complicated process with several steps before it arrives on the analyst dashboard or review platform. In order for logs to be trustworthy and useful, it is important to understand the process, methods, and the makeup of a consumable log record that can be relied on to support the investigative process. In order to fully support *proactive monitoring*, it is as important to ensure that data sources that *should be* reporting are, and that the source systems provide the full breadth of log records possible.

There is also a difference between general logging and system auditing. An Audit record will meet a higher standard, is not immediately discardable, provides documentary evidence of a specific activity, and is usually associated with a named user or process.

Log Record Data Elements

Minimum data for an Audit record: There are several components of a reliable log record explained here.

1. **When:** Event Time: This is the time *on the local system* when the event occurred. It is ideal to get the local time zone, if possible, or the UTC offset with the log record.
2. **When:** Log Time: This is the time when the log record itself was written, and *may be* different than the event time.

3. **Where From:** Source System Address, Device Identifier, or FQDN: Each record should preserve the system identifier of the source of the event at the time when the event occurred. This is particularly important with the proliferation of network-based systems.
4. **Where Occurred:** Acting system Address and/or Name: The name and/or address of the computer system that "caused" or is the "source" of the event. This relationship exists for Network Intrusion Detection Systems, or other observational systems where the observational system sees traffic between two (or more) hosts.
5. **Who**: Acting User: The account for the user who performed the event. This data should, whenever possible, tie back to a known account within at least one directory in the organization.
6. **Who**: Application Data: This data would identify the application, a database instance, a module name or a service. Essentially, this data element must clearly identify what process, component, or module on the system produced the record. Note that in many cases this can be inferred from the log name.
7. **What**: Event Occurrence: Sufficient information that explains what the acting user or process did, the action taken, and the outcome of that action.
8. **What**: Changed Data: This type of data is highly variable, and amplifies what the event is. For example, a user may add a purchase order to a system for the "event occurrence" and the "changed data" would be the PO number. In the case of account or group management events, these may be a right granted or revoked.
9. **Why**: Conditions: Logging often indicates conditional information – information, success, failure, success, error, critical. These conditionals should be consistent in what they 'mean' across the system.

Account Management: Account Life Cycle Events (ALCEs) relate to creation, assignment, modification, disable, and removal of user accounts and the rights or permissions assigned to those accounts.

1. **Who**: Acted Upon User: The user's account that is affected by the ALCE.
2. **Who**: Acting User: The user who made the change to the acted upon user.
3. **Who**: Modeled User: Some systems allow for one user to be transformed to match or model another user's account. For systems that allows this, the originating, or modeled user, should be part of the audit record. Note this condition establishes a three-way user relationship. Administrator Bob cloned user Alice's account based on the current rights and group membership held by Charlie. This is often called permissions cloning.
4. **What**: Right/Role/Permission/Group: Users gain or lose access based on a role, which may be implemented through group membership or having a specific right assigned. That right must be recorded.

Network Device Data: Network devices have access to spatial type data at Layer 3 and 4 of the OSI model. While an application may not have access to a TCP port, a network device does. When it comes to *recording* network activity data, the source numerical information should be recorded, *not* the usual service or protocol. More specifically, a firewall should record port 80/TCP, and not assume HTTP to describe a packet flow. It is ideal when the network device can see into the protocol in use, and provide that in addition to numerical port and protocol numbers because that analysis provides significant *context*.

Logging System Components

There are several components in a log management system which will be described and illustrated here.

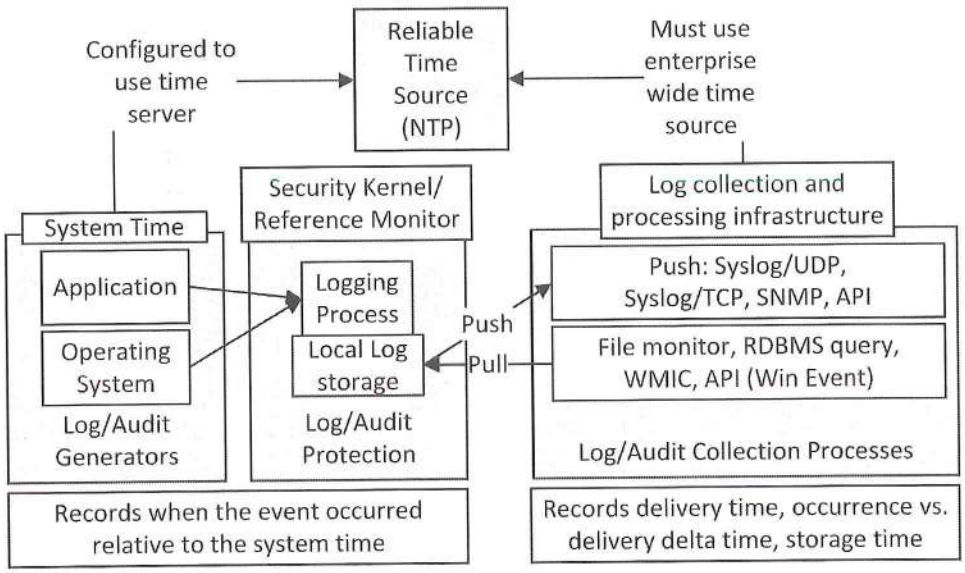


Figure 16 Logging Generation, Timestamps, and Collection Components

Log Generator: Proper log management begins with *instrumentation* of the source system, application, network device, or database. In order to have a log record, there needs to be a log generator, which means that the *best possible source component* must be configured to create a record, at a minimum, for a "critical transaction", and all of the other useful activities that may be logged. Many systems must have logging enabled. Once enabled, logging must be configured in various decision-making components of a system. For example, a firewall is configured with a policy that controls traffic between multiple interfaces. When the state manager checks network traffic flowing through the firewall, it may or may not actually log the record. This configuration may be a conscious decision or may be an error.

Log Protection: Once the fact data is created, it is most often stored locally for the application or system. Log data must be forwarded off the system as soon as possible. The part of the system that stores the log must be secure, meaning that non-authorized 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 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 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, on the other hand, by default, will roll over the entries in the event log once the log reaches a specific size which is 20 MB by default at install time. In a domain, a group policy object 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

The are many examples when a site can safely not log all traffic. The decision to *not log* traffic should focus on exchanging a *lower resolution source* for a higher resolution source, particularly one that understands the application level protocol and is user attributable. Examples:

1. A site may choose not to log *blocked inbound traffic* at the perimeter firewall because this data will be excessive and it is very well known that the "Internet is always knocking on the door".
2. A site may be using Microsoft DNS servers and have a usable DNS monitoring solution, like PassiveDNS in place, so they do 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.

