Ark Jain

27th October, 2025

Theoretical Knowledge

1. Threat Hunting Methodologies

At its heart, threat hunting is a paradigm shift from a reactive to a proactive security posture. It operates on the assumption that determined attackers are already inside the network and focuses on finding them before they can achieve their objectives.

- **Proactive Threat Hunting vs. Reactive Incident Response:**
  - **Reactive Incident Response:** This is the traditional approach to security, where actions are triggered by an alert from a security tool like an antivirus program, an intrusion detection system, or a SIEM. The focus is on damage control after a potential threat has been identified.
  - **Proactive Threat Hunting:** This is an analyst-driven process that doesn't wait for an alert. It involves actively and iteratively searching through networks and datasets to detect and isolate advanced threats that have evaded existing security solutions. The primary goal of threat hunting is to significantly reduce the "dwell time" – the period between the initial compromise and its discovery.
- **Hypothesis-Driven Hunting:** This is a core tenet of modern threat hunting. Instead of aimlessly looking for "anything suspicious," hunters start with a specific, testable hypothesis based on their knowledge of attacker behaviors. A hypothesis acts as a guide for the investigation. For example:
  - **Based on Threat Intelligence:** "A threat actor known to target our industry uses scheduled tasks for persistence (MITRE ATT&CK T1053). Let's search for anomalous task creations in our environment."
  - **Based on TTPs:** A common tactic for attackers is to gain legitimate credentials and use them to blend in with normal network activity. This is known as "Valid Accounts" misuse (T1078). A hunt could be initiated with the hypothesis: "An attacker has compromised a user's credentials and is using them to access systems they don't normally interact with."
  - **Example: Hunting for Anomalous Privilege Escalation:** An attacker who has gained initial access to a system will often try to escalate their privileges to gain administrative control. A threat hunter can form a hypothesis like: "An attacker is exploiting a known vulnerability to

escalate privileges by manipulating Windows registry keys." The hunt would then involve searching for unusual modifications to sensitive registry keys, especially those related to autorun or service creation, and correlating them with other suspicious activities like abnormal process injections.

2. Threat Hunting Frameworks

To provide structure and ensure that hunts are repeatable and efficient, several frameworks have been developed. These frameworks guide analysts through the entire hunting process, from forming a hypothesis to responding to a discovery.

- **SqRR (Search, Query, Retrieve, Respond):** While not as formally documented as others, the principles of SqRR are foundational to many hunting loops. It represents a continuous, iterative process:
    - **Search:** Proactively look for signs of malicious activity.
    - **Query:** Ask specific questions of your data to validate or invalidate your hypothesis.
    - **Retrieve:** Pull the relevant data and analyze it for anomalies and patterns.
    - **Respond:** If a threat is found, initiate the incident response process. If not, refine your hypothesis and begin the loop again.
- **TaHiTI (Targeted Hunting integrating Threat Intelligence):** Developed by a consortium of Dutch financial institutions, TaHiTI provides a structured, step-by-step process that heavily integrates threat intelligence. Its core idea is to use threat intelligence not just as a starting point for a hunt but also to enrich and contextualize findings throughout the investigation. The TaHiTI methodology encourages a focused, risk-driven approach where hunting activities are prioritized based on the threats most relevant to the organization.

3. Data Sources for Hunting

Effective threat hunting is impossible without rich and comprehensive data. Hunters need to be able to pull information from a wide variety of sources to get a complete picture of activity in their environment. Key data sources include:

- **Endpoint Detection and Response (EDR) Logs:** EDR solutions provide deep visibility into endpoint activities, including process execution, file modifications, registry changes, and network connections. This is often the richest source of data for hunting attacker behaviors.
- **Network Traffic:** Full packet capture (PCAP) and network flow data are invaluable for identifying suspicious communication patterns, command-and-control (C2) traffic, and data exfiltration.
- **SIEM and Log Aggregators:** Centralized log platforms that collect data from firewalls, proxies, domain controllers, and applications are essential for correlating events across different systems.

- **Threat Intelligence Feeds:** These feeds provide up-to-date information on Indicators of Compromise (IOCs) like malicious IP addresses and file hashes, as well as intelligence on the TTPs of various threat actors.

Key Objectives

The primary goal of adopting these methodologies is to develop the skills necessary to proactively identify and mitigate threats before they cause significant damage. Key objectives include:

- **Proactively Identify Hidden Threats:** Move beyond a reliance on automated alerts to find adversaries that are actively trying to blend in with normal traffic.
- **Utilize Structured Methodologies:** Employ frameworks like TaHiTI to conduct efficient, repeatable, and intelligence-driven hunts.
- **Leverage Diverse Data Analysis:** Gain proficiency in querying and correlating data from multiple sources to uncover the full story of an attack.
- **Reduce Attacker Dwell Time:** By actively searching for threats, significantly shorten the time that an adversary can operate undetected within the network.

How to Learn

- **Study Frameworks and Methodologies:** The SANS Institute offers numerous white papers and webcasts that delve into the theory and practice of threat hunting, including detailed discussions of various hunting loops and maturity models. These resources provide a strong theoretical foundation.
- **Explore Real-World Case Studies:** The MITRE ATT&CK® framework is an invaluable resource that breaks down the TTPs of numerous threat actor groups. Studying a group like APT29 (also known as Cozy Bear) provides a deep understanding of how a sophisticated adversary operates, from initial access to achieving their objectives. Analyzing their techniques will generate numerous hypotheses for practice hunts.
- **Review Practical Guides:** Tool-specific guides, such as Elastic's threat hunting guide, offer practical, hands-on approaches to implementing threat hunting concepts. These guides often provide sample queries and workflows that can be adapted to your own environment, bridging the gap between theory and real-world application.

2. Advanced SOAR Automation

To effectively leverage SOAR, it's essential to understand its foundational components, how to codify processes into playbooks, and how it integrates into the broader security ecosystem.

1. SOAR Components: The Three Pillars of Modern Response

SOAR technology is built on three interconnected capabilities that work together to streamline security operations.

- **Security Orchestration:** This refers to the coordination of disparate security tools and systems. In a typical investigation, an analyst might need to manually pivot between a SIEM, an EDR platform, a threat intelligence feed, and a ticketing system. Orchestration connects these tools via APIs, allowing them to share information and trigger actions in a unified workflow. For example, an alert in a SIEM can be used to automatically query an EDR tool for more endpoint data.
- **Automation:** This is the machine-based execution of security tasks that would otherwise be performed manually. Automation is driven by "playbooks," which are digital codifications of incident response procedures.
  - **Example (Auto-Ticketing):** Instead of an analyst manually creating a ticket for every alert, a SOAR platform can be configured to automatically ingest an alert from the SIEM, create a corresponding ticket in a system like TheHive, and assign it to the appropriate analyst or queue.
- **Response:** This component provides analysts with the tools and information needed to investigate and react to an incident from a centralized platform. It involves case management features, threat intelligence consolidation, and the ability to trigger containment actions.
  - **Example (Auto-Containment):** For a high-confidence malware alert, a SOAR playbook could automatically trigger a response action, such as instructing an EDR tool to quarantine the infected endpoint, without waiting for human intervention.

2. Playbook Development: Codifying Your Response

Playbooks are the heart of SOAR, transforming manual, step-by-step runbooks into automated workflows. Designing effective playbooks requires a deep understanding of your incident response processes.

