README: Task 1 - Threat Hunting Practice

1. Objective

This exercise is designed to provide hands-on practice with the entire threat hunting lifecycle. The primary goal is to develop and test a security hypothesis using a combination of SIEM queries, endpoint forensic analysis, and external threat intelligence. By the end of this task, you will be able to proactively search for signs of advanced adversary behavior that may have evaded automated security alerts.

2. Core Concepts

- **Hypothesis-Driven Hunting:** The practice of starting a hunt with a specific, testable theory about adversary activity (e.g., "An attacker is using PowerShell for lateral movement") rather than aimlessly searching through logs.
- **Threat Intelligence Integration:** Using external data (e.g., IOCs from AlienVault OTX) to inform and validate hunting activities.
- **Endpoint Triage:** Using tools like Velociraptor to perform deep, real-time analysis on endpoints to confirm or deny suspicions raised by log analysis.

3. Tools Required

- **Elastic Security:** As the central SIEM for log aggregation and querying.
- **Velociraptor:** For live endpoint analysis and forensic data collection.
- **AlienVault OTX Account:** To access threat intelligence feeds and IOCs.

4. Setup and Prerequisites

1. **Data Ingestion:** Ensure that Windows event logs, particularly those related to security (Event IDs 4624, 4625, 4672, etc.), are being ingested into Elastic Security from your test endpoints.
2. **Velociraptor Agent:** The Velociraptor agent must be deployed and running on the target endpoints you intend to investigate.
3. **OTX Account:** Have your AlienVault OTX credentials ready for accessing threat intelligence.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Formulate the Hypothesis:**
    - **Hypothesis:** "An attacker has successfully compromised a standard user account and has escalated their privileges to a domain administrator, bypassing our standard security controls."

- **Rationale:** Privilege escalation is a critical step for an attacker to gain control over a network. Windows Event ID 4672 ("Special privileges assigned to new logon") is a high-fidelity indicator of an administrative logon. By hunting for this event from unexpected user accounts, we can find evidence of this TTP.

2. **Query Elastic Security:**
    - Navigate to the **Discover** tab in Kibana (within Elastic Security).
    - Enter the following query to search for Event ID 4672, while filtering out known legitimate administrator accounts to reduce noise:

codeKql

event.code : 4672 and not user.name : ("*legit_admin1*" or "*svc_admin*")

    - Set the time range to the period of interest (e.g., last 24 hours).
3. **Analyze and Document Findings:**
    - The query returns a result showing that a user named testuser, known to be a standard user, triggered Event ID 4672. This is a significant anomaly.
    - Document this finding immediately in your investigation notes or a dedicated log.

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.

1. **Gather Intelligence:**
    - Log in to **AlienVault OTX**. Search for recent pulses or IOCs related to threat actors known to use stolen credentials (T1078).
    - Identify a suspicious IP address (e.g., 198.51.100.123) known to be a C2 server from a recent campaign.
2. **Cross-Reference with Velociraptor:**
    - Pivot to the **Velociraptor** UI.
    - Create a new hunt targeting all relevant endpoints.
    - Use the following Velociraptor Query Language (VQL) query to search for any live process that is communicating with the suspicious IP address:

codeSQL

SELECT Name, Pid, Cmdline, Connections

FROM processes()

WHERE Connections.DestIP = '198.51.100.123'

    - **Rationale:** This query interrogates the live state of every endpoint, which is something log analysis might miss. It directly links a known malicious IP to a specific running process on a specific host.

3. **Validate Findings:**
    - The hunt results show that on the same machine used by testuser, a PowerShell process (powershell.exe) has an active, established connection to 198.51.100.123. This strongly corroborates the initial hypothesis and indicates a likely compromise.

1. **Summarize the Findings:**
    - Synthesize the evidence from both the hypothesis-driven hunt and the intelligence-driven hunt into a single, concise report. The report must be clear, actionable, and mapped to the relevant MITRE ATT&CK framework.

2. **Draft the Report:**

Threat Hunt Summary: 2025-08-18

A proactive threat hunt has uncovered strong evidence of a domain compromise involving unauthorized privilege escalation and the misuse of valid credentials. The hunt began by validating a hypothesis for privilege escalation, identifying a standard account ('testuser') that was granted administrative privileges (Windows Event ID 4672).

A parallel intelligence-driven hunt, based on IOCs from OTX, discovered this same user's workstation actively communicating with a known malicious C2 server (198.51.100.123) via a PowerShell process. This activity directly maps to **MITRE ATT&CK T1078 (Valid Accounts)**. We assess with high confidence that the 'testuser' account is compromised. Immediate incident response is required.

---

6. Expected Outcomes and Deliverables

- A documented finding of an anomalous administrative logon.
- Evidence of a live endpoint communicating with a known malicious IP address.
- A final, 100-word threat hunting report summarizing the findings and recommending next steps (incident response).

