Blue Team Handbook: SOC, SIEM, and Threat Hunting Use Cases

Security Operations Center (SOC): "A centralized team in a single organization that monitors the information technology environment for vulnerabilities, unauthorized activity, acceptable use/policy/procedure violations, intrusions into and out of the network, and provides direct support of the cyber incident response process."

Modern data collection should be user attributable, be as close the application as possible, and provide execution context because the modern attacker must live off the land, which means they need to change the OS and use scripting languages present on the system.

Once operating system data is collected, then focus on valuable network level trace data. Network level instrumentation should focus on chokepoints, flow data, and application support intelligence such as DNS activity, web browsing activity, and network flows between network segments. For example, workstation to workstation traffic is highly unlikely in most corporate networks.

A use case is "a set of actions or steps which define the interactions between an actor, which can be a person, a system, or a service, to a system in order to achieve a particular objective." The security focused use case development process involves evaluating data by establishing normal baselines and other analysis dimensions. Characteristics to understand include volume, peaks/lulls, outliers, averages, frequencies of types of data or specific elements, duration of normal behavior, and how do you find something "new".

Today, the most likely avenue of attack is against the end user through some mechanism that entices them to visit a site, download a file, or click a link in an email and then ignore security warnings. Collectively, these attacks all fall under the umbrella term "social engineering".

Below is a list of the top ten security use cases that a SOC team should implement as early as possible:

1. Privileged Entity Monitoring.
2. Brute force Authentication failures.
3. Authentication Anomalies a. Service Accounts used for interactive logon. b. Service Accounts used from non-authorized source systems. c. Interactive User authentication from multiple source systems.
4. Session Anomalies. a. The typical end user should have a session beginning and ending with ten (or less) hours from each other. b. Significant profile change in web browsing habits. c. Spike in outbound firewall denies. d. Workstation network to workstation network communication.
5. Account Anomalies
6. Data Exfiltration indicators a. HTTP(S) Send/Receive mismatch. Data received from a site is often many times data sent to a site, by byte volume, as most of the time the browser is downloading a file and rendering it for the user.
7. Signature Matches to known Vulnerability Scan Results.
8. Any excessive 'service failures'.
9. Insider Threat Indications.
10. Security Log Data failure conditions.

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

Email Use Cases:

1. The only systems that should be communicating to any one of the messaging TCP ports should be well known and understood (such as the internal messaging systems). For continuous monitoring, the system should create an alarm for traffic outbound on these ports from a non-authorized source.

2. Significant volume changes: From a threat hunting perspective, the team should look for volume-based changes, such as a user who rarely sends attachments suddenly sends a large number of attachments to a competitor may indicate intellectual property theft or industrial espionage.

3. Autoforwarding: Users who send large amounts of data to their home email addresses may expose the organization to an unacceptable risk.

4. Email with competitors: Most, but not all, organizations do not routinely send a large portion of email with a competitor. This pattern is also subjective, but it may reveal an insider threat.

5. Users generating numerous Non-Delivery Reports: This condition may indicate their account is being used to probe for valid email addresses at a particular domain or some operational issue.

6. Constant email transmission: Users sending email every hour of the day, which may indicate something on their system is attempting to use an email capability for covert communications.

Antivirus Use Cases:

1. The immediate web, email, and USB device history right before malware observance should be checked. A ".scr" file is most often a screen saver, which are often malicious. Realize that a screensaver, by its very nature, captures the username and password.

2. Repeat A/V offenders or reinfections: Users who routinely get an infection notice need more attention. This use case is longitudinal in nature, meaning that you want to know if a machine is being re-infected over time, say 14 to 60 days.

3. Escalation based on Asset Value: Assuming that your SOC team has valid intelligence on the value of an asset, an infection on a "critical" asset such as one that falls under the Payment Card Industry Data Security Standard warrants more rapid attention than an asset with a "medium" value.

4. A user with Elevated Access logging into an infected system: This use case requires that you maintain an inventory of users with elevated access, or that all of these users have a particular naming convention so elevated access accounts can be more readily detected.

DNS Monitoring Use Cases and Detection Patterns:

1. Young (< 7d old) or recently registered domains (and thus, websites): Malware is increasingly using sophisticated DNS lookups and query types to signal their command and control network. Attackers, and in particular Phishers, are using recently registered domains as spreader points.

2. Names not in the Top 1 Million List.

3. Long, misshapen, or weird second level domain names: Most second level names should be less than 24 characters. DNS names have a maximum of 255 characters in total. In practice, some analysis should be performed on DNS names that are 72 characters or longer.