3. A site may have configured process monitoring or local file system auditing using sysmon and native Windows events. At some point they may implement a high-quality Endpoint Detection and Response platform like Carbon Black or Tanium in favor of sysmon collection. EDR applications allow them to investigate alarm conditions more completely because EDR provides more usable data. with a more structured user interface to for exploring process, network, and file system changes.
4. For sites that implement a proxy server like Squid, BlueCoat, or WebSense, the firewall can be configured not to log outbound firewall "accept" and NetFlow data originating from the proxy servers, because the proxy provides far more valuable application level data than the firewall. Note, however, for this particular case, the site should log accept and deny traffic that is normally covered by the proxy for systems other than the proxy.
5. Network level flow data to and from specific types of infrastructure systems like the SAN, NAS, and backup/recovery systems. These systems are constantly exchanging data often at high volume. Capturing this type of flow data can be considered extraneous. For user shares, logging user and process actions against a file systems contents are significantly more valuable than flow because it is user attributable and flow data is machine attributable.

Log Times

Of particular importance are the timestamps involved, because they are used to order the events when the timeline of an incident is constructed. Event generators are rarely aware of a time other than the system time for an event from their own perspective. A system's time is initially derived from the CMOS clock and then is updated by a time synchronization OS component like an NTP agent.

More often these days, the system time will be offset by the time zone setting for user display, but stored in UTC. That practice allows a user to see times that correspond to their wrist watch or cell phone. If that time is significantly off from the network time, a system may have problems fully connecting to the network OS. For example, if system clocks are off by more than five or ten minutes, Kerberos based systems like Active Directory will not properly authenticate users and grant tickets.

Log Arrival Time: The logging infrastructure should record the event arrival time. Some SIEM's are actually capable of recording timestamps as an event is accepted, stored, processed, and turns into an alarm. Refer to the section titled Implement Synthetic Transactions on page 193.

Log Ingest Time: Many systems take a store and forward approach, meaning that they store up events and send them along. For SecOps, we really want the time delay between event generation and receipt in the SIEM to be minimized. By recording the time that an event is consumed into the analysis cycle the overall delay can be measured. In the real world, make every effort to minimize the time between event generation and ingest wherever possible, balanced with environmental factors.

Detecting NTP Issues Use Case

NTP outbound synchronization should occur *from* the sites authorized NTP servers, and none others. This simple technique is a great way of identifying rogue appliances recently installed on the network because they attempt to get time from the Internet, systems which are not properly configured, or unmanaged Windows systems plugged into the LAN attempting to get time from "time.windows.com".

You can also do some analysis and find systems that are not synchronizing. You would do this by finding IP addresses that are not in a windows domain, not using a time source, and are present on the network either by a network scan and not reaching out through the perimeter to an Internet time service.

Windows domain participants will naturally synchronize with the domain controller, so if they are running properly, they should not visit "time.microsoft.com" or another time source.

Use Case:

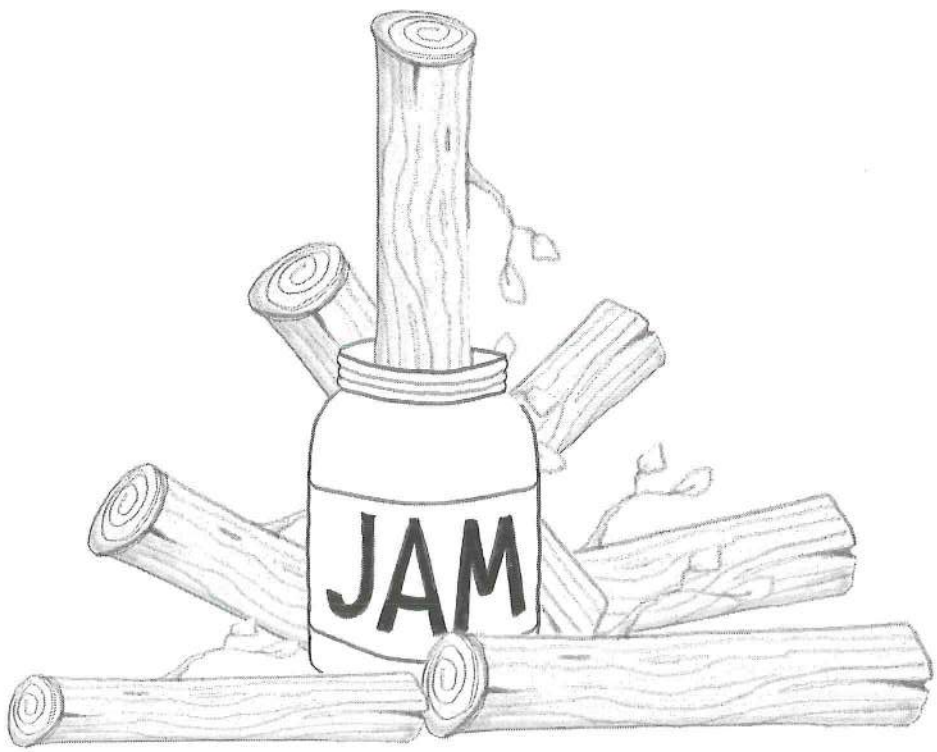
1. In the perimeter firewall, allow outbound traffic from the internal servers *to* the selected time servers over port 123/UDP (NTP uses port 123 over UDP). Then configure a separate rule to log traffic that doesn't fit this pattern, and then create an alarm condition or a daily report in your SIEM platform to detect systems not using the authorized time source.
2. Report on systems connecting to time.microsoft.com.

