Ark Jain

Theoretical Knowledge

1. Alert Priority Levels

Part 1: Theoretical Knowledge of Alert Priority Levels

Section 1.1: Core Concepts of Alert Priority

At its core, alert prioritization is about categorizing security incidents based on their potential impact and urgency to ensure that the most severe threats are addressed immediately. A typical framework categorizes alerts into four main levels: Critical, High, Medium, and Low.

- **Critical (P1):** These are the most severe incidents, posing an immediate and significant threat to the organization. They often involve active breaches, significant data loss, or widespread disruption of critical services.
    - **Example:** A successful ransomware attack encrypting files on a critical production server.
- **High (P2):** High-priority incidents are severe and have the potential to cause significant damage if not addressed promptly. They may not be an active breach yet, but could quickly escalate.
    - **Example:** Malware detected on a key system that has not yet spread, or unauthorized administrative access to a sensitive database.
- **Medium (P3):** These incidents carry a moderate risk with a limited or localized potential impact. They do not pose an immediate threat to core business functions but require attention.
    - **Example:** Suspicious but inconclusive network activity on a non-critical system.
- **Low (P4):** Low-priority alerts represent minimal risk. These could be informational alerts or minor policy violations that need to be addressed but are not time-sensitive.
    - **Example:** A single failed login attempt from an unknown IP address.

Section 1.2: Criteria for Assigning Priority

To assign a priority level, SOC analysts evaluate an alert against a set of criteria. This risk-based prioritization is crucial for focusing on what matters most. Key factors include:

- **Asset Criticality:** The business importance of the affected system or data. An alert on a mission-critical server will always have a higher priority than the same alert on a development machine.
- **Business Impact:** The potential damage to the organization, including financial loss, data breach, reputational harm, or disruption of services.
- **Threat Context & Likelihood:** The specific type of threat and how likely it is to succeed. Factors include the sophistication of the threat actor and the stage of the attack in the cyber kill chain. An alert indicating data exfiltration is more severe than one for initial reconnaissance.
- **Data Sensitivity:** Alerts involving sensitive or regulated data, such as personally identifiable information (PII) or financial records, are treated with a higher priority.

Section 1.3: Mastering Scoring Systems

Scoring systems provide a standardized way to quantify the severity of vulnerabilities and, by extension, the alerts they may trigger.

Common Vulnerability Scoring System (CVSS): CVSS is the industry standard for rating the severity of software vulnerabilities. It provides a numerical score from 0 to 10, which can be translated into qualitative severity levels.

- **CVSS Score Ranges:**
  - **None:** 0.0
  - **Low:** 0.1 - 3.9
  - **Medium:** 4.0 - 6.9
  - **High:** 7.0 - 8.9
  - **Critical:** 9.0 - 10.0
- **CVSS Metric Groups:**

1. **Base Metrics:** These represent the intrinsic qualities of a vulnerability that do not change over time or in different environments. They include factors like the attack vector, attack complexity, and the impact on confidentiality, integrity, and availability.
2. **Temporal Metrics:** These reflect characteristics of a vulnerability that may change over its lifetime, such as the availability of an exploit or a patch.
3. **Environmental Metrics:** These allow an organization to customize the CVSS score based on its specific environment, such as the criticality of the affected asset.

Other Scoring Systems: Many SOCs use their own or tool-specific risk-scoring mechanisms. For instance, a Security Information and Event Management (SIEM) system might generate a risk score for an alert by correlating multiple data points, including threat intelligence and asset value.

Part 2: How to Learn and Apply This Knowledge

Here is a step-by-step guide to developing the skills necessary to assess and prioritize alerts effectively.

Step 1: Study the Common Vulnerability Scoring System (CVSS)

- **Action:** Visit the website of the Forum of Incident Response and Security Teams (FIRST.org) to access the official CVSS documentation, including the user guide and examples.
- **Focus On:**
  - **Base Metrics:** Understand how factors like Attack Vector (is it remote or local?), Attack Complexity, Privileges Required, and User Interaction affect the score.
  - **Impact Metrics:** Grasp the concepts of Confidentiality, Integrity, and Availability (the "CIA Triad").
  - **Temporal and Environmental Metrics:** Learn how these can be used to adjust a base score to better reflect the risk to your specific organization.

Step 2: Review NIST Incident Handling Guidance

- **Action:** Read through the National Institute of Standards and Technology (NIST) Special Publication 800-61, the "Computer Security Incident Handling Guide
- **Focus On:**
  - **Incident Response Life Cycle:** Understand the phases of Preparation, Detection & Analysis, Containment, Eradication & Recovery, and Post-Incident Activity.
  - **Incident Classification:** Pay close attention to how NIST recommends classifying and prioritizing incidents based on their functional and informational impact and recoverability.