README: Task 2 - SOAR Playbook Development

1. Objective

The objective of this task is to design, build, test, and document a Security Orchestration, Automation, and Response (SOAR) playbook. This exercise will provide hands-on experience in automating a repetitive and time-consuming incident response workflow—specifically, the handling of phishing alerts. The goal is to improve response time, reduce manual analyst workload, and ensure a consistent response to every alert.

2. Core Concepts

- **Security Orchestration:** The integration and coordination of different security tools via APIs to create a unified workflow.
- **Automation:** The machine-based execution of predefined tasks (e.g., checking an IP's reputation, blocking an IP).
- **Playbooks:** The digital codification of an incident response process, defining the triggers, logic, and actions for an automated workflow.

3. Tools Required

- **Splunk Phantom (or Splunk SOAR):** The SOAR platform used to build the playbook.
- **TheHive:** The Security Incident Response Platform (SIRP) for automated ticket creation.
- **CrowdSec (or a similar IPS/firewall with API access):** The tool used for automated IP blocking.
- **Wazuh (or any SIEM):** The source of the initial phishing alert.
- **Google Docs:** For playbook documentation.

4. Setup and Prerequisites

1. **API Integration:** Ensure that Splunk Phantom has been configured with the necessary "apps" and API credentials to communicate with TheHive and CrowdSec.
2. **Alert Ingestion:** Configure Phantom to ingest alerts from your SIEM (Wazuh). This often involves setting up a dedicated data source or using an API/webhook.
3. **Threat Intelligence:** Configure a threat intelligence app in Phantom (e.g., VirusTotal, AbuseIPDB) with a valid API key.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Design the Workflow:**
    - Before building, map out the logical flow of the playbook on a whiteboard or in a diagram.

- **Trigger:** The playbook will start when an alert with the tag "phishing" is ingested.
- **Step 1: Enrichment.** Check the reputation of the source IP address from the alert.
- **Step 2: Decision.** If the IP is malicious, proceed to containment. If not, flag for manual review.
- **Step 3: Containment.** Block the malicious IP address using CrowdSec.
- **Step 4: Ticketing.** Create a new incident case in TheHive with all collected information.

2. **Build the Playbook in Splunk Phantom:**
    - In the Phantom UI, create a new playbook.
    - **Start Block:** Configure the on_start block to trigger on alerts with label=phishing.
    - **Action Block 1 (Enrichment):** Add an "IP Reputation" action. Select your threat intelligence app (e.g., virustotal). Configure it to take the sourceAddress artifact from the initial alert as input.
    - **Decision Block:** Add a "Decision" block. Configure it to check the output of the IP reputation action. The condition should be something like action_result.summary.malicious == True.
    - **Action Block 2 (Containment):** On the "True" path from the decision block, add a "Block IP" action. Select your CrowdSec app. Configure it to take the same sourceAddress artifact as input.
    - **Action Block 3 (Ticketing):** After the "Block IP" action, add a "Create Case" action. Select your TheHive app. Map the artifacts from the alert (source IP, description) and the enrichment data (the threat intel report) into the fields for the new TheHive case.
    - Save and activate the playbook.

1. **Simulate a Phishing Alert:**
    - Generate a test alert in Wazuh that mimics a phishing event. Ensure it contains a source IP address (e.g., 192.168.1.102) that you know will be flagged as malicious by your threat intelligence provider (you may need to add it manually to a local list for the test).
    - Ensure this alert is correctly forwarded to and ingested by Splunk Phantom.

2. **Verify Execution:**

- In Phantom, navigate to the "Running Playbooks" or "Mission Control" view. You should see your new playbook has been triggered by the test alert.
- Monitor the playbook's progress. Each block should turn green as it executes successfully.
- **Crucially, verify the external actions:**
  - Check CrowdSec to confirm that 192.168.1.102 has been added to the blocklist.
  - Log in to TheHive to confirm that a new case has been created and populated with the correct information from the alert.

3. **Document the Test Results:**

Playbook Step	Status	Notes
Check IP	Success	The IP 192.168.1.102 was successfully checked and flagged as malicious by the threat intelligence provider.
Block IP	Success	A successful API call was made to CrowdSec. Confirmed that CrowdSec 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.

1. **Write the Playbook Summary:**
   - Create a new document in Google Docs. The summary should be brief but comprehensive, allowing any analyst to understand the playbook's purpose at a glance.
2. **Draft the Summary:**

Playbook Summary: Automated Phishing IP Containment

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

---

6. Expected Outcomes and Deliverables

- A functional and activated Splunk Phantom playbook.
- Documented proof of a successful end-to-end test run.

- A 50-word summary document explaining the playbook's function.

