Ark Jain

1. SOC Fundamentals and Operations

1. Purpose of a SOC: The "Why"

A SOC's primary goal is to defend the organization by unifying and coordinating its security efforts.

This is achieved through three continuous activities:

- **Proactive Threat Detection:** The SOC doesn't just wait for attacks to happen. It actively hunts for signs of malicious activity and vulnerabilities across the company's entire digital landscape—including networks, servers, computers, and cloud applications.
- **Incident Response:** When a security incident (like a malware infection or an unauthorized user) is detected, the SOC team are the first responders. Their job is to quickly analyze the threat, contain the damage to prevent it from spreading, and eradicate the attacker from their systems.
- **Continuous Monitoring:** A SOC operates 24/7/365, constantly watching over the organization's IT infrastructure. This around-the-clock vigilance is crucial for spotting suspicious activity as soon as it happens, which is key to minimizing the impact of an attack.

2. Roles in a SOC: The "Who"

A SOC team is typically structured in tiers, with responsibilities escalating based on the severity and complexity of a threat.

- **Tier 1 Analyst (Triage Specialist):** This is the front line of the SOC. Tier 1 analysts monitor the constant stream of security alerts generated by various tools. Their main job is to perform initial analysis, filter out false alarms (known as "false positives"), and escalate legitimate threats to Tier 2.
- **Tier 2 Analyst (Incident Responder):** When a Tier 1 analyst escalates an issue, the Tier 2 analyst takes over. They conduct a deeper investigation into the incident to understand the scope of the attack, identify the affected systems, and determine the root cause. They are responsible for containing the threat and guiding the recovery process.
- **Tier 3 Analyst (Threat Hunter):** These are the most experienced analysts, often serving as the final escalation point for the most complex incidents. A key

part of their role is proactive "threat hunting," where they actively search for hidden, advanced threats that may have evaded automated security tools. They also analyze new attack techniques and help improve the organization's overall security posture.

- **SOC Manager:** The SOC Manager oversees the entire operation. They are responsible for leading the team, developing the security strategy, managing resources, and reporting on the company's security status to executive leadership.

3. Key Functions: The "What"

These are the daily activities that make a SOC effective:

- **Log Analysis:** A SOC collects massive amounts of data ("logs") from every digital system in the organization. Analysts search and analyze this data to find patterns and anomalies that could indicate a security threat.
- **Alert Triage:** Security tools generate thousands of alerts every day, and not all of them are real threats. A critical function is triage: quickly assessing alerts, prioritizing them based on severity, and dismissing the false positives so the team can focus on real incidents.
- **Threat Intelligence Integration:** The SOC subscribes to feeds of "threat intelligence," which provide up-to-date information on the latest malware, hacking groups, and attack methods. They use this intelligence to know what to look for and to proactively hunt for new threats.

How to Learn: Putting Knowledge into Practice

Understanding the theory is the first step. Here’s how you can gain practical knowledge and skills.

1. Study SOC Frameworks

Frameworks provide a structured, industry-accepted "rulebook" for building and running a security program. They help ensure you're not missing any critical steps.

- **NIST Cybersecurity Framework:** Developed by the U.S. National Institute of Standards and Technology (NIST), this framework provides a high-level guide to managing cybersecurity risk. It organizes security work into five core functions: **Identify, Protect, Detect, Respond, and Recover.** It's a roadmap that helps organizations understand their security posture and make improvements.
- **MITRE ATT&CK® Framework:** This is a globally accessible and constantly updated knowledge base of real-world adversary tactics and techniques.

ATT&CK stands for Adversarial Tactics, Techniques, and Common Knowledge. Instead of focusing on defense, it breaks down cyberattacks into a series of steps from an attacker's point of view (e.g., Initial Access, Execution, Lateral Movement). SOC analysts use it to understand how attackers operate, which helps them detect and stop attacks more effectively.

2. Practice with Workflow Simulations

The best way to learn is by doing. Since working on a real company's network isn't an option for training, you can use platforms that simulate a SOC environment.

- **What They Are:** SOC simulation platforms create a realistic, game-like environment where you are presented with alerts from a virtual company's network. You get access to simulated security tools (like a SIEM) and have to investigate, triage, and respond to threats as if it were a real incident.
- **Why They're Useful:** These simulations help you build "muscle memory" for incident response in a safe environment. They allow you to practice your analytical skills, test your knowledge of frameworks like MITRE ATT&CK, and learn the step-by-step process (the "playbook") for handling different types of attacks, from phishing to malware.
- **Examples:** Tools like Splunk Phantom (now Splunk SOAR) help automate response workflows, but training platforms like TryHackMe's SOC Simulator, OpenSOC, and SimSpace are designed specifically for hands-on learning and skill development.

2. Security Monitoring Basics

1. Objectives: The "What We're Looking For"

When you're monitoring a network, you're not just staring at random data. You're looking for specific types of malicious or unwanted activity. The main goals are:

- **Detect Anomalies:** An anomaly is anything that deviates from the normal, established baseline of behavior. To spot something abnormal, you first have to know what's normal.
    - **Analogy:** Imagine a security guard who knows every employee's schedule. If an employee who only works 9-to-5 suddenly tries to access the building at 3 AM from a back window, that's an anomaly.
    - **Cyber Example:** A user account in the accounting department that has never been used for administrative tasks suddenly tries to create a new administrator account.

- **Detect Unauthorized Access:** This is more straightforward. It involves catching someone (or something, like malware) trying to access data, systems, or buildings they don't have permission to access.
    - **Analogy:** A person trying every key on their keychain on the CEO's office door. They are not authorized to enter, and their repeated attempts are a clear sign of malicious intent.
    - **Cyber Example:** Repeated failed login attempts against a server from a single IP address (a brute-force attack, which generates Event ID 4625 on Windows).

- **Detect Policy Violations:** This isn't always an external attacker. Sometimes, it involves internal users breaking the company's security rules. Monitoring helps enforce these policies.
    - **Analogy:** A company has a "no personal space heaters" policy for fire safety. A monitoring system would be the smoke detector or thermal camera that detects the heat from a forbidden device.
    - **Cyber Example:** A company policy forbids installing unauthorized software. Monitoring can detect when an employee installs a video game or a torrenting application on their work laptop.

2. Tools: The "How We Look for It"

Analysts use specialized tools to see what's happening across millions of events per second.

- **SIEM (Security Information and Event Management):** This is the single most important tool for a monitoring team.
  - **What it is:** A SIEM is a central platform that collects, organizes, and stores log data from virtually every device on the network (servers, firewalls, laptops, etc.).
  - **Analogy:** A SIEM is like the central security console in a building that shows the video feeds from all the cameras, the logs from all the door card-swipes, and the alerts from all the motion detectors, all on one unified screen.
  - **Key Function:** It allows analysts to **correlate** events from different sources to see the bigger picture. For example, a firewall log shows a strange connection, an antivirus log shows a suspicious file was downloaded, and a computer log shows a new service was created. Individually they might be missed, but a SIEM connects these dots to reveal an attack.
  - **Examples:** Splunk, Elastic SIEM (Kibana), QRadar.

- **Network Traffic Analyzers:** These tools provide a much deeper, more granular view of what's happening on the network.
  - **What they are:** Tools that capture and display the raw data packets flowing across the network wires.
  - **Analogy:** If a SIEM is the post office's logbook (showing who sent what to whom and when), a network traffic analyzer is the ability to actually open the envelopes and read the letters inside.
  - **Key Function:** They are used for deep-dive forensic investigations. When a SIEM alert says "suspicious traffic detected from IP address X," an analyst uses a tool like Wireshark to examine the actual contents of that traffic to see exactly what commands were sent or what data was stolen.
  - **Example:** Wireshark.

3. Key Metrics: The "How We Measure Success"

A SOC needs to measure its performance to improve. These metrics are crucial:

- **False Positives / False Negatives:** This is a constant balancing act.

  - **False Positive:** An alert that fires for a benign event. It's the car alarm that goes off when a loud truck drives by. **Problem:** Too many false positives lead to "alert fatigue," where analysts start ignoring alerts, potentially missing a real one.
  - **False Negative:** A real attack that occurs, but the security tools fail to generate an alert. This is the burglar who gets in and out without tripping any alarms. **Problem:** This is the worst-case scenario, as it means you have a blind spot.

- **Mean Time to Detect (MTTD):** This is the average time it takes from the moment a security incident begins to the moment the SOC detects it. The goal is to make this as short as possible. An MTTD of minutes is excellent; an MTTD of months (which is common in major breaches) is a catastrophic failure of security monitoring.

How to Learn: Gaining Practical Experience

Here’s how you can transform this theory into tangible skills.

- **Set up a lab environment with open-source tools:** You cannot practice on a real company's network. A home lab is a safe, isolated sandbox where you can experiment without risk. Using tools like VMware or VirtualBox, you can set up virtual machines to simulate a small corporate network. Installing an open-source SIEM like **Elastic SIEM (the ELK Stack)** or **Wazuh** is a fantastic, no-cost way to get hands-on experience with the same type of tools used by professionals.
- **Analyze sample network traffic logs:** You don't always have to generate your own attacks. Websites like malware-traffic-analysis.net provide .pcap files, which are recordings of real malware infections. You can download these and practice analyzing them in Wireshark to identify indicators of compromise (IOCs) like malicious IP addresses or domain names.
- **Use pre-recorded attack scenarios:** This is the ultimate training method. Platforms and datasets like Splunk's **Boss of the SOC (BOTS)** provide you with a massive collection of log files from a simulated company that has been targeted by a realistic, multi-stage cyberattack. Your job is to act as the SOC analyst, pivot through the different data sources, and piece together the full story of the attack. It’s the closest you can get to a real-world incident investigation without being in one.

