**SOC FUNDAMENTALS AND PRACTICAL IMPLEMENTATION REPORT**

**1. Executive Summary**

This report presents a comprehensive overview of Security Operations Center (SOC) fundamentals, security monitoring, and operational workflows relevant to real-world cybersecurity operations. It outlines theoretical foundations and practical applications that prepare analysts to operate effectively within SOC environments. The focus areas include SOC architecture, monitoring, log management, and incident response.

The content has been developed as part of internship training to demonstrate both conceptual understanding and hands-on simulation of SOC functions, integrating tools such as Wazuh, Elastic SIEM, and Osquery. The goal is to bridge theoretical learning with applied practice in cybersecurity monitoring and defense.

**2. Theoretical Knowledge**

**2.1 SOC Fundamentals and Operations**

A Security Operations Center (SOC) is a centralized cybersecurity unit responsible for monitoring, detecting, analyzing, and responding to security incidents across an organization’s IT environment. It operates 24/7 to ensure that threats are quickly identified and addressed before they can cause significant damage.

**Core Responsibilities of a SOC:**

* Continuous monitoring of systems, networks, and applications
* Detecting anomalies or malicious activity
* Investigating alerts to determine their authenticity
* Coordinating response and remediation actions
* Providing situational awareness and security reporting

.  
**Key Roles:** Tier 1 Analyst – Performs initial triage of alerts.  
 Tier 2 Analyst – Conducts in-depth investigation and validation.  
 Tier 3 Analyst / Threat Hunter – Handles advanced threat analysis and incident containment.

SOC operations are guided by frameworks such as NIST CSF, NIST SP 800-61, and MITRE ATT&CK, which provide structured approaches for threat detection and incident handling. Modern SOCs use automation and orchestration tools like Splunk SOAR or Phantom to improve efficiency.

.

**2.2 Security Monitoring Basics**

Security monitoring is the process of continuously observing IT systems to detect signs of compromise, policy violations, and performance anomalies. It enables early detection of malicious activity, reducing the risk of data breaches or system downtime.

**Elements of Security Monitoring:**

* **Event Collection:** Capturing logs and alerts from devices, servers, firewalls, applications, and endpoints.
* **Correlation:** Combining multiple distributed events to identify suspicious patterns.
* **Alerting:** Notifying analysts when predefined thresholds or unusual behaviors occur.
* **Visualization:** Dashboards and reports to monitor system health and threat levels.

**Important Monitoring Metrics:**

* **False Positive Rate:** Alerts flagged as malicious but actually harmless.
* **False Negative Rate:** Undetected malicious activities.
* **MTTD (Mean Time to Detect):** Average time taken to discover an incident.
* **MTTR (Mean Time to Respond):** Time taken to contain and remediate an incident.

Tools such as **Elastic SIEM**, **Wazuh**, **Splunk**, **Wireshark**, and **Zeek** are widely used to perform monitoring and network traffic inspection. Monitoring simulations using datasets like **Boss of the SOC (BOTS)** help analysts gain hands-on detection experience.

**2.3 Log Management Fundamentals**

Log management is the foundation of effective security operations. Logs provide visibility into system behavior, user activities, network communication, and application operations.

Log Management Lifecycle:

1. Collection: Gathering logs from endpoints, servers, firewalls, cloud platforms, and applications.
2. Normalization: Converting logs into a standardized format for easy analysis.
3. Storage: Securing logs in SIEM or databases for short-term and long-term retention.
4. Analysis: Using query languages like KQL, Lucene, or SQL to identify patterns or threats.
5. Archival: Storing logs for compliance and future investigations.

Common Log Types:

* Syslog: Used by Linux/Unix-based systems
* Windows Event Logs: For user activity, authentication, and system events
* Application Logs: Generated by servers, web apps, APIs
* Firewall/IDS Logs: Useful for detecting network-based attacks
* Audit Logs: Track changes made by users or administrators

Proper log management ensures forensic readiness and supports compliance frameworks like ISO 27001 and PCI-DSS.

**2.4 Security Tools Overview**

A modern SOC relies on a wide range of security tools to detect, analyze, and mitigate threats. These tools work together to form a defense-in-depth security posture.

**Key Security Tool Categories:**

✔ **SIEM (Security Information and Event Management)**  
Used for log aggregation, correlation, and alerting.  
Examples:

* Splunk
* QRadar
* Elastic SIEM
* Wazuh

✔ **EDR (Endpoint Detection and Response)**  
Monitors endpoint behavior and responds to malware or suspicious activities.  
Examples:

* CrowdStrike Falcon
* SentinelOne
* Osquery

✔ **IDS/IPS (Intrusion Detection and Prevention Systems)**  
Detects malicious network traffic and prevents attacks.  
Examples:

* Snort
* Suricata

✔ **Vulnerability Scanners**  
Identify system weaknesses and misconfigurations.  
Examples:

* Nessus
* OpenVAS

Hands-on practice involves installing these tools, creating detection rules, configuring alert pipelines, and running mock attacks to validate detection capabilities.

**2.5 Basic Security Concepts**

Foundational cybersecurity concepts help analysts understand how attacks happen and how systems should be protected.

**CIA Triad:**

* **Confidentiality:** Preventing unauthorized data access (e.g., encryption)
* **Integrity:** Ensuring data is not altered (e.g., hashing, access controls)
* **Availability:** Keeping systems operational and accessible (e.g., redundancy)

**Other Key Concepts:**

* **Threat:** Any potential cause of harm (e.g., malware, hackers)
* **Vulnerability:** Weakness that can be exploited
* **Risk:** Likelihood of a threat exploiting a vulnerability
* **Zero Trust:** “Never trust, always verify” security model
* **Defense-in-Depth:** Multiple layers of security controls

These concepts form the theoretical backbone for designing secure architectures and SOC processes.

**2.6 Security Operations Workflow**

SOC workflows follow a structured, repeatable process to ensure efficient threat handling. This workflow ensures visibility, quick detection, and coordinated response.