README: Task 3 - Post-Incident Analysis

1. Objective

The objective of this exercise is to master the post-incident analysis process, which is a critical phase of the incident response lifecycle. This task focuses on moving beyond simple incident closure to a state of continuous improvement. You will learn to conduct a thorough Root Cause Analysis (RCA), visualize contributing factors, and calculate key performance metrics to identify and address systemic weaknesses in security.

2. Core Concepts

- **Root Cause Analysis (RCA):** A structured method to identify the fundamental cause of an incident, not just the immediate symptoms.
- **5 Whys:** A simple but powerful RCA technique for drilling down to the root cause by repeatedly asking "Why?".
- **Fishbone (Ishikawa) Diagram:** A visual tool for brainstorming and categorizing the potential causes of a problem.
- **SOC Metrics (MTTD/MTTR):** Key Performance Indicators used to measure the effectiveness and efficiency of security operations.

3. Tools Required

- **Google Sheets:** For documenting the 5 Whys analysis.
- **Draw.io:** For creating the Fishbone diagram.

4. Setup and Prerequisites

- A mock incident scenario to analyze. For this exercise, we will use a common scenario: "A user's workstation was compromised by malware delivered via a phishing email."
- Defined timestamps for the mock incident to allow for metrics calculation.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Define the Problem:**
    - Start with a clear, concise problem statement.
    - **Problem:** "A user's workstation was compromised after they clicked a link in a phishing email."
2. **Conduct the 5 Whys Analysis:**

- In a **Google Sheet**, create two columns: "Question" and "Answer."
- Begin with the problem and ask the first "Why." Record the answer. Use that answer as the basis for the next "Why." Continue this process until you arrive at a root cause—typically a process or technology failure, not a person.

3. **Document the Results:**

Question	**Answer**
Problem Statement:	**A user's workstation was compromised by malware from a phishing email.**
1. Why was the workstation compromised?	Because the user clicked a malicious link in a phishing email that bypassed endpoint protection.
2. Why was the link clicked?	Because the user did not recognize the email as a phishing attempt.
3. Why did the user not recognize the threat?	Because the security awareness training is generic, infrequent (annual), and does not cover the latest social engineering tactics.
4. Why did the malicious email reach the user's inbox?	Because the email security gateway's filters failed to identify the URL as malicious.
5. Why did the email filtering fail?	**(Root Cause)** Because the gateway relies on a simple blocklist and lacks advanced features like URL sandboxing or dynamic analysis to detect zero-day phishing sites.

1. **Create the Diagram in Draw.io:**
   - Go to app.diagrams.net (Draw.io) and create a new diagram.
   - Draw the central "spine" and write the problem statement at the "head" of the fish: "Phishing Incident Compromise."
   - Add the main category "bones" branching off the spine: **People**, **Process**, **Technology**.
2. **Map the Causes:**
   - Brainstorm the contributing factors identified during the 5 Whys and place them on the appropriate branches.

- **People:** "User clicked link," "Lack of specific training."
- **Process:** "Annual-only security training," "No clear process for reporting suspicious emails."
- **Technology:** "Weak email filtering," "No URL sandboxing," "Endpoint protection bypassed."
- This visual aid makes it easy to see that the incident was not a single failure but a combination of factors across different domains. Save the diagram as an image (PNG or SVG).

1. **Establish the Timeline:**
   - For this mock incident, we will use the following timestamps:
     - **Time of Compromise:** 2025-08-18 10:00 AM (When the user clicked the link).
     - **Time of Detection:** 2025-08-18 12:00 PM (When the SOC received an EDR alert about suspicious PowerShell activity).
     - **Time of Resolution:** 2025-08-18 4:00 PM (When the infected host was isolated, cleaned, and brought back online).
2. **Calculate MTTD and MTTR:**
   - **Mean Time to Detect (MTTD):** The time from compromise to detection.
     - 12:00 PM - 10:00 AM = 2 hours
   - **Mean Time to Respond (MTTR):** The time from detection to resolution.
     - 4:00 PM - 12:00 PM = 4 hours
3. **Summarize the Findings:**
   - Draft a brief summary that explains what the metrics mean in a business context.

Incident Metrics Summary

For the recent phishing incident, our Mean Time to Detect (MTTD) was **2 hours**, and our Mean Time to Respond (MTTR) was **4 hours**. The 2-hour detection delay indicates a gap in our proactive alerting capabilities, as we relied on post-exploitation alerts. While the 4-hour response is within our SLA, it can be improved with automation.

---

6. Expected Outcomes and Deliverables

- A completed 5 Whys analysis in a Google Sheet.
- A saved Fishbone diagram from Draw.io.

- A 50-word summary of the calculated MTTD and MTTR for the mock incident.