- **Phishing Response Playbook:** Phishing is a common, high-volume threat, making it an ideal candidate for automation. A typical phishing playbook might include the following automated steps:
  1. Ingest a user-reported phishing email from a dedicated inbox.
  2. Extract observables (URLs, file attachments, sender IP address).
  3. Detonate any attachments in a sandbox for behavioral analysis.

4. Check the reputation of URLs and IPs against threat intelligence feeds like VirusTotal.
5. If observables are confirmed malicious, automatically search all other user inboxes for similar emails and delete them.
6. Block the malicious URL or IP at the firewall or web proxy.
7. Create a ticket with all enriched findings and notify the security team.

- **Malware Response Playbook:** When an EDR tool detects malware, a SOAR playbook can immediately orchestrate a response.
  1. Ingest the malware alert from the EDR.
  2. Enrich the alert with threat intelligence on the malware hash.
  3. Retrieve process and network connection data from the infected endpoint via the EDR agent.
  4. **Example (Automate IP Blocking for C2 Traffic):** If the endpoint data shows the malware is communicating with a known Command and Control (C2) server, the playbook can automatically take the malicious IP and add it to a blocklist on the organization's firewalls.
  5. Isolate the infected host from the network to prevent lateral movement.
  6. Escalate the incident to an analyst for remediation and recovery.

3. Integration with SIEM/EDR: Creating a Unified Workflow

SOAR platforms do not replace SIEM or EDR tools; they enhance them by acting as a centralized brain for the entire security stack.

- The workflow typically begins with detection. A SIEM (like Wazuh or Elastic) correlates logs to identify a potential threat and generates an alert.
- This alert is forwarded to the SOAR platform.
- The SOAR playbook then takes over, orchestrating actions across other tools, such as querying an EDR agent for deeper endpoint visibility or cross-referencing indicators with threat intelligence platforms. This seamless integration transforms a simple alert into a fully enriched case, ready for an analyst to make a decision or for the playbook to continue its automated response.

Key Objectives

Mastering advanced SOAR automation is about more than just learning a new tool; it's about fundamentally changing how a SOC operates. The key objectives are:

- **Improve Efficiency:** Automate the repetitive, time-consuming tasks that lead to analyst burnout, freeing up human experts to focus on complex threat hunting and investigation.
- **Drastically Reduce Response Times:** By codifying incident response procedures in playbooks, SOAR can execute containment and remediation actions in seconds or minutes, a process that could take hours for a human analyst.

- **Increase Consistency and Reduce Human Error:** Automation ensures that every incident is handled according to the organization's established best practices, reducing the risk of missed steps or errors that can occur during a manual investigation.

How to Learn

- **Study SOAR Concepts via Splunk SOAR Documentation:** Splunk is a leader in the SOAR space, and their documentation provides a wealth of information on the core concepts of orchestration and automation. Their guides offer detailed explanations of playbook development and how to integrate with a wide range of security tools.
- **Review Playbook Examples in TheHive Project:** TheHive is a popular open-source incident response platform that functions as the case management hub in many SOAR workflows. While not a full SOAR platform itself, analyzing how it is used in conjunction with automation tools like Cortex provides excellent, real-world examples of how incident response playbooks are structured and executed.
- **Analyze Automation Case Studies:** The U.S. Cybersecurity and Infrastructure Security Agency (CISA) has released guidance for organizations implementing SOAR. These documents, aimed at security practitioners, detail practical workflows and automation strategies for common threats like phishing, offering a government-endorsed perspective on best practices for automation.

3. Post-Incident Analysis and Continuous Improvement

A robust post-incident process is built on three pillars: understanding the fundamental cause, learning from the experience, and measuring performance to track progress.

1. Root Cause Analysis (RCA): Beyond the Symptoms

Root Cause Analysis is a systematic method for looking past the immediate, superficial symptoms of an incident to uncover the fundamental issue that allowed it to occur. The goal is to implement solutions that prevent the problem from ever happening again.

- **5 Whys:** This is one of the simplest yet most effective RCA techniques. It involves repeatedly asking the question "Why?" to peel back the layers of causality.
    - **Example: RCA for a Phishing Breach**
        1. **Problem:** An employee's account was compromised via a phishing email.
        2. **Why?** The employee clicked a malicious link and entered their credentials.
        3. **Why?** They did not recognize it as a phishing attempt.
        4. **Why?** The phishing email was very convincing and bypassed the company's technical controls.
        5. **Why?** The existing email filters were not configured to block this new type of sophisticated phishing attack.
        6. **Why?** The email security gateway's threat intelligence was outdated and its rule set was too permissive.
    - **Root Cause:** The incident was not just due to human error, but a systemic failure in the email security gateway's configuration and threat intelligence capabilities.
- **Fishbone Diagram (Ishikawa Diagram):** This visual tool helps teams brainstorm and categorize the potential causes of a problem. It is particularly useful for complex incidents where multiple factors may have contributed to the failure. The "bones" of the fish represent different categories of potential causes, such as People, Processes, Technology, and Environment.

2. The Lessons Learned Process: Institutionalizing Knowledge

The "lessons learned" or "post-mortem" phase is a structured review of an incident, what went well, what could have been done better, and what will be changed as a result. According to NIST SP 800-61, this is one of the most important parts of incident response, though it is often overlooked.

This process involves:

- **Conducting a Blame-Free Post-Mortem:** The primary goal is to understand process and technology failures, not to assign blame to individuals. A blame-free environment encourages honest and open discussion, which is essential for uncovering the truth.
- **Documenting the Incident Timeline:** A precise, chronological timeline of events—from detection and analysis to containment and recovery—is created.
- **Improving Processes, Tools, and Training:** The output of the lessons learned meeting should be a list of concrete, actionable improvements. This could include updating incident response playbooks, reconfiguring security tools for better detection, or developing new training modules for employees.

3. Metrics and KPIs: Measuring What Matters

To gauge the effectiveness of a SOC and track improvements over time, it's crucial to measure performance with key performance indicators (KPIs). Two of the most important metrics in incident response are:

- **Mean Time to Detect (MTTD):** This metric measures the average time it takes for the security team to identify that a security incident has occurred. A lower MTTD indicates a more effective monitoring and detection capability, which is critical for minimizing an attacker's dwell time in the network.
- **Mean Time to Respond (MTTR):** This metric measures the average time it takes to contain, remediate, and recover from an incident *after* it has been detected. MTTR is a key indicator of the efficiency and effectiveness of the incident response team and their playbooks.

By consistently tracking these metrics, a SOC can set performance goals, justify investments in new tools or personnel, and demonstrate improvement to leadership.

Key Objectives: Driving Continuous Improvement

The ultimate goal of post-incident analysis is to create a feedback loop that drives continuous improvement throughout the SOC and the broader organization. The key objectives are to:

- **Prevent Incident Recurrence:** By identifying and addressing the true root cause, you can fix systemic weaknesses and significantly reduce the likelihood of a similar incident happening again.
- **Strengthen Security Defenses:** The findings from post-mortems should directly inform changes to security controls, leading to a more resilient and hardened environment.
- **Enhance SOC Performance:** By refining playbooks, improving communication, and leveraging metrics like MTTD and MTTR, the SOC can become faster and more effective in its response to future incidents.

How to Learn: A Path to Mastery

- **Study RCA Techniques via SANS Reading Room:** The SANS Institute provides a wealth of white papers, webcasts, and articles that explore various methods for conducting effective Root Cause Analysis in the context of cybersecurity incidents.