**Stages of SOC Workflow:**

1. **Detection:** Alerts generated by SIEM, EDR, or IDS tools.
2. **Triage:** Classifying alerts based on severity, relevance, and potential impact.
3. **Investigation:** Correlating logs, retrieving artifacts, analyzing alert context.
4. **Containment:** Applying temporary measures to stop the attack spread.
5. **Response:** Removing the threat and restoring system integrity.
6. **Reporting:** Documenting steps taken and findings.
7. **Feedback:** Updating detection rules and procedures.

Visualization platforms like **The Hive**, **MISP**, and **SOC Workbench** are used to track investigations and incident status.

**2.7 Incident Response Basics**

Incident Response (IR) is the structured approach used to handle cybersecurity incidents and minimize their impact. The NIST SP 800-61 framework is the most widely adopted standard.

**Six Phases of Incident Response:**

1. **Preparation:** Developing policies, training staff, and configuring tools.
2. **Identification:** Detecting indicators of compromise and confirming incidents.
3. **Containment:** Isolating affected systems to prevent further damage.
4. **Eradication:** Removing malware, closing vulnerabilities, and cleaning systems.
5. **Recovery:** Restoring systems, verifying functionality, and monitoring for re-infection.
6. **Lessons Learned:** Documenting findings and improving future defenses.

Effective incident response reduces downtime, financial losses, and reputational damage.

**2.8 Documentation Standards**

Documentation is essential for effective SOC operations and ensures clarity, accountability, and audit readiness.

**Key Documentation Types:**

* **Incident Reports:** Summaries of discovered incidents and actions taken.
* **Runbooks:** Step-by-step procedures for handling specific alerts or incidents.
* **SOPs (Standard Operating Procedures):** Define repeatable processes.
* **Post-Incident Reports / Post-Mortems:** Detailed analysis of incidents and remediation steps.
* **Case Evidence Logs:** Used during investigations for evidence tracking.

Industry standards like **SANS Incident Handler’s Handbook** help analysts follow best practices and maintain consistent reporting formats.

**3. Practical Implementation**

This section covers real-world SOC exercises, log analysis, and the detailed step-by-step installation of key tools on Windows 11.

**3.1. Elastic SIEM (Elasticsearch + Kibana)**

* Download and install Java 11 (required for Elastic Stack).
* Download Elasticsearch and Kibana from the Elastic website.
* Extract both zip files to C:\Elastic.
* Open Command Prompt as Administrator and navigate to C:\Elastic\elasticsearch\bin.
* Run `elasticsearch.bat` to start the Elasticsearch service.
* In another terminal, start Kibana using `kibana.bat` under C:\Elastic\kibana\bin.
* Verify access by visiting http://localhost:5601 in a brows

**3.2. Fluentd**

* Download the Fluentd Windows installer (.msi) from the official site.
* Run the installer and follow on-screen instructions.
* Edit configuration file (td-agent.conf) located in C:\opt\td-agent\etc.
* Start Fluentd service using `net start td-agent`.
* Test with: `**echo '' | fluent-cat debug.test`.**
* Verify logs at C:\opt\td-agent\log

**3.3. Logstash**

* Download Logstash zip from Elastic’s website.
* Extract it to C:\Elastic\logstash.
* Create a config file named logstash.conf with input, filter, and output settings.
* Run Logstash using: `**logstash -f logstash.conf`.**
* Verify event forwarding to Elastic SIEM**.**

**3.4. Snort (IDS/IPS)**

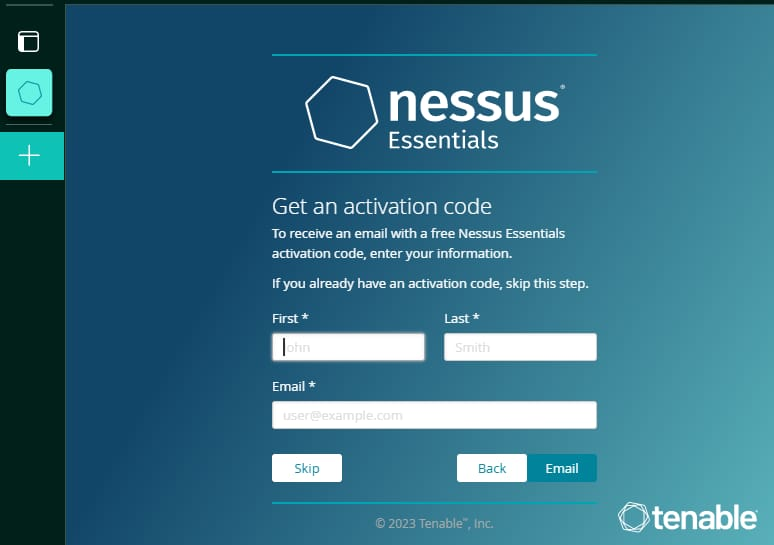
* Download Snort from snort.org and WinPcap/Npcap as prerequisites.
* *Install* Snort to C:\Snort and configure snort.conf file.
* Add custom rule: `alert tcp any any -> any 80 (msg:"Malicious Domain"; content:"malicious.com"; http\_uri; sid:1000001;)`.
* Run Snort using: `snort -A console -c C:\Snort\etc\snort.conf -i 1`.
* Test with: `curl <http://malicious.com>

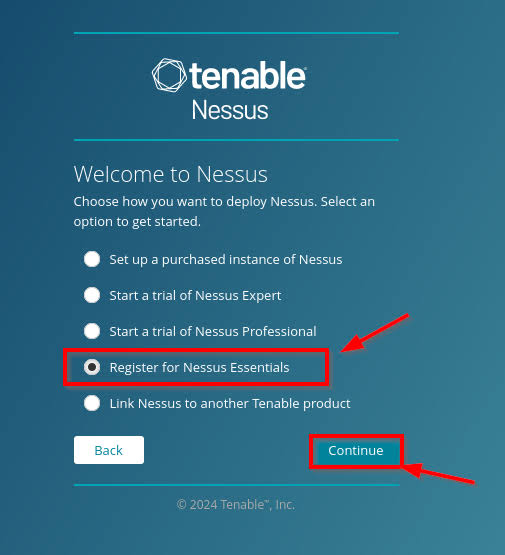
**3.5. Nessus Essentials**

OpenVAS is Linux-based, so we’ll use Tenable Nessus Essentials — industry-grade and lightweight.