README: Task 4 - Alert Triage with Automation

1. Objective

The primary objective of this task is to streamline the initial alert triage process by integrating automated validation. This exercise will demonstrate how to connect a Security Incident Response Platform (SIRP) with an external threat intelligence source to automatically enrich alerts, reducing the manual workload on Tier 1 analysts and accelerating the decision-making process. The goal is to move from a manual lookup process to an automated, "intelligence-at-your-fingertips" workflow.

2. Core Concepts

- **Alert Triage:** The initial process of reviewing, prioritizing, and validating security alerts to determine if they represent a genuine threat requiring further investigation.
- **Automated Enrichment:** The process of using automation to append contextual data (e.g., threat intelligence, user information) to a security alert.
- **SIRP and Analysis Engines:** The use of a platform like TheHive in conjunction with an analysis engine like Cortex to orchestrate enrichment tasks.

3. Tools Required

- **Wazuh:** As the SIEM/XDR source for the security alert.
- **TheHive:** The SIRP used for case management.
- **Cortex:** The analysis engine that connects TheHive to external tools.
- **VirusTotal:** The external threat intelligence service for validating file hashes.

4. Setup and Prerequisites

1. **TheHive/Cortex Integration:** Ensure TheHive and Cortex are installed and can communicate with each other. TheHive needs to be configured with the Cortex API key.
2. **VirusTotal Analyzer:** In Cortex, add and enable the VirusTotal analyzer. This requires a valid VirusTotal API key.
3. **Alert Forwarding:** Configure Wazuh to automatically forward relevant alerts (e.g., via a webhook or script) to TheHive's API, creating a new case for each alert.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Generate a Mock Alert:**

- Simulate an event that would be detected by Wazuh's File Integrity Monitoring (FIM). For example, create a file named invoice.exe in the C:\Users\TestUser\Downloads directory on a monitored endpoint.
- This action should trigger a "Suspicious File Download" or "New Executable Created" alert in Wazuh.

2. **Review the Alert in Wazuh:**
    - Navigate to the Wazuh dashboard and locate the new alert.
    - Review the initial details: the hostname, the user, the full path to the file (C:\Users\TestUser\Downloads\invoice.exe), and the file's SHA256 hash (Wazuh automatically calculates this).
3. **Document the Initial Triage:**
    - In your incident log or ticketing system, create an initial entry for the alert. This step simulates the manual process before automation is fully trusted.

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

1. **Configure TheHive for Auto-Analysis:**
    - In TheHive's administration settings, navigate to the "Analyzers" configuration.
    - Enable the "Auto-run" feature for the Cortex VirusTotal analyzer.
    - Configure it to automatically run on any new observable that has the data type of hash.
2. **Trigger the Automated Workflow:**
    - The alert forwarded from Wazuh automatically creates a new case in TheHive.
    - The integration script is configured to parse the Wazuh alert and add the SHA256 hash of the downloaded file as a new "observable" in the case.
3. **Verify the Enrichment:**
    - Open the newly created case in **TheHive**.
    - Navigate to the "Observables" tab. You will see the file hash listed.
    - Because of the auto-run configuration, you will also see a small icon or tag next to the hash indicating that an analysis has been performed.

    - Click on the observable to view the analysis report. A report from the VirusTotal analyzer will be displayed directly within TheHive's UI, showing how many security vendors flagged the file as malicious (e.g., "55/70").

4. **Summarize the Results:**
    - Draft a brief summary of the value this automation provides.

Automated Triage Summary

By integrating TheHive with a VirusTotal analyzer, we have automated the critical file hash reputation check. Now, when a file-based alert is ingested, its malware status is validated instantly without any analyst intervention. This eliminates manual lookups, accelerates triage, and provides immediate, actionable intelligence within the case file.

---

6. Expected Outcomes and Deliverables

- A documented initial alert from Wazuh.
- A configured TheHive/Cortex instance that automatically enriches file hash observables with VirusTotal data.
- A 50-word summary explaining the benefits and results of the automated validation process.

README: Task 5 - Evidence Analysis

1. Objective

The objective of this task is to practice the fundamentals of digital forensic evidence analysis and the critical process of maintaining a chain of custody. This exercise focuses on performing live analysis of a system to identify indicators of compromise and on correctly documenting the collection of digital evidence to ensure its integrity and admissibility.

2. Core Concepts

- **Live Analysis:** The forensic examination of a running computer system. This is crucial for analyzing volatile data (like active network connections) that would be lost if the system were shut down.
- **Volatile Data:** Data that exists in a temporary state and is lost when a system loses power (e.g., data in RAM, active processes, network connections).
- **Chain of Custody:** A chronological documentation trail showing the collection, control, transfer, and analysis of evidence. It is essential for proving that evidence has not been tampered with.
- **Cryptographic Hashing:** The process of creating a unique digital fingerprint (e.g., SHA256) of a file or piece of data to ensure its integrity.