- **Review NIST SP 800-61 for Post-Incident Guidelines:** The National Institute of Standards and Technology's Special Publication 800-61, "Computer Security Incident Handling Guide," is the foundational document for incident response in the United States. Its section on "Post-Incident Activity" provides a comprehensive framework and set of best practices for conducting lessons learned sessions and using the data to improve security posture.
- **Explore SOC Metrics via CISA's Resources:** The Cybersecurity and Infrastructure Security Agency (CISA) offers guidance and best practices for running a security operations center, including information on the most critical metrics to track. Their resources can provide a practical understanding of how to implement and interpret KPIs like MTTD and MTTR to drive performance.

4. Adversary Emulation Techniques

Adversary emulation is a sophisticated security assessment that goes beyond traditional penetration testing. Instead of just looking for vulnerabilities, it involves mimicking the specific behaviors and attack chains of real-world adversaries to test an organization's detection and response capabilities in a controlled manner. This threat-focused approach provides a holistic view of security by simulating how an actual attacker would operate within the network, including attempts at lateral movement and maintaining stealth.

- **Simulating Attacker TTPs:** The process is deeply rooted in cyber threat intelligence. Emulation plans are meticulously crafted based on public reports and frameworks like MITRE ATT&CK® to model the behavior of specific threat actors, such as Advanced Persistent Threat (APT) groups. For instance, to test defenses against a particular threat actor known for using phishing and exploiting remote services, an emulation plan would include:
  - **T1566 - Phishing:** Simulating a spear phishing campaign to see if malicious emails are blocked by filters, if employees click the links, and if the resulting endpoint activity is detected.
  - **T1210 - Exploitation of Remote Services:** Attempting to exploit a known vulnerability in a public-facing application to test patching cadence, firewall rules, and intrusion detection systems.

2. Emulation Frameworks: Automating the Adversary

To execute these complex simulations in a consistent and scalable way, specialized frameworks have been developed. These tools allow security teams to automate adversary behaviors and measure the defensive response.

- **MITRE Caldera:** Caldera is an open-source adversary emulation platform that enables security teams to launch automated security assessments. It is built on the MITRE ATT&CK® framework and can be used to:
  - **Test and Validate Defenses:** Caldera can execute a sequence of TTPs against a network to see if security controls and sensors detect and alert on the malicious behavior.
  - **Automate Red Teaming:** It can be used for both fully automated engagements and to assist manual red team operations.
  - **Example: Simulating a Spear phishing Attack:** An operator can use Caldera to create an "adversary" profile that includes techniques associated with a phishing campaign. The platform can then automate the delivery of a benign payload and the subsequent actions an attacker would take post-compromise, such as reconnaissance or establishing persistence. This allows the blue team to see a realistic attack chain and validate their detection and response playbooks.

3. Red-Blue Team Collaboration: The "Purple Team" Mindset

Adversary emulation is most effective when it's a collaborative effort between the offensive (Red Team) and defensive (Blue Team) sides of a security organization. This collaborative approach is often referred to as "Purple Teaming."

- **Informing Defensive Strategies:** The Red Team executes the emulation plan, simulating the attacker's TTPs. The Blue Team's job is to detect and respond to this activity. The immediate feedback loop is invaluable. If the Blue Team misses a particular technique, both teams can work together to understand why. Was it a lack of logging, a misconfigured tool, or a poorly written detection rule?
- **Improving Detection Rules:** Based on the results of the emulation, the Blue Team can tune their security tools and write new, more robust detection analytics. For example, if the emulation showed that a particular PowerShell command used for reconnaissance went undetected, the Blue Team can develop a new rule in their SIEM to specifically alert on that behavior in the future. The exercise is then repeated to validate that the new control works as expected.

Key Objectives: Enhancing SOC Preparedness

The primary goal of adversary emulation is to move beyond theoretical and compliance-based security to a state of continuous, evidence-based improvement. Key objectives include:

- **Simulate Adversary Behaviors:** Develop the capability to mimic the TTPs of threats that are most relevant to the organization, based on solid threat intelligence.
- **Enhance SOC Preparedness:** Go beyond standard penetration testing to train the SOC on detecting and responding to multi-stage, sophisticated attacks that mirror real-world campaigns.
- **Validate Security Controls:** Use emulation to empirically test whether security technologies, processes, and personnel are effective at detecting and stopping known adversary behaviors.

How to Learn: A Practical Path to Emulation Expertise

- **Explore MITRE Caldera:** The best way to understand an emulation framework is to use it. The official Caldera documentation and GitHub repository provide extensive information on installation, creating adversary profiles, and launching operations. This hands-on experience is critical for learning how TTPs are mapped to executable actions.
- **Study Adversary Emulation Case Studies:** MITRE has published detailed adversary emulation plans for groups like APT3 and APT28 (also known as Fancy Bear). These plans break down the group's known TTPs into specific, command-by-command actions, offering a blueprint for how to construct a

realistic emulation. Analyzing these case studies provides deep insight into how threat intelligence is translated into an actionable test plan.

- **Review Practical Emulation Guides:** Organizations like Red Canary produce a wealth of practical content on threat detection and response. Their guides and open-source tools, such as Atomic Red Team, provide a library of simple, executable tests mapped to ATT&CK techniques. These resources are excellent for getting started with testing individual TTPs and understanding how to build up to more complex emulations.

5. Security Metrics and Executive Reporting

To truly evaluate SOC performance, teams must move beyond simple output metrics (e.g., "number of alerts") and focus on those that measure efficiency, effectiveness, and overall impact.

- **Dwell Time:** This is one of the most critical security metrics. It measures the total time from the moment a compromise occurs to the moment it is detected. A long dwell time gives an adversary a larger window to escalate privileges, move laterally, and exfiltrate data. Reducing dwell time is a primary objective of any mature security program.
- **False Positive Rate:** This metric tracks the percentage of alerts that, upon investigation, turn out to be benign. A high false positive rate can lead to "alert fatigue," where analysts become desensitized and may overlook genuine threats. Tracking this helps in fine-tuning detection rules and systems. A healthy range is often considered to be between 1% and 5%.
- **Incident Resolution Rate:** This KPI measures the percentage of identified incidents that are successfully contained and remediated within a given period. A high resolution rate (ideally above 90%) indicates an efficient and effective incident response process.
- **Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR):**
    - **MTTD:** Measures the average time it takes for the SOC to *identify* a security incident. A shorter MTTD reflects strong threat hunting and detection capabilities.
    - **MTTR:** Measures the average time from the *detection* of a threat to its complete mitigation. A low MTTR demonstrates a rapid and efficient incident response process, which is crucial for minimizing damage.

2. Executive Reporting: Translating Technical Data into Business Impact

Executive leadership and the board of directors are not interested in the technical minutiae of cybersecurity; they care about business risk, financial impact, and compliance. Effective executive reporting bridges this communication gap.

- **Clear Visualizations and Narratives:** Instead of presenting raw data, use visual dashboards, heatmaps, and trend lines to tell a story. For example, a chart showing a consistent decrease in MTTR over several quarters is a powerful visual that demonstrates improving efficiency.
- **Focus on Business Alignment:** Frame security metrics in the context of business objectives. Don't say, "We blocked 2 million malicious emails." Instead, say, "Our enhanced email security controls prevented an estimated X number of business email compromise attempts this quarter, safeguarding our financial assets."
- **Quantify Risk in Financial Terms:** Whenever possible, translate cyber risk into financial impact. Models like FAIR (Factor Analysis of Information Risk)

can help assign a dollar value to potential risks, making the importance of security investments immediately clear to non-technical stakeholders.