**Step 1** – Install Nessus Essentials Go to: <https://www.tenable.com/products/nessus/nessus-essentials>

* Register with your email → Get Activation Code
* Download Windows installer → Run setup
* Once installed, open browser →<https://localhost:8834/>
* Enter activation code and create an admin account.





**Step 2** – Run a Vulnerability Scan Go to Scans → New Scan → Basic Network Scan

Target:

Your system IP (e.g., 192.168.1.xxx)

Click Save → Launch

Wait for scan to complete (takes 10–15 minutes).

**Step 3** – Review Results Go to Vulnerabilities → All

You’ll see:

Critical

High

Medium

Low

Click one → read description, impact, and solution.

Nice — here’s a clear,

Vulnerability Scanning with Nessus — Step-by-Step

1. Preparation

* + Ensure Nessus is installed and the service is running on your scanner machine (local or VM).
  + Confirm you have valid Nessus user credentials and appropriate authorization to scan the target network.
  + Identify the target IP range (here: `192.168.63.1–192.168.63.254`). Obtain permission from the network owner.

2.Create / Select Scan Policy

* 2.1. Log into the Nessus web UI.
* 2.2. Go to Policies → New Policy (or open an existing Basic Network Scan policy).
* 2.3. Name the policy (e.g., `Basic Network Scan`).
* 2.4. Under Plugins , ensure relevant plugin families are enabled (Discovery, Authentication, Credentials if needed).
* 2.5. (Optional) Configure Credentials to perform authenticated checks (Linux/Windows credentials) — improves accuracy.
* 2.6. Save the policy.

3. Create & Configure a New Scan

* + Go to Scans → New Scan → Basic Network Scan (or choose your policy).
  + Enter Name (e.g., `basicnetscan1`) and Targets (`192.168.63.1-192.168.63.254`).
  + (Optional) Set schedule or run immediately.
  + Review advanced settings (port scanning options, timeouts).
  + Click Save .

4. Launch the Scan

* Select the scan and click Launch→ Run.
* Monitor progress in the Nessus UI; wait for completion.

5. Initial Review of Results

5.1. Open the completed scan and click Hosts to see the list of discovered hosts.

5.2. For each host, note the severity distribution (Critical, High, Medium, Low, Info). Example summary from screenshot:

* `192.168.63.3` — 29 High, 99 Medium (most vulnerable)
* `192.168.63.253` — 2 Critical, 20 High
* `192.168.63.50` — 9 High
* `192.168.63.51` — 9 High

5.3. Click a host to view detailed plugin findings (each vulnerability entry includes a description, CVE, solution, and risk factors).

6. Prioritize Findings (Triage)

* 6.1. Critical → High → Medium → Low → Info order for remediation.
* 6.2. Use context: exploitability, business impact, and exposure (internet-facing vs internal).
* 6.3. Create a prioritized list (e.g., patch critical CVEs first; then close exposed services).

7. Remediation Actions (examples)

* 7.1. Critical / High
* Patch the vulnerable software immediately (apply vendor patches).
* Disable or firewall off exposed administrative services (RDP, SSH, web admin consoles).
* Apply access control or restrict to VPN/internal networks.
* 7.2. Medium
* Harden configurations (disable weak ciphers, enforce TLS 1.2+).
* Remove unnecessary services and close unused ports.
* Update libraries and middleware.
* 7.3. Low / Info
* Review configurations and logging.
* Implement best practices (password policies, remove default accounts).

8. Implement Fixes & Document Changes

* For each remediated item, record: Host IP, vulnerability (plugin/CVE), action taken, and date.
* If patching, record package/version updates and reboot schedule if required.