Log Retention, Audit, and Compliance Considerations

There are many regulations that affect the modern business. Privately owned businesses need audit records just to run their businesses, while publicly traded business have this same requirement and they must also support a financial control program such as Sarbanes Oxley in the US. For healthcare organizations the plot thickens, because they must support HIPAA and HITECH regulation. Lastly, after California's breach notification law was enacted in 2003, most states in the US have followed and enacted their own unique notification

requirements. The majority of these regulations relate to a specific aspect of operating the business – record keeping and a controls program focused on personally identifiable information, the accuracy of financial information, and the access controls surrounding that information. In other words, regulatory legislation is scope limited. In order to support regulation, it is *very important* to attribute assets to the type of governing regulation and understand, for your organization, what type of log and audit data supports your regulatory requirements as part of IT General Controls. SOC should leverage organizationally specific and contextual requirements so that they can better monitor and protect those assets. The point has been made elsewhere in this book and it bears repeating. The SOC must be able to connect its operations to the business context, and monitoring the system security state of SOX, HIPAA, and breach notification data sources are an excellent example of how to do that. The converse is also true: when an incident occurs, can SOC provide assurance that those assets are current 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 auditor. Auditors review data for a "prior period" when they arrive on site. That period could be the prior quarter or the prior 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 then long-term compliance drivers. The first time period should support immediate investigations on a quarter boundary, meaning that data should be online and available for the past 93 days (365/4, rounded up one). The second time period must fully support your regulatory requirement(s), and will be much longer. This means that the log management capability should have ready access to backup media (disks, tapes) past that the immediate investigation frame *and have the ability to import those log archives* for the organizations *entire* records retention and regulatory period in a "timely manner" (think within one day).

Table 33 Example Compliance and Regulatory In Scope Log Retention Periods

Standard/Legislation	Period	Notes impacting Logging
Sarbanes Oxley USA (July 2002)	7 yrs	See SOX 302 and 404 controls. SOX is focused on record retention that supports *financial systems, financial reporting, communications, and audit records.*
PCI DSS 3.2	1 yr	Limit cardholder data to support business, legal, regulatory. (3.1). Visitor logs for 3 months.
Graham Leech Bliley Act USA (Nov 1999)	6 yrs	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.
European Union Data Retention Europe, (Mar 2006 to April 2014)	2 yrs	Relates to telecom data, with LEA/LEO's capable of requesting IP address for email, phone, and text message. 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 yrs, 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)	6 yrs	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 which have contracts with a government agency, review NIST SP 800-92 "Guide to Security Log Management" and NIST SP 800-53 (Rev 4) "Security and Privacy Controls for Federal Information Systems and Organizations".

Logging and SOC Program Maturity from NIST

Organizations go through a growth process in their SOC programs as they do in every other aspect of how they operate. The forward-thinking organization will achieve a certain level of maturity within SOC program, balancing the needs of the organization with the FTE, CapEx, and OpEx costs of improving this capability. For reference, the USA National Institute of Standards or NIST[75] has a standardized an information security program model. Entire books and websites are devoted to these processes. This section is specific to the log management program from a NIST perspective.

Table 34 NIST's Security Maturity Levels and SecOps

NIST Security Maturity Level	SOC and Log Management Support
IT Security Maturity Level 1: Policies	"Required Logging" policy is in place and known to IT. This policy would require security and operational focused logging be enabled on each system. Logging enablement would also be part of the change control process, in order to ensure that logging is functional before a system is put into production.
IT Security Maturity Level 2: Procedures	"Configured Logging Procedure" is followed by system admins in order to ensure that the right level of logging is enabled. The SOC team is staffed and operating, has access to log sources as needed, is well known in the organization, even if it is on a part time basis. SOC "services" are defined.
IT Security Maturity Level 3: Implementation	At this level, NIST states "procedures and controls are implemented in a consistent manner". At this level you can make the case that a centralized log management solution be in place with a structured review

[75] https://csrc.nist.gov/Projects/Program-Review-for-Information-Security-Assistance/Security-Maturity-Levels

NIST Security Maturity Level	SOC and Log Management Support
	process so that analysts can detect and respond security issues. SOC needs to be identifying incidents, actively engaging with IT, the business, and management to ensure that incidents do not have adverse impact. The SOC is closing incidents with an appropriate "Lessons Learned" activity. In other words, by the time an organization aspires to reach level 4, SOC should be in "full swing".
IT Security Maturity Level 4: Test	At this level, NIST states "Effective corrective actions are taken". SOC should be capable of deploying log collection capabilities on its own. Note, though, since SOCs are usually a "monitoring" function, specific individuals should be authorized to make system changes, not the SOC team as a whole.
IT Security Maturity Level 5: Integration	The SOC monitoring program is continually implementing more advanced alerting and automation. As new systems come online the SOC is engaged from Day One to design and test conditions that identify risks and security incidents.

Level zero of a log management program would be accepting the default logging and auditing configuration of an operating system, database, or an application. Following that, an organization moves through a graduated process to mature its log program to the point of continuous monitoring, and then being capable of using logs to support a threat hunting program.

Security Onion: Effective Network Security Monitoring

If you haven't heard of Security Onion, head on over to securityonion.net and read about the distribution. Doug Burks (GSE #24) has bundled several best of breed open source tools and components together to build out a Network Security Monitoring platform that can easily be deployed and can hold its own against many commercial systems.

There are several books which discuss NSM. First, Richard Bejtlich wrote The Tao of Network Security Monitoring: Beyond Intrusion Detection back in 2004, then The Practice of Network Security Monitoring: Understanding Incident Detection and Response in 2013. Second Jason Smith and Chris Sanders wrote Applied Network Security Monitoring in 2013. These are excellent books on the subject. If there is one open source security tool your SOC team should have (besides tcpdump and tshark), it is the current version of Security Onion, as the distribution includes the tools described in these respected books.

With cost of commodity hardware as low as it is today, a moderately sized site can deploy an enterprise grade NSM solution for under $10,000 in capital expense (CapEx) and a weeks' worth of work and have an effective solution. Over time, that solution will need tuning, like any NIDS system. Once that's in place, the site will need to invest in training and education in order to make the best use of the platform, such as Security Onion's own course.

Several years ago, a network TAP solution would cost several thousand dollars. In Q4/2016, a good quality network tap such as a USR 4503 with gigabit capability and two monitor ports costs less than $700 on Amazon.

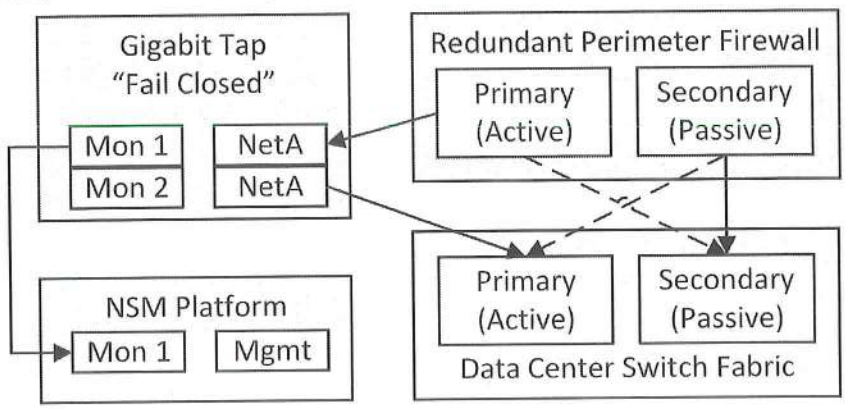


Figure 17 NSM Schematic

Security Onion can run either Snort or Suricata as the NIDS engine. To make best use of this tool, a subscription to the either the Talos ruleset ($400) or the Emerging Threats ruleset must be purchased). With more and more organizations implementing fault tolerant configurations, such as the one illustrated here, all you would need to do is add in a second TAP for the secondary side and another NIC in the NSM platform, assuming you actually need this.