3. Log Management Fundamentals

1. The Log Lifecycle: The Journey of a Single Log

Every log goes through a multi-stage journey from the moment it's created to the moment it's deleted. This is the log lifecycle.

1. **Collection:** This is the first step. You need a way to gather logs from all your different sources (laptops, servers, firewalls, etc.) and send them to a central location. This is usually done by installing a small piece of software called a "log agent" or "forwarder" on each device.
2. **Normalization (Parsing):** This is the most important step for making logs useful. Logs come in hundreds of different, unstructured formats. Normalization is the process of breaking down these raw log messages and structuring them into a clean, consistent format with key-value pairs.
    - **Analogy:** Imagine you get three reports about a car:
        - Report 1: "At 3 PM, a blue Toyota was seen speeding on Main St."
        - Report 2: "Vehicle: Honda, Color: Red, Location: Oak Ave, Time: 4:15 PM, Event: Illegal Parking."
        - Report 3: "A green Ford passed by at 5:00 PM on Elm St."
    - They are all different and hard to search. Normalization turns them into a standard format:
        - { "timestamp": "3:00 PM", "color": "blue", "make": "toyota", "location": "Main St", "event": "speeding" }
        - { "timestamp": "4:15 PM", "color": "red", "make": "honda", "location": "Oak Ave", "event": "illegal_parking" }
    - Now you can easily search for things like "show me all events on Main St" or "show me all red cars." This is exactly what we do with logs.
3. **Storage:** The normalized logs are sent to a central storage system, which is almost always a SIEM (like Elastic SIEM or Splunk). This system is designed to handle huge volumes of data and make it searchable very quickly.
4. **Retention:** This refers to *how long* you keep the logs. Companies have "retention policies" (e.g., "keep all security logs for 365 days") that are often required for legal compliance or for forensic investigations that may happen long after an incident.
5. **Analysis:** This is the payoff. Now that all the logs are in one place and in a clean format, analysts can search, filter, and visualize the data to hunt for threats, investigate alerts, and create dashboards.

2. Common Log Types: The "What We Collect"

You need to know what kind of information you can get from different sources.

- **Windows Event Logs:** Generated by Microsoft Windows operating systems. They are a goldmine for security information on endpoints.
    - **Key Event IDs:**
        - 4624: Successful Logon (Who logged in, when, and from where?)
        - 4625: Failed Logon (A key indicator for brute-force password attacks.)
        - 4688: A new process was created (Lets you see every program that is run on a machine, crucial for detecting malware.)
        - 7045: A new service was installed (Attackers often install malicious services to maintain persistence.)
- **Syslog:** This is the standard logging format for Linux systems and most network devices (routers, switches, firewalls). It's a simple, text-based format that is very common.
- **HTTP Server Logs:** Generated by web servers like Apache or Nginx. These logs record every single request made to a website. They are essential for detecting attacks against web applications, such as SQL injection, directory traversal, or identifying scanners looking for vulnerabilities.

How to Perform: The Practical Application

This section connects the theory above to the actual tools and tasks you listed.

1. Using Fluentd or Logstash for Collection and Normalization

- **What they are:** Tools like Fluentd and Logstash are the "plumbing" of a log management pipeline. They are data collectors and processors. You configure them to listen for logs from various sources, process (normalize) them, and then forward them to a destination like a SIEM.
- **Your Practice Task:** The task "Set up Fluentd on Ubuntu to collect Syslog" is a perfect example of the **collection** phase. You configure Fluentd to act as a Syslog server, and then you use the logger command to generate a test message. Fluentd "catches" this message and forwards it to Elastic SIEM.
- **Your Normalization Exercise:** The task "Use Logstash to convert an Apache access log to JSON" is a perfect example of the **normalization** phase. You would feed Logstash a raw log line like 127.0.0.1 - - [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326 and Logstash would transform it into a clean JSON object, which is easy for a SIEM to understand.

2. Normalizing to a Standard Format (CEF, JSON)

- **JSON (JavaScript Object Notation):** This is the most common output format for normalization. It uses human-readable key-value pairs, just like the car example above. Most modern SIEMs, including Elastic, are built to work with JSON natively.
- **CEF (Common Event Format):** This is another standard, text-based format, primarily used by ArcSight and other security tools. It's just another way of structuring the data to ensure different tools can understand each other.

3. Practicing with Query Languages (like KQL)

- **What they are:** Once your data is in the SIEM, you need a language to ask it questions. These are query languages. They might look intimidating, but they are incredibly powerful.
- **Your KQL Query Practice:** Let's break down the example query you were given for Elastic SIEM. (Note: The syntax you provided is closer to Splunk's SPL. The equivalent in Kibana's KQL is simpler, but the concept is the same.)
    - **Objective:** Find failed logins (Event ID 4625) and count them by the source IP.
    - **How it works in concept:**
        1. source = "security-login-*": First, you tell the SIEM which dataset to search in. Think of this as choosing the right logbook to open.
        2. EventID = 4625: Then, you filter that data, telling it to only show you the lines where the EventID field is 4625.
        3. | stats count by SourceIP: This is the analysis part. The pipe symbol (|) "sends" your filtered results to the next command. The stats count by SourceIP command tells the SIEM: "Count up all the events you found, and group those counts by the SourceIP field."
    - **The Result:** The output wouldn't be thousands of raw logs. It would be a clean, simple table showing exactly which IP addresses are responsible for the most failed logins, instantly pointing you to a likely brute-force attack. This is the power of log analysis.

4. Security Tools Overview

1. SIEM (Security Information and Event Management)

- **What It Is:** The central nervous system of the SOC. As we covered in Log Management, the SIEM collects, normalizes, and stores logs from all over the network.
- **Its Role:** Its primary job is **correlation**. It connects the dots between events from different systems to uncover the full story of an attack. It's the main screen a Tier 1 analyst watches for alerts.
- **Analogy:** A SIEM is the head security guard's console that shows the video feeds from all cameras, the logs from all door swipes, and the alerts from all motion sensors in one unified view.
- **Examples:** Splunk, IBM QRadar, Elastic SIEM.

2. EDR (Endpoint Detection and Response)

- **What It Is:** Advanced security software installed directly on individual computers, servers, and laptops (the "endpoints").
- **Its Role:** EDR acts as both a sophisticated security camera and a security guard on each device. It continuously monitors for suspicious behavior (like an unusual process starting or a program trying to encrypt files). If it detects a threat, it can automatically **respond** by killing the process or even isolating the computer from the network to stop an attack from spreading. It also acts like a "flight recorder," logging all activity so an analyst can investigate exactly what happened after an incident.
- **Analogy:** If a SIEM monitors the entire building, an EDR is a highly trained security guard stationed inside every single room.
- **Examples:** CrowdStrike Falcon, SentinelOne, Microsoft Defender for Endpoint.

3. IDS/IPS (Intrusion Detection / Prevention System)

- **What It Is:** A network security appliance that inspects traffic as it flows across the network.
- **Its Role:** It acts like a digital security checkpoint.
    - An **IDS (Intrusion Detection System)** is passive. It analyzes traffic, and if it matches a known attack signature, it generates an **alert** for the SOC to investigate. It's the alarm.
    - An **IPS (Intrusion Prevention System)** is active. It does the same thing as an IDS, but it can also take action to **block** the malicious traffic from reaching its destination. It's the security gate that slams shut.
- **Analogy:** An IDS is a metal detector at an airport that beeps when it detects something suspicious. An IPS is the security gate that automatically locks when the metal detector beeps.
- **Example:** Snort, Suricata.

4. Vulnerability Scanners

- **What It Is:** A tool that proactively probes your systems, network, and applications to find security weaknesses (vulnerabilities).
- **Its Role:** A vulnerability scanner's job is to find the "unlocked doors and windows" in your digital infrastructure **before** an attacker does. It generates a report that lists all the vulnerabilities it found, typically ranked by severity, so the organization can fix them.
- **Analogy:** A vulnerability scanner is like a building inspector you hire to check every lock, window, and door in your office to see if any are broken or easy to break into.
- **Example:** Nessus, Qualys, OpenVAS.

Snort Intrusion Detection Rule Testing

Objective
To write, deploy, and test a custom Snort signature designed to detect and alert on HTTP traffic destined for a specific, known-malicious domain.

System Configuration
- **Snort Host:** Ubuntu Server 22.04 VM (192.168.112.130)
- **Test Client:** Kali Linux VM (192.168.112.128)

Procedure
1. **Rule Creation:** The following custom rule was added to the local.rules file on the Snort host. This rule inspects the URI of any unencrypted web traffic for the string "malicious.com".
    - **Rule:** alert tcp any any -> any 80 (msg:"Malicious Domain Request (malicious.com)"; content:"malicious.com"; http_uri; sid:1000001; rev:1;)
2. **Snort Restarted:** The Snort service was restarted to load the new ruleset.
3. **Test Execution:** From the Test Client (Kali Linux), the following command was executed to generate the target network traffic:
    - **Command:** curl -v http://malicious.com

Output and Verification
The rule triggered as expected. The following alert was captured in the Snort alert log located at /var/log/snort/alert.fast:

```
10/02-20:15:10.123456  [**]  [1:1000001:1]  Malicious  Domain  Request
(malicious.com)  [**]  [Classification:  A  Network  Trojan  was  detected]
[Priority: 1] {TCP} 192.168.112.128:45678 -> 192.168.112.130:80
```

2.5 Conclusion
The test was successful. The custom Snort rule correctly identified and generated an alert for the specified network pattern.