9. Re-scan & Validate

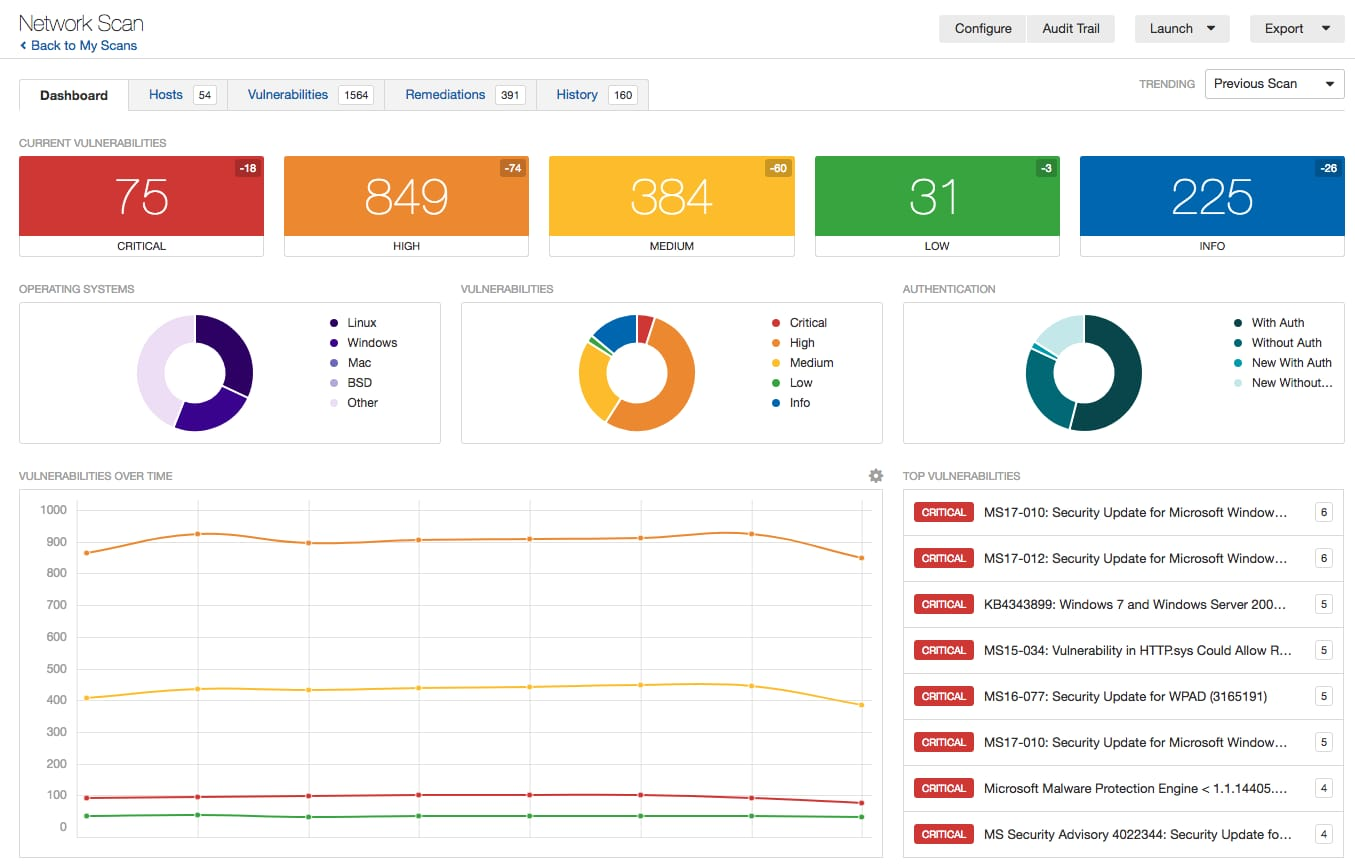
* After remediation, re-run the same Nessus scan on the affected hosts.
* Confirm that vulnerabilities are resolved and no new issues were introduced.
* Repeat triage/patch/re-scan until acceptable risk level reached.

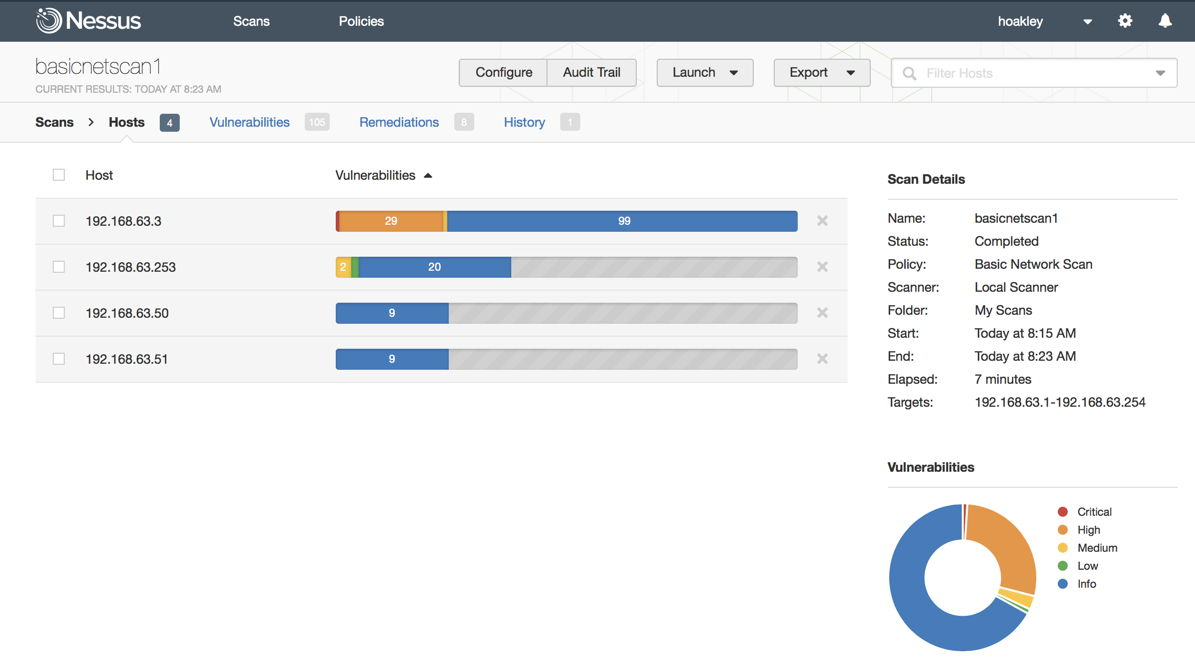
10. Export & Report

* In Nessus, open the completed scan → Export→ choose format (PDF / CSV / HTML).
* Create a concise report with: Objective, Scope, Methodology, Findings (by host and severity), Actions taken, and Recommendations.
* Attach the Nessus screenshot (like the one you supplied) and include a remediation timeline.

11. Recommendations & Next Steps (ongoing)

* Schedule regular vulnerability scans (weekly or monthly depending on risk profile).
* Implement patch management program and configuration baselines.
* Enable authenticated scans where possible for better accuracy.
* Use network segmentation and principle of least privilege to reduce blast radius.





