- **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

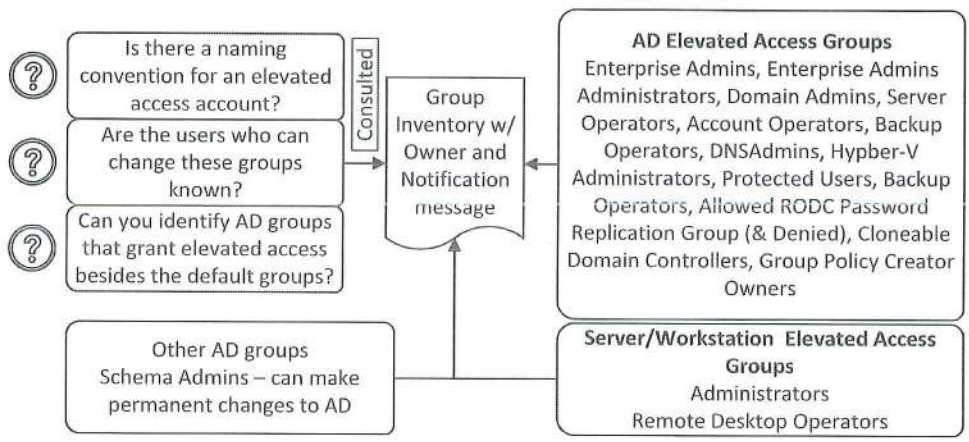


Figure 6 Maintaining Inventory of Elevated Access Groups

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.

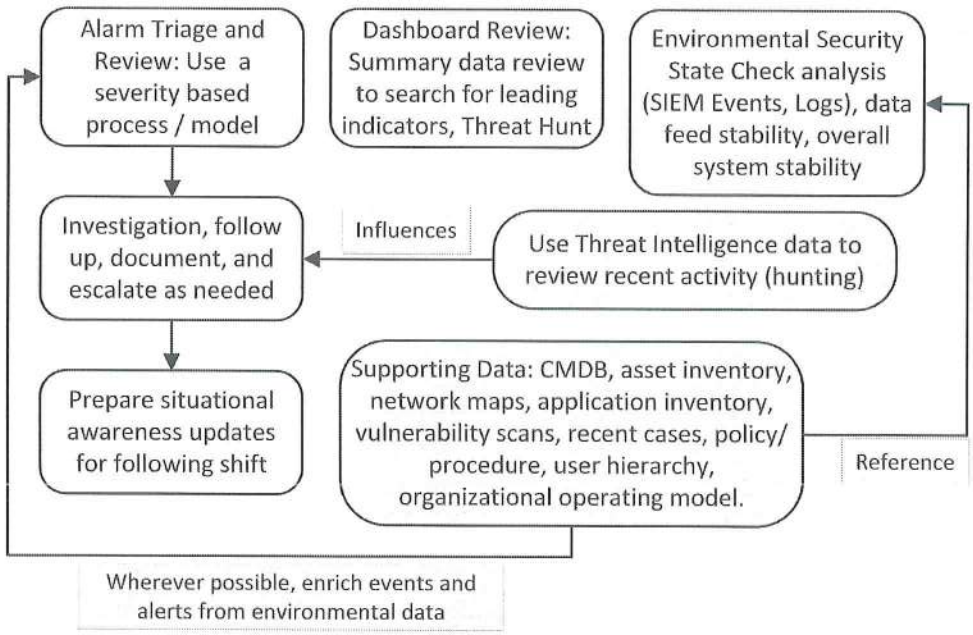


Figure 7 Daily Analysis Overview

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.

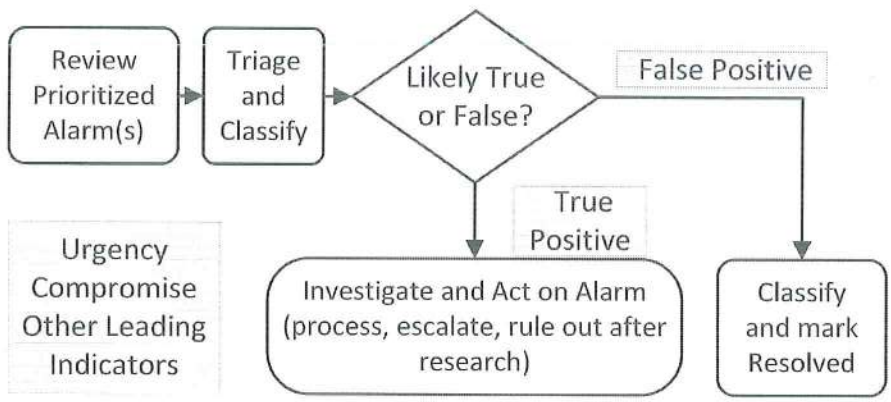


Figure 8 Alarm Triage Overview

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.