3. Continuous Improvement: Using Metrics to Drive Action

Metrics are not just for reporting; they are essential tools for identifying weaknesses and driving continuous improvement in SOC operations.

- **Identifying Gaps:** Metrics provide clear, data-driven evidence of where processes or technologies are failing. For example, a consistently high **MTTD** is a strong indicator that the SOC's detection capabilities are lagging. It points to a need for more advanced detection tools, better threat intelligence, or more proactive threat hunting.
- **Proposing Solutions:** With metric-driven evidence, security leaders can make a much stronger case for investment. Instead of saying, "We need a new EDR solution," a CISO can present data showing that "Our MTTD is 30% higher than the industry average for our sector, and an EDR solution is projected to reduce that time by 40%, significantly lowering our risk of a major breach."
- **Tracking Progress:** Once a solution is implemented, the same metrics are used to track its effectiveness over time, demonstrating a clear return on investment (ROI).

Key Objectives: From Measurement to Mastery

The ultimate goal is to build proficiency in both the science of measurement and the art of communication to elevate the SOC's role within the organization.

- **Build Proficiency in Measuring SOC Performance:** Go beyond surface-level numbers to track meaningful KPIs that reflect the true effectiveness and efficiency of security operations.
- **Communicate Results to Leadership:** Master the ability to translate complex technical data into clear, concise, and compelling reports that align with business goals and inform strategic decision-making.

How to Learn: A Path to Proficiency

- **Study SOC Metrics via SANS Reading Room:** The SANS Institute offers numerous whitepapers and resources, such as "Measuring SOC Success," that provide in-depth analysis of which metrics are most valuable and how to implement them effectively.
- **Review CISA’s Cybersecurity Metrics:** The Cybersecurity and Infrastructure Security Agency (CISA) provides guidance and frameworks for government and private sector organizations. Their publications offer a structured approach to what to measure and how to align those metrics with national standards like the NIST Cybersecurity Framework.
- **Explore Executive Reporting Templates:** SANS provides a variety of policy and plan templates, including those for incident response, which can be adapted for executive reporting. These templates provide a solid foundation

for structuring reports to include key sections like an executive summary, impact analysis, and recommendations in a format that is accessible to leadership.

Practical Application

1. Threat Hunting Practice

This exercise simulates a real-world threat hunt, leveraging a suite of powerful, industry-standard tools to uncover signs of advanced adversary behavior.

Core Activities

- **Tools:**
    - **Elastic Security:** A powerful SIEM and log analysis platform that will serve as the primary tool for querying large-scale log data from across the enterprise.
    - **Velociraptor:** An advanced digital forensics and incident response (DFIR) tool used for deep, real-time inspection and data collection from endpoints.
    - **AlienVault OTX:** A community-driven threat intelligence platform used to enrich the hunt with data on known malicious indicators.
- **Tasks:** The core of the exercise is to follow the threat hunting loop: develop a specific, testable hypothesis about potential adversary activity, query logs and other data sources to investigate that hypothesis, and use additional tools and intelligence to validate any findings.

Enhanced Tasks: Step-by-Step Hunting Scenarios

The following tasks provide a detailed walkthrough of a complete threat hunt, combining different methodologies to increase the likelihood of discovering sophisticated threats.

A. Hypothesis Development: Hunting for Unauthorized Privilege Escalation

One of the most common goals for an attacker after gaining initial access is to escalate their privileges to gain administrative control over a system or the entire domain. This hunt focuses on finding evidence of this critical post-exploitation step.

- **Hypothesis Formulation:** The hunt begins with a clear, specific hypothesis: **"An attacker has compromised a standard user account and has successfully escalated their privileges to a domain administrator role without authorization."** This hypothesis is based on the common attacker TTP of privilege escalation.
- **Querying and Investigation:**
  To test this hypothesis, the hunt will focus on a specific, high-fidelity indicator in Windows environments: **Event ID 4672**, which logs when "Special privileges [are] assigned to a new logon." This event is generated when an account with administrative privileges logs on. While legitimate admins will trigger this event, an unexpected account generating it is a major red flag.

Using Elastic Security, the threat hunter would execute a query against the collected Windows event logs:

```
event.code : 4672
```

This query looks for the specific event ID but filters out known, legitimate administrator accounts to reduce noise and focus on anomalies.

- **Documentation:**
  During the query process, the hunter discovers a suspicious event. This finding is immediately documented to maintain a clear record of the investigation.

Timestamp	User	Event ID	Notes
2025-08-18 15:00:00	testuser	4672	The 'testuser' account, which should have standard user permissions, was unexpectedly assigned an administrative role upon logon. This is a strong indicator of successful privilege escalation.

B. Threat Intelligence Hunt: Tracking Valid Account Misuse

This task pivots from a behavior-based hunt to one initiated by external threat intelligence. It focuses on MITRE ATT&CK T1078 - Valid Accounts, where attackers use stolen credentials to blend in with normal network activity.

- **Leveraging Threat Intelligence:**
  The hunt starts in AlienVault OTX. The hunter searches for threat intelligence "pulses" (reports) related to recent campaigns known to use stolen credentials. From these reports, the hunter extracts a list of Indicators of Compromise (IOCs), such as IP addresses of known malicious command-and-control (C2) servers.
- **Cross-Referencing with Endpoint Data:**
  The IOCs alone are just a starting point. The crucial next step is to see if any of these indicators are present in the live environment. The hunter pivots to Velociraptor to query all endpoints in real-time. A targeted Velociraptor Query Language (VQL) query is used to search for any active processes that are making network connections to the suspicious IP addresses identified from OTX.

The VQL query might look like this:
SELECT Name, Pid, Cmdline, Uid FROM processes() WHERE find(array=network_connections, value=~'suspicious_ip_from_otx')

This query asks Velociraptor to return the process name, ID, and command line for any process on any endpoint that has an active network connection to the malicious IP, providing immediate, actionable evidence of a compromise.

C. Hunting Report: Summarizing and Communicating Findings

The final and most critical step of any hunt is to summarize the findings in a clear, concise report. This report serves as the basis for initiating a formal incident response and provides valuable data for improving detections in the future.

- **Objective:** To write a 100-word report that synthesizes the findings from both the hypothesis-driven and intelligence-driven hunts, mapping the activity to the MITRE ATT&CK framework.
- **Sample Hunting Report:**

Threat Hunt Summary: 2025-10-30

A proactive threat hunt has uncovered evidence of a likely domain compromise. The hunt began by validating a hypothesis for privilege escalation, identifying a standard account ('testuser') that was granted administrative privileges (Event ID 4672).

A parallel intelligence-driven hunt, based on IOCs from OTX, discovered this same user account's workstation communicating with a known malicious C2 server. This activity strongly indicates the misuse of valid accounts, mapping directly to **MITRE ATT&CK T1078**. We assess with high confidence that the 'testuser' account is compromised and being used by an attacker. Immediate incident response is required.

2. SOAR Playbook Development

This exercise simulates the end-to-end process of developing a SOAR playbook, a critical skill for any modern security operations professional looking to enhance efficiency and reduce response times.

Core Activities

- **Tools:**
    - **Splunk Phantom:** A leading SOAR platform that will be used to build, test, and execute the automated playbook. (Note: Splunk Phantom is now known as Splunk SOAR).
    - **TheHive:** An open-source Security Incident Response Platform (SIRP) used for case management. The playbook will integrate with TheHive to automate ticket creation.
    - **Google Docs:** A collaborative tool for documenting the playbook's purpose and design.
- **Tasks:** The primary objective is to design and test a functional playbook that automates the key steps of a phishing incident response, from initial alert to containment and ticketing.