Step 3: Analyze a Real-World Case Study: The Log4Shell Vulnerability

- **Action:** Research the Log4Shell vulnerability (CVE-2021-44228). This is a prime example of a critical threat that required immediate and widespread attention.
- **Analysis Points:**
  - **The CVSS Score:** Log4Shell was assigned a CVSS score of 10.0—the highest possible rating.
  - **Why it was Critical:** The vulnerability was easy to exploit remotely, allowing attackers to execute arbitrary code and take full control of affected systems. It existed in a widely used Java logging library, Log4j, meaning hundreds of millions of devices were potentially at risk.
  - **Prioritization in Action:** CISA, the FBI, and other agencies strongly urged all organizations to immediately patch their systems, prioritizing those that were internet-facing. This case perfectly illustrates how a high CVSS score, combined with widespread impact and ease of exploitation, leads to a critical alert priority.

2. Incident Classification

Part 1: Theoretical Knowledge of Incident Classification

Section 1.1: Core Concepts of Incident Classification

Effective incident classification involves systematically categorizing security events. This ensures a common understanding, facilitates accurate reporting, and enables efficient response.

- **Incident Categories:** The first step is to group incidents by their nature. Common categories include:
  - **Malware:** Malicious software designed to disrupt operations or gain unauthorized access. Examples range from ransomware like WannaCry to spyware.
  - **Phishing:** A form of social engineering where attackers deceive individuals into divulging sensitive information. This is often the initial step in a larger attack.
  - **Denial-of-Service (DoS/DDoS):** Attacks that overwhelm a system's resources, making it unavailable to legitimate users.
  - **Insider Threat:** A security risk originating from within the organization, such as an employee, former employee, or contractor. This can be malicious (e.g., unauthorized data export by a disgruntled employee) or unintentional (e.g., an employee misconfiguring a server).
  - **Data Exfiltration:** The unauthorized transfer of data from a computer or network. This is the ultimate goal of many cyberattacks.
  - **Improper Usage:** A violation of an organization's acceptable use policies by an authorized user.
- **Taxonomy Frameworks:** To standardize classification, SOCs rely on established frameworks. These provide a common language to describe incidents consistently.
  - **MITRE ATT&CK®:** A globally accessible knowledge base of adversary tactics and techniques based on real-world observations. It is not just a list of attacks but a framework that helps you understand an adversary's goals.
    - **Tactics:** The "why" of an attack (the adversary's technical goal). Examples include *Initial Access*, *Execution*, and *Exfiltration*.
    - **Techniques:** The "how" an adversary achieves a tactic. For example, within the *Initial Access* tactic, a specific technique is **T1566 – Phishing**.
  - **ENISA Incident Taxonomy:** The European Union Agency for Cybersecurity (ENISA) provides a comprehensive taxonomy that covers

a wide range of incident types, including nefarious activity, abuse, and information content security.
    - **VERIS (Vocabulary for Event Recording and Incident Sharing):** A set of metrics designed to provide a common language for describing security incidents in a structured and repeatable way. It focuses on the "Four A's":
        - **Actor:** Who was behind the incident? (e.g., external threat, internal employee).
        - **Action:** What did they do? (e.g., hacking, malware, social engineering).
        - **Asset:** What was affected? (e.g., server, user device, data).
        - **Attribute:** How was the asset compromised? (e.g., confidentiality, integrity, availability).
- **Contextual Metadata:** An incident is more than just a category; it's a collection of data points that provide context for the investigation. Enriching an incident with this metadata is crucial. Key elements include:
    - **Affected Systems:** Hostnames, IP addresses, and the function of the impacted assets (e.g., web server, domain controller).
    - **Timestamps:** When the event occurred, when it was detected, and when the investigation began.
    - **Source and Destination IPs:** Where the attack originated from and what it was targeting.
    - **Indicators of Compromise (IOCs):** Forensic data that identifies malicious activity, such as file hashes, malicious domains, or specific registry key changes.

Section 1.2: Key Objectives of Incident Classification

The primary goal of mastering incident classification is to **streamline the investigation and response process**. Proficiently categorizing, labeling, and enriching incidents allows a SOC to:

- **Prioritize Effectively:** Quickly identify the nature of a threat to better assess its potential impact.
- **Improve Efficiency:** Route incidents to the correct teams and automated playbooks faster.
- **Enhance Threat Intelligence:** Structured data allows for better analysis of trends, helping the organization understand its threat landscape and proactively strengthen defenses.
- **Standardize Communication:** Use a common language to report on incidents to both technical and non-technical stakeholders.