Note, as illustrated, a TAP is installed inline so there is no need to consume an expensive switch port or to ask the switch to SPAN or mirror traffic. These configurations will "fail closed" so that if the TAP loses power data will still flow.

The SOC team can also deploy multiple NSM platforms off a single tap and dedicate them to specific uses. For example, at the network perimeter, a multiport tap device can feed the NSM platform and a second port can be used to record full content PCAP data for one off analyses without impacting the NSM platform itself.

NSM Platform Advice from the Field

This section provides advice on implementing NSM.

1. Set your retention time on the platform. Packet capture data can consume vast amounts of disk space, so only keep what you need. Note that meta data, as in the kind produced by the Bro IDS, provides tremendous intel on network data and it compresses very well.
2. Consider carefully trimming historical packet capture data. There is a tool called TrimCAP from netresec.com that can be used to trim out packet data from the end of a flow so you could keep the first few hundred KB on hand for data that past a reasonable threshold – for example, 14 days.
3. Use flash or SSD disk for your primary data source, immediate PCAP storage, Bro logs, Elasticsearch database, and at least "Enterprise grade" SAS or SATA for long term historical data and historical storage. Avoid using home user drives you buy from Amazon for $100 or so. There is a significant difference in the sustainable transfer speed of an Enterprise SATA drive when compared to a home user commodity drive in practice. You need to use SSD storage for the primary PCAP storage device because it will support being able to read data independently from the drive as it is being written to without the contention caused by disk head movement – the disk head is a bottleneck.
4. Add a caching disk controller to the implementation (especially if you virtualize). Hardware based disk access for read/write is not normally managed, they provide simple read/write. By adding a dedicated and purpose-built controller, you can significantly improve read/write speeds.

Even a controller as inexpensive as an Avago 3108 can result in significant performance boost, depending on how the controller is configured.

5. Implement and maintain a "filter out" clause for full content capture. Routinely check the top talkers on the NSM platform in order to determine if there are high volume, low to no value sources that can be eliminated from full PCAP capture. For example, if you have a backup/recovery capability, you could safely eliminate capturing data for a specific conversational flow. This doesn't mean that you ignore capturing session level data – that flow data will be significantly more compact than PCAP data.
6. Push DNS data collection as close to the client as possible, then data reduce, and then feed it to the NSM. While perimeter-based DNS capture is certainly valuable, it often lacks one key element – the true client source. In other words, you can detect all manner of misuse, but you won't necessarily be able to capture the client. Instead, deploy a full NSM solution like Security Onion at the perimeter and a PassiveDNS on the "inside". If you detect malicious sites at the perimeter, then you can query PassveDNS on the interior to determine which client attempted to visit that site.
7. Offload compressed Bro logs for the long term. The cost of a pair of mirrored SATA drives in an enclosure is very inexpensive when compared to the value of Bro logs. Once per day (perhaps at 2 AM) script out a SCP of "yesterdays" Bro logs onto a storage volume.

Continuous Monitoring

Continuous monitoring (CM) is focused on applying sound processes and technology to detect compliance and risk issues with both the financial and operational environment[76]. In effect, CM assesses administrative and technical controls for their effectiveness as part of an organization's governance program. NIST 800-137 offers this definition for CM:

"Information security continuous monitoring (ISCM) is defined as maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decision."

The SIEM system and the SOC team is an integral support to CM. Specifically:

1) Continuous Monitoring can bring security issues that need to be solved to the SOC team. There is a large inventory of technical controls in NIST 800-53 v4. Alerting provides a more dynamic view of the technical state of system security, and clearly ties the expenditures of the SIEM and the FTE cost of the SOC team to an IT General Controls program. For example, by enabling USB device monitoring and log collection, the SOC function can be notified within minutes of someone violates NIST 800-53 v4 "Media Protection Control Family" set of controls.
2) Account Life Cycle Management processes and alerting are key supports to the governance aspects of ISCM. Unless there is a full-fledged account management program implemented through an identify and access management solution, lapses in access control can easily occur. Many organizations can't implement full-fledged IdM. Instead, they rely on service desk systems where a user requests access and hopefully have a 1 over 1 manager authorization before access is granted. Providing an alarm to a *resource owner* when users are added or removed from groups that grant elevated access to resources under their control is an example of supporting these controls.

Getting CM Right There are several key elements in a CM program summarized below, as adapted from NIST 800-137:

1. *Asset categorization* and intelligence is *central* to CM. The SOC must know which assets have the most value in order to fully monitor them, their

[76] Paraphrased from https://en.wikipedia.org/wiki/Continuous_monitoring (Oct 2016).

communication patterns, and usage patterns of elevated privilege accounts used to manage these assets.

2. Applying security controls to systems should be based on operational risk which is derived from overall enterprise risk. Enterprise risk tolerance influences policy and implementation. Therefore, the frequency that SOC should focus attention on threat hunting would be higher for critical assets, and lower for low value assets.
3. Each aspect of the "security stack" should *strongly correlate* to your policy. If you have spent any time with the SABSA model, you will know that each component must show traceability up through the system architecture at each layer. One of the tenants of the SABSA model is to demonstrate traceability from the individual component at the lowest layer, through the physical and logical data layer, then to the business conceptual and contextual model.
4. Controls, monitoring, alerting: all of these components should work together and be created to support one another, with data arriving into the SIEM based on the critically of the monitored system and the control family

When a continuous monitoring program and the change detection components of the program are operating correctly, the program can be a tremendously effective way to detect issues and anomalies in the environment. Some examples, using NIST 800-53 v4 as an example:

1. If data arrives to the SIEM outside of a defined reporting window, such as 2 hours late or 3 hours early and there is no explainable reason why, that condition violates AU-8 TIME STAMPS control. This control requires 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.
2. If there are "generic" or "default accounts" in use, the organization can be violating many of the controls defined in the Access Control Control Family of controls, and in particular AC-2 Account Management. This control family has numerous requirements like ensuring accounts have identified owners or custodians, account usage monitoring, and intended system usage.
3. There are numerous reasons to monitor, and then build alerts, for node to node traffic. For example, detecting connections from workstation network to workstation network, DHCP traffic from an unauthorized host, remote access via SSH or RDP to many server types from other than an IT network, etc. By being able to review of flow 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.