4. Private IP addresses returned: Name server queries to Internet sites should rarely return private (RFC 1918) IP addresses.

5. DNS queries not from authorized servers: An enterprise should only have a small number of internal DNS servers that can forward queries to servers on the Internet. Any DNS query outside of this boundary should be investigated.

6. Volume and volume profile changes: Establish a baseline profile for DNS traffic. These indicators can become alarm conditions once baselines are established.

7. Foreign countries: You should study your organization's communication and operating model to determine how much communication occurs to countries outside of your own country.

8. Traffic to external IP without DNS query: Direct HTTP, HTTPS, FTP, SSH, and likely other protocols directly to an IP address is suspicious.

End Point Detection Use Cases:

1. IoC hit: IoC hits when the EDR system detects a connection to a suspicious or nefarious IP address or domain name, or a file that matches a known bad by hash value.

2. Binary first observed: A "first occurrence" of a binary, never seen before in the environment, once baselining is done can detect unauthorized software installs, malicious software, unauthorized downloads, or software executing from removable media.

3. Registry Key: Modification of a specific registry key used to establish persistence, such as the Run, Run Once, or RunOnceEx.

4. Specific directories: Modification of directory or file within a file system.

Account lifecycle use cases:

1. Short cycle account create and account delete events: This use case catches accounts that are created and removed within a very short time window. As a bonus, the severity would be raised if the account was used, such as a logon event between the create and delete event.

2. Short Cycle elevated group add and group remove events: This use catches accounts added to highly privileged groups like "Domain Admins" and then quickly removed from the group. Extensive damage can be done in a short time.

3. Accounts created/modified/disabled by staff other than designated account managers: This condition helps identify policy violation, rogue admins, or attackers who gain access to a domain admin level credential.

4. Accounts that do not follow an established naming convention: Detection can be accomplished through regular expression pattern matching or account length checking. In the weakest case, simple account name length checks may work, or a daily human review of accounts created, enabled, disabled, or removed from the network and the AD forest.

Account Logon Use Cases:

1. Concurrent console logons (4624, type 2) from multiple sources within a short timeframe: This condition indicates an account is being used from multiple systems. For most users, any count above two is out of the ordinary.

2. Logons from internal and external, within a short window, not over RDP (4624, type 10): This condition may indicate account misuse, credential theft, account sharing, or a behavioral issue.

Operating system stability Use Cases:

1. Security service failures: The use case relates to security focused services failing, because that can indicate the environment cannot be properly monitored or active tampering is occurring.

2. New Services: Windows records a new service installation with Event ID 4697. These are infrequent events and should be supported with a change control item.

3. New Scheduled Task: Windows records this event using ID 4698 in the security log a very common and almost 100% reliable indicator of lateral movement when there are local logons (4624), new service (4697) and new task (4698) within two minutes of each other.

Some consistent percentage of your user population will forget their account credentials routinely, every Monday morning, and will repeatedly try to login will lock their accounts, wait a little bit, and eventually call the Service Desk for assistance.

Outside of that window, repeated account lockout conditions that repeatedly occur indicate one of a few things: • A completely misconfigured system or application • A user with a device that has an old credential that needs to be updated • The account that is repeatedly locked out or failed to logon and is under a password guessing attack.

Brute Force authentication use cases:

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

2. For monitoring account lockout, start with 5 consecutive lockouts before you trigger a brute force alarm, assuming the account lockout policy is set to 5 failed attempts and then the account locks for a short period of time like 1 minute.

3. Password Spray: This attack attempts to use a single likely to be true password during the authentication attempt against all accounts in the domain, one time per account. This type of attack can be found by searching for failed logons for multiple unique user accounts from the same source machine (name or IP address).

NIDS/NIPS Use Cases:

1. Same alert, high volume, single target: Repeated alerts directed towards the same "target" need to be either a) tuned because they are most likely a false positive orb) should be investigated because the rule is firing on a serious condition.

2. Same alert, multiple targets: When an alarm arrives for multiple targets, the same general rule applies - determine if the rule can be tuned based on an understandable condition, and if not, investigate.

3. Multiple alarms, same system: There are several rule conditions which can "stack" on one another or relate to part of the kill chain. When a system has multiple different alarms, especially if those alerts indicate that a machine is both the source and destination, then it is a sure sign of compromise.