Part 2: How to Learn and Apply This Knowledge

Here is a step-by-step guide to developing your incident classification skills.

Step 1: Explore the MITRE ATT&CK Framework

- **Action:** Navigate to the official MITRE ATT&CK website. It features an interactive matrix of tactics and techniques for enterprise, mobile, and industrial control systems.
- **Practice:** Take a known incident, such as a phishing attack that led to a ransomware infection. Map the incident's progression through the ATT&CK framework:
    1. **Initial Access:** T1566 - Phishing
    2. **Execution:** T1204.002 - Malicious File
    3. **Defense Evasion:** T1070.004 - File Deletion
    4. **Impact:** T1486 - Data Encrypted for Impact
    This exercise helps you think from the attacker's perspective and understand the entire lifecycle of an incident.

Step 2: Study the ENISA and VERIS Frameworks

- **Action:** Review the documentation for the ENISA Incident Taxonomy and the VERIS framework. The VERIS website provides access to the VERIS Community Database, which contains thousands of real-world incidents classified using the framework.
- **Focus On:** Understand the different goals of these frameworks. While ATT&CK is focused on adversary behavior, VERIS is excellent for recording incident details in a structured way for later analysis and metrics. For example, using VERIS, you would classify an incident by its actor (External), action (Hacking), asset (Server), and attribute (Confidentiality compromised).

Step 3: Review Case Studies to Practice Adding Metadata

- **Action:** Read through public breach reports and case studies from reputable sources like the SANS Institute's Reading Room, Verizon's Data Breach Investigations Report (which uses VERIS), or news articles on major cyberattacks.
- **Practice:** As you read a case study (for example, the 2017 Equifax breach), create your own incident report.
    - **Categorize it:** Data Breach, Hacking.
    - **Apply a framework:** Map the attacker's techniques to MITRE ATT&CK (e.g., T1190 - Exploit Public-Facing Application).
    - **List the metadata:**
        - **Affected Asset:** Equifax web application server.

- **IOCs:** The specific vulnerability exploited (Apache Struts CVE-2017-5638).
- **Data Compromised:** Personal data of approximately 147 million consumers.
- **Timeline:** Note the date of the initial compromise and the date of discovery.

This practice will build your proficiency in extracting and organizing the critical information needed for a swift and effective incident response.

3. Basic Incident Response

Part 1: Theoretical Knowledge of Basic Incident Response

Section 1.1: Core Concepts: The Incident Response Lifecycle

The incident response process is a cyclical framework that guides an organization's actions from preparation through learning from an incident. The most widely adopted models are from NIST (National Institute of Standards and Technology) and the SANS Institute. While they may label the phases slightly differently, the core activities are aligned.

Here are the six logical phases of the incident response lifecycle:

- **1. Preparation:** This foundational phase happens *before* an incident occurs. Its goal is to ensure the organization is ready to respond.
  - **Examples:** Developing and practicing incident response plans (playbooks), establishing a formal Computer Security Incident Response Team (CSIRT), and ensuring the necessary tools (like SIEM, EDR) are in place and configured correctly.
- **2. Identification (or Detection & Analysis):** This is where a potential security incident is first noticed and investigated.
  - **Examples:** A SIEM alert fires for multiple failed logins, an EDR system flags a malicious process, or a user reports a suspicious email. The SOC analyst's job is to analyze data from logs, alerts, and other sources to confirm if an event is a genuine incident.
- **3. Containment:** Once an incident is confirmed, the immediate goal is to stop it from spreading and causing further damage.
  - **Examples:** Isolating a compromised host from the network to prevent malware from moving to other systems, blocking a malicious IP address at the firewall, or disabling a compromised user account.
- **4. Eradication:** This phase focuses on removing the threat and its artifacts from the environment.

    - **Examples:** Deleting malicious files, removing malware, patching the vulnerability that was exploited, and ensuring the attacker has no lingering access.
- **5. Recovery:** The goal of this phase is to safely restore affected systems and services to normal operation.
    - **Examples:** Restoring systems from clean backups, rebuilding compromised machines, and monitoring the environment closely to ensure the threat does not return.
- **6. Lessons Learned (or Post-Incident Activity):** This is a critical, and often overlooked, phase that takes place after recovery.
    - **Examples:** Conducting a post-mortem meeting to review what happened, what went well, what could be improved, and updating playbooks, policies, and security controls based on the findings. This ensures the organization becomes more resilient against future attacks.