Enhanced Tasks: A Step-by-Step Guide to Playbook Creation

The following tasks provide a detailed walkthrough of the playbook development lifecycle, culminating in a ready-to-deploy automation that can significantly improve SOC efficiency.

A. Playbook Creation: Automating the Phishing Response

- **Objective:** Create a Splunk Phantom playbook designed to automatically handle alerts related to phishing attempts. The playbook will enrich the alert, perform a containment action, and create an incident ticket.
- **Playbook Design and Logic:**
  The playbook is designed as a linear workflow that triggers automatically when an alert tagged as "phishing" is ingested into Splunk Phantom.

Step 1: Ingest and Extract IOCs

- The playbook starts when it receives an alert, for example, from a SIEM like Wazuh, indicating a user has clicked on a link in a known phishing email.
- The first action is to parse the alert data and extract key Indicators of Compromise (IOCs), such as the source IP address of the malicious website.

Step 2: Check IP Reputation (Enrichment)

- The extracted IP address is automatically sent to a threat intelligence service (e.g., VirusTotal, AbuseIPDB) via an API call.

- The playbook waits for the response. A decision point is configured: if the IP is flagged as malicious by the threat intelligence provider, the playbook proceeds to the next step. If not, the alert may be marked as a false positive and closed or flagged for manual review.

Step 3: Block via CrowdSec (Containment)

- If the IP is confirmed to be malicious, the playbook executes a containment action. It makes an API call to CrowdSec, a collaborative intrusion prevention system, with a command to add the malicious IP to its blocklist. This action automatically updates firewall rules to block all traffic to and from that IP.

Step 4: Create TheHive Ticket (Ticketing and Escalation)

- Finally, the playbook creates a formal incident ticket in TheHive. It populates the ticket with all the information gathered so far: the original alert, the threat intelligence findings, and a note confirming that the IP has been blocked. This provides a full audit trail and allows an analyst to perform any necessary post-incident follow-up.

B. Playbook Test: Verifying the Automation

- **Objective:** To simulate a phishing alert in a controlled environment and verify that each step of the Phantom playbook executes correctly and in the intended order.
- **Testing Procedure:**
  1. **Simulate an Alert:** A test alert is manually created or sent from Wazuh to Splunk Phantom. This alert is crafted to mimic a real phishing event and includes a mock malicious IP address (e.g., 192.168.1.102).
  2. **Monitor Playbook Execution:** In the Splunk Phantom interface, the security analyst watches the playbook execute in real-time. They verify that each action (the "block" in the playbook's visual editor) turns green, indicating successful completion.
  3. **Validate Actions:** The analyst checks the external systems to confirm the playbook's actions were successful:
     - They check the CrowdSec command-line interface or dashboard to confirm that the IP 192.168.1.102 is now in the blocklist.
     - They log into TheHive to confirm that a new, correctly populated incident ticket has been created.
  4. **Document Results:** The results of the test are documented to confirm the playbook is ready for production.

- **Playbook Execution Documentation:**

Playbook Step	Status	Notes
Check IP	Success	The IP address was checked against threat intelligence and was flagged as malicious.
Block IP	Success	A successful API call was made to CrowdSec, which has now blocked the IP 192.168.1.102.
Create Ticket	Success	A new case (ID: #12345) was successfully created in TheHive with all relevant alert and enrichment data.

C. Documentation: Explaining the Playbook's Purpose

- **Objective:** To write a brief, 50-word summary in Google Docs that clearly explains the playbook's function, triggers, and key actions. This documentation is essential for training new analysts and for maintaining the playbook over time.
- **Playbook Summary:**

Phishing IP Auto-Block Playbook

This playbook automates the response to high-confidence phishing alerts. When triggered, it extracts the source IP from the alert, validates its reputation using threat intelligence, and, if malicious, automatically blocks the IP using CrowdSec. It concludes by creating a ticket in TheHive for tracking and final review.

3. Post-Incident Analysis

This exercise simulates the critical "lessons learned" phase of the incident response lifecycle, focusing on the techniques and documentation necessary to strengthen an organization's defenses for the future.

Core Activities

- **Tools:**
  - **Google Sheets:** An accessible and collaborative spreadsheet tool, perfect for documenting the structured question-and-answer format of a 5 Whys analysis.
  - **Draw.io:** A free and powerful online diagramming tool, ideal for creating visual aids like Fishbone diagrams to explore and categorize the causes of a complex incident.
- **Tasks:** The primary objectives are to conduct a formal Root Cause Analysis, document the lessons learned from the process, and calculate the core SOC metrics that measure the performance of the incident response team.

Enhanced Tasks: A Step-by-Step Guide to Post-Incident Review

The following tasks provide a detailed walkthrough of a comprehensive post-incident analysis for a mock phishing incident.

A. Root Cause Analysis: Using the 5 Whys Method

- **Objective:** To use the 5 Whys technique to move beyond the surface-level cause of a phishing incident and uncover the fundamental process or technology failure that allowed it to succeed.

- **Procedure:**
  The security team convenes for a blame-free post-mortem. Using a shared Google Sheet, they collaboratively answer a series of "Why?" questions to drill down to the root cause.

- **Documented 5 Whys Analysis:**

Question	**Answer**
Problem Statement:	**A user's workstation was infected with malware from a phishing email.**
1. Why was the user's workstation infected?	Because the user opened a malicious attachment from a phishing email.
2. Why was the email opened and the attachment clicked?	Because the user did not recognize the email as a phishing attempt; it appeared to be a legitimate invoice.
3. Why did the user not receive training on this type of phishing?	Because the security awareness training is only conducted annually and does not cover the latest, most sophisticated phishing TTPs.
4. Why did the malicious email reach the user's inbox?	Because the email security gateway's filters did not flag the attachment as malicious.
5. Why did the email filtering fail?	Because the gateway's rules were too generic and it lacked advanced features like sandboxing to analyze unknown attachments.
Root Cause:	**The incident was caused by a combination of inadequate security awareness training and a weak email filtering technology that was not equipped to handle modern threats.**

B. Fishbone Diagram: Visualizing the Causes

- **Objective:** To create a Fishbone (Ishikawa) diagram in Draw.io to visually map out the various contributing factors that led to the phishing incident.
- **Procedure:**
  1. **Create the Diagram:** In Draw.io, the team creates a central "spine" pointing to the problem statement: "Phishing Incident."
  2. **Define Categories:** They add the main "bones" to the diagram, representing key categories. Common categories for a security incident are **People**, **Process**, **Technology**, and **Environment**.
  3. **Brainstorm Causes:** The team brainstorms the specific causes identified during the 5 Whys analysis and places them on the appropriate branches:
     - **People:** "User clicked link," "Insufficient training frequency."

- **Process:** "Annual-only security training," "No process for reporting suspicious emails."
- **Technology:** "Weak email filtering," "No endpoint sandboxing," "Outdated AV signatures."
- **Environment:** "High volume of daily emails," "Remote work increases phishing risk."

This diagram provides a powerful, one-page visual summary of all the contributing factors, making it easy to present the findings to management.

C. Metrics Calculation: Measuring Performance

- **Objective:** To calculate the Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR) for the mock incident to establish a performance baseline.
- **Procedure:**
  The team reviews the incident timeline to determine the key timestamps:
  - **Time of Compromise:** 10:00 AM (When the user clicked the link).
  - **Time of Detection:** 12:00 PM (When the SOC received an EDR alert).
  - **Time of Resolution:** 4:00 PM (When the infected host was isolated and remediated).