3. Tools Required

- **Velociraptor:** For performing live analysis and remote data collection from an endpoint.
- **FTK Imager (for context):** While not used for the live analysis part of this task, it represents the toolset used for analyzing static, non-volatile evidence like disk images.

4. Setup and Prerequisites

1. **Velociraptor Agent:** The Velociraptor agent must be deployed and running on the target Windows VM ("Server-Z") that you will be analyzing.
2. **Test Environment:** Have a Windows VM available for the live analysis. To make the exercise more realistic, you can generate some benign network traffic by opening a web browser.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Initiate Live Analysis:**
    - From the **Velociraptor** server UI, select the client ID corresponding to your target Windows VM, "Server-Z."
    - Navigate to the "Shell" or "VQL Collector" section to run a live query.
2. **Collect Volatile Data:**
    - To collect the list of active network connections, you will use the netstat artifact, which is a built-in capability of Velociraptor.
    - Enter the following Velociraptor Query Language (VQL) query:

codeSQL

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

    - Launch the query. Velociraptor will instruct the agent on the endpoint to gather the netstat information and stream the results back to your server console in real-time.
3. **Analyze the Results:**
    - The output will be a table of all active network connections. Carefully examine each row, paying close attention to the following columns: Name (the process name), DestIP (the remote IP address), DestPort, and Status.
    - **Look for anomalies:**

        - Are there connections from unexpected processes (e.g., svchost.exe, powershell.exe) to external IP addresses?
        - Are there connections on non-standard ports?
        - Are there connections to IP addresses known to be malicious (you would cross-reference these with a threat intelligence source)?
    - **Identify a Suspicious Connection:** In your analysis, you identify that a PowerShell process is making an outbound connection to an unknown IP address on port 443. While the port is common, the process initiating the connection is suspicious. This is a key finding.

4. **Export the Evidence:**
    - Once the query is complete, use the Velociraptor UI to export the results as a CSV file. Name it Server-Z_netstat_log.csv. This file is now considered digital evidence.

1. **Preserve Evidence Integrity:**
    - The moment the evidence file (Server-Z_netstat_log.csv) is created, its integrity must be sealed.
    - Open a terminal or command prompt and use a hashing utility to calculate the SHA256 hash of the file.

codeBash

sha256sum Server-Z_netstat_log.csv

    - The command will output a long alphanumeric string. This is the hash value. Copy it.

2. **Document the Collection:**
    - Open your official chain-of-custody log (this can be a spreadsheet or a dedicated tool).
    - Create a new entry for the evidence you just collected. Fill in every field meticulously. The hash value is non-negotiable and serves as the proof of integrity.

3. **Create the Log Entry:**

Item	Description	Collected By	Date	Hash Value (SHA256)

Network Log	A CSV export of active network connections from Server-Z, collected via a live Velociraptor query.	SOC Analyst	2025-08-18	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

(Note: The example hash is for an empty file. Your hash will be different.)

---

6. Expected Outcomes and Deliverables

- A CSV file containing the network connection data collected from the target VM.
- Identification of at least one suspicious connection during the analysis.
- A fully documented chain-of-custody log entry for the collected evidence, including a valid SHA256 hash.

README: Task 6 - Adversary Emulation Practice

1. Objective

The objective of this task is to provide practical experience in adversary emulation. This exercise will teach you how to use an automated framework to simulate the tactics, techniques, and procedures (TTPs) of a real-world adversary in a controlled environment. The goal is to proactively test and validate your Security Operations Center's (SOC) detection capabilities against known threat behaviors, identify gaps, and drive improvements to your security posture.

2. Core Concepts

- **Adversary Emulation:** The practice of mimicking the TTPs of a known threat actor to test defensive capabilities. It focuses on *behavior* rather than just exploiting vulnerabilities.

- **MITRE ATT&CK® Framework:** A globally accessible knowledge base of adversary tactics and techniques that serves as the foundation for planning and executing emulation exercises.
- **Purple Teaming:** A collaborative approach where the offensive team (Red Team, executing the emulation) and the defensive team (Blue Team, monitoring for detections) work together to improve security.

3. Tools Required

- **MITRE Caldera:** An open-source adversary emulation platform for automating TTP execution.
- **Wazuh:** The SIEM/XDR platform used by the Blue Team to monitor for and detect the simulated attack.