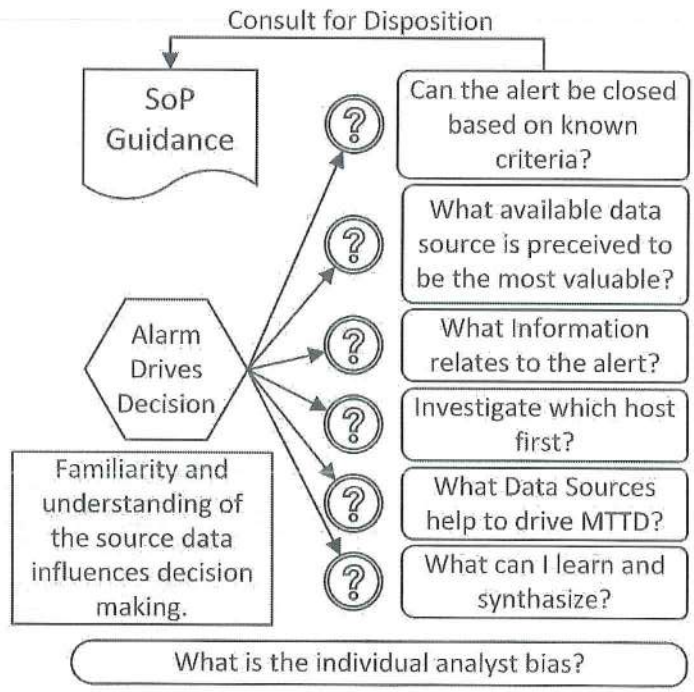


Figure 9 Decisions Driving the Opening Move

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.

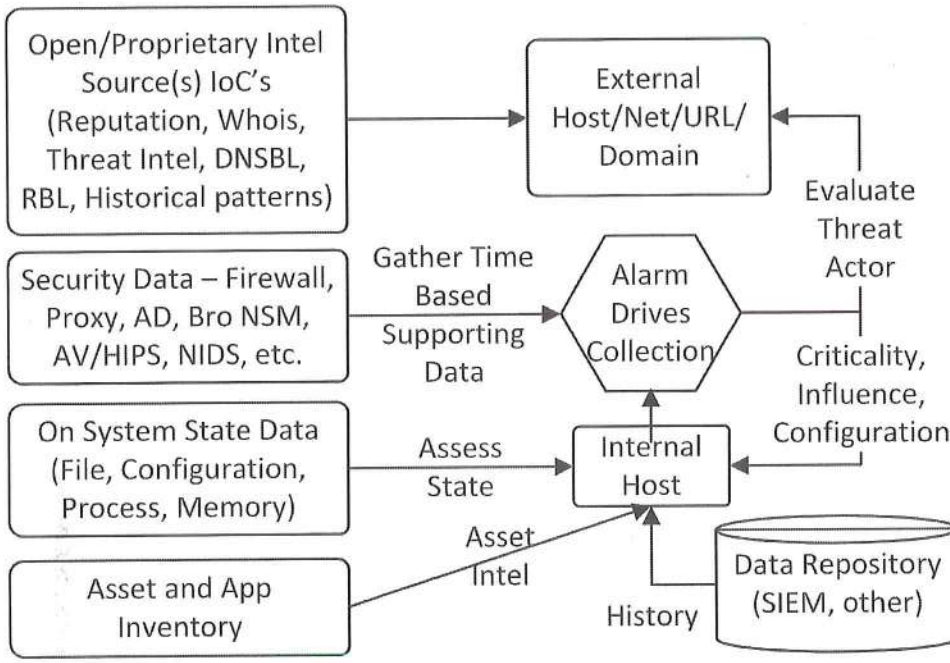


Figure 10 Review Data Sources

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.