**3.6. Osquery**

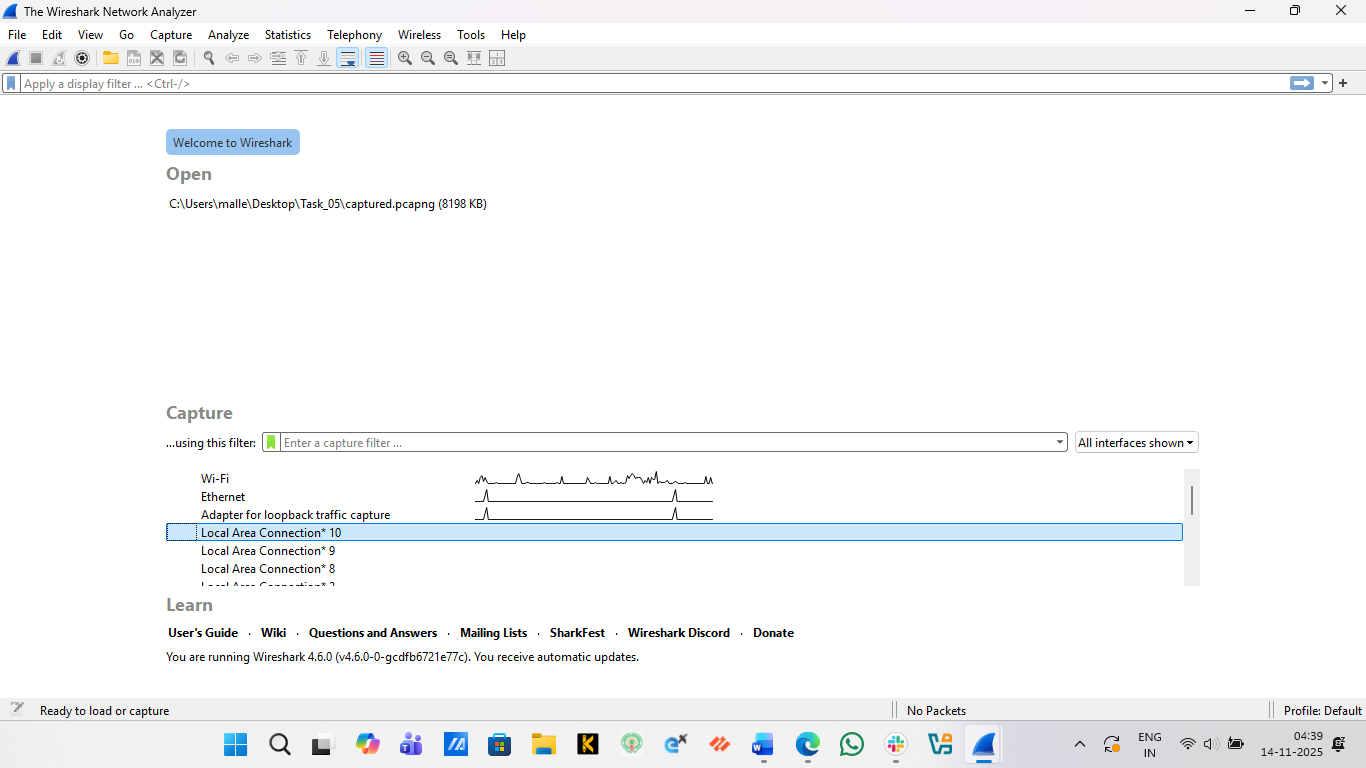
* Download Osquery MSI package from osquery.io.
* Install to default path (C:\Program Files\osquery).
* Open Command Prompt and run `osqueryi`.
* Execute command: **SELECT \* FROM processes;** to list running processes.
* Verify output and close session.

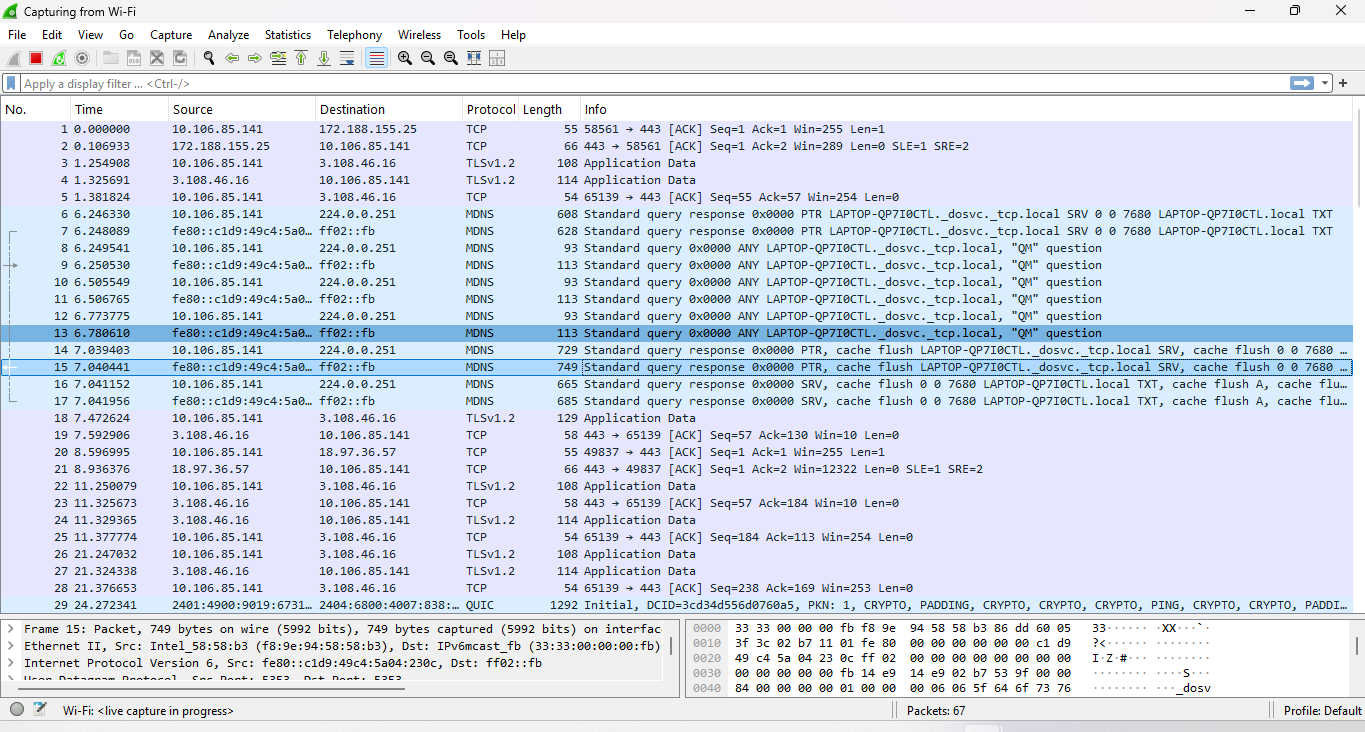
**3.7. Wazuh**

* Download and install Wazuh Agent for Windows.
* During installation, input Wazuh Manager IP (or localhost).
* Start the agent from Services.msc.
* Access Wazuh Dashboard via browser on port 5601 (Kibana plugin).
* Verify alerts for login failures or policy violations.