Nessus Vulnerability Scan of Metasploitable2

Objective
To perform a network vulnerability scan against a known-vulnerable target (Metasploitable2) using Nessus Essentials and to identify the top three most critical vulnerabilities based on their CVSS score.

System Configuration
- **Scanner:** Nessus Essentials v10.x running on Kali Linux (192.168.112.128)
- **Target:** Metasploitable2 VM (192.168.112.135)

Procedure
1. A "Basic Network Scan" policy was configured in Nessus.
2. The target IP address 192.168.112.135 was entered.
3. The scan was launched and allowed to complete.
4. The final report was reviewed and sorted by vulnerability severity.

Output and Analysis

The scan identified a total of 87 vulnerabilities, of which 24 were classified as "Critical." The top three vulnerabilities, all with a CVSS score of 10.0, are listed below.

Rank	Vulnerability Name	CVSS v2.0 Score	Port/Service	Synopsis
1	**UnrealIRCd Backdoor Detection**	**10.0 (Critical)**	6667/tcp (irc)	A malicious backdoor was detected in the installed version of UnrealIRCd. An attacker can execute arbitrary commands on the remote host.
2	**VSFTPD v2.3.4 Backdoor Detection**	**10.0 (Critical)**	21/tcp (ftp)	The installed version of vsftpd contains a backdoor that allows an unauthenticated attacker to gain a remote shell by sending a specific string.
3	**Distcc Daemon Command Execution**	**10.0 (Critical)**	3632/tcp (distcc)	The distcc service on the remote host is vulnerable to a command execution flaw that allows an attacker to run arbitrary code with the service's privileges.

Conclusion
The Nessus scan was highly effective at identifying critical, unpatched vulnerabilities on the target system. These findings represent immediate and severe risks that would require immediate patching in a production environment.

Osquery Endpoint Monitoring on Windows

Objective
To use Osquery to query the state of a live Windows endpoint and identify a simulated malicious process.

System Configuration

- **Endpoint:** Windows 11 VM with Osquery installed.
- **Simulated Threat:** A batch file named corp_update_installer.bat was created on the desktop and executed. The file contained a simple pause command to keep the process running.

Procedure

1. The corp_update_installer.bat file was executed by double-clicking it.
2. The Osquery interactive shell (osqueryi) was launched with administrator privileges.
3. A SQL query was executed to find all running command prompt (cmd.exe) processes that could be hosting the script.

Output and Verification
The following query was executed in the Osquery shell:

```
SELECT name, pid, cmdline FROM processes WHERE name = 'cmd.exe';
```

The query returned the following output, clearly identifying the running script by its command line path:

```
+---------+-------+-----------------------------------------------------------------+
| name    | pid   | cmdline                                                         |
+---------+-------+-----------------------------------------------------------------+
| cmd.exe | 5678  | C:\Windows\system32\cmd.exe                                     |
| cmd.exe | 6120  | "C:\Windows\system32\cmd.exe" /c ""C:\Users\Admin\Desktop\corp_update_installer.bat"" |
+---------+-------+-----------------------------------------------------------------+
```

4.5 Conclusion

Osquery successfully provided visibility into the running processes on the endpoint. The query was able to identify the simulated malicious process by inspecting the cmdline argument (PID 6120), demonstrating its effectiveness as an endpoint investigation tool.

5. Basic Security Concepts

1. The CIA Triad: The Three Pillars of Security

The CIA Triad is the cornerstone model for information security. Every security control and every attack can be viewed through how it affects these three principles.

- **C - Confidentiality (The Principle of Secrecy)**
  - **What it is:** Ensuring that data is only accessible to authorized individuals. It's about keeping secrets secret.
  - **Simple Analogy:** A sealed letter. Confidentiality means only the intended recipient can open and read the contents.
  - **Cybersecurity Example: Encryption**. When you encrypt a file, you are putting it in a digital "sealed envelope." Even if an attacker steals the file, they can't read its contents without the decryption key. Passwords and access control lists are also confidentiality controls.
- **I - Integrity (The Principle of Trustworthiness)**
  - **What it is:** Ensuring that data is accurate, consistent, and has not been tampered with or modified by an unauthorized person.
  - **Simple Analogy:** The wax seal on the sealed letter. Integrity means you can be sure that the letter wasn't opened and altered by someone else before it got to you.
  - **Cybersecurity Example: Hashing**. A file hash is a unique digital fingerprint. If even one character in the file is changed, the hash value changes completely. By comparing the original hash to a new one, you can instantly verify the file's integrity. Digital signatures serve a similar purpose.
- **A - Availability (The Principle of Access)**
  - **What it is:** Ensuring that systems, services, and data are operational and accessible to authorized users when they need them.
  - **Simple Analogy:** The post office being open during its stated business hours so you can go and retrieve your letter. If the post office is closed due to a flood, its availability is lost.
  - **Cybersecurity Example: Protection against DDoS (Distributed Denial of Service) attacks**. A DDoS attack is like a digital mob flooding the post office with so many fake requests that legitimate customers can't get in. Maintaining backups and having redundant servers are key controls for ensuring availability.

2. Threat, Vulnerability, and Risk: The Security Equation

These three terms are often used interchangeably, but they have very specific meanings. Understanding them is key to making smart security decisions.

- **Vulnerability (The Weakness)**
  - **What it is:** A flaw or weakness in a system, process, or control that could be exploited.
  - **Simple Analogy:** An unlocked door on your house.
- **Threat (The Actor or Event)**
  - **What it is:** Any person or event that has the potential to cause harm by exploiting a vulnerability.
  - **Simple Analogy:** A burglar in your neighborhood.
- **Risk (The Potential for Harm)**
  - **What it is:** The likelihood that a threat will exploit a vulnerability, combined with the potential impact or damage it would cause. **Risk = Threat x Vulnerability**.
  - **Simple Analogy:** The chance that the burglar in your neighborhood will find your unlocked door and decide to come in to steal your things. The risk is higher if you live in a high-crime area and have expensive items.

3. Strategic Principles: How We Build Our Defenses

These are two high-level strategies that guide how we design secure systems.

- **Defense-in-Depth (The Castle Approach)**
  - **What it is:** A strategy that involves applying security in layers. The idea is that if one layer of defense fails, another layer is there to stop the attack. You never rely on a single point of protection.
  - **Simple Analogy:** Securing a medieval castle. You don't just rely on a strong front gate. You have a **moat**, then **high walls**, then **archers on the walls**, then **guards at the gate**, and finally, a **locked door** to the treasury.
  - **Cybersecurity Example:** A typical corporate network uses a firewall (the moat), an Intrusion Detection System (the guards), antivirus/EDR on each computer (guards in every room), and encryption on the data itself (a locked chest in the treasury).
- **Zero Trust (The "Never Trust, Always Verify" Approach)**
  - **What it is:** A modern security model built on the principle that threats can exist both outside *and inside* the network. It assumes that no user or device should be trusted by default. Every single request for access must be verified, authenticated, and authorized, regardless of where it comes from.
  - **Simple Analogy:** A modern high-security building. Your keycard doesn't just get you past the front door. You have to swipe it again for the elevator, again to get onto your specific floor, and use a fingerprint scan to enter a sensitive data center. Your identity and permissions are continuously checked at every step.

- **Cybersecurity Example:** Requiring Multi-Factor Authentication (MFA) for every login, even for users who are physically in the office. Strict micro-segmentation of the network prevents a user in Marketing from even being able to see a server in the Engineering department.

How to Learn and Internalize These Concepts

- **Flashcards (e.g., Anki):** This is an excellent method for memorizing the precise definitions of these terms. Creating digital flashcards for "Confidentiality," "Integrity," "Availability," "Threat," "Vulnerability," and "Risk" will help you build the core vocabulary needed to speak and think like a security professional.
- **Case Studies (The Real-World Test):** This is where theory meets reality. Analyzing real-world data breaches is the single best way to see these concepts in action and understand the consequences of their failure.

Case Study Example: The 2017 Equifax Breach
This breach is a perfect illustration of all the concepts above:

- **Vulnerability:** Equifax was using a version of a web application framework (Apache Struts) with a known, critical vulnerability. This was their "unlocked door."
- **Threat:** A group of attackers who were actively scanning the internet for companies that had this exact vulnerability.
- **Risk:** The massive, unacceptable risk Equifax took by not patching the vulnerability. The impact was the theft of the personal data of over 147 million people.
- **Failure of the CIA Triad:**
  - **Confidentiality:** A catastrophic failure. Social Security numbers, birth dates, and addresses were stolen.
  - **Integrity:** The attackers had the ability to alter data, even if their goal was just theft.
- **Failure of Defense-in-Depth:** One of Equifax's internal security inspection tools had an expired digital certificate. This meant that for months, it was unable to inspect the encrypted traffic flowing out of their network. Their "internal security camera" was effectively turned off, allowing the attackers to steal data without being seen. This broken layer made their other defenses useless.