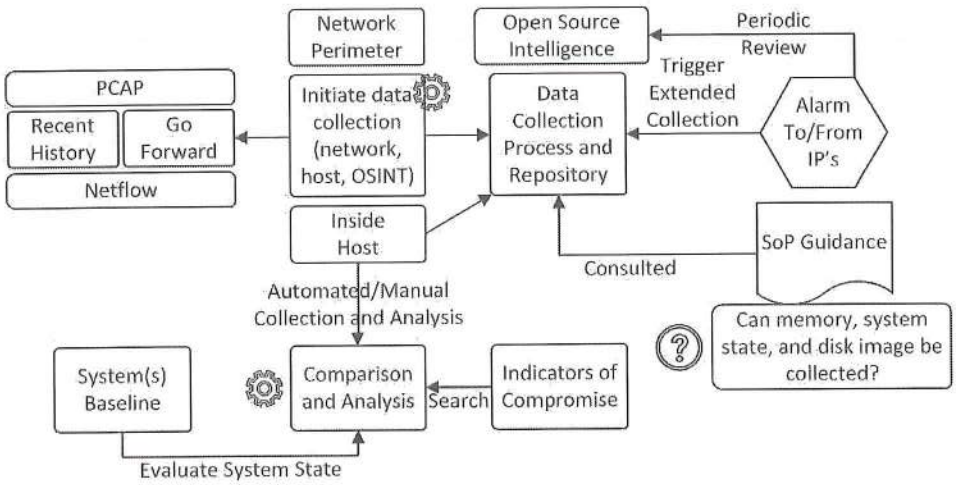


Figure 11 Data Analysis Processes

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/

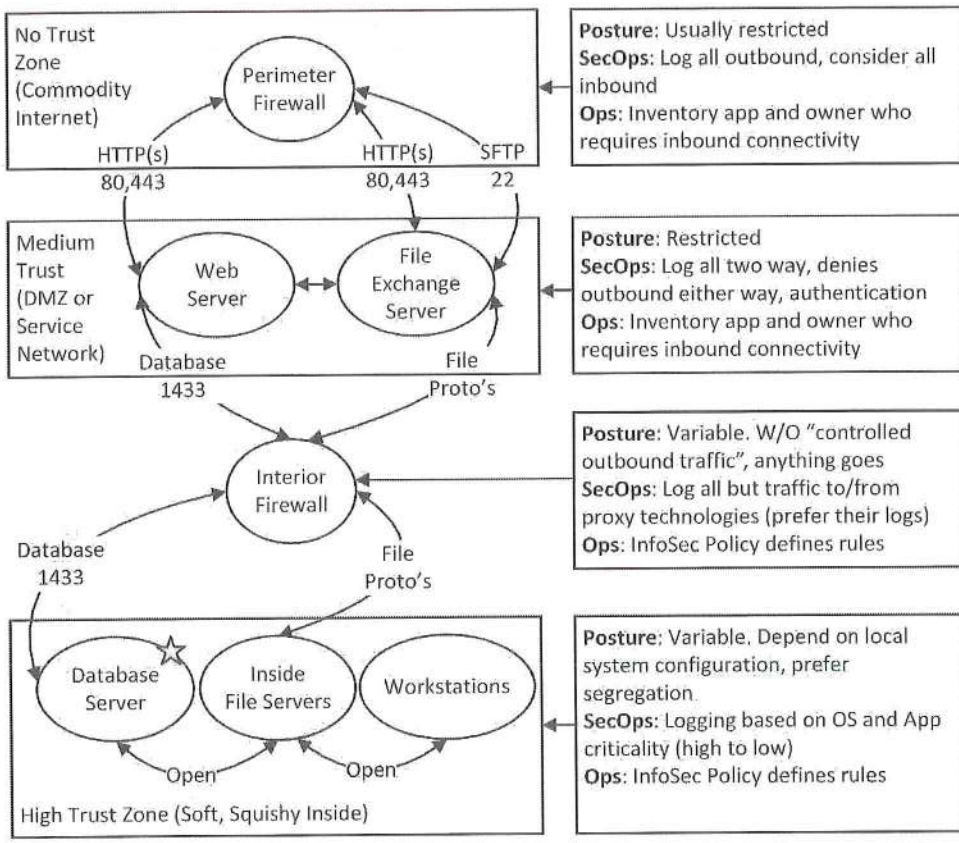


Figure 12 Graph Theory Illustrated

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

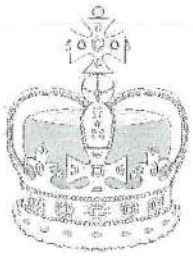
Threat hunting, for the purposes of this book, is defined as "leveraging information to proactively search out and identify if an attacker was successful in compromising your network, applications, data sources, or systems on an iterative basis". In effect, threat hunting seeks to *proactively* leverage 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

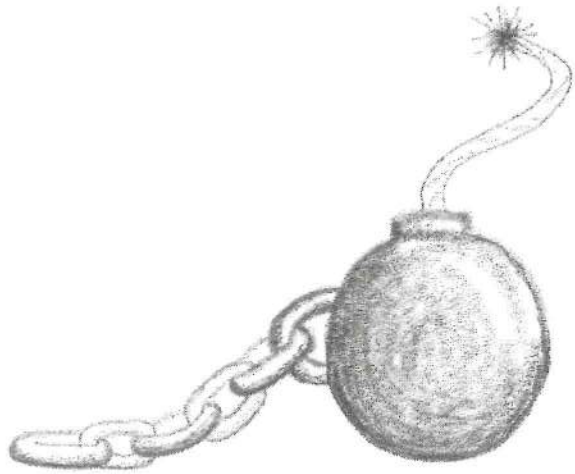
System to System communication that doesn't support the target systems usage pattern. Examples include:

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.