4. Setup and Prerequisites

1. **Caldera Server:** A running instance of the MITRE Caldera server.
2. **Caldera Agent:** The Caldera agent must be deployed on a target endpoint that is also being monitored by a Wazuh agent. This endpoint will be the "victim" in the simulation.
3. **Wazuh Monitoring:** Ensure the Wazuh agent on the target endpoint is configured to monitor for relevant events, such as command-line activity, file creation, and process execution.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Plan the Emulation (Red Team):**
    - **Objective:** Simulate the actions an attacker would take *after* a user has been tricked by a spearphishing email. We will simulate the payload execution phase of T1566.
    - In **MITRE Caldera**, navigate to the "Adversaries" tab and create a new adversary profile.
    - Add an "ability" to this profile that corresponds to T1566. For example, use a built-in ability that creates a file and then executes it using PowerShell, mimicking a common phishing payload.
2. **Prepare the Defenses (Blue Team):**
    - In **Wazuh**, ensure the detection rules are ready. The key rule for this test would be one that monitors for suspicious PowerShell execution. Wazuh has default rules for this, but you can create a custom rule to look for specific command-line arguments or parent processes.

- Example Wazuh rule logic: "Alert if powershell.exe is launched with an encoded command (-enc) or a non-interactive flag (-NonI)."

3. **Execute the Emulation:**
    - In Caldera, create a new "Operation."
    - Select your target endpoint (the machine with the Caldera agent), your newly created adversary profile, and launch the operation.
    - Caldera will instruct the agent on the endpoint to execute the steps defined in the T1566 ability.
4. **Monitor for Detection:**
    - The Blue Team actively monitors the **Wazuh** dashboard in real-time.
    - As Caldera executes its commands on the endpoint, the Wazuh agent will send the relevant logs to the manager.
    - A successful detection will result in a new alert appearing in the dashboard, for example, "Suspicious PowerShell Command Detected."
5. **Document the Results:**
    - The outcome of each step of the emulation must be documented. This forms the basis of the final report. The documentation should be clear about what was tested and what the result was.

Timestamp	TTP	Detection Status	Notes
2025-08-18 17:00:00	T1566	Detected	The simulated execution of a PowerShell payload was successfully detected by Wazuh rule #80790. However, the initial simulated email delivery itself was not visible, as we do not have email gateway logs in Wazuh.

1. **Analyze the Results:**
    - The most important part of the exercise is analyzing *why* certain actions were detected or missed. In the example above, the endpoint action was detected, but there was a visibility gap at the email gateway layer.
2. **Draft the Report:**
    - Create a concise report that summarizes the exercise, celebrates the successes, and, most importantly, is transparent about the gaps and provides actionable recommendations.
3. **Write the Summary:**

Adversary Emulation Report: Phishing Payload Execution Test

An adversary emulation was conducted to validate our defenses against the execution phase of a spearphishing attack (MITRE T1566). Using MITRE Caldera, we simulated a common PowerShell-based payload.

Our Wazuh EDR monitoring successfully **detected** the suspicious PowerShell activity on the endpoint, which is a key success.

However, the exercise highlighted a **detection gap**: our current logging does not provide visibility into the initial email delivery itself. **Recommendation:** We must integrate our email security gateway logs with our SIEM to ensure we can detect the full attack chain, from delivery to execution.

---

6. Expected Outcomes and Deliverables

- A successfully executed Caldera operation simulating an adversary TTP.
- Documented results of the detection status for each TTP tested.
- A 100-word emulation report that summarizes the results and clearly identifies at least one detection gap with a corresponding recommendation for improvement.

README: Task 7 - Security Metrics and Executive Reporting

1. Objective

The objective of this task is to master the critical skills of measuring Security Operations Center (SOC) performance and communicating those results effectively to leadership. This exercise will provide hands-on practice in creating a metrics dashboard, analyzing key performance indicators (KPIs), and drafting a concise executive report that translates technical data into strategic business insights and actionable recommendations.

2. Core Concepts

- **SOC Metrics:** Quantifiable measurements used to track the performance and effectiveness of security operations. Key metrics include MTTD, MTTR, False Positive Rate, and Dwell Time.
- **Executive Reporting:** The practice of summarizing complex technical information into a clear, concise, and business-focused narrative for non-technical stakeholders.
- **Data-Driven Improvement:** Using metrics to identify weaknesses, justify investments, and track progress over time.

3. Tools Required

- **Elastic Security:** As the SIEM and data visualization platform for creating the dashboard.

- **Google Sheets:** For ad-hoc metrics analysis and calculations.
- **Google Docs:** For drafting the executive report.