Security Architecture Considerations

This section summarizes, as succinctly as possible, many critical aspects of network and system security architecture that are necessary for effective Security Operations, incident response, continuous monitoring, and threat hunting. The majority of these architectural components and practices are no to low cost, meaning that they do not require significant additions of technology.

This section *assumes* that systems are generally hardened using well known guides such as the CI Security benchmark or the DISA STIGs.

Buy as many look a like domains as possible: As discussed in Email and Web: Interactions with Look a Like or Doppelganger Domains, look a like or doppleganger domains represent a threat to the organizations brand identity and can be used to coopt an unsuspecting or unaware user.

Use Ron Van O's SOC-CMMI and MAGMA frameworks: One of the leaders in the SOC/SIEM space is Ron Van Os. Ron has put together a CMMI based SOC assessment tool[77] and a solid framework for developing use cases[78].

Outermost Perimeter Router: This is the farthest out point or edge device. There are a variety of network level (L3/L4) controls that should be in place.

1. Ensure that SOC is aware of NAT translations implemented at the perimeter router.
2. The SOC should know the external IP addresses, ranges, and DNS A, AAAA, and CNAME records. It is much better to say "we are detecting malicious traffic coming from the external SFTP server that hosts these three sites" than "our public IP A.B.C.D" is being malicious. Enter all externally owned IPs and domain names as a modeled "asset" in the system.
3. The SOC should have an overview of the security posture of the outermost perimeter. If the SOC detects a condition that the perimeter router should stop, then an immediate notification needs to be made.

Default Deny Firewall Policy: A permissive firewall that permits all outbound traffic is not only an ineffective security control, it will not assist in detecting threats and compromises. In contrast, as an absolute minimum, a firewall that logs the final deny rule after the permit traffic rules will identify any

[77] https://www.SOC-cmm.com/about/

[78] https://www.betaalvereniging.nl/wp-content/uploads/FI-ISAC-Use-Case-Framework-Full-Documentation.pdf

unauthorized traffic. Firewall logging should not stop there, though. Much of the traffic permitted through a firewall should be logged. The SOC will need access to the origination documents that were submitted for a firewall rule because they need to know who or what the observed traffic should support. Containment procedures will be improved and much better informed. Further, if malicious traffic coming from a business partner is observed and Incident Command decides to disable the traffic, then the business partner can be engaged to a) respond and b) develop an alternative communication method while the issue is resolved.

Deploy Security Onion, NetFlow or some form of Bro IDS at the server and workstation aggregation and edge points: Collecting session data is important to having internal network awareness. However, capturing session data with NetFlow may be challenging, given that the purpose of the network is to move data, and the amount of data generated. One should be judicious where NetFlow data is captured. If at all 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. Other TCP ports will be useful, but these are the minimum If there are choices to be made based on capacity. Once workstation networks are monitored, then collect data from aggregation points that record traffic to the server segments from workstations and between server segments.

Implement separate elevated (Admin, root, application) and user accounts: Regardless of the security controls built into the operating system, those controls are ineffective when a user runs with elevated access on a routine basis. The easiest mitigation is to issue secondary elevated accounts for users who have a business justified reason for elevated access. For example, Joe Smith has an account named jsmith04. If Joe is authorized for elevated access, generate an account named "jsmith04sa" or "jsmith04.sec" as their secondary account. There is also value in separating out the type of elevated access for just the "Domain Admins" group. To shore this practice up one step further, do not create email addresses or permit web access through the proxy server for secondary accounts. This second step inexpensively closes two of the most common avenues of potential compromise which are clicking on a malicious email and a process owned by an elevated account being granted outbound network access.

Rotate passwords for service accounts: Frequently, a service account is setup on a "fire and forget" basis as a member of the "Domain Admins" group, or worse, a group nested in the Domain Admins group. A typical justification is that changing the password may disrupt the service and the service requires "admin access". In Active Directory, the "Domain Admins" group is used to manage the

domain, not manage the servers themselves. There are other groups which can manage servers and workstations quite effectively, while not putting the entire domain at risk should any elevated account become compromised. These accounts also often have some form of elevated access, and thus, they are the targets of attackers. Ensuring that IT can rotate service account passwords and grants an appropriate level of access not only aids during an incident, it also provides assurance that when staff leave a role where they knew these credentials the risk of that user using the service account can be mitigated.

Service accounts are nearly always targeted by an intruder. Inventory your service accounts, attempt to enforce a naming convention, and be sure that you know the names of servers where connections should originate from. Service accounts must be configured to prevent interactive login, which can be done with group policy, so that if there is a particular condition where a service account is needed for an interactive login, *deliberate action must be taken* to enable that right and it is not normally enabled. To achieve this objective, all service accounts need to be in a group, put the accounts in an OU, and apply a GPO to that OU. Edit the GPO. Disable User Configuration. Under Computer Configuration/ Windows Settings/ Security Settings/ Local Policies/ User Rights Assignment, add the "Service Accounts' Security Group to 'Deny log on locally" and "Deny log on through Terminal Services".

Centralize Recurring File Transfers: Data exfiltration is a *serious* concern, because it is the data itself that is of value to an attacker or rival. If at all possible, permit file transfer (FTP, SCP, SFTP, etc.) to occur from a set of centralized servers and monitor for exceptions.

DNS: For internal DNS, configure reverse lookups and scavenging[79] so that you have a very good chance of readily identifying a current system name when all you have is an IP address. Prefer a short scavenging time over a long one.

DNS Chokepoint: In the overall security architecture, provide for a very small number (at least two) internal DNS servers that can resolve queries to the Internet or root name servers. These servers should sit "*above the Windows Active Directory and DNS Hierarchy*", if you will. For example, all of the clients on the network point to the Active Directory domain controllers, and all other devices point to the nearest AD DC for local resolution. Remote DC's point to a central DC. Each centralized AD DC points to a pair of domain servers that can then, in turn, query the Internet root name servers. Ideally, the DNS software should have a more robust logging capability such as BIND (and at least 20 others!). *Only these top-level DNS servers should be allowed out of the network*

[79] One of the better articles on the Microsoft site is: https://social.technet.microsoft.com/wiki/contents/articles/21724.how-dns-aging-and-scavenging-works.aspx

perimeter via a firewall rule. The configuration should support collecting Internet outbound queries, not the local Windows client lookups for AD based server resource records (SRVRR's) and local NetBIOS style short name queries. This restriction will give you the ability to more easily deploy a passive DNS monitoring solution. By controlling outbound DNS access, there is a much better chance of catching outbound DNS exfiltration.