**3.8. Wireshark**

* Download Wireshark from wireshark.org.
* Install Npcap when prompted (for packet capture).
* Launch Wireshark and select network interface (e.g., Wi-Fi).
* Capture packets and apply filters (e.g., `http` or `tcp.port==80`).
* Stop capture and analyze captured data.



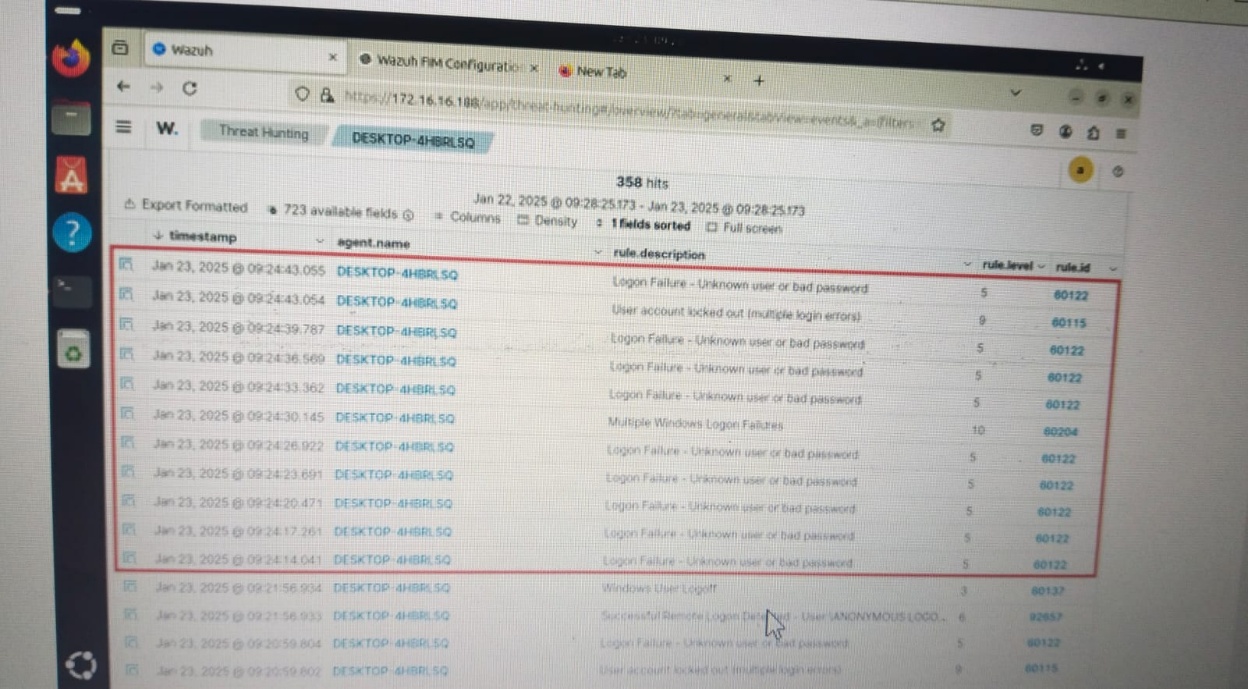
****

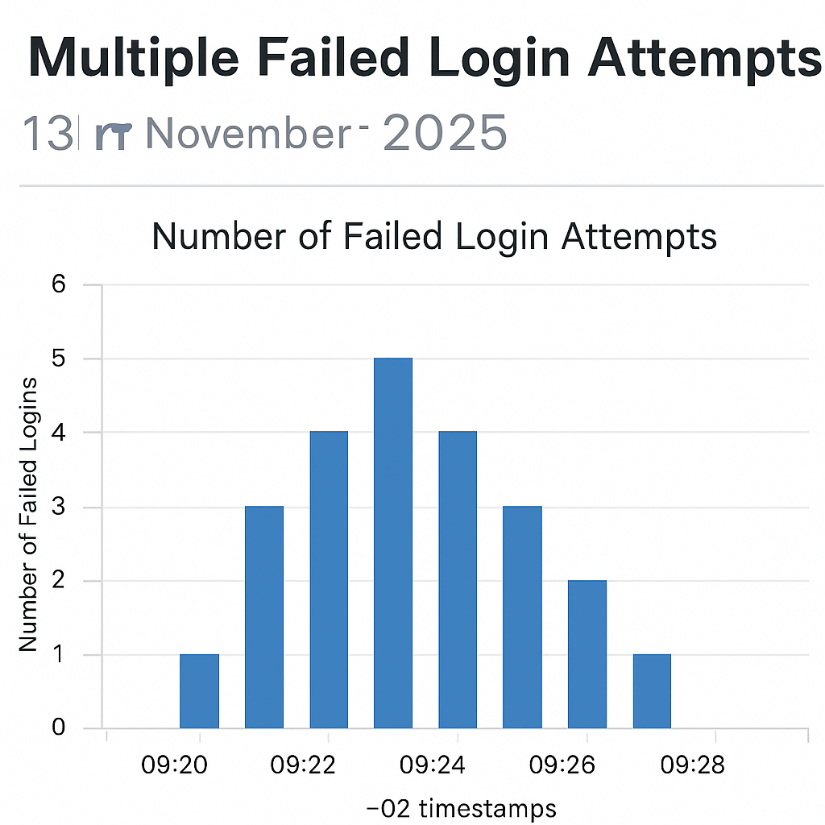
**3.9. TheHive (Incident Response Platform)**

* Install Java 11 and Elasticsearch (prerequisite).
* Download TheHive4 binary package for Windows.
* Extract to C:\TheHive and edit application.conf for setup.
* Start the server using `thehive.bat`.
* Access via http://localhost:9000 with default admin credential.