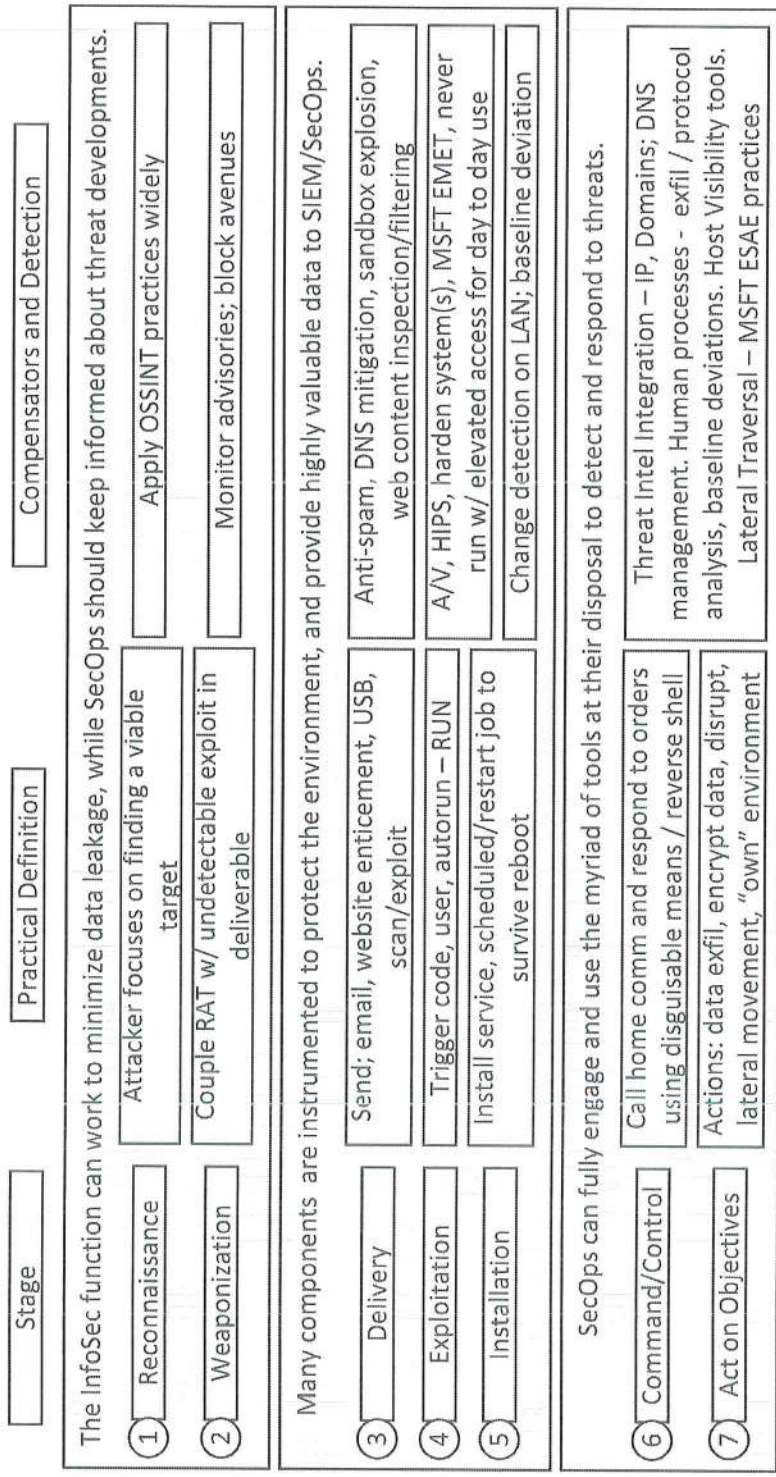


Figure 13 Lockheed Martin Cyber Kill Chain and Security Controls

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.

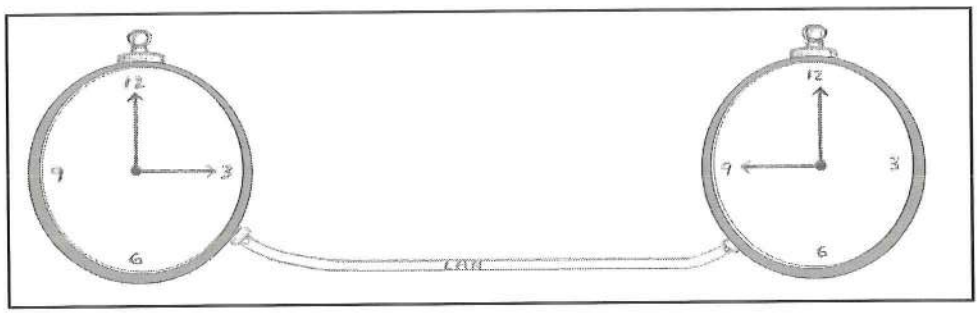
Summary

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/.

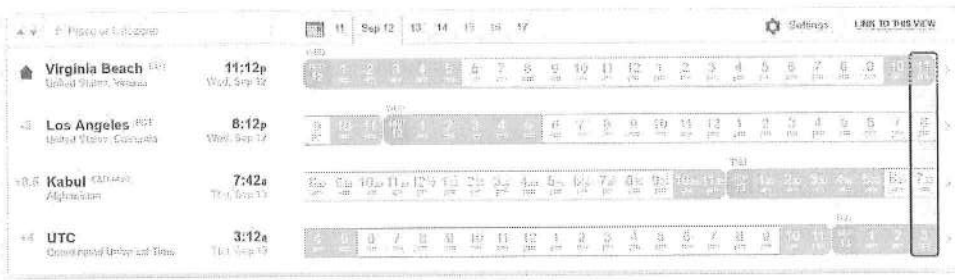


Figure 15 Time differences by time zones

- 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

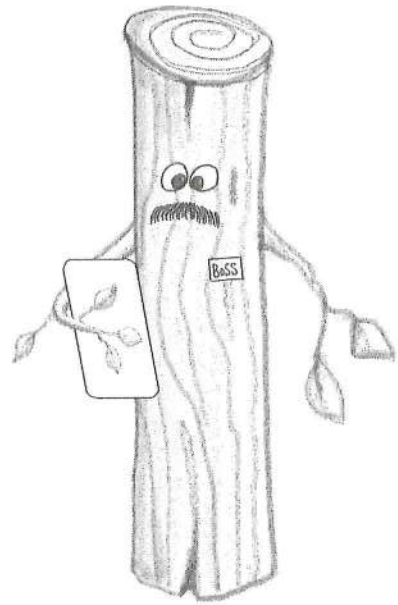
Step	Action
	is can be more expedient to disprove a theory and then further instrument the system to detect if the event happens again.
10	**Summarize findings:** Since you have significant amounts of time to investigate, gather, and perform an analysis, cross communicate your process to at least one 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.