Section 1.2: Key Procedures and Tools
Within the lifecycle, analysts must execute specific technical and procedural tasks:

- **System Isolation:** A primary containment strategy. This can be done by disconnecting the network cable, using an EDR tool to quarantine the host, or placing the system on an isolated VLAN.
- **Evidence Preservation:** It's crucial to collect and preserve evidence for forensic analysis and potential legal action.
    - **Examples:** Creating a forensic image (a bit-by-bit copy) of a hard drive, capturing a memory dump of a running system, and using hashing algorithms to verify the integrity of collected evidence.
- **Communication Protocols:** A clear communication plan is vital. This outlines who to notify (e.g., management, legal, HR), when, and how. It ensures all stakeholders are informed and that the response is coordinated.
- **SOAR (Security Orchestration, Automation, and Response):** These tools are used to automate repetitive incident response tasks.
    - **Example:** A SOAR playbook could be triggered by a phishing alert. It could automatically detonate the suspicious attachment in a sandbox, check the sender's reputation, and if malicious, block the sender's domain at the email gateway and search for other instances of the email in the organization—all before an analyst even begins their manual investigation.

Part 2: How to Learn and Apply This Knowledge
Here is a step-by-step guide to developing the skills necessary to respond to security incidents.

Step 1: Study the NIST SP 800-61 Incident Handling Guide

- **Action:** Read NIST Special Publication 800-61, the "Computer Security Incident Handling Guide." It is the foundational document for incident response in the industry.

- **Focus On:** Pay close attention to the detailed breakdown of the four main phases: Preparation; Detection and Analysis; Containment, Eradication, and Recovery; and Post-Incident Activity. Understand the goals and recommended actions for each.

Step 2: Use the SANS Incident Handler's Handbook

- **Action:** Access the SANS Institute's "Incident Handler's Handbook." This resource provides practical, real-world guidance and checklists.
- **Focus On:** The handbook's six-phase model (Preparation, Identification, Containment, Eradication, Recovery, Lessons Learned) offers a slightly more granular view. Use its templates and checklists as a guide for what to do during an actual incident.

Step 3: Gain Hands-On Experience with Let's Defend

- **Action:** Sign up for an account on Let's Defend, a training platform that provides a simulated SOC environment.
- **Practice:** This platform allows you to investigate real-world cyber-attack scenarios in a hands-on manner. You will be able to analyze alerts, create cases, and follow playbooks, mirroring the workflow of a real SOC analyst. This practical experience is invaluable for reinforcing theoretical knowledge and understanding how SOAR concepts are applied to automate response actions.

Practical Application

1. Alert Classification System

Objective: To develop a structured system for classifying incoming alerts by mapping them to the MITRE ATT&CK® framework, thereby standardizing analysis and response.

Tools Used: Google Sheets

Methodology: A table was created to map alert IDs and types to a priority level and a corresponding MITRE ATT&CK Tactic/Technique. This provides immediate context to an alert, helping an analyst understand the potential adversary behavior behind it.

Output: Alert Classification Mapping Table

Alert ID	Alert Type	Priority	MITRE Tactic	MITRE Technique	Description
001	Phishing Email: Suspicious Link	**High**	Initial Access	T1566: Phishing	User reported a suspicious email with a link to a known malicious domain.
002	Multiple Failed Logins	**Medium**	Credential Access	T1110: Brute Force	150 failed login attempts from a single IP against the admin account.
003	Port Scan Detected	**Low**	Reconnaissance	T1046: Network Service Scanning	An external IP address scanned multiple ports on our external firewall.
004	Log4Shell Exploit Detected	**Critical**	Initial Access	T1190: Exploit Public-	A WAF alert triggered for a pattern matching the

				Facing Application	Log4Shell vulnerability (CVE-2021-44228).
005	Powershell Empire Execution	**High**	Execution	T1059.001: PowerShell	EDR flagged the execution of a suspicious PowerShell command consistent with the Empire framework.
006	Data Exfiltration Anomaly	**Critical**	Exfiltration	T1041: Exfiltration Over C2 Channel	An unusual amount of outbound traffic (2 GB) was detected from a database server to an unknown external IP.

2. Alert Prioritization Practice

Objective: To practice prioritizing a queue of simulated alerts based on potential impact and urgency, using the Common Vulnerability Scoring System (CVSS) where applicable.

Tools Used: Google Sheets

Methodology: A list of simulated alerts was generated. For vulnerability-related alerts, the CVSS 3.1 score was used to assign a severity level. For other alerts, priority was assigned based on asset criticality and potential business impact.