4. Vulnerability Correlation: when a signature detects traffic against a port or service that has a known vulnerability validated from a vulnerability scanner, you have a high value alarm.

5. Known command and control: Botnet or command and control triggers based on IP addresses, domain names, pulse patterns, user agents, and other conditions. Note that malicious domain names are frequently updated.

6. TLS/SSL Blacklists: There are known certificates which are used by malicious software and botnets. This type of alarm can be applied to HTTPS traffic by matching the certificate necessary to setup the TLS connection.

7. High alarm counts and singleton events: Alerts at either end of the spectrum need specific attention. The top 3 to 5% of ID/PS alerts should be checked daily in order to tune them so they occur less often, or can be disabled. In contrast, there is hidden gold in 'singleton' alerts, particularly if there are a few unique singleton related IDS signatures for the same source or destination.

Perimeter use case requirements:

1. NAT: The SOC must know all of the external DNS entries and NAT translations in order to have the best possible awareness and ability to correlate internal private IP addresses to external IPs. A usage mapping is also essential.

2. Top 1M: Determine your daily baseline, such as top 10,000 external IPs and whether they are in the a top 1M domain/ IP list.

3. Protocols in use and protocol volume: From there, you can compare each day against this baseline to locate potential deviations, abnormal flows, or systems communicating to potential suspects using an unusual protocol.

4. Persistence: the ability to detect persistence, such as a connection lasting more than 24 hours is a key capability.

5. Forgery and Private IP: Forged IPs, private IPs attempting to egress, and other traffic anomalies.

Top Ten IP Address use cases:

1. Top Ten "outbound connections"+ Top Ten "data flow"+ Top Ten "Workstations" or "Servers": This condition may indicate data exfiltration, and can be used to profile the overall network activity. When an IP appears in all three lists, spend time confirming if the machine is compromised.

2. Top ten outbound with a connection lasting more than 24 hours: This is another example of a possible data exfiltration, particularly if it comes from the workstation side.

3. Top Ten (or more) DNS requests for newly registered domains: If you have a DNS logging capability or have implemented PassiveDNS, then you can take "yesterdays" DNS queries and compare it to newly registered DNS names.

4. Top Ten outbound denies by source and/or destination port. This type of a threat hunting activity can also help identify operational issues, like systems not using proper DNS, SMTP, proxy, or NTP servers.

Web proxy use cases:

1. Suspicious user agents: End user desktop user agent strings should identify the browser, operating system, and may identify some key features. The list of user agents should be reviewed using long tail analysis. You are looking for user agents that are not used by your installed browser base, such as a web spider, a scanner, misspellings of browser user agents, the Python programming language, or applications self-identifying by the user agent string.

2. First time use sites: Users within an organization will display a habit of using the same set of sites, so the ability to detect a new site can be very useful. The very first time a site is seen may be a result of a user clicking on a spam my link, a banner add that takes a user to a malicious site, or some other condition.

3. Consistent, repeatable browser pings or beacons: Small sets of data going to the same site (meaning DNS name) over time indicate that a persistence mechanism or something nefarious is in place on the sending system.

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

1. How is remote administration and what are the remote execution tools used to perform remote management by system custodians? Common methods include direct RDP, WMIC, WinRM, psexec, ssh access, Group Policy, and package build and deployment.

2. How is remote access to servers granted? Are users added directly to a local group, is there an AD group, or is there a privileged access solution?

3. What are the network flows for remote access? For example, are jump boxes used, specific segments for IT management, are end users granted RDP or SSH access, and what are the user accounts.

“Living off the Land” (LOLBAS) of interest:

- at.exe: Used to schedule a job.
- attrib.exe: Can be used to hide files and directories.
- cmd.exe: There are a number of oddities that you can detect when cmd.exe is the parent process. Of note - cscript.exe. wscript.exe, and powershell.exe can be used by attackers. Further, when a productivity application such as Word, Adobe or Excel launch cmd.exe, the source file is most likely up to no good.
- cscript.exe/wscript.exe: These are older scripting tools, predating PowerShell, and are still viable today.
- dsquery.exe: used to extract user and group information.
- fsinfo.exe: Used to get the list of connected drives
- ipconfig.exe: Get the NIC and DNS configuration.
- .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.
- ping.exe: Test connectivity using ICMP.
- 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.
- 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.
- 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
- sdelete.exe: This Sysinternals tool is used to securely wipe the contents of a file.
- 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.
- vssadmin.exe: The volume shadow service administration tool. Adversaries use this tool to create, disable and/or delete volume shadow copies.
- 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: Has hundreds of query capabilities about a system and can also interact with remote systems.