Implement highly instrumented Jump Boxes for server network access and enable event logging to SIEM: Windows provides native console logins via the Remote Desktop protocol carried over 3389/TCP. 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 jumpbox, 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 the full command line is captured for the 4688 Event ID. Install system on them as well. Also, a localized version of the SIEM Windows event log reader should be installed directly on each jump box to minimize network connections to the jump box.

Prefer short DHCP lease times, reverse DNS Integration, and DNS scavenging: DHCP traffic is very light on the LAN and is highly valuable when it comes to identifying authorized systems, providing IP to name lookups through DHCP integration with DNS, and can improve the detection time of a rogue DHCP server 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 when a reverse lookup query is made it should be as accurate as possible.

Automate external IP information and threat intel: More and more resources are appearing today that can be used to investigate an IP address or a domain name. Along with more resources appearing, resources also disappear.

If you want to learn about how this all works, visit the AlienVault Open Threat Exchange threat dashboard. Then pull off an IP address that OTX has found, or dig into an individual address and see if there are other IoC's such as a domain name, and run it through one of the sites identified below.

Understand Open Source Intelligence: The next step is to spend some time at http://www.osintframework.com/, curated and maintained by Justin Nordine. This site maintains a very well-organized directory of numerous Open Source INTelligence (OSINT) sites and tools. To use this site, navigate through the paths and then click on the text to the right of the rightmost entry, which will navigate to the OSINT site or tool.

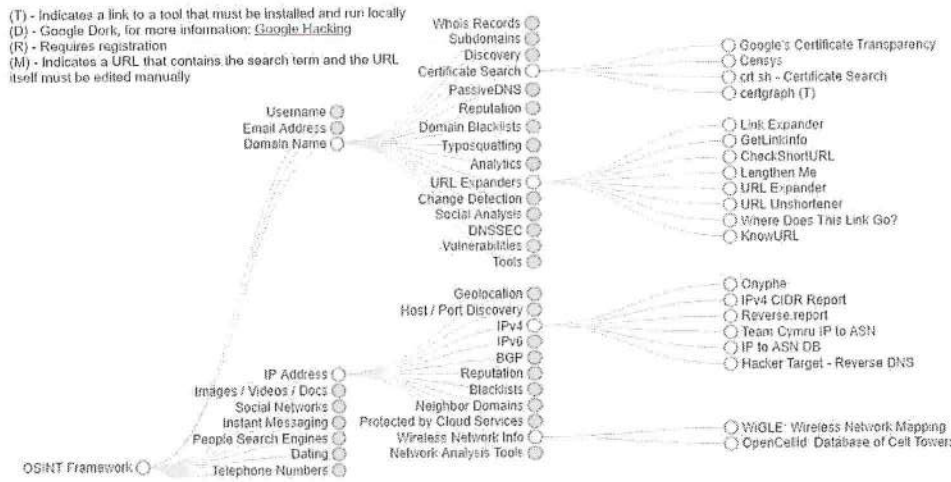


Figure 18 www.osintframework.com with Legend

There are numerous sites on the Internet that can provide information about IP addresses, domain names, and URL's. Many of these come and go. Some of the more reliable ones are listed below. In order to get a handle on learning how to use threat intelligence sites, start with Threat Connect and sign up for a free account. From there search start exploring the operations dashboard, and search for IP addresses that you would find by reviewing any of the lists on Firehol. From there, you can branch out to the OTX and other services.

ThreatCrowd/AlienVault OTX. **otx.alienvault.com**	Firehol - IP lists can be found on a per list basis, under the "source" link. **iplists.firehol.org**
Threat Connect – Free and paid versions. **www.threatconnect.com**	Symantec maintains an IP reputation service. **ipremoval.sms.symantec.com/lookup/**
Spamhaus – well known blocklist antispam site. **www.spamhaus.org/lookup**	Majestic maintains a free list of the top 1M websites – use to find sites not in the list. **majestic.com/reports/majestic-million**

Once you get a solid understanding of how these services work you will make a better threat intelligence purchase decision.

Zero Trust Network Model: John Kindervag from Forrester has published numerous articles and is strong advocate for implementing a network model where "Security Professionals must stop trusting packets as if they were people" and "all network traffic is untrusted." For reference: one of the more complete papers is titled "Build Security into Your Network's DNA: The Zero Trust Network Architecture", dated Nov. 5, 2010. This model has, at its core, several

concepts. First, there are no more 'trusted' or 'untrusted' interfaces on the security devices. Second, all networks are 'untrusted'. Third, there are no longer trusted and untrusted users. Mr. Kindervag's premise is that the activities of the malicious insider and the realities of today demand a new model, not that users and systems are inherently untrustworthy. For the rest of the story, I'd encourage you to look for presentations by John Kindervag and consider engaging with Forrester on this topic.

Useful Reports, References, and Standards

There are several useful standards that affect IT provide significant guidance to a security operations team. This section describes several of them.

Industry Reports and Organizations of Note

There are numerous resources which 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: Institute for Applied Network Security is an industry advisory and consulting firm, provides access to some of the best in the security industry, and helps teams achieve address and solve information, computer, and network security issues. IANS hosts over 100 end user security focused events, worldwide, per year.

Mandiant: **M-Trends Annual Report** is based on a synthesis of real word cases. M-Trends provides real world statistical information on compromises, dwell time, and trends. www.fireeye.com.

Ponemon: Most known for annual **Cost of Data Breach** analysis, The Ponemon Institute has a wide variety of studies and research available. Much of the research is focused on current state issues facing businesses and IT, with some emphasis on healthcare. www.ponemon.org.

SANS Publications and the Reading Room: The SANS Institute provides top quality training, research, guidance, posters, blogs, the Ouch email newsletter, and various webcasts focused on computer, network, and information security topics. SANS also runs a graduate school with two Master's degree programs and several graduate certificate programs. The SANS Reading Room has current articles written by people who have earned Gold level certification or are STI graduate students.