Output: Simulated Alert Prioritization Queue

Alert Name	CVSS Score	Data Source	Assigned Priority	Justification
Log4Shell Exploit Detected	**9.8**	WAF	**Critical**	CVE-2021-44228. A publicly known, easily exploitable RCE vulnerability affecting a critical web server.
Ransomware Behavior Detected	N/A	EDR	**Critical**	Active encryption detected on a file server. Immediate impact on data availability and business operations.
Unauthorized Admin Access	N/A	SIEM	**High**	A successful login to a domain controller from an unusual geographic location.
Brute Force on SSH	5.3	Firewall Logs	**Medium**	Attempted brute force on a non-critical server. High volume but unsuccessful.
Port Scan from External IP	N/A	IDS	**Low**	Common reconnaissance activity. No indication of a successful breach. Informational.

3. Dashboard Creation in Wazuh
Objective: To create a dashboard visualization in Wazuh for monitoring alert priorities at a glance.
Tools Used: Wazuh
Methodology: Logged into the Wazuh dashboard and navigated to the Visualization section. A new pie chart widget was created and configured. The data source was set to the security alerts index. The chart was configured to aggregate alerts based on the rule.level field, which corresponds to alert severity. Levels were grouped into Critical (12-15), High (8-11), Medium (5-7), and Low (3-4) for clear visualization.
Result: The created dashboard now features a pie chart titled **"Live Alert Priorities."** This chart provides an immediate, real-time visual breakdown of all incoming alerts by severity. This allows the SOC team to instantly gauge the current threat level and focus their attention on the most critical threats without needing to parse raw logs.

4. Incident Ticket Creation in TheHive

Objective: To draft a formal incident ticket in TheHive for a critical security event, ensuring all necessary information is included for investigation and tracking.

Tools Used: TheHive (simulated)

Methodology: An incident ticket was drafted based on a critical ransomware detection alert. The ticket includes a clear title, a concise description with key indicators, the appropriate priority level, and is assigned to the primary response group.

Output: Incident Ticket Draft

- **Title:** [Critical] Ransomware Detected on Server-X
- **Description:**
  - EDR agent on **Server-X (192.168.1.50)** triggered a high-severity alert for ransomware-like behavior at 14:30 UTC.
  - Multiple files on shared drives are being rapidly encrypted with the .locked extension.
  - Initial analysis points to a malicious executable.
  - **Indicators of Compromise (IOCs):**
    - **File Name:** crypto_locker.exe
    - **File Hash (SHA-256):** e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
    - **Source IP:** 192.168.1.50
- **Priority:** Critical
- **Assignee:** SOC Analyst
- **Tags:** Ransomware, Active-Breach, Server-X, Containment-Needed

5. Escalation Role-Play

Objective: To practice the formal communication procedure for escalating a critical incident to the next tier of response.

Tools Used: Email Client (simulated)

Methodology: An email was drafted to escalate the previously created ransomware incident to the Tier 2 SOC Analyst or Incident Response team. The email is concise, informative, and clearly states the required action.

Output: Escalation Email Draft

To: Tier 2 SOC Analysts t2-soc@company.com

From: Tier 1 SOC Analyst <arkjain@company.com>

Subject: URGENT: Escalation of Critical Ransomware Incident - IOCs Included

Tier 2 Team,

This is a formal escalation of a critical ransomware incident currently in progress.

At 14:30 UTC, EDR detected active ransomware on **Server-X (192.168.1.50)**. Files are being encrypted. We have confirmed this is a legitimate incident and have placed the host in network quarantine as a preliminary containment step.

Key IOCs:

- **Malicious File:** crypto_locker.exe

- **File Hash (SHA-256):** e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Please take ownership of this incident for immediate eradication and recovery. The full ticket is available in TheHive (Case #2025-058).
Thank you,
Ark Jain
SOC Analyst, Tier 1

2. Response Documentation

1. Incident Response Template
Objective: To document a mock phishing incident using a standardized template, demonstrating the ability to create a comprehensive report suitable for both technical and executive review.
Tools Used: Google Docs
Methodology: The SANS Institute's incident response template structure was used to document a simulated phishing attack from initial detection to final resolution.
Output: Final Incident Report - IR-2025-071
1. Executive Summary
On October 6, 2025, a SOC analyst detected a malware infection on a user workstation. The infection originated from a phishing email where the user clicked a malicious link. The response team immediately contained the threat by isolating the endpoint. The investigation confirmed an infostealer was deployed, but no evidence of successful data exfiltration was found. The endpoint was remediated, and additional security controls were implemented. The business impact was minimal, limited to the temporary loss of productivity for one user.
2. Timeline of Events (Chronology)

Date & Time (UTC)	Event
2025-10-06 10:00	User receives a phishing email with the subject "Action Required: Invoice Update."
2025-10-06 10:05	User clicks the embedded link.
2025-10-06 10:06	EDR system generates an alert for suspicious PowerShell execution on the user's workstation.
2025-10-06 10:10	SOC Analyst begins investigation of the EDR alert.
2025-10-06 10:15	Analyst confirms malicious activity and isolates the endpoint using the EDR console.