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.

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.

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.

2. Design the question that the use case should answer. How would the attacker gain needed access, cause damage, exfiltrate data, or what accounts would they need to use?

3. Determine and test the data sources and the data elements that provide the visibility needed to answer the question.

4. Evaluate the data by establishing normal baselines and other analysis dimensions. Characteristics to understand include volume, peaks/lulls, outliers, averages, frequencies of types of data or specific elements, duration of normal behavior, and how you find something "new".

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.

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.

Blue Team Handbook: A Day in the Life of a SOC Analyst

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.

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 Alarm 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.

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.

Analyst Action Examples:

1. Assess and close alerts that are non-actionable with a supporting indicator or reason code
2. Close alerts that are confirmed as a false positive.
3. Escalate the alarm when it is beyond skill level to assess.
4. Process the alert, based on current skill level.

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.

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. 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.

2. Check for Threat Intel activity. 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.

3. As any environment changes, so will the security event data.

4. Inspect assets for vulnerabilities that may have appeared.

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. There is nothing worse than working an alarm or an incident, looking for data, only to find out that it is not available.

When the SOC team is not responding directly to alerts, they should take on other support and maintenance tasks. Event data review can also provide operational awareness, point out issues, and be used to keep systems running well. 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.

Blue Team Handbook: Alarm Investigation Process

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.

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".

Check for supporting data directly relating to the "suspect" (the system that caused the alarm.) 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?

2. Endpoint Detection and Response (EDR): a. "First Run Binaries" - an executable that has not been seen in the organization. b. Active and recent network connections. c. Watchlist hits and submissions. 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: a. Did the suspect system generate more than a few NXDOMAIN responses? b. Consistent DNS communication to a specific domain, most often not in the top 1M domain lists, or a newly created domain(< 30d old). c. DNS queries where the hostname domain name (not the TLD) score low for the entropy score.

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?

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? 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? c. Perform top one million site checks.

7. Authentication sources (AD, database, application, email): a. What user accounts authenticated from the suspect IP in the last hour? Day? 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).

8. HIDS (such as OSSEC, sysmon, 4688 events, and OsQuery): a. Have there been registry key or file system changes that cannot be explained? b. Are there process command lines t hat 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. Is the asset (host) under recurring attack?

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.

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? 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.

The initial decision, or the opening move, during the analysis process affect the close rate and time of an alarm. The first stage is the opening move, or the immediate triage phase where quick decision is made whether or not to investigate the alarm.

Assuming the alarm will be investigated, then there are decisions on what data to retrieve and how to go about getting that information. Analysts needs to determine what conditions must be present for the alarm to be a "true positive".

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.

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.

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.

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.

One particular issue that every analyst must come to grips with is cognitive bias. 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."

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.

John Lambert (Microsoft): ""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."

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".

Blue Team Handbook: 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).

How to Generate Hypotheses for Successful Threat Hunting:

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. Using IoC's and Tools, Tactics, and Procedures (TTP's) of an adversary 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.

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.

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.

Anomalous Device Communications: Devices normally communicate in well-known patterns, such as ephemeral TCP to 80/443 for web traffic, 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.

Example Threat Hunt Check List:

1. Prepare and execute threat hunting a. Search for signs of Command and Control i. Look for beacons ii. Review the top 20 IPs with the greatest number of connections, the longest connection time, and the most about of data moved. 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. Review the event types and alerts generated from systems that contain the most sensitive data. b. Search out 4624 authentication events from within systems on the network and look for odd patterns. c. 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.

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.

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.

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.

A typical value is between 1:10 to 1:20, depending on the site. 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.

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.

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.

Beaconing 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.

Process execution: There are numerous tools available to attackers which can manipulate a system and establish persistence. In order to view these events, collect 4688 and make sure that Detailed Tracking 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.

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.

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.

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. 2. ICMP traffic between workstation networks.

When the SOC team detects activity at that matches one of the steps in the chain, they can immediately pivot based on the Iocal alarm and look backwards in current event and alarm data while the issue is resolved.

"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 (IoA)." An IoA differs from an IoC because IoAs focus on what an attacker is attempting to accomplish.

Blue Team Handbook: 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.

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.

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.

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.

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.

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.
b. Asset Criticality.
c. App to Server to Storage mapping relationship.
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 don't 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.
i. External network ranges, NAT translations, and DNS names.

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.

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.