Verizon: Data Breach Investigation Report, based on incident response plus validated information from leading commercial and government security organizations. Theme of the DBIR varies from year to year, as well as the analysis and information presentation. www.verizonenterprise.com.

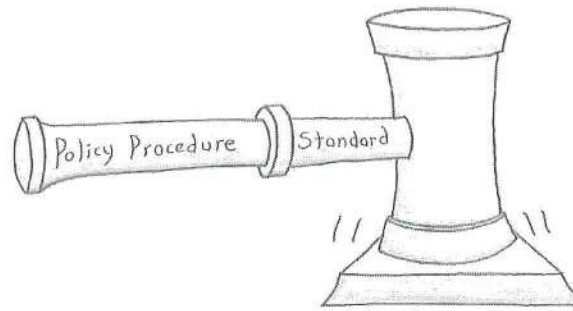
MITRE ATT&CK

MITRE Adversarial Tactics, Techniques & Common Knowledge (ATT&CK): This is one of the best resources to understand how attacks progress from initial

access through establishing persistence, lateral movement, and performing command and control. ATT&CK defines eleven major steps through the process. Under each tactic category are several examples of a technique used to accomplish that specific tactic. Each tactic has several example tools and mitigations for each of the tactics.

InfoSec Standards of Note

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



ASD: The Australian Signals Directorate maintains a list of 37 prioritized strategies to guide technical security professionals to mitigate cyber security incidents. SOC should consult the ASD and match it up to the security program and use these strategies for gap analysis. These strategies are listed by relative effectiveness security rating, user resistance, upfront cost, and ongoing maintenance. ASD also has a Top 4 list of strategies to implement as early as possible. The ASD website states: "Properly implementing application whitelisting, patching applications, patching operating systems and restricting administrative privileges (referred to as the Top 4) continues to mitigate over 85% of adversary techniques".

CIS 20 Critical Controls: The Center for Internet Security maintains a set of twenty **Critical Controls** that represent the most important action any organization can take to improve their security posture. The Twenty CC's are updated every few years based on current security issues, and as of mid-2018, are at Version 7.0. If your organization does not have a control framework, start with the Twenty CC, as every control fully maps to more comprehensive frameworks like the ISO and NIST.

ISO 2700X: The International Organization for Standardization publishes a number of standards relating to information security. Most notable is the **ISO 27001:2013** and the inventory of security focused controls. The ISO controls provide an excellent library of measurable technical controls and is often the basis for an independent audit of an organization.

ISACA: ISACA is a worldwide professional organization that started with an emphasis on information security auditing, and has grown to cover IT Governance. ISACA is well known for Control Objectives for Information and Related Technologies. CobiT is a general IT management framework, not just a security framework.

NIST: The US National Institute of Standards has recently produced the Cybersecurity Framework which is a policy framework designed to help organizations assess and improve their security posture around cyber-attacks. NIST has also produced close to two hundred special publications relating to computer, network, and information security. One of the primary Special Publications is 800-53 v4, Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans. At over 440 pages, this document is is written to facilitate security control assessments and privacy control assessments conducted within an effective risk management framework. Like the ISO 27001, this standard is often used as a basis for auditing an organizations security program and technical controls.

Common TCP and UDP Ports

Note: Some terms are abbreviated and edited for space. "P" stands for protocol in nearly all acronyms. This list was put together from PacketLife, the /etc/services file, life experience, and the IANA ports list.

Encrypted ports: shadowed. Chat traffic ports: ++	Streaming ports: ** Peer to Peer ports: !
TCP MUX	1 TCP/UDP
Echo	7 TCP/UDP
FTP data	20 TCP
FTP control	21 TCP
SSH	22 TCP/UDP
Telnet	23 TCP
SMTP	25 TCP
TIME protocol	37 TCP/UDP
nameserver or WINS	42 TCP/UDP
WHOIS	43 TCP
TACACS Login Host	49 TCP/UDP
DNS	53 TCP/UDP
Route Access Protocol	56 TCP/UDP
DHCP	67-68 UDP
TFTP	69 UDP
Finger	79 TCP
HTTP	80 TCP/UDP
Torpark	81-82 TCP
Kerberos	88 TCP/UDP
POP3	110 TCP
ident/auth	113 TCP/UDP
SFTP (Simple File Transfer)	115 TCP
NNTP (NetNews Transfer)	119 TCP
NTP (Network Time)	123 UDP
DCE/RPC and DCOM	135 TCP/UDP
NetBIOS Name Service	137 TCP/UDP
NetBIOS Datagram Svx	138 TCP/UDP
NetBIOS Session Svc	139 TCP/UDP
IMAP (Internet Message Access)	143 TCP/UDP
SNMP (Simple Network Mgmt)	161 UDP
XDMCP (X Display Manager Ctrl)	177 TCP/UDP
BGP (Border Gateway Protocol)	179 TCP
IRC (Internet Relay Chat)	194 TCP/UDP
IMAP3 (Internet Message Access)	220 TCP/UDP
BGMP (Border Gateway Multicast)	264 TCP/UDP
LDAP (Lightweight Direcc. Access)	389 TCP/UDP
Direct Connect Hub	411-412 TCP
Service Location Protocol (SLP)	427 TCP/UDP

Common TCP and UDP Ports

Encrypted ports: shadowed.	Streaming ports: **
Chat traffic ports: ++	Peer to Peer ports: !
HTTPS	443 TCP
HTTP – occasionally on	8443 TCP
SMB File Sharing	445 TCP
Kerberos	464 TCP/UDP
SMTPS (SMTP over SSL)	465 TCP
Internet Security Association and Key Management Protocol (ISAKMP)	500 TCP/UDP
Rexec (Remote Process Exec.)	512 TCP
rlogin	513 TCP
Syslog/Syslog-ng	514 UDP/TCP
LPD (Line Printer Daemon)	515 TCP
Routing Information Protocol (RIP)	520 UDP
UUCP (Unix-to-Unix Copy Proto)	540 TCP
HTTP RPC	593 TCP/UDP
IPP (Internet Printing Protocol)	631 TCP/UDP
LDAPS (LDAP over TLS/SSL)	636 TCP/UDP
MSDP (Multicast Source Discov.)	639 TCP/UDP
Doom	666 UDP
MS Exchange Routing	691 TCP
OLSR (Optimized Link State)	698 UDP
Kerberos	749-754 TCP/UDP
rsync	873 TCP
VMware	901-904 TCP/UDP
FTPS (FTP over TLS/SSL)	989-990 TCP/UDP
TELNET over TLS/SSL	992 TCP/UDP
IMAPS (IMAP over SSL)	993 TCP
POP3S (POP3 over TLS/SSL)	995 TCP
NFS or IIS	1025 TCP
MS-DCOM	1026 1029 TCP
SOCKS proxy	1080 TCP
Kazaa	1214 TCP !
VLC media player - UDP/RTP	1234 UDP
WASTE	1337 TCP !
MSFT SQL Server	1433 TCP
MSFT SQL Server	1434 UDP
WINS (MSFT Win Name Service)	1512 TCP/UDP
Oracle DB	1521 TCP
Layer 2 Tunneling L2TP	1701 UDP
MSFT Pnt-to-Pnt Tunneling (PPTP)	1723 TCP/UDP
MSFT Media Server	1755 TCP/UDP **
RADIUS authentication protocol	1812 TCP/UDP
NFS (Network File System)	2049 UDP
Oracle DB	2483-2484 TCP/UDP
Symantec AntiVirus Corp. Edition	2967 TCP
Xbox LIVE and/or Games for Win.	3074 TCP/UDP