2025-10-06 10:30	Malicious domain and associated IP address are blocked at the firewall.
2025-10-06 11:00	A search via the email gateway confirms no other users received this specific phishing email.
2025-10-07 09:00	The affected workstation is reimaged from a known good backup and returned to the user.
2025-10-07 14:00	Incident post-mortem completed. Case closed.

3. Impact Analysis

- **Business Impact:** Low. One user's productivity was impacted for less than one business day. No core business services were affected.
- **Data Impact:** Contained. The malware was identified as an infostealer. A forensic review of network logs showed no signs of successful data exfiltration. As a precaution, the user's corporate credentials were reset.
- **Technical Impact:** Limited to a single endpoint, which was successfully remediated.

4. Remediation Steps

- **Containment:** The compromised endpoint was immediately isolated from the network.
- **Eradication:** The affected endpoint was completely wiped and reimaged.
- **Hardening:** The malicious domain was added to the web proxy blocklist. The hash of the malware was added to the EDR blacklist.
- **User Action:** The user's password was reset. The user was re-enrolled in phishing awareness training.

5. Lessons Learned

- **Success:** The EDR tool was highly effective at detecting the post-exploitation technique, allowing for a rapid response that prevented data loss.
- **Improvement Area 1:** The phishing email bypassed existing email security filters. A review of email gateway rules is recommended to better detect sophisticated lures.
- **Improvement Area 2:** The user reported the incident after the EDR alert had already fired. Continued user education is needed to encourage proactive reporting of suspicious emails.

2. Investigation Steps Documentation

Objective: To log the specific actions taken during an investigation, creating a clear audit trail for quality control and future analysis.

Methodology: A chronological log was created for the mock phishing incident, detailing each step taken by the responding analyst.

Output: Analyst Action Log for IR-2025-071

Timestamp (UTC)	Action	Notes
2025-10-06 10:10:15	Acknowledged EDR Alert ID 7834-B and started investigation.	Alert for "Suspicious PowerShell Activity."
2025-10-06 10:12:45	Analyzed process tree in EDR. Confirmed PowerShell was launched by Chrome.	Indicates a browser-based exploit or download.
2025-10-06 10:15:30	**Isolated endpoint** WKSTN-108 via EDR console.	Containment action to prevent lateral movement.
2025-10-06 10:25:00	Extracted IOCs (domain: secure-document-portal.xyz, IP: 123.45.67.89).	Submitted IOCs to VirusTotal for analysis.
2025-10-06 10:30:00	Blocked malicious domain and IP on firewall and web proxy.	Proactive remediation step.
2025-10-06 10:45:00	Contacted user to gather context on their activity.	User confirmed clicking a link in an email.
2025-10-06 11:00:10	Performed enterprise-wide email search for the sender/subject.	Search returned only one result (the affected user).
2025-10-06 11:30:00	**Collected memory dump** from the isolated endpoint for forensic analysis.	Evidence preservation.
2025-10-06 11:45:00	Escalated ticket to IT for endpoint reimaging.	Remediation hand-off.

3. Phishing Analysis Checklist
Objective: To create a standardized, repeatable checklist for the initial analysis of a reported phishing email.
Tools Used: Google Docs (simulated)
Output: Tier 1 Phishing Analysis Checklist

Phase 1: Initial Triage & Verification

- [ ] Obtain the original email as an .eml or .msg file.
- [ ] Create a ticket in the incident management system.
- [ ] Examine email headers to verify the Return-Path and Received-SPF fields. Note any signs of spoofing.

Phase 2: Indicator of Compromise (IOC) Analysis

- [ ] Extract all URLs. Check link reputation using VirusTotal and other threat intelligence tools. **DO NOT CLICK THE LINKS.**

- [ ] Extract all attachment hashes. Check hash reputation in VirusTotal and EDR. **DO NOT OPEN ATTACHMENTS.**
- [ ] Analyze the sender's email address and domain. Check domain age and reputation.