The metrics are then calculated:

- **MTTD (Mean Time to Detect):** Time of Detection - Time of Compromise = 12:00 PM - 10:00 AM = **2 hours**.
- **MTTR (Mean Time to Respond):** Time of Resolution - Time of Detection = 4:00 PM - 12:00 PM = **4 hours**.

- **Summary of Metrics (in 50 words):**

For this incident, the MTTD was 2 hours, and the MTTR was 4 hours. The 2-hour detection time highlights a need for faster alerting on malicious email activity. The 4-hour response time is within our target, but automation could further reduce it. These metrics will serve as a baseline for measuring future improvements.

4. Alert Triage with Automation

This exercise simulates the process of enhancing a standard alert triage workflow with automated threat intelligence validation. The goal is to reduce the manual burden on analysts and accelerate the incident response lifecycle.

Core Activities

- **Tools:**
  - **Wazuh:** An open-source SIEM and XDR platform that will serve as the source of the initial security alert.
  - **VirusTotal:** A comprehensive threat intelligence service that will be used to validate the reputation of file hashes.
  - **TheHive:** An open-source Security Incident Response Platform (SIRP) that, along with its analysis engine, Cortex, will be used to automate the validation process.

- **Tasks:** The primary objective is to take a standard triage workflow and supercharge it with automation, ensuring that every relevant alert is automatically enriched with external threat intelligence before it is even seen by a human analyst.

Enhanced Tasks: A Step-by-Step Guide to Automated Triage

The following tasks provide a detailed walkthrough of how to build and leverage an automated validation workflow for a common security alert.

A. Triage Simulation: Analyzing a Suspicious File Download

- **Objective:** To perform the initial manual triage of a mock alert for a "Suspicious File Download" within the Wazuh dashboard and document the key details.
- **Procedure:**
  1. **Alert Generation:** Wazuh's File Integrity Monitoring (FIM) or a custom rule detects that a new executable file has been downloaded into a temporary directory on a user's workstation. This generates a high-priority alert.
  2. **Initial Review:** The SOC analyst sees the new alert in the Wazuh dashboard. They review the basic information to understand the context: which user, on which machine, downloaded what file, and from what source IP.
  3. **Documentation:** The analyst begins the incident documentation process, logging the initial alert details. This first step is crucial for establishing a timeline and an audit trail.

- **Documented Alert:**

Alert ID	Description	Source IP	Priority	Status
005	File Download	192.168.1.102	High	Open

B. Automated Validation: Integrating TheHive and VirusTotal

- **Objective:** To configure TheHive and its analysis engine, Cortex, to automatically check the file hash of any observable added to a case against the VirusTotal database.
- **Procedure:**
  1. **Configure Cortex:** Cortex is a powerful analysis engine that integrates with TheHive. The first step is to configure a Cortex "analyzer" for VirusTotal. This involves entering the VirusTotal API key into Cortex, enabling the VirusTotal_GetReport_3_1 analyzer.
  2. **Automate the Workflow:**
     - An alert from Wazuh is automatically forwarded to TheHive, creating a new case.
     - The SHA256 hash of the downloaded file is automatically extracted from the Wazuh alert and added as an "observable" to the case in TheHive.
     - TheHive's integration with Cortex is configured to automatically run the VirusTotal analyzer on any new observable of the hash data type.
  3. **View Enriched Data:** The analyst opens the case in TheHive. The "observables" tab now shows the file hash, and next to it is a "report" generated by Cortex. With a single click, the analyst can see the full VirusTotal report, including how many antivirus engines flagged the file as malicious.

- **Summary of Results (in 50 words):**

By integrating TheHive with a VirusTotal analyzer, we've automated a critical triage step. Now, when a file hash is added to a case, its malware status is checked instantly. This eliminates manual lookups, reduces human error, and provides analysts with immediate, actionable intelligence directly within their case management platform.

5. Evidence Analysis

This exercise simulates the core tasks of a digital forensics analyst: examining collected artifacts for signs of malicious activity and documenting the process in a forensically sound manner.

Core Activities

- **Tools:**
  - **Velociraptor:** An advanced endpoint incident response tool used for live analysis, allowing investigators to query and analyze the state of a running system without altering it.
  - **FTK Imager:** A standard forensic tool used for creating and analyzing disk images. While Velociraptor is used for live analysis in this exercise, FTK Imager would be used to analyze a static, non-volatile disk image collected from a system.
- **Tasks:** The primary objectives are to analyze the collected evidence to identify threat actor activity and to maintain a strict, unbroken chain of custody for every piece of evidence handled.

Enhanced Tasks: A Step-by-Step Guide to Evidence Analysis and Documentation

The following tasks provide a detailed walkthrough of how to analyze a critical piece of volatile data—live network connections—and how to properly document the evidence to ensure its integrity.

A. Evidence Analysis: Investigating Live Network Connections

- **Objective:** To use Velociraptor to perform a live analysis of the network connections on a potentially compromised Windows Virtual Machine (VM) and identify any suspicious or unauthorized connections.
- **Step-by-Step Procedure:**
  1. **Initiate the Collection:** From the Velociraptor server interface, the analyst targets the specific Windows VM under investigation ("Server-Z"). A new "hunt" is created to collect data from this endpoint.
  2. **Execute the VQL Query:** The analyst uses the Velociraptor Query Language (VQL) to retrieve the list of active network connections. This is a crucial first step in identifying active command-and-control (C2) channels or data exfiltration.

     The query used is:

     ```
     SELECT * FROM netstat()
     ```

3. **Analyze the Output:** The query returns a detailed table of all active network connections, including the local and remote IP addresses, ports, the state of the connection (e.g., ESTABLISHED, LISTENING), and the process ID (PID) responsible for the connection. The analyst systematically examines this output, looking for anomalies such as:
   - Connections to IP addresses with a known poor reputation (cross-referencing with threat intelligence feeds).
   - Connections on non-standard ports for common services (e.g., an HTTP connection over port 8080 instead of 80).
   - Connections from unexpected processes (e.g., svchost.exe or powershell.exe making persistent connections to external IPs).
4. **Identify Suspicious Connections:** During the analysis, the investigator identifies a connection from a PowerShell process (PID 4172) to an external IP address 198.51.100.55 on port 443. While the port is standard for HTTPS, the fact that PowerShell is maintaining this connection is highly suspicious and warrants immediate further investigation as a potential C2 channel. The collected data is then exported as a CSV file for documentation.

B. Chain-of-Custody: Documenting Evidence Collection

- **Objective:** To create and maintain a formal chain-of-custody log for the collected evidence (the network connection log) to ensure its integrity is preserved.
- **Procedure:**
  1. **Evidence Hashing:** Immediately after exporting the network connection log from Velociraptor (e.g., Server-Z_netstat.csv), the analyst calculates a cryptographic hash of the file using a standard algorithm like SHA256. This hash serves as a unique digital fingerprint. Any modification to the file, no matter how small, will produce a different hash, making tampering immediately obvious.
  2. **Logbook Entry:** The analyst records every detail of the collection in the official chain-of-custody log. This document tracks who handled the evidence, what it is, when it was collected, and its verification hash. This meticulous record-keeping is essential for legal and formal proceedings.