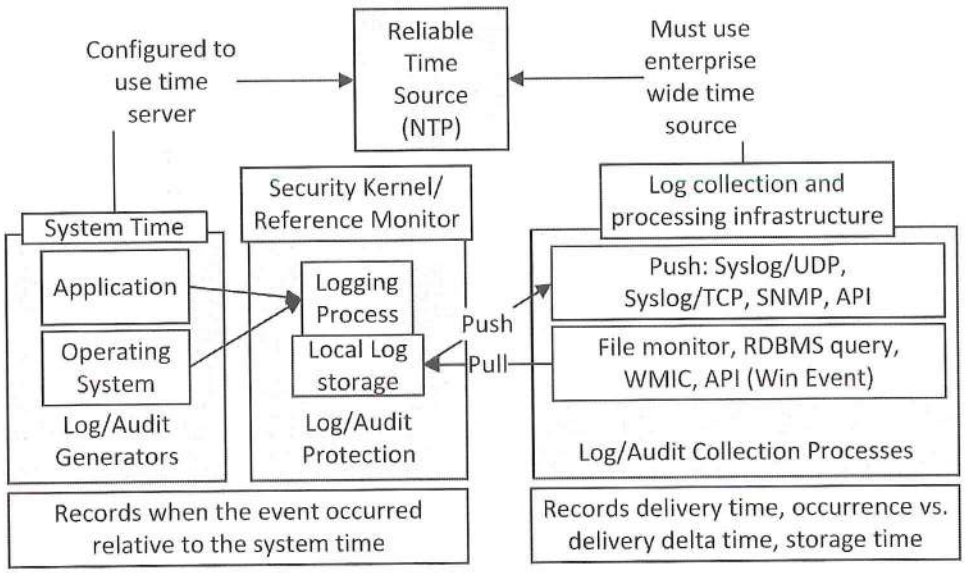


Figure 16 Logging Generation, Timestamps, and Collection Components

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:

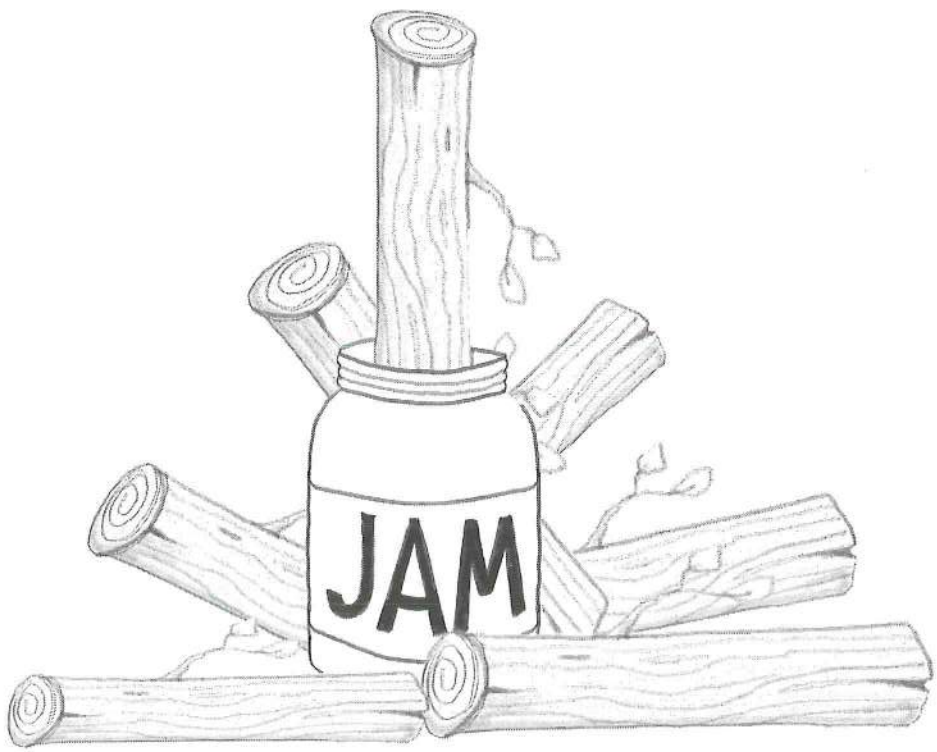
1. In the perimeter firewall, allow outbound traffic from the internal servers *to* the selected time servers over port 123/UDP (NTP uses port 123 over UDP). Then configure a separate rule to log traffic that doesn't fit this pattern, and then create an alarm condition or a daily report in your SIEM 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.