Phase 3: Impact Assessment & Containment

- [ ] Search email logs to identify all other users who received the same email.
- [ ] If links were clicked or attachments opened, identify affected user(s) and endpoint(s).
- [ ] Quarantine any confirmed malicious emails from user inboxes.
- [ ] Block confirmed malicious senders, domains, IPs, and hashes.

Phase 4: Closure

- [ ] Document all findings in the ticket.
- [ ] Notify affected user(s) with remediation instructions.
- [ ] Escalate to Tier 2 if an endpoint compromise is confirmed.

4. Mock Post-Mortem Summary

Objective: To concisely summarize the key lessons learned from a simulated incident, focusing specifically on actionable process improvements.

Methodology: A 50-word summary was drafted following a simulated breach scenario where the response was delayed due to a failure in the alerting process.

Output: Post-Mortem Key Finding

The simulated breach revealed a critical gap in our alert notification process, as the primary analyst missed the initial page. **Improvement:** We will implement a redundant alerting system that automatically escalates to the secondary analyst and the SOC manager if a critical alert is not acknowledged within 15 minutes.

3. Alert Triage Practice

1. Triage Simulation in Wazuh
Objective: To simulate the initial investigation and documentation of a common security alert generated by Wazuh.
Tools Used: Wazuh (simulated)
Methodology: A mock alert for "Brute-force SSH Attempts" was selected for analysis. The standard procedure for initial triage involves documenting the key details of the alert in the case management system to create a record of the investigation. The alert's priority is assigned based on the potential impact and the fidelity of the alert signature.
Output: Triage Documentation

Alert ID	Rule ID	Description	Source IP	Destination Asset	Priority	Status	Analyst Notes
002	5712	SSHD brute force trying to get access to the system.	185.222.58.64	linux-web-srv01	**Medium**	**Open**	Multiple (20+) failed SSH login attempts detected in a short period. The source IP is external. Escalating for threat intelligence validation before further action.

2. Threat Intelligence Validation
Objective: To validate the IOCs from the simulated alert by cross-referencing them with a public threat intelligence platform to determine if they are associated with known malicious activity.

Tools Used: AlienVault OTX, VirusTotal (simulated)
Methodology: The source IP address (185.222.58.64) from the Wazuh alert was submitted to AlienVault OTX for analysis. The platform was queried for any existing "pulses" (collections of IOCs), associated malware, or reports of malicious activity linked to this IP.
Output: Threat Intelligence Findings Summary
The IP address 185.222.58.64 was cross-referenced in AlienVault OTX. The IP is associated with multiple recent threat pulses related to SSH scanning and brute-force attacks. Community reports confirm it is a known bad actor. This validates the alert as a true positive, representing a real threat.

4. Evidence Preservation

1. Volatile Data Collection
Objective: To collect volatile data, specifically active network connections, from a live Windows virtual machine. This type of data is lost when a system is powered off and is crucial for understanding an endpoint's activity at the time of an incident.
Tools Used: Velociraptor
Methodology:
A connection was established to the target Windows VM using the Velociraptor agent. A Velociraptor Query Language (VQL) query was executed to collect the output of the netstat command, which provides a list of all active TCP/IP network connections and listening ports.

VQL Query Executed:

```
SELECT * FROM netstat
```

Result:
The query successfully executed on the endpoint. The results, containing details such as protocol, local address, foreign address, state (e.g., ESTABLISHED, LISTENING), and the associated process ID (PID), were collected and exported to a CSV file named network_connections_20251007.csv for analysis and preservation. This file provides a snapshot of the machine's network communications during the incident.

2. Evidence Collection & Chain of Custody
Objective: To acquire a full memory dump from a compromised server for forensic analysis and to document the collection process using a chain-of-custody log to ensure evidence integrity.
Tools Used: Velociraptor, sha256sum utility
Methodology:

1. **Acquisition:** The Velociraptor artifact Artifact.Windows.Memory.Acquisition was used to perform a live memory acquisition from the target machine, identified as

"Server-X". This artifact safely captures the entire contents of the system's RAM and packages it into a single file.

2. **Hashing:** Upon successful collection, the sha256sum utility was used to calculate the SHA256 hash of the resulting memory dump file. This hash value serves as a digital fingerprint, ensuring the integrity of the evidence can be verified at any point in the future.
3. **Documentation:** All relevant details of the collection were recorded in a formal chain-of-custody table.