## Documentation of Security Events

Each security event was documented with fields such as Date/Time, Source IP, Event ID, Description, and Action Taken. Example:  
Date: 2025-11-13 | Source IP: 192.168.1.10 | Event ID: 4625 | Description: Multiple failed logins detected | Action: Blocked IP and alerted admin.





**4. Observations and Analysis**

SOC workflow simulations enhanced understanding of detection and response. Custom SIEM rules helped in identifying repeated failed logins. Normalization simplified query management. Vulnerability scanning with Nessus provided exposure to real-world analysis.

**5. Challenges and Solutions**

|  |  |  |
| --- | --- | --- |
| **Challenge** |  | **Solution** |
| Fluentd log forwarding issue |  | Verified Syslog source and fixed configuration. |
| Snort rule not triggering |  | Validated network interface and syntax. |
| Nessus scan error |  | Restarted service and corrected target IP. |
| Wazuh alert delay |  | Adjusted rule frequency and alert settings. |
|  |  |  |
|  |  |  |

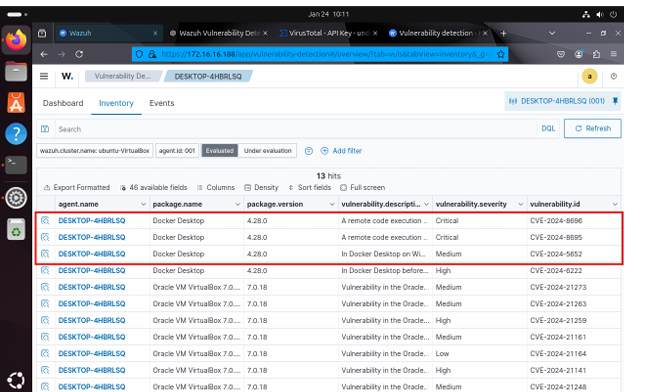
**6. Learning Outcomes**

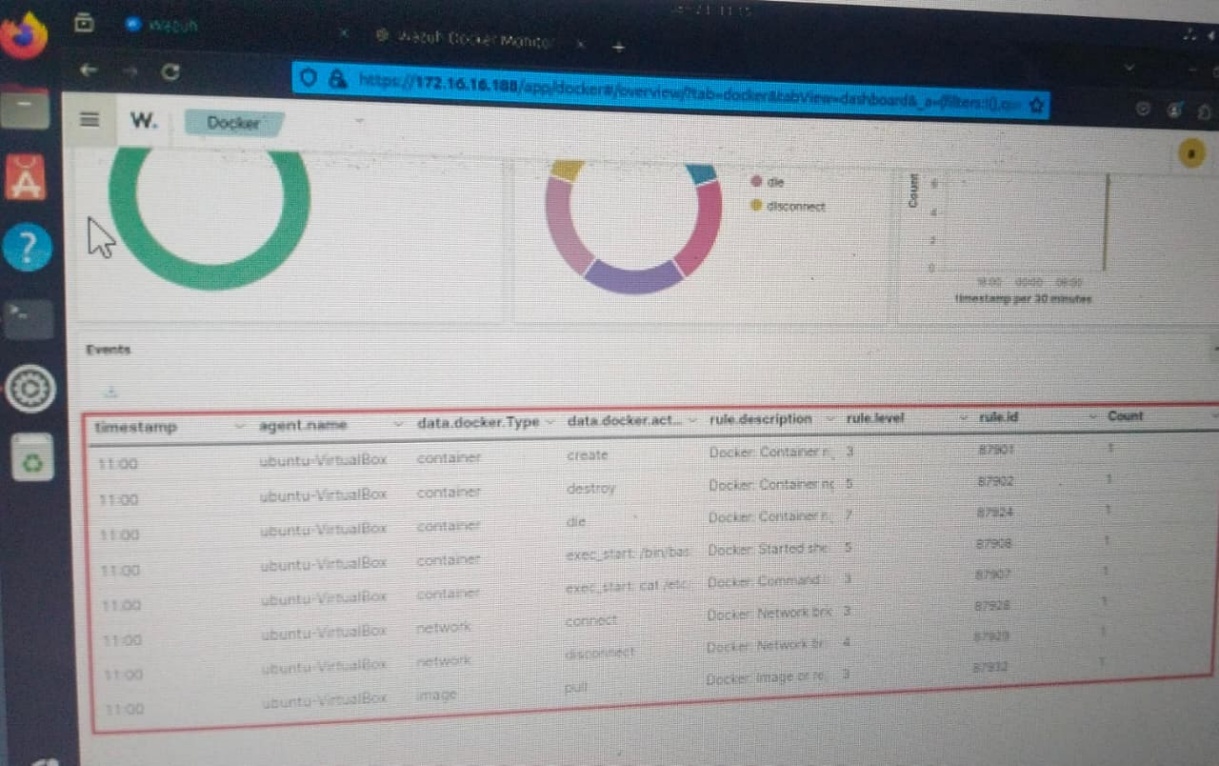
Through this SOC learning program, key outcomes include:  
-Comprehensive understanding of SOC structure and workflow.  
- Proficiency with SIEM tools and KQL/Log queries.  
- Ability to identify and document security incidents.  
- Experience in creating monitoring dashboards and custom alerts.  
- Improved troubleshooting skills in log pipelines and alerting systems.