- **Chain-of-Custody Log:**

Item	**Description**	**Collected By**	**Date**	**Hash Value (SHA256)**
Network Log	A CSV export of active network connections from Server-Z, collected via Velociraptor.	SOC Analyst	2025-10-30	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

6. Adversary Emulation Practice

6. Adversary Emulation Practice: From Simulation to Validation
This exercise simulates a controlled cyberattack to provide a data-driven assessment of the SOC's ability to defend against a common threat vector.

Core Activities

- **Tools:**
  - **MITRE Caldera:** An open-source adversary emulation platform that will be used to automate the execution of attacker TTPs in a safe and controlled manner.
  - **Wazuh:** The SIEM/XDR platform that serves as the primary defensive tool, responsible for monitoring the environment and detecting the simulated attack.
- **Tasks:** The core objective is to simulate a series of adversary TTPs using Caldera and then to rigorously test and measure the SOC's ability to detect these actions using Wazuh.

Enhanced Tasks: A Step-by-Step Guide to Emulation and Analysis
The following tasks provide a detailed walkthrough of how to plan and execute an emulation, document the results, and generate actionable insights for security improvement.

A. Emulation Simulation: Testing Defenses Against Phishing

- **Objective:** To use MITRE Caldera to simulate a spearphishing attack, a technique categorized as T1566 in the MITRE ATT&CK framework. The parallel objective is to ensure that Wazuh is configured to detect this simulated attack and to document the outcome.
- **Step-by-Step Procedure:**
  1. **Configure Caldera:** Within the Caldera framework, the security engineer (acting as the "Red Team") creates a new operation. They build an "adversary" profile that includes the specific ability for T1566 - "Spearphishing Attachment" or "Spearphishing Link." This involves instructing a Caldera agent, deployed on a target workstation, to simulate a user receiving and opening a malicious file or link.
  2. **Configure Wazuh for Detection:** The security analyst (acting as the "Blue Team") ensures that Wazuh is properly configured to detect the downstream effects of a phishing attack. This includes:
     - **File Integrity Monitoring (FIM):** To detect the creation of suspicious files dropped by the phishing payload.

        - **Command Monitoring:** To alert on suspicious commands, such as PowerShell execution with encoded payloads, that are often launched after a user clicks a malicious link.
        - **Integration with Email Security Gateways:** To ingest logs that can show the delivery of the malicious email.
    3. **Execute the Emulation:** The Caldera operation is launched. The Caldera agent on the target machine executes the steps defined in the T1566 ability.
    4. **Monitor for Detection:** The Blue Team monitors the Wazuh dashboard in real-time to see if alerts are generated in response to Caldera's actions.
    5. **Document the Results:** Every step of the emulation is meticulously documented, noting which TTP was executed and whether it was successfully detected, blocked, or missed entirely.

- **Emulation Documentation:**

Timestamp	TTP	Detection Status	Notes
2025-08-18 17:00:00	T1566	Detected	The initial delivery of the simulated phishing email was successfully blocked and alerted on by the integrated email security gateway. The payload did not reach the endpoint.

B. Emulation Report: Summarizing Results and Identifying Gaps

- **Objective:** To summarize the results of the emulation exercise in a concise, 100-word report. This report's primary purpose is to communicate the findings to stakeholders and, most importantly, to highlight any identified detection gaps that require attention.
- **Procedure:**
  The analyst synthesizes the documented results into a high-level summary. The report should celebrate successes but be transparent about weaknesses to drive improvement.

- **Emulation Report:**

Adversary Emulation Report: Q3 Phishing Test
An adversary emulation was conducted to test our defenses against spearphishing (T1566). The simulation, performed using MITRE Caldera, showed that our email security gateway successfully detected and blocked the initial malicious email, which is a significant success for our perimeter defenses.
However, a secondary phase of the test, simulating a scenario where the email bypassed the filter, revealed a detection gap. Our endpoint monitoring rules failed to alert on the specific type of PowerShell command execution used by the payload. This highlights a critical area for improvement in our endpoint detection and response (EDR) rule set.

7. Security Metrics and Executive Reporting

This exercise simulates the process of transforming incident data into strategic assets. It focuses on the creation of visual dashboards for real-time performance tracking and the crafting of concise reports that inform executive decision-making.

Core Activities

- **Tools:**
  - **Elastic Security:** The SIEM and security analytics platform used to collect incident data and build visual metrics dashboards.
  - **Google Sheets:** A spreadsheet tool used for ad-hoc analysis and calculation of specific incident metrics.
  - **Google Docs:** A collaborative word processor used for drafting and finalizing formal executive reports.
- **Tasks:** The primary objectives are to calculate advanced SOC performance metrics, visualize them in a dashboard, and create a clear, concise executive report that summarizes the findings and provides strategic recommendations.

Enhanced Tasks: A Step-by-Step Guide to Data-Driven Reporting
The following tasks provide a detailed walkthrough of how to create a metrics dashboard, analyze a critical metric like dwell time, and communicate the results to a non-technical audience.

A. Metrics Dashboard: Visualizing SOC Performance

- **Objective:** To create a dedicated dashboard in Elastic Security to provide an at-a-glance view of the SOC's key performance indicators: Mean Time to Detect (MTTD), Mean Time to Respond (MTTR), and the false positive rate.

- **Step-by-Step Procedure:**
  1. **Data Preparation:** Ensure that incident data within Elastic Security contains the necessary timestamp fields (e.g., compromise_time, detection_time, resolution_time) and a field to mark alerts as false positives (e.g., status: "false_positive").
  2. **Create Visualizations:** In Kibana (the visualization component of Elastic), create a new visualization for each metric:
     - **MTTD/MTTR:** Use the "Metric" visualization type. To calculate the average time, use a scripted field or a runtime field that calculates the duration between the relevant timestamps (e.g., doc['detection_time'].value - doc['compromise_time'].value). Set the aggregation to

"Average." For the example, this would yield MTTD = 2 hours and MTTR = 4 hours.
    - **False Positive Rate:** Use the "Pie Chart" or "Metric" visualization. Create a filter for alerts within a specific time frame (e.g., last 30 days). For a pie chart, slice by the status field. For a metric, use a filter ratio to calculate the percentage of alerts where status is "false_positive" versus all other statuses.
3. **Assemble the Dashboard:** Create a new dashboard and add the newly created visualizations. Arrange them in a logical layout so that the most critical metrics are immediately visible. Save the dashboard as "SOC Performance Overview."

B. Executive Report: Communicating Findings to Leadership

- **Objective:** To draft a 150-word executive summary in Google Docs that presents the key metrics and provides clear, actionable recommendations for improving the SOC's performance.
- **Procedure:**
  The report is written for a non-technical audience, focusing on business impact and strategic goals.

- **Executive Summary Draft:**

Subject: Q3 Security Operations Performance and Recommendations

This report summarizes the performance of our Security Operations Center (SOC) for the third quarter. Our team's average time to respond to a detected threat (MTTR) is a strong **4 hours**, demonstrating our ability to contain incidents efficiently. However, our average time to detect a threat (MTTD) stands at **2 hours**, which presents an opportunity for improvement.

The key factor influencing our detection time is a high volume of alerts, with a current false positive rate of 15%.

Recommendations:

To reduce detection time and improve focus, we recommend investing in advanced alert correlation and SOAR (Security Orchestration, Automation, and Response) technology. This will automate the triage of low-level alerts, allowing our analysts to proactively hunt for threats and significantly decrease our overall risk exposure. We project this could reduce our MTTD by up to 50%.