Output: Chain of Custody Log

Item	Description	Collected By	Date	Hash (SHA256)
Memory Dump	Full memory dump from Server-X (192.168.1.100) following a critical EDR alert.	SOC Analyst	2025-10-07	a34b9e7c5b2a8f8d9b9e1c2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2c

5. Capstone Project: Full Alert-to-Response Cycle

1. Attack Simulation
Objective: To simulate a realistic attack by exploiting a known vulnerability to test the SOC's detection and response capabilities in a controlled environment.
Tools Used: Metasploit Framework, Metasploitable2 VM
Methodology:
The vsftpd_234_backdoor vulnerability on the Metasploitable2 lab server (192.168.1.150) was targeted from an attacker machine (192.168.1.100). The Metasploit Framework was used to configure and launch the exploit, which provides a remote command shell upon successful execution.
Steps Executed in Metasploit:

1. Launched msfconsole.
2. Selected the exploit: use exploit/unix/ftp/vsftpd_234_backdoor
3. Set the target host: set RHOSTS 192.168.1.150
4. Executed the exploit: exploit
5. Result: A command shell session was successfully opened on the target, confirming a successful compromise.

---

2. Detection and Triage
Objective: To detect the simulated attack using the SIEM and correctly triage the resulting alert.
Tools Used: Wazuh
Methodology:
The Wazuh agent installed on the Metasploitable2 VM monitored system activity. Upon execution of the exploit, the malicious connection and subsequent commands triggered a pre-configured rule in Wazuh. The generated alert was analyzed by the SOC analyst to determine its nature, source, and severity.
Output: Alert Triage Documentation

Timestamp (UTC)	Source IP	Alert Description	MITRE Technique	Priority
2025-10-07 11:00:15	192.168.1.100	VSFTPD backdoor exploit detected on metasploitable2.	T1190: Exploit Public-Facing Application	**Critical**

---

3. Response
Objective: To execute the standard containment and blocking procedures to neutralize the threat.
Tools Used: Hypervisor (VMware/VirtualBox), CrowdSec

Methodology:
The response was twofold: immediate isolation of the asset and blocking of the malicious source IP.

1. **Isolation:** The Metasploitable2 virtual machine's network adapter was immediately reconfigured in the hypervisor to an "Internal Only" network. This action severed its connection to the main network, preventing any potential for lateral movement from the compromised host.
2. **IP Blocking:** The attacker's IP address was added to the CrowdSec blocklist using the command line interface: sudo cscli decisions add --ip 192.168.1.100 --type ban --reason "Manual ban following VSFTPD exploit."
3. **Verification:** A ping 192.168.1.150 test was initiated from the attacker's machine. The test failed with "Destination Host Unreachable," confirming that both the VM isolation and the CrowdSec IP ban were effective.

---

4. Reporting (SANS-Style Incident Report)
Objective: To create a formal incident report documenting the event from start to finish.
Tools Used: Google Docs (simulated)
Output: Incident Report IR-2025-072
Executive Summary
At 11:00 UTC on October 7, 2025, security monitoring tools detected an active exploit against a non-production lab server, Metasploitable2. The attack originated from an internal IP (192.168.1.100) and successfully compromised the server via a known backdoor in its VSFTPD service. The incident response plan was activated immediately. The SOC team contained the threat by isolating the server and permanently blocking the source IP. The investigation confirmed the compromise was limited to the single lab server. There was no impact on production systems or business data. The incident was fully remediated within 30 minutes of detection.
Timeline

- **11:00:15:** Wazuh alert for VSFTPD backdoor exploit.
- **11:02:00:** Analyst confirms alert is a true positive.
- **11:05:00:** Metasploitable2 VM isolated from the network.
- **11:10:00:** Attacker IP 192.168.1.100 blocked via CrowdSec.
- **11:30:00:** Incident contained and remediation confirmed.

Recommendations

1. The vulnerable Metasploitable2 VM should be decommissioned and removed from the network permanently.
2. A network-wide vulnerability scan should be conducted to ensure no other hosts are running the vulnerable version of VSFTPD.

---

5. Stakeholder Briefing
Objective: To provide a clear, concise, non-technical summary of the incident for management.
Tools Used: Email Client (simulated)
Output: Briefing Draft
Subject: Summary of Today's Security Simulation

Team,
Today, our security team ran a successful test to validate our defense systems. We simulated an attack on a vulnerable lab server, which was immediately detected by our monitoring tools.
Our response plan worked exactly as designed: we instantly isolated the affected server to prevent any spread and blocked the attacker.
This was a controlled exercise on a non-production system, so there was zero risk to our business operations or data. The test confirms that our security alerts and containment procedures are effective at stopping threats quickly. We are now removing the vulnerable test server.