Encrypted ports: shadowed.	Streaming ports: **
Chat traffic ports: ++	Peer to Peer ports: !
MySQL database system	3306 TCP/UDP
RDP (Microsoft Terminal Server)	3389 TCP/UDP
Teredo tunneling	3544 UDP
Subversion version control system	3690 TCP/UDP
Battle.net	3723 TCP/UDP
Ventrilo VoIP program	3784-3785 TCP/UDP
Smartcard-TLS	4116 TCP/UDP
Rwhois (Referral Whois)	4321 TCP
IP Sec NAT Traversal	4500 UDP
Slingbox	5001 TCP/UDP **
RTP (Real-time Transport Protol)	5004 TCP/UDP **
RTP (Real-time Transport Protol)	5005 TCP/UDP **
NAT Port Mapping Protocol	5351 TCP/UDP
mDNS (Multicast DNS)	5353 UDP
LLMNR (Link-Local Mcast Name)	5355 TCP/UDP
PostgreSQL	5432 TCP/UDP
VNC over HTTP	5800 TCP
VNC (Virtual Network Computing)	5900 TCP/UDP
DameWare Remote Control	6129 TCP
gnutella-svc	6346 TCP/UDP
IRC	6660-6669 TCP ++
IRC SSL	6679 6697 TCP ++
BitTorrent	6888-6999 TCP/UDP !
Windows Live (chat)	6891-6901 TCP ++
Cu See Me	7648 TCP/UDP ++
Cu See Me	7649 TCP/UDP ++
HTTP	8008 8080 TCP
HTTP – Proxies may be here	8080 TCP
Cold Fusion	8500 TCP
TeamSpeak3 - Voice	9987 UDP ++
Tor	9050-9051 TCP

Bibliography and References

NIST SP 800-92 Guide to Security Log Management, NIST. URL: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf (9/3/16)

Grunzweig, Josh (others). Palo Alto Networks. *New Wekby Attacks Use DNS Requests As Command and Control Mechanism*. URL: http://researchcenter.paloaltonetworks.com/2016/05/unit42-new-wekby-attacks-use-dns-requests-as-command-and-control-mechanism/ (1/7/17)

CrowdStrike, "Indicators of Attack vs. Indicators of Compromise", URL: https://www.crowdstrike.com/blog/indicators-attack-vs-indicators-compromise/ (6/10/18)

Hutchins, Cloppert, Amin. Lockheed Martin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Lockheed Martin. URL: http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf (9/16/16)

Wiegers, Karl. Writing Quality Requirements. www.processimpact.com. URL: http://www.processimpact.com/articles/qualreqs.html (12/18/16)

Australian Signals Directorate, Australian Signals Directorate Strategies to Mitigate Targeted Cyber Intrusions URL: http://www.asd.gov.au/infosec/mitigationstrategies.htm (12/16/16)

Microsoft. Well-known security identifiers in Windows operating systems, URL: https://support.microsoft.com/en-us/kb/243330 (12/26/2016)

50 HR and Recruiting Statistics for 2016. 6URL: https://b2b-assets.glassdoor.com/50-hr-and-recruiting-stats-for-2016.pdf (1/2/17)

Nathans, David. Designing and Building a Security Operations Center. Syngress, 2014.

Sanders, Chris. "The Effects of Opening Move Selection on Investigation Speed". URL: http://chrissanders.org/2016/09/effects-of-opening-move-investigation-speed/

https://digitalguardian.com/blog/seek-evil-and-ye-shall-find-guide-cyber-threat-hunting-operations

http://windowsir.blogspot.com/2015/06/hunting-and-knowing-what-to-huntnot-for.html

http://blogs.gartner.com/anton-chuvakin/2016/03/21/antons-favorite-threat-hunting-links/

https://www.sans.org/reading-room/whitepapers/analyst/who-what-where-when-effective-threat-hunting-36785

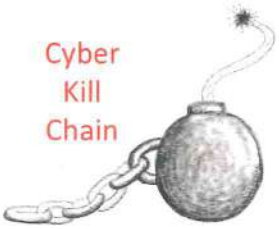
Index

Have you ever asked a security product vendor this question: "What should we monitor?", only to get the answer "That is something your organization needs to decide" or words to that effect? This book answers that question. More importantly, it provides a proven model on how to document your security use case.

Are you ... a Cyber Security SOC member, charged with protecting your company's network against malicious forces both outside and inside? A SOC manager who needs to know how to build and grow your technical team?

Do you have these questions ...? How do you find the bad actor? How do you use the data at your disposal in order to derive information? What matters? What do you monitor?

Author Don Murdoch is a top information security professional with 17 years of corporate, nonprofit, and academic InfoSec experience capturing malware, herding botnets, responding to search warrants, designing and building technical monitoring solutions. Join forces with him and this book, the second in the Blue Team Handbook series. Together, you will release your Cyber Security inner hero.



Act on Objectives
Windows Event Log
Long Tail Analysis
Top 10 IP's
Adversary
Skills, Traits, and Knowledge
Command Line Arguments
Nefarious Process Execution
Threat Hunt
Darknet
NIDS
Use Cases
5 Ws
Firewall
SOP Checklist
Sysmon + Swift
Alarm Triage
Evidence Collection
Credential Theft
Data Reduction
Initial Access
Persistence
Credential Theft
Nextgen Firewall
Countering SIEM Failure
Limited Sleep
don-cu-ment
Account Life Cycle
Monitoring
Proxy
4688

ISBN 9781091493896