C. Metrics Analysis: Understanding Dwell Time

- **Objective:** To analyze the dwell time for a specific mock incident using Google Sheets and summarize the security implications of the finding. Dwell time is the period an attacker goes undetected in a network (Time of Detection - Time of Initial Compromise).

- **Procedure:**
  In a Google Sheet, the analyst creates a simple table to calculate the dwell time for a recent incident.

Metric	**Timestamp**
Time of Initial Compromise	2025-10-30 09:00:00
Time of Detection	2025-10-30 11:00:00
Dwell Time (Hours)	**50**

- **Summary of Findings (in 50 words):**

The analysis of the "Server-Z" incident revealed a dwell time of 50 hours. This extended period provided the adversary with a significant window to conduct reconnaissance and escalate privileges. This highlights a critical need to enhance our threat hunting capabilities to find adversaries faster and reduce our overall risk.

8. Capstone Project

This project simulates a complete incident lifecycle, providing an invaluable opportunity to practice and refine the entire spectrum of incident response capabilities in a controlled environment.

Core Activities

- **Tools:** A full suite of security tools will be utilized, including:
  - **Offensive:** Metasploit, MITRE Caldera
  - **Defensive:** Wazuh, CrowdSec, Elastic Security
  - **Response & Management:** TheHive, Google Docs
- **Tasks:** The primary objective is to simulate a complex, multi-stage incident and execute every phase of the response plan: detect, triage, respond, analyze, emulate, and report.

Enhanced Tasks: The End-to-End Incident Response Workflow

The following tasks provide a detailed, step-by-step execution of the capstone project.

1. Attack Simulation & Adversary Emulation: The Threat

- **Objective:** To simulate a realistic initial compromise using both a direct exploit and a TTP-based emulation to test defenses against different attack methodologies.
- **Attack Simulation Procedure (Metasploit):**
  1. **Target:** A Metasploitable2 VM, an intentionally vulnerable Linux machine.
  2. **Tool:** The Metasploit Framework.
  3. **Exploit:** The Samba usermap_script vulnerability (CVE-2007-2447), a classic remote code execution flaw.
  4. **Execution:** Following the **Metasploit Unleashed** guide, the attacker uses the exploit/multi/samba/usermap_script module, sets the target IP, configures a reverse shell payload, and executes the exploit to gain root-level access to the target.
- **Adversary Emulation Procedure (MITRE Caldera):**
  1. **Objective:** To test the detection of the *technique* itself, not just a specific exploit signature.
  2. **Tool:** MITRE Caldera.
  3. **TTP: T1210 - Exploitation of Remote Services**.
  4. **Execution:** An adversary profile is created in Caldera that includes an "ability" representing T1210. An operation is launched targeting a machine with an open, vulnerable service. Caldera's agent executes the technique, and the results are monitored.

5. **Detection Documentation:** The Blue Team monitors Wazuh for alerts related to the emulation.

Timestamp	Source IP	Alert Description	MITRE Technique
2025-08-18 16:00:00	192.168.1.102	Samba exploit	T1210

2. Detection, Triage, Response, and Containment: The Defense

- **Objective:** To detect the attack, perform initial triage, and execute immediate containment actions to stop the threat.
- **Procedure:**
  1. **Detection:** Wazuh, configured with rules to monitor Samba logs and detect anomalous shell creation, generates a high-priority alert for the exploit.
  2. **Triage:** The alert is automatically forwarded to TheHive. A Tier 1 analyst reviews the alert, confirms it is a true positive based on the source IP and the nature of the alert, and declares a formal incident.
  3. **Containment:** The analyst immediately takes two actions:
     - The compromised Metasploitable2 VM is moved to an isolated network VLAN via the hypervisor.
     - The attacker's IP (192.168.1.102) is added to the CrowdSec blocklist, which automatically propagates a firewall rule to block the IP at the network edge.
  4. **Verification:** The analyst verifies the block by attempting to ping from the attacker's machine to another host on the network; the test fails.

3. SOAR Automation: The Efficiency Multiplier

- **Objective:** To use a playbook to automate the initial triage and response actions, significantly reducing the response time.
- **Procedure:**
  1. **Playbook Creation:** A playbook is created in a SOAR tool (e.g., TheHive's Cortex integration or Splunk SOAR) with the following logic:
     - **Trigger:** On ingestion of a Wazuh alert with "Samba Exploit" in the description.
     - **Action 1:** Automatically create a new high-severity case in TheHive.
     - **Action 2:** Extract the source IP address from the alert.
     - **Action 3:** Make an API call to CrowdSec to ban the extracted IP address.
  2. **Verification:** During the exercise, the analyst verifies that the playbook executed successfully by confirming that TheHive case was created and the IP was blocked *before* any manual intervention was required.

4. Post-Incident Analysis & Metrics Reporting: The Learning Loop

- **Objective:** To conduct a thorough post-mortem to understand the root cause of the incident and to calculate key performance metrics.
- **Post-Incident Analysis:**
  - A **5 Whys** analysis is conducted, revealing that the root cause was not just the exploit, but a failure in the patch management process that left the vulnerable Samba service exposed.
  - A **Fishbone Diagram** is created in **Draw.io** to visually map the causes, including "Process" (no vulnerability scanning) and "Technology" (outdated Samba version).
- **Metrics Reporting:**
  - The incident timeline is used to calculate key metrics: **MTTD, MTTR, and Dwell Time**.
  - A dashboard is created in **Elastic Security** to visualize these metrics, providing a clear view of SOC performance during the incident.

5. Reporting & Stakeholder Briefing: The Communication

- **Objective:** To communicate the findings of the incident to both technical and non-technical leadership.
- **Comprehensive Incident Report (300 Words):**
  A formal report is drafted in **Google Docs** using a **SANS template**.

Executive Summary: On October 30, 2025, the SOC detected and contained a critical security incident involving the compromise of a development server (Metasploitable2) via an external exploit. The incident was successfully contained with no impact on production systems or data. The root cause was identified as a gap in our patch management process. Recommendations are outlined below to prevent recurrence.

Incident Timeline:

- **16:00:** Attacker exploits Samba vulnerability (T1210).
- **16:02:** Wazuh detects the exploit and alerts TheHive.
- **16:03:** SOAR playbook automatically blocks the attacker's IP.
- **16:10:** Analyst isolates the VM.
- **18:00:** Incident fully contained and remediation begins.

Root Cause Analysis (RCA): The primary cause was a failure to patch a known critical vulnerability (CVE-2007-2447) on a non-production server. This was compounded by a lack of routine vulnerability scanning on development assets, leaving the system exposed.

Recommendations:

6. Immediately decommission or patch the vulnerable server.
7. Expand the scope of our vulnerability management program to include all development and testing environments.

8. Implement a quarterly adversary emulation exercise to continuously validate our detection and response capabilities against common TTPs.

- **Stakeholder Briefing (150 words):**
  A concise summary is drafted for a non-technical executive.

Subject: Summary
This briefing is to inform you of a security incident that our team successfully managed yesterday. We detected and immediately blocked an external attack on a non-production server used for development.
Our automated defense systems worked as designed, stopping the threat in under three minutes and preventing any access to our core network or sensitive data. There was zero impact on our customers or business operations.
The investigation revealed the issue stemmed from an out-of-date server, highlighting a gap in our internal processes. We have already taken steps to fix the immediate issue and are implementing a broader plan to ensure all our systems, including those in development, are consistently scanned and updated. This event has provided a valuable opportunity to further strengthen our security posture, and we are confident in the improvements being made.