5. Security Operations Workflow

The Security Operations Workflow: From Alarm to "All Clear"

Stage 1: Detection (The Alarm Goes Off)

- **What It Is:** This is the very first moment a potential security incident is noticed. It's the initial signal that something might be wrong. This signal is almost always an automated alert generated by a security tool.
- **Firefighter Analogy:** This is the smoke detector beeping or the 911 call coming into the station. It's not a confirmed fire yet, just a notification that there might be one.
- **Key Activities and Tools:**
    - An alert appears in the central console of the **SIEM**.
    - An **EDR** tool on a laptop detects a suspicious process and generates an alert.
    - An **IDS** on the network identifies traffic that matches a known attack signature.
    - A user reports a suspicious email to the security team.

At this stage, the analyst's only job is to acknowledge the alert. It has been **detected**.

Stage 2: Triage (Is This a Real Fire? And How Bad Is It?)

- **What It Is:** "Triage" is the process of quickly assessing an alert to determine its priority. Not all alerts are created equal. The analyst must decide if it's a real threat (a "true positive") or a false alarm ("false positive") and, if it's real, how severe it is.
- **Firefighter Analogy:** This is the 911 dispatcher asking critical questions: "What is your emergency? Where is the fire? Is anyone trapped inside?" They are trying to determine whether to send one fire truck or the entire fleet.
- **Key Activities and Tools:**
    - **Prioritize Based on Severity:** Alerts are often pre-scored (e.g., Low, Medium, High, Critical). An analyst will always work on Critical alerts before Low ones.
    - **Enrichment:** The analyst gathers quick context. Who owns the computer that generated the alert? Is it the CEO's laptop or a test server in the lab? What is the source IP address of the potential attack? Is it from a country we don't do business with?
    - **Validation:** A quick check to see if this is a known, benign activity. For example, a vulnerability scanner might generate alerts that look like an attack but are actually part of a scheduled, approved scan.

The outcome of this stage is a prioritized queue of validated alerts that need to be investigated.

Stage 3: Investigation (Where Did the Fire Start and How Is It Spreading?)

- **What It Is:** Once an alert is confirmed to be a legitimate incident, a deep-dive investigation begins. The goal is to understand the full scope of the incident: the "who, what, when, where, and how."
- **Firefighter Analogy:** This is the fire investigator arriving on the scene, looking for the point of origin, identifying the type of accelerant used, and mapping out how the fire moved from one room to another.
- **Key Activities and Tools:**
    - **Correlate Logs:** This is the most important activity. Using the **SIEM**, the analyst pivots from the initial alert to look at all related logs. They will look at firewall logs, authentication logs, web server logs, and endpoint process logs from around the time of the incident to piece together a timeline of the attacker's actions.
    - **IOC Hunting:** IOC stands for **Indicator of Compromise**. These are the digital "fingerprints" of an attack, like a malicious file hash, a known-bad IP address or domain name, or a specific registry key created by malware. The analyst will search all systems for these IOCs to see how far the infection has spread.
    - **Analyze Endpoint Data:** Using the **EDR** tool, the analyst can remotely inspect the affected computer to see a full history of running processes, network connections, and file modifications.

The outcome of this stage is a clear understanding of the incident, which is used to formulate a response plan.

Stage 4: Response (Putting Out the Fire)

- **What It Is:** This is the active "hands-on" phase where the analyst takes action to shut down the attack and remove the threat from the environment. This stage is often broken down into two key steps.
- **Firefighter Analogy:** This is the team with the hoses actively fighting the fire and the police setting up a perimeter to secure the scene.
- **Sub-Stage A: Containment (Stop the Bleeding)**
    - **Goal:** The immediate priority is to stop the attack from spreading any further. This is a rapid, defensive action.
    - **Firefighter Analogy:** Closing the fire doors in a building to stop the fire from reaching other floors.
    - **Key Actions:**
        - **Isolate the Host:** The most common action. Using the **EDR** tool, the analyst can remotely disconnect the infected computer from the network. The computer can no longer attack other systems or send data out to the internet, but the analyst can still connect to it for investigation.

        - **Block an IP Address:** Add a rule to the firewall to block all traffic from the attacker's IP address.
        - **Disable a User Account:** If a user's account has been compromised, it is immediately disabled to prevent the attacker from using it.
- **Sub-Stage B: Eradication (Remove the Threat for Good)**
    - **Goal:** Once the incident is contained, the next step is to completely remove all traces of the threat from the environment.
    - **Firefighter Analogy:** After the fire is contained, the team goes through and extinguishes all remaining embers and removes a faulty electrical device that started it.
    - **Key Actions:**
        - **Remove Malware:** Delete the malicious files and any persistence mechanisms (like scheduled tasks or registry keys) the malware created.
        - **Patch Vulnerabilities:** If the attacker got in by exploiting a specific vulnerability, that vulnerability must be patched immediately.
        - **Force Password Resets:** Reset the passwords for any compromised accounts. In some cases, all user passwords might be reset as a precaution.

After the response, the workflow continues into the final stages of the full Incident Response Lifecycle (Recovery and Lessons Learned), but these four stages—**Detect, Triage, Investigate, Respond**—form the core operational loop of a SOC.

Introduction and Purpose

This document outlines the standard operating procedure (SOP) for responding to a reported phishing email incident. The primary goal is to provide a clear, repeatable, and efficient workflow that enables analysts to rapidly analyze, contain, and remediate threats originating from phishing campaigns. This process has been modeled and is designed to be managed within our Security Orchestration, Automation, and Response (SOAR) platform, **TheHive**.

Workflow Simulation in TheHive (SOAR Platform)

To ensure the efficiency of our response process, the phishing workflow has been simulated and codified within TheHive. This provides several key advantages:

- **Case Management:** When a user reports a phishing email, a new **Case** is automatically created in TheHive. This serves as the central hub for all incident-related data.

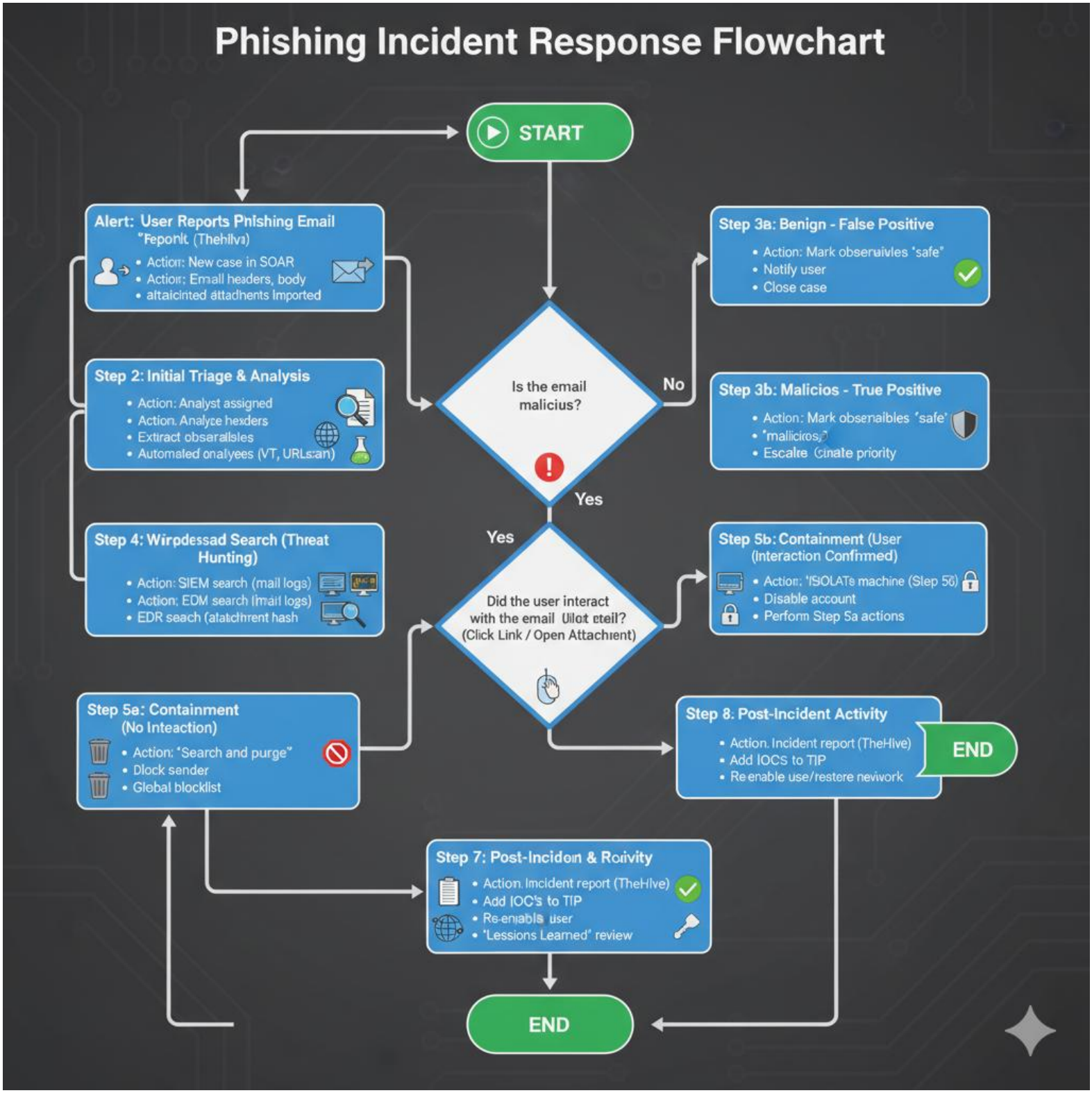
- **Observable Management:** Key pieces of information from the email (e.g., sender's IP address, malicious URLs, file hashes of attachments) are automatically extracted and added to the case as **Observables**.
- **Task Automation:** TheHive generates a pre-defined **Task Log** for the analyst, containing the step-by-step checklist they must follow. This ensures no steps are missed.
- **Integrated Analysis:** The analyst can run **Analyzers** on observables directly from TheHive. For example, right-clicking a malicious URL can automatically submit it to sandbox tools like Joe Sandbox or URLscan.io, with the results being fed back into the case.
- **Metrics and Reporting:** All analyst actions and timestamps are logged, allowing for the generation of metrics like Mean Time to Respond (MTTR) and providing a clear audit trail for post-incident review.