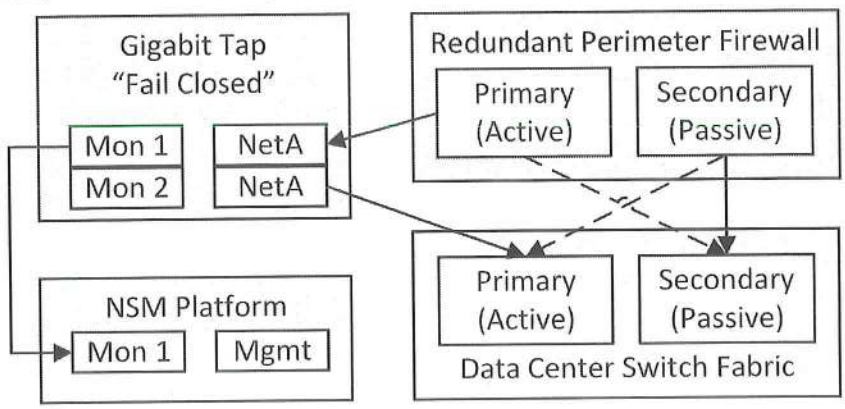


Figure 17 NSM Schematic

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.

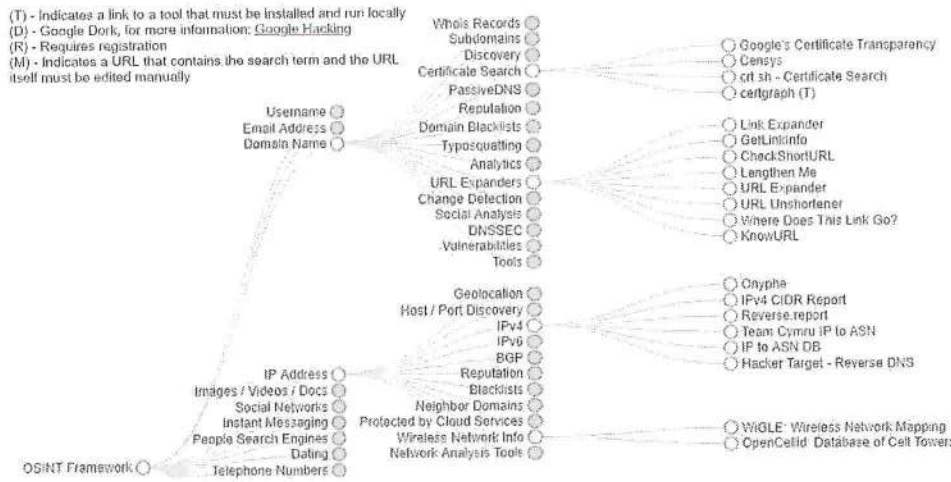


Figure 18 www.osintframework.com with Legend

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.

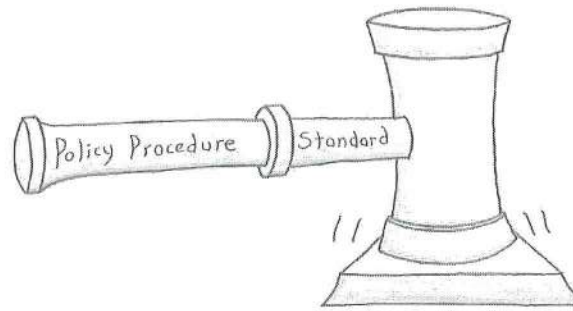
MITRE ATT&CK

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

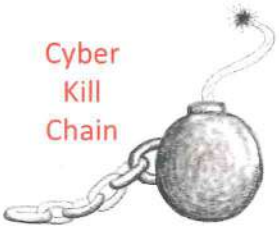
Index

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