4. Setup and Prerequisites

1. **Incident Data:** Your Elastic Security instance should contain historical incident data. For this exercise, you can use mock data, but it must include the following fields for each incident:
    - compromise_time: The timestamp of the initial breach.
    - detection_time: The timestamp when the SOC first identified the incident.
    - resolution_time: The timestamp when the incident was fully contained and remediated.
    - status: A field that can be marked as "false_positive."
2. **Defined Mock Incident:** Have a specific mock incident with defined timestamps to analyze for the Dwell Time calculation.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

1. **Define the Metrics:**
    - **MTTD (Mean Time to Detect):** average(detection_time - compromise_time)
    - **MTTR (Mean Time to Respond):** average(resolution_time - detection_time)
    - **False Positive Rate:** count(status: "false_positive") / count(all_alerts) * 100
2. **Build Visualizations in Elastic (Kibana):**
    - Navigate to the **Dashboard** app in Kibana and create a new dashboard.
    - For each metric, create a new visualization using the "Metric" visualization type.
    - **MTTD Visualization:**
        - Use a Lens formula or a scripted field to calculate the duration: average(detection_time) - average(compromise_time). Format the output as a duration (e.g., hours). Title it "Mean Time to Detect (MTTD)." For this example, the result is **2 hours**.
    - **MTTR Visualization:**
        - Use a similar method to calculate average(resolution_time) - average(detection_time). Title it "Mean Time to Respond (MTTR)." For this example, the result is **4 hours**.
    - **False Positive Rate Visualization:**

        - Use the "Metric" visualization with a "Filter Ratio" calculation to get the percentage of documents that match status: "false_positive". Title it "False Positive Rate."

3. **Assemble and Save the Dashboard:**
    - Arrange the three metric visualizations at the top of the dashboard for an at-a-glance view.
    - Save the dashboard with a descriptive name like "SOC Performance KPIs."

1. **Calculate Dwell Time:**
    - **Dwell Time** is the total time an adversary is undetected in the network (detection_time - compromise_time). It is the same calculation as MTTD but is often discussed for a specific, significant incident.
    - In **Google Sheets**, create a simple table for a mock incident.
    - **Mock Incident Timeline:**
        - Initial Compromise: An employee clicks a phishing link on Monday at 9:00 AM.
        - Detection: The SOC detects C2 traffic on Wednesday at 11:00 AM.
    - **Calculation:**
        - The time between the two events is **50 hours**.

2. **Summarize the Findings:**
    - Analyze the business implication of this dwell time and write a brief summary.

Dwell Time Analysis Summary

The analysis of our recent mock phishing incident revealed a dwell time of 50 hours. This extended period provided the adversary with a significant window to perform reconnaissance and move laterally undetected. This highlights a critical need to enhance our proactive threat hunting capabilities to find threats faster.

1. **Draft the Report in Google Docs:**
    - The report should be concise, professional, and focused on risk and recommendations. Use the metrics from your dashboard and analysis to support your narrative.

2. **Write the Executive Summary:**

Subject: Q3 Security Operations Performance and Strategic 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, reflecting an efficient containment process. However, our average time to detect a threat (**MTTD**) stands at **2 hours**, indicating a key area for strategic improvement.

Analysis of a recent simulated incident revealed a potential attacker **dwell time** of over two days, which presents a significant business risk. This is largely driven by a high **false positive rate** (15%) that consumes analyst time.

Recommendation:
To reduce detection time and overall risk, we recommend a strategic investment in a SOAR platform. This will automate the handling of false positives and low-level alerts, freeing up our expert analysts to proactively hunt for threats, which we project could reduce our MTTD by 50% within two quarters.

---

6. Expected Outcomes and Deliverables

- A functional Elastic Security dashboard displaying MTTD, MTTR, and the false positive rate.
- A 50-word summary analyzing the business impact of dwell time.
- A 150-word executive report that presents key metrics and provides a clear, data-driven recommendation for SOC improvement.

README: Task 8 - Capstone Project: Comprehensive SOC Incident Response

1. Objective

This capstone project is a comprehensive, end-to-end simulation designed to integrate all the skills learned in the previous tasks. The objective is to manage a complex security incident from the initial attack simulation to the final executive briefing. This exercise will test your ability to utilize a full suite of offensive and defensive tools, follow a structured incident response process, leverage automation, conduct a thorough post-mortem, and communicate findings effectively to all stakeholders.

2. Core Concepts

This project covers the entire incident response lifecycle, including:

- **Attack Simulation & Adversary Emulation:** Simulating real-world threats.
- **Detection and Triage:** Identifying and prioritizing alerts.
- **Response and Containment:** Taking immediate action to stop the threat.
- **SOAR Automation:** Using playbooks to accelerate response.
- **Post-Incident Analysis:** Identifying root causes to prevent recurrence.
- **Metrics and Reporting:** Measuring performance and communicating value.

3. Tools Required

- **Metasploit & MITRE Caldera:** For attack simulation and emulation.
- **Wazuh & Elastic Security:** For detection, analysis, and metrics visualization.
- **CrowdSec:** For automated network-level containment.
- **TheHive:** For case management and workflow orchestration.
- **Draw.io:** For creating post-mortem diagrams.
- **Google Docs:** For formal reporting and briefing documents.