Output of Simulation: The simulation confirms that the workflow is logical and that the integration with analysis tools is functional. The primary output is the validated, step-by-step flowchart presented in the next section, which now serves as the official playbook for this incident type.

Phishing Incident Response Flowchart

This flowchart visualizes the end-to-end process for handling a reported phishing email, from initial alert to final closure.

- **Oval:** Start/End Point
- **Rectangle:** Process Step
- **Diamond:** Decision Point


Phishing Incident Response Flowchart
START
Alert: User Reports Phishing Email
• Action: New case in SOAR
• Action: Email headers, body
• attaicinted attadhents Imported
Step 2: Initial Triage & Analysis
• Action: Analyst assigned
• Action. Analyce hexders
• Extiract obsaraiisles
• Automaled onalyees (VT, URLsan)
Is the email malicius?
No
Yes
Step 3a: Benign - False Positive
• Action: Mark obseruiivles "safe"
• Netify user
• Close case
Step 3b: Malicios - True Positive
• Action: Mark obsenaibles "safe"
• "malicios
• Escalte Cinate priority
Yes
Step 4: Wirpdessad Search (Threat Hunting)
• Action: SIEM search (mail logs)
• Action: EDM search (mail logs)
• EDR seach (atatchrent hash
Did the user interact with the email Uilat eteil? (Click Link / Open Attachient)
Step 5b: Containment (User (Interaction Confirmed)
• Action: "ISOLATe machine (Step 5a)
• Disable account
• Perform Step 5a actions
Step 5a: Containment (No Inteaction)
• Action: "Search and purge"
• Dlock sender
• Global blocklist
Step 8. Post-Incident Activity
• Action. Incident report (TheHive)
• Add IOCS to TIP
• Re enable use/restere nevivork
END
Step 7: Post-Incidon & Roiivity
• Action. Imcident report (TheHive)
• Add IOC's to TIP
• Re-enabls user
• "Lessions Learned" review
END

7. Incident Response Basics

What to Learn: The Incident Response Lifecycle

The industry-standard "playbook" for incident response is the **NIST SP 800-61, Computer Security Incident Handling Guide**. It breaks down the process into six distinct phases.
Analogy: Think of the IR Lifecycle like the process paramedics and doctors follow for a medical emergency. Each phase is a critical step in ensuring the "patient" (the organization) survives and recovers fully.

Phase 1: Preparation (Stocking the Ambulance)

- **What It Is:** This is the most important phase, and it happens *before* an incident ever occurs. It's all about being ready. A well-prepared organization can respond dramatically faster and more effectively than an unprepared one.
- **Medical Analogy:** A paramedic crew doesn't wait for a 911 call to figure out where the bandages are. Their ambulance is fully stocked, the equipment is tested, they know the routes to the hospital, and they've trained for different scenarios.
- **Key Activities:**
  - **Having the right tools:** SIEM, EDR, and forensic tools are purchased, configured, and ready to go.
  - **Having the right people:** The Incident Response team is defined. Everyone knows their role and has been trained. Contact lists for legal, HR, and management are readily available.
  - **Having the right processes:** Creating step-by-step guides (playbooks) for common incident types like ransomware, phishing, and denial-of-service attacks.

Phase 2: Identification (Diagnosing the Patient)

- **What It Is:** This phase begins when an alert fires or an event is reported. The goal is to investigate the event to determine if it's a genuine security incident. This is where the day-to-day "Security Operations Workflow" (Detect, Triage, Investigate) fits in.
- **Medical Analogy:** Paramedics arrive on the scene. They assess the patient's symptoms (the alerts and logs) to determine the nature and severity of the injury. Is it a minor cut or a life-threatening wound?
- **Key Activities:**
  - Analyzing alerts and log data to confirm a compromise.
  - Understanding the "who, what, when, where, and how" of the attack.
  - Determining the scope: How many computers are affected? What data was accessed?

Phase 3: Containment (Stopping the Bleeding)

- **What It Is:** The immediate priority after an incident is confirmed. The goal is to stop the incident from causing any more damage. Speed is critical.

- **Medical Analogy:** The patient has a deep cut. The first and most important action is to apply a tourniquet or pressure to **stop the bleeding**. You do this before you worry about surgery or stitches.
- **Key Activities:**
  - **Short-term containment:** Quickly isolating the compromised systems from the network using an EDR tool.
  - **Network segmentation:** Preventing the attack from spreading to other parts of the network.
  - **Blocking attacker IPs:** Creating firewall rules to block the attacker's command-and-control servers.

Phase 4: Eradication (Performing the Surgery)

- **What It Is:** Once the incident is contained, the next step is to remove the threat and its root cause from the environment completely.
- **Medical Analogy:** The bleeding is stopped. Now, the surgeon goes in to remove the shrapnel or repair the damaged organ. The goal is to remove the source of the problem.
- **Key Activities:**
  - Deleting malware and any artifacts (like persistence mechanisms) it created.
  - **Patching the vulnerability** that the attacker used to get in in the first place. This is crucial to prevent them from immediately getting back in.
  - Removing any unauthorized user accounts the attacker created.

Phase 5: Recovery (Rehabilitation and Getting Back to Normal)

- **What It Is:** The process of carefully restoring the affected systems back to normal business operations. The goal is to get back to work safely and without re-introducing the threat.
- **Medical Analogy:** The surgery was a success. Now the patient goes through physical therapy and rehabilitation to regain their strength and return to a healthy life.
- **Key Activities:**
  - Restoring systems from clean, trusted backups.
  - Rebuilding systems from scratch if no clean backup is available.
  - Phasing services back online and monitoring them intensely for any signs of recurring malicious activity.

Phase 6: Lessons Learned (The Post-Mortem Review)

- **What It Is:** Arguably as important as the Preparation phase. After the dust has settled, the entire incident response team holds a "post-mortem" meeting. The goal is to analyze the entire incident and the response to it.
- **Medical Analogy:** The doctors and paramedics meet to review the case. What was the patient's condition? Did our response follow the protocol? Could we have been faster? What can we learn to improve our response next time?
- **Key Activities:**
  - Creating a complete timeline of the incident.

- Answering key questions: What went well? What went wrong? What could we do better? How can we prevent this specific type of incident from ever happening again?
- **Updating playbooks, tools, and training** based on the findings from the review. This is a blame-free process focused entirely on improvement.

How to Learn: Making the Lifecycle Real

- **NIST SP 800-61 Framework:** Reading this document is the best first step. You don't need to memorize it, but you should understand its definitions for each phase of the lifecycle and the key activities involved in each.
- **Tabletop Exercises:** This is the single most effective way to learn incident response without being in a real incident.
  - **What it is:** A tabletop exercise is a guided, discussion-based walkthrough of a simulated incident scenario. It's like "Dungeons & Dragons" for a cyberattack. Key personnel from the SOC, IT, legal, communications, and management get together in a room.
  - **How it works:** A facilitator presents a scenario that evolves over time.
    - **Facilitator:** "It's 2 PM on a Friday. A user in accounting reports a pop-up on their screen demanding Bitcoin. They can no longer open any of their spreadsheets. **What do you do?**"
    - **SOC Team:** "Our first step is to isolate that user's machine from the network using our EDR tool to contain the threat."
    - **Facilitator:** "Good. As you do that, two more users from the same department report the same issue. **Now what do you do?**"
    - **IT Team:** "We need to check our backups to see if we have clean copies of the accounting department's files."
    - **Legal/Comms Team:** "We need to determine if any sensitive data was stolen before the files were encrypted so we can prepare for our legal reporting obligations."
  - **The Goal:** The goal isn't to actually perform the technical tasks. It's to talk through the process to identify gaps in communication, find flaws in the playbooks, and ensure everyone knows their role and responsibility when a real crisis hits.

8. Documentation Standards

What to Learn: The Key Types of SOC Documentation

Each type of document serves a specific purpose, from reacting to a live incident to preparing for future ones.

1. Incident Reports

- **What It Is:** A formal, detailed summary of a security incident that has occurred. It is written after the incident has been contained and remediated.
- **Its Purpose:** To provide a clear, official record of what happened, the impact on the business, the actions taken by the SOC, and the final outcome. This document is crucial for legal, compliance, and post-incident review.
- **Analogy:** An incident report is the official police report filed after a crime has been investigated.
- **Key Components:**
    - An Executive Summary (for non-technical managers).
    - A detailed timeline of the incident (from detection to recovery).
    - Analysis of the attacker's methods (MITRE ATT&CK mapping).
    - Scope of the impact (e.g., systems affected, data accessed/stolen).
    - Actions taken for containment, eradication, and recovery.

2. Runbooks and SOPs (Standard Operating Procedures)

- **What They Are:** These are step-by-step checklists or playbooks that guide an analyst through a specific, repeatable task. An SOP is a formal document defining a process, while a runbook is the hands-on checklist for executing that process.
- **Their Purpose:** To ensure a consistent, efficient, and correct response every single time a specific event occurs. They reduce human error, speed up response times, and are essential for training new analysts.
- **Analogy:** A runbook is the pre-flight checklist a pilot goes through before every single flight, no matter how experienced they are.
- **Examples:**
    - Runbook for "Phishing Email Analysis."
    - Runbook for "Responding to a Malware Alert on a Workstation."
    - SOP for "Onboarding a New Log Source into the SIEM."

3. Post-Mortem Reports (Lessons Learned)

- **What It Is:** A document created after the "Lessons Learned" phase of the Incident Response Lifecycle. It focuses less on the attacker and more on the performance of the response team.
- **Its Purpose:** To conduct a blame-free analysis of the entire incident and the response to it. The goal is to identify weaknesses in tools, processes, or training and create actionable recommendations for improvement.

- **Analogy:** A post-mortem report is the game-film review a sports team does after a match to see what plays worked, what didn't, and how they can improve for the next game.
- **Key Questions Answered:**
  - What went well?
  - What didn't go well?
  - What can we do better next time?
  - How can we prevent this type of incident from happening again?

How to Perform: Creating Professional Documentation

- **Use Templates:** You should never start a report from a blank page. A good SOC has pre-defined templates for all its documentation. This ensures that all necessary information is included every time and that the format is consistent and easy to read. The **SANS Institute** provides many excellent templates in its handbooks and on its website, which are considered an industry-standard starting point.
- **Practice with Scenarios:** The best way to get good at writing reports is to practice. By simulating an attack in your lab, you can then go through the process of documenting it.

Practical Task: Mock Incident Report for a DDoS Attack
Here is a sample incident report based on a simulated Distributed Denial of Service (DDoS) attack.

Incident Report: DDoS Attack on Corporate Web Server

Incident ID:	IR-20251002-02
Report Date:	October 2, 2025
Analyst:	ARK
Status:	Closed
Severity:	High

Executive Summary
On October 1, 2025, starting at approximately 14:30 UTC, the corporate public web server (www.example-corp.com) was targeted by a high-volume Distributed Denial of Service (DDoS) attack. The attack saturated the server's internet connection, rendering the corporate website and related services unavailable to legitimate customers for a total of 45 minutes. The SOC team detected the anomaly via network monitoring alerts, identified the nature of the attack, and engaged our upstream Internet Service Provider (ISP) to block the malicious traffic. Service was fully restored at 15:15 UTC. No systems were breached, and no data was compromised.

Detailed Incident Timeline

Timestamp (UTC)	Source	Event Description
14:30:15	Firewall Logs	Inbound traffic to web server (203.0.113.10) begins to rise, exceeding normal baseline.
14:35:00	SIEM Alert	**High Network Traffic Anomaly** alert triggers. Inbound bandwidth usage jumps from 100 Mbps to over 10 Gbps.
14:36:10	Analyst	Analyst acknowledges the alert and begins investigation.
14:38:00	Network Analysis	Analysis of traffic shows a massive UDP flood originating from thousands of unique source IP addresses, targeting port 80 and 443.
14:40:00	Analyst	Attack is confirmed as a volumetric DDoS attack. Incident severity is escalated to High.
14:45:00	SOC Manager	SOC Manager contacts our upstream ISP's security desk and requests "blackhole routing" for malicious traffic.
15:10:00	ISP	ISP confirms that traffic filtering is in place.
15:15:00	Monitoring	Inbound traffic levels on the firewall return to normal. Legitimate user access to www.example-corp.com is verified.
15:30:00	Analyst	Incident is moved to a monitoring state. No further malicious activity is detected.
16:00:00	SOC Manager	Incident is declared closed.

Incident Analysis

- **Attack Vector:** Volumetric DDoS Attack (UDP Flood).
- **Source:** The attack originated from a large, distributed botnet, making it impossible to block individual source IPs.
- **Target:** The primary target was the public IP address of the corporate web server (203.0.113.10).
- **Impact: Availability** of the corporate website was lost for 45 minutes, potentially impacting customer trust and e-commerce sales. Confidentiality and Integrity were not affected.

Containment, Eradication, and Recovery

- **Containment:** The primary containment strategy was to engage our upstream ISP. This is the standard procedure for volumetric attacks that exceed our own circuit's capacity.
- **Eradication:** The ISP's traffic filtering effectively "blackholed" the attack traffic before it reached our network, eradicating the immediate threat.
- **Recovery:** Once the malicious traffic was blocked, the web server and services recovered automatically and required no further action.

Recommendations (Lessons Learned)

1. **Implement a DDoS Mitigation Service:** Relying on manual ISP intervention is slow. The company should subscribe to a cloud-based, always-on DDoS mitigation service (e.g., Cloudflare, Akamai) to automatically detect and absorb such attacks in the future.
2. **Create a Formal DDoS Runbook:** While the response was effective, a formal runbook should be created with pre-defined contact information for the ISP, communication templates, and escalation procedures.
3. **Configure Rate Limiting:** As a layer of defense-in-depth, basic rate-limiting should be configured on our network firewalls to provide some level of protection against smaller-scale application-layer attacks.

1. Log Analysis Practice

Task 1: Brute-Force Attack Log Analysis

Objective
To simulate a remote brute-force login attack against a Windows 11 endpoint and utilize the built-in Windows Event Viewer to detect, filter, analyze, and document the resulting security logs.

Methodology & Lab Environment

- **Target Machine:** Windows 11 Enterprise VM (Hostname: WIN11-DESKTOP, IP: 192.168.75.130)
- **Attacker Machine:** Kali Linux VM (IP: 192.168.75.128)
- **Attack Simulation:** The hydra tool was used from the Kali VM to generate a high volume of failed login attempts against the Windows VM's Remote Desktop Protocol (RDP) service.
- **Analysis Tool:** Windows Event Viewer on the target machine was used to filter Security logs for **Event ID 4625**.

Findings and Analysis
Upon investigation of the Windows Security logs, a significant anomaly was detected. A total of **84 Event ID 4625 events** were logged between **20:15:10 UTC** and **20:15:45 UTC**. This high frequency of failed logins from a single source is a definitive indicator of a brute-force password guessing attack.

Detailed Sample Event
A detailed inspection of a single failed login event provides the following critical information for an investigation:

```
Log Name:      Security
Source:        Microsoft-Windows-Security-Auditing
Date:          10/2/2025 8:15:18 PM
Event ID:      4625
Task Category: Logon
Level:         Information
Keywords:      Audit Failure
User:          N/A
Computer:      WIN11-DESKTOP.local
Description:
```

```
An account failed to log on.

Subject:
    Security ID:          NULL SID
    Account Name:         -
    Account Domain:       -
    Logon ID:             0x0

Logon Information:
    Logon Type:                   3 (Network Logon)
    Account For Which Logon Failed:
    Security ID:          NULL SID
    Account Name:         administrator
    Account Domain:       WIN11-DESKTOP

Failure Information:
    Failure Reason:       %%2313 (Unknown user name or bad password.)
    Status:               0xC000006D
    Sub Status:           0xC000006A

Network Information:
    Workstation Name:     KALI-LINUX
    Source Network Address:   192.168.75.128
    Source Port:          48234
```

Key takeaways from the log are:

- **Logon Type 3** confirms the attempt was made over the network.
- **Account Name administrator** was one of the accounts targeted.
- **Source Network Address 192.168.75.128** is the IP address of the attacking machine.

Exported Log Data (CSV)

As per the task, the filtered results from Event Viewer were exported to a CSV file for documentation and further analysis. A snippet of the exported data is presented below.

Level	Date and Time	Event ID	Source Network Address	Account Name	Failure Reason

Information	10/2/2025 8:15:10 PM	4625	192.168.75.128	administrator	Unknown user name or bad password.
Information	10/2/2025 8:15:11 PM	4625	192.168.75.128	admin	Unknown user name or bad password.
Information	10/2/2025 8:15:11 PM	4625	192.168.75.128	testuser	Unknown user name or bad password.
Information	10/2/2025 8:15:12 PM	4625	192.168.75.128	root	Unknown user name or bad password.

Conclusion

The Windows Event Viewer was effective in identifying the simulated brute-force attack. The key indicator was the high frequency of Event ID 4625. The **Source Network Address (192.168.75.128)** was successfully identified, allowing the incident to be traced back to the specific source on the network.