4. Setup and Prerequisites

1. **Integrated Lab Environment:** A lab environment where all the tools are installed and integrated is essential. This includes:
    - A Metasploitable2 VM as the target.
    - An attacker VM with Metasploit and Caldera.
    - A security monitoring stack (Wazuh/Elastic) with agents on the target.
    - A response stack (TheHive/CrowdSec/SOAR tool) that can receive alerts and execute actions.
2. **Pre-configured Rules and Playbooks:** Have the necessary detection rules in Wazuh and the automated IP-blocking playbook (from Task 2) in your SOAR tool ready and active.

---

5. Enhanced Tasks: Step-by-Step Walkthrough

- **Action:** Using the **Metasploit Framework**, execute the exploit/multi/samba/usermap_script against your Metasploitable2 VM to gain a root shell.
- **Action:** In **MITRE Caldera**, create an operation to simulate **T1210 (Exploitation of Remote Services)** against the same target to test behavioral detection.
- **Documentation:** Record the timestamp and source IP of your attack (192.168.1.102).
- **Action:** Monitor **Wazuh** for the incoming alert related to the Samba exploit.
- **Action:** The alert should automatically trigger your SOAR playbook. Verify that a case is created in **TheHive**.
- **Action:** Verify that the playbook successfully blocks the attacker's IP (192.168.1.102) in **CrowdSec**. Test this by trying to ping another machine from the attacker VM; it should fail.

- **Action:** Manually isolate the Metasploitable2 VM by moving it to a quarantined network segment in your hypervisor.
- **Action:** Conduct a **5 Whys** analysis for the incident. The final root cause should point to a process failure.
    - **Root Cause Example:** "The incident's root cause was a failure in our patch management process, as the critical Samba vulnerability (CVE-2007-2447) was known but remained unpatched on a server believed to be non-critical."
- **Action:** Open **Draw.io** and create a **Fishbone Diagram** visualizing the contributing causes, including "Process" (No vulnerability scanning on dev systems) and "Technology" (Outdated Samba version).
- **Action:** Determine the key timestamps for the incident:
    - Time of Compromise: 2025-08-18 16:00:00
    - Time of Detection (Wazuh alert): 2025-08-18 16:02:00
    - Time of Resolution (VM isolated, IP blocked): 2025-08-18 16:10:00
- **Action:** Calculate the metrics:
    - **Dwell Time / MTTD:** 2 minutes
    - **MTTR:** 8 minutes
- **Action:** In **Elastic Security**, update your "SOC Performance KPIs" dashboard with the data from this incident.
- **Action:** In **Google Docs**, draft a comprehensive, 300-word incident report using a SANS template. This report is for a technical audience and for archival.

Executive Summary: On August 18, 2025, the SOC detected and contained a critical incident involving the compromise of a development server (Metasploitable2). The attack was contained in under 10 minutes due to automated defenses, with no impact on production systems. The root cause was a patch management failure. Recommendations are outlined to prevent recurrence.

Incident Timeline:

- **16:00:00:** Attacker exploits Samba vulnerability (T1210).
- **16:02:00:** Wazuh detects exploit, triggers SOAR playbook.
- **16:03:00:** SOAR playbook automatically blocks attacker IP.
- **16:10:00:** Analyst manually isolates VM; incident contained.

Root Cause Analysis (RCA): The fundamental cause was a procedural failure in our vulnerability management program, which excluded development assets from routine

scanning and patching schedules. This allowed a known critical vulnerability (CVE-2007-2447) to remain exposed.

Recommendations:

5. **Immediate:** Decommission the vulnerable Metasploitable2 server.

6. **Tactical:** Expand the scope of our vulnerability management program to include all development and staging environments, with a mandate to patch critical vulnerabilities within 72 hours.

7. **Strategic:** Implement quarterly adversary emulation exercises, using frameworks like Caldera, to continuously validate that our security controls and detection rules are effective against the TTPs most relevant to our organization.

- **Action:** Draft a separate, 150-word briefing for a non-technical executive.

Subject: Summary and Outcome of August 18 Security Incident

This memo is to update you on a security incident our team successfully managed yesterday. We detected and immediately blocked an external cyberattack targeting a non-production server used for development purposes.

Our automated defense systems performed exceptionally, stopping the threat in under three minutes and preventing any further access to our network or data. I can confirm there was zero impact on our customers, production services, or sensitive information.

The investigation revealed the issue stemmed from an out-of-date server, highlighting a gap in our internal processes for non-production systems. We have already fixed the immediate issue and are expanding our patch management program to ensure all systems are protected. This incident has served as a valuable test, validating our investment in security automation and providing clear direction for further strengthening our defenses.

---

6. Expected Outcomes and Deliverables

- Successful execution of a multi-stage attack and response simulation.
- Completed post-incident analysis documents (5 Whys, Fishbone Diagram).
- An updated metrics dashboard in Elastic Security.
- A comprehensive, 300-word technical incident report.
- A concise, 150-word non-technical executive briefing.