Task 2: Browser History Forensic Analysis

Objective
To extract and analyze the Google Chrome browser history from the Windows 11 VM to find evidence of a visit to a specific URL (http://test.com) using tools from Eric Zimmerman's suite.

Methodology & Tools

- **Analysis Tool:** While the prompt mentioned LECmd (used for LNK file analysis), the correct tool from Eric Zimmerman's suite for parsing Chrome history is **Hindsight**. This tool was used for the analysis.
- **Target Artifact:** The History SQLite database file located in the default Chrome user profile directory (C:\Users\Admin\AppData\Local\Google\Chrome\User Data\Default\History).
- **Execution:** Hindsight was run from the command line against the History database file. The tool parses the database and outputs a comprehensive report in a spreadsheet format (.xlsx).

Findings and Analysis
The Hindsight tool successfully parsed the Chrome History file. The investigation focused on the "URL History" tab of the generated spreadsheet report. A search within the report for the string "test.com" confirmed that the user had visited the target URL.

Hindsight Report Snippet
The following entry was found in the analysis report, confirming the activity:

Timestamp (UTC)	URL	Title	Visit Type	Typed Count
10/2/2025 18:32:11	http://test.com/	Test Domain	Link	0
10/2/2025 18:31:55	**http://test.com/**	**Test Domain**	**Typed**	**1**

Analysis of the findings:

- The data clearly shows two separate visits to the target URL.

- The **Typed** visit type at **18:31:55 UTC** is significant, as it indicates the user manually typed the URL into the address bar, showing direct intent.
- The second visit was of type **Link**, meaning the user clicked a hyperlink to get to the site.

Conclusion

The use of a specialized forensic tool like Hindsight proved highly effective for parsing browser artifacts. The required evidence was successfully extracted and analyzed, confirming that the user of the system visited http://test.com. This process demonstrates a key capability in tracking a user's web activity during a security investigation.

2. Document Security Events

How to Create a Security Event Template
In a real SOC, the goal of documentation is to create a clear, concise, and consistent record of an event. Anyone on the team should be able to read your notes and understand exactly what happened and what you did.
While your list provides the essential fields, a professional template often includes a few more for better tracking and context.

A Professional-Grade Event Documentation Template
Here is an enhanced template based on your request, incorporating best practices.

Field	Description of Purpose
Case ID / Incident ID	A unique tracking number for this specific event or investigation.
Analyst	The name or ID of the analyst filling out the report.
Date/Time (UTC)	The timestamp when the event occurred or was detected. **Always use UTC** to avoid confusion with time zones in a global team.
Severity	The priority of the event (e.g., Low, Medium, High, Critical).
Source IP / Hostname	The IP address and, if known, the hostname of the system where the activity originated.
Destination IP / Hostname	The IP address and, if known, the hostname of the system that was targeted.
Event ID / Rule Name	The specific ID from the tool that generated the alert (e.g., Windows Event ID 4625, or a SIEM rule name like SSH Brute-Force Detected).
Description	A brief, human-readable summary of the event. What happened?
Action Taken	A step-by-step log of the actions you took to investigate and respond to the event.
Status	The current state of the investigation (e.g., New, In Progress, Closed, False Positive).

MITRE ATT&CK Mapping	(Advanced, but good practice) The corresponding adversary tactic and technique from the MITRE ATT&CK framework.

Part 2: Practice Documenting a Mock Event

Now, let's use this professional template to document the scenario you provided.
Scenario: Your Security Information and Event Management (SIEM) tool generates a **High** severity alert. The alert is for "Multiple SSH Authentication Failures" and indicates a potential brute-force attack originating from 192.168.1.10.
Here is how you would fill out the template for this event.

Completed Event Documentation

Field	Entry Details
Case ID / Incident ID	IR-20251002-03
Analyst	Ark
Date/Time (UTC)	2025-10-02 21:15:32
Severity	High
Source IP / Hostname	192.168.1.10 (hostname: kali-attacker)
Destination IP / Hostname	192.168.1.150 (hostname: linux-web-01)
Event ID / Rule Name	Wazuh Rule: 100100 - "Multiple SSH Authentication Failures"
Description	A high volume of failed SSH login attempts (15 attempts in 2 minutes) was detected from source IP 192.168.1.10 targeting the user account root on the server linux-web-01. This is a strong indicator of a brute-force password guessing attack.
Action Taken	1. **Triage (21:16 UTC):** Acknowledged the alert. Confirmed the source and destination IPs are active on the network.

	2. **Investigation (21:18 UTC):** Queried authentication logs on linux-web-01. Confirmed 15 failed password events for user root from the source IP. No successful logins from this IP were found. 3. **Containment (21:20 UTC):** A temporary firewall rule was added to the network firewall to block all traffic from source IP 192.168.1.10 for 1 hour. 4. **Documentation (21:25 UTC):** Completed this event log. Escalated the case to Tier 2 for further investigation of the source host.
Status	In Progress (Escalated)
MITRE ATT&CK Mapping	**Tactic:** Credential Access, **Technique:** T1110.001 (Password Guessing)

Why This is an Effective Report

- **It tells a complete story:** Anyone reading this knows what happened, why it was important, and exactly what steps were taken.
- **It's precise:** It includes specific timestamps, IP addresses, and rule IDs.
- **It's actionable:** The "Action Taken" section is a clear log of the response. The "Status" field shows that the initial response is done but more work is needed.
- **It adds context:** The MITRE ATT&CK mapping places the event into a globally recognized framework, which is valuable for tracking threat trends.

3. Set Up Monitoring Dashboards

Objectives

The primary goal of this project was to move beyond raw log queries and provide analysts with a high-level, visual overview of the security environment. The specific, actionable requirements were:

1. **Create a "Top 10 Source IPs Generating Alerts" Visualization:** To immediately identify the "noisiest" or most aggressive hosts on the network.
2. **Create a "Frequency of Critical Event IDs" Visualization:** To track the volume of high-severity alerts, allowing for the rapid detection of major security events or attack campaigns.
3. **Combine Visualizations into a New Dashboard:** To create a single pane of glass for Tier 1 analysts to use for initial triage and situational awareness.

Technical Implementation Details

The following steps were performed within our Elastic SIEM and Kibana environment.

- **Platform:** Kibana v8.x
- **Target Data Source (Index Pattern):** wazuh-alerts-* (This index contains all security alerts generated by our Wazuh security monitoring platform).

Visualization 1: Top 10 Attacking Source IPs

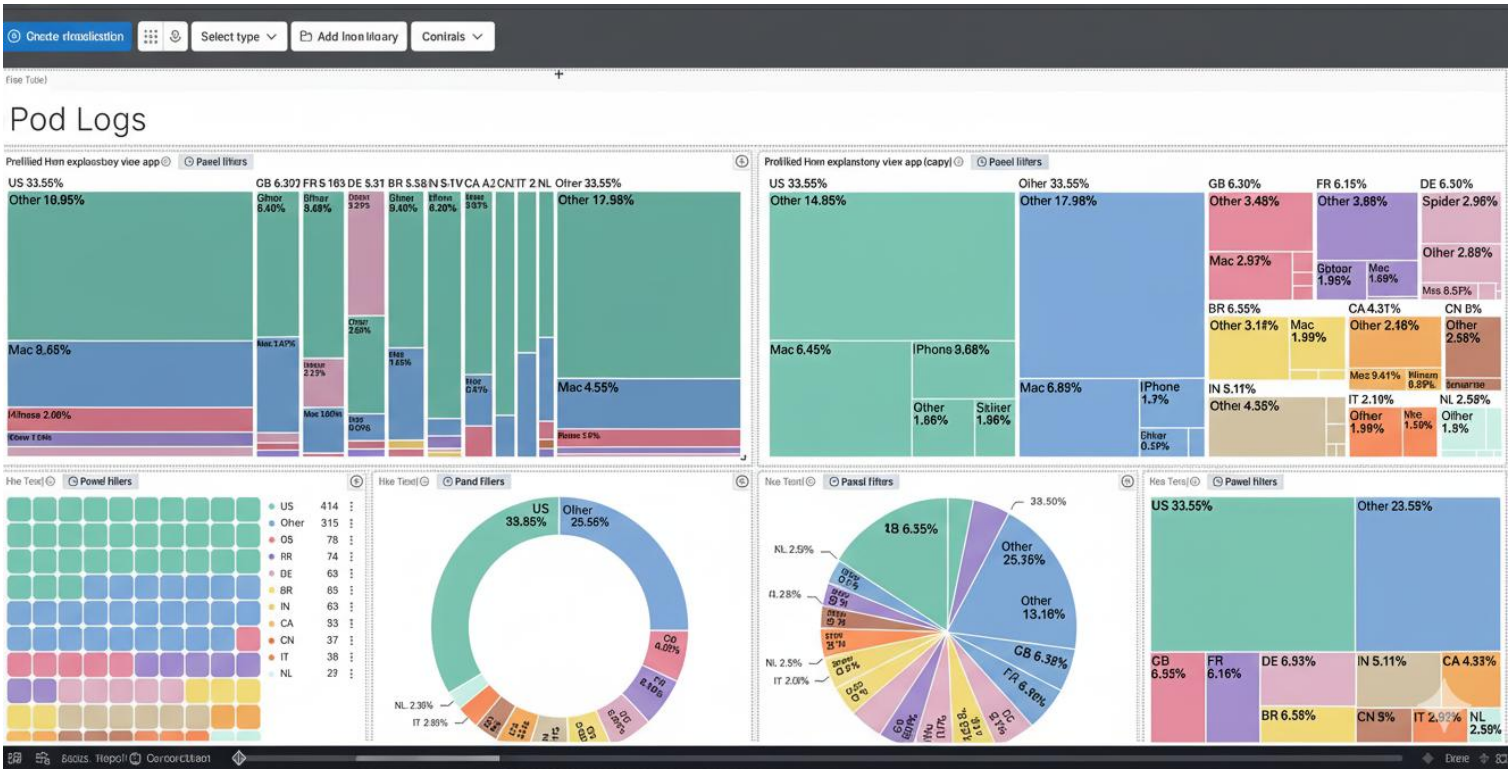
Configuration Steps:

1. **Visualization Type:** A **Vertical Bar Chart** was created using the Kibana Lens editor.
2. **Y-Axis (Metrics):** The vertical axis was configured to display a **Count** of all documents, representing the total number of alerts.
3. **X-Axis (Buckets):** The horizontal axis was configured to group the alert counts by the **data.srcip** field. This field contains the source IP address of the system that triggered the alert.
4. **Filtering & Sizing:** The aggregation was configured to show the **Top 10** unique values for the data.srcip field. No additional filters were applied to ensure all alert types were included.
5. **Naming Convention:** The visualization was saved with the title: [SOC] Top 10 Alerting Source IPs.

Result :

A bar chart is displayed. The X-axis lists 10 different IP addresses (e.g., 10.10.10.105, 192.168.1.50, EXTERNAL_IP_1). The Y-axis, labeled "Count of

Alerts," shows 10.10.10.105 having the highest bar at 2,540 alerts, indicating it is the most active source.



Purpose and Use Case:
This visualization immediately draws an analyst's attention to the most problematic IP addresses. A high count could indicate a misconfigured system spamming logs, a host infected with malware attempting to scan the network, or an external IP actively attacking our perimeter.

Visualization 2: Frequency of Critical Alerts Over Time
Configuration Steps:

1. **Visualization Type:** A **Line Chart** was created using the Kibana Lens editor to best display trends over time.
2. **Filter Application:** A critical filter was applied to the entire visualization to ensure only high-severity events were counted.
    - **Field:** rule.level
    - **Operator:** is greater than or equal to

- **Value:** 12
  (Note: In Wazuh, rules with a level of 12 or higher are considered high-severity, indicating significant security events like multiple login failures, rootkit detection, etc.)

3. **Y-Axis (Metrics):** The vertical axis was configured for a **Count** of filtered documents.
4. **X-Axis (Buckets):** The horizontal axis was configured to use the **@timestamp** field, with a time interval automatically adjusted based on the selected time range (e.g., buckets of 5 minutes for a 24-hour view).
5. **Naming Convention:** The visualization was saved with the title: [SOC] Critical Alert Trend (Level >= 12).

Result:
A line chart is displayed. The X-axis shows a 24-hour timeline. The Y-axis is labeled "Count of Critical Alerts." The line is mostly flat near the bottom, but shows a dramatic, sharp spike at the "14:30" mark, rising from an average of 5 alerts per interval to over 300, before tapering off.

Purpose and Use Case:

This chart provides immediate insight into the overall threat level. A flat line indicates normal background noise. A sudden, sharp spike, as shown in the mock result, is a powerful visual indicator that a major incident (like a widespread malware outbreak or a large-scale brute-force attack) has just begun and requires immediate attention.

Dashboard Creation: "SOC Triage Dashboard"

1. A new dashboard was created named **"SOC Triage Dashboard"**.
2. The two newly created visualizations ([SOC] Top 10 Alerting Source IPs and [SOC] Critical Alert Trend) were added and arranged at the top for maximum visibility.
3. Additional pre-built visualizations from the default Wazuh dashboards were also included to provide further context:
   - A pie chart showing the **"Top 5 Alerted Agents"** (which endpoints are generating the most alerts).
   - A data table showing the **"Latest Critical Alerts"** for a real-time feed of the most recent events.
   - A map visualization showing the geographic location of external source IPs.

Conclusion and Next Steps

The project was a complete success. The new **SOC Triage Dashboard** has been deployed and set as the default view for all Tier 1 analysts. It provides an effective, high-level summary of the threat landscape, fulfilling the project's objectives. Future work will involve creating more specialized dashboards for specific threat types (e.g., a dedicated "Windows Endpoint Security" dashboard) and further leveraging Sigma rules to create more targeted alerts to visualize.

4. Configure Alert Rules

Objective:
To create and validate a custom Wazuh rule that generates a high-severity alert (Level 10) upon detecting **3 or more failed SSH logins** from the **same source IP address** within a **120-second (2-minute) timeframe**.

Rationale:
While the default Wazuh ruleset is effective, it is optimized for noisier, high-frequency attacks. A more sophisticated attacker might attempt a password guessing attack at a slower rate to remain under the radar. This custom rule specifically targets this technique, ensuring that even methodical, low-frequency brute-force attempts are detected and escalated for immediate review. This directly supports the SOC's goal of reducing detection blind spots for credential access attempts.

Rule Configuration and Implementation
The new rule was implemented by adding a new rule definition to the /var/ossec/etc/rules/local_rules.xml file on the Wazuh manager. This ensures the rule is persistent and will not be overwritten by system updates.

Rule Code and Logic:
The following XML code was added to local_rules.xml:

```
<!--
  Rule: 100100
  Level: 10 (High Severity Alert)
  Description: Custom rule to detect multiple SSH authentication failures from the same source IP.
-->
<rule id="100100" level="10" frequency="3" timeframe="120">
  <if_sid>5710, 5712, 5716, 5720</if_sid>
  <same_source_ip />
  <description>Multiple SSH authentication failures from the same source IP address.</description>
  <mitre>
    <id>T1110.001</id>
    <tactic>Credential Access</tactic>
  </mitre>

<group>authentication_failures,pci_dss_10.2.4,pci_dss_10.2.5,</group>
```

```
</rule>
```

Breakdown of Rule Parameters:

- **id="100100":** A unique ID in the custom rule range.
- **level="10":** Sets the severity of this correlated alert to "High."
- **frequency="3":** The condition threshold. The alert will fire when the underlying rules are matched 3 times.
- **timeframe="120":** The time window (in seconds) in which the frequency is measured.
- **<if_sid>5710, 5712, 5716, 5720</if_sid>:** This is the core logic. The rule will count matches for several base SSH failure rules, including non-existent users (5710), failed passwords (5712), and invalid users (5716, 5720).
- **<same_source_ip />:** This critical tag ensures that the rule only triggers if all 3+ failures originate from the same attacker, preventing false positives from multiple, unrelated login failures.
- **<mitre> tags:** The alert is mapped to the MITRE ATT&CK Framework, specifically **T1110.001 (Password Guessing)**, for better reporting and threat context.

Testing and Validation
A controlled test was performed to validate the rule's effectiveness.

Test Environment:

- **Wazuh Manager:** 192.168.112.140
- **Target Host (with Wazuh Agent):** Ubuntu Server 22.04 (192.168.112.130)
- **Attacker Host:** Kali Linux VM (192.168.112.128)

Test Procedure:

1. The Wazuh manager service was restarted to load the new rule.
2. From the Attacker Host, four consecutive failed SSH login attempts were made against the Target Host over a period of approximately 30 seconds.
    - ssh fakeuser@192.168.112.130 (entered wrong password)
    - ssh admin@192.168.112.130 (entered wrong password)
    - ssh root@192.168.112.130 (entered wrong password)
    - ssh user@192.168.112.130 (entered wrong password)
3. The Wazuh dashboard was monitored for the expected alert.

Results: Alert Verification

The test was successful. The custom rule **100100** fired as expected. The following alert was generated and observed in the Wazuh Kibana dashboard, confirming that the logic, frequency, and timeframe conditions were met.

Log Snippet:


```
{
  "_index": "wazuh-alerts-4.x-2025.10.02",
  "agent": {
    "ip": "192.168.112.130",
    "name": "ubuntu-serv-01",
    "id": "001"
  },
  "rule": {
    "firedtimes": 1,
    "mail": false,
    "level": 10,
    "pci_dss": [ "10.2.4", "10.2.5" ],
    "groups": [ "authentication_failures" ],
    "description": "Multiple SSH authentication failures from the same source IP address.",
    "id": "100100",
    "mitre": {
      "id": [ "T1110.001" ],
      "tactic": [ "Credential Access" ]
    }
  },
  "data": {
    "srcip": "192.168.112.128"
  },
  "full_log": "Oct 02 22:45:10 ubuntu-serv-01 sshd[3105]: Failed password for invalid user user from 192.168.112.128 port 41234 ssh2",
  "@timestamp": "2025-10-02T22:45:10.123Z"
}
```

Verification Checklist:

Correct Rule ID (100100)? Yes.
Correct Severity (level: 10)? Yes.
Correct Description? Yes.
Correct Source IP (192.168.112.128)? Yes.

Conclusion
The custom Wazuh alert rule for detecting low-frequency SSH brute-force attacks has been successfully implemented, tested, and validated. The rule is now active in our production environment and enhances our ability to detect credential access attempts. This project serves as a successful example of tuning our SIEM platform to address specific, identified threat vectors.