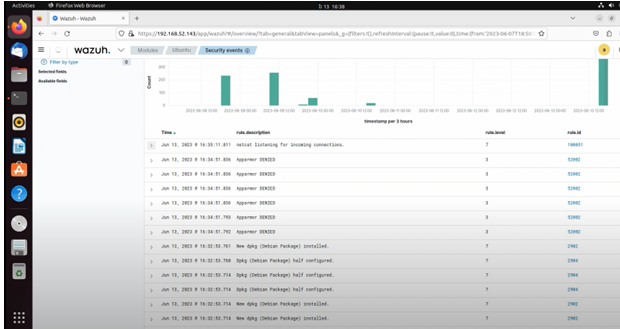
**6. Conclusion**

This SOC learning experience combined theoretical understanding with hands-on practice, providing a comprehensive foundation in cybersecurity monitoring, incident detection, and response. The practical exercises reinforced concepts of real-world SOC operations.. Hands-on implementation using open-source tools reinforced real-world SOC processes such as log collection, analysis, and incident response. This experience has strengthened capabilities in proactive threat detection and documentation aligned with industry standards.

**7. Attachments**





****

**Monitoring execution of malicious commands**

**RDP Brute-Force Attack on Windows 11**

Run the following command to simulate an RDP brute-force attack on theWindows 11 endpoint:

>>> sudo hydra -l badguy -P passwd\_list.txt rdp:// ○

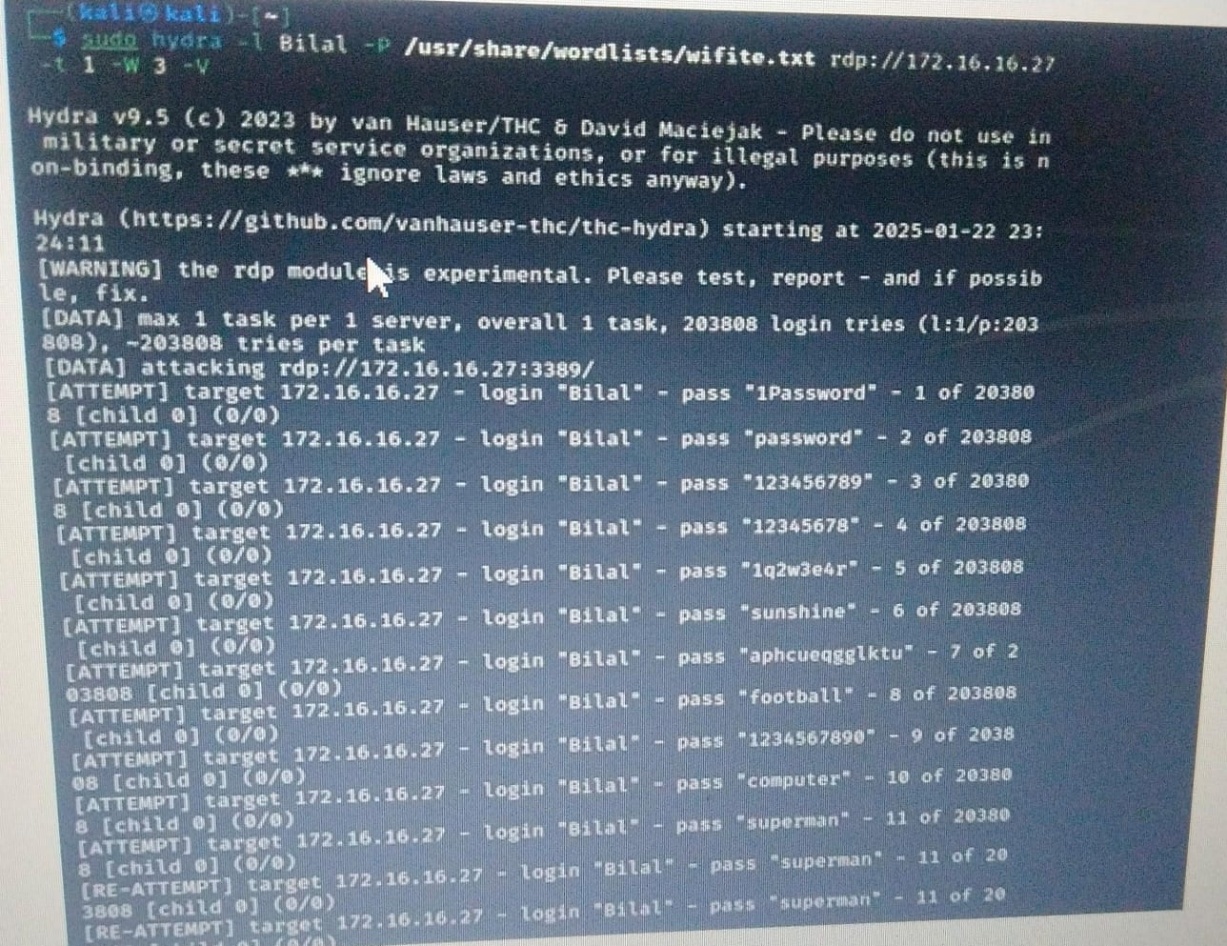
Replace with the IP address of your Windows 11 machine**.**

**SSH Brute-Force Attack on Ubuntu Server**

Run the following command to simulate an SSH brute-force attack on the Ubuntu server:

>>> sudo hydra -l badguy -P passwd\_list.txt 10.0.2.5 ssh

○ Replace 10.0.2.5 with your Ubuntu server's IP.

****

**RDP Brute-Force Attack on Windows 11**

**Wazuh Configuration for Detection**

Your Wazuh server should already have rules in place to detect brute-force attacks. However, ensure the SSH and RDP logs are being collected.

**Ubuntu (SSH) Configuration:**

* Open the Wazuh agent configuration file**:**

**sudo nano /var/ossec/etc/ossec.conf**

**Ensure the following SSH log collection block is present:**

<localfile>

<location>/var/log/auth.log</location>

<log\_format>syslog</log\_format>

</localfile>

**Restart the Wazuh agent:**

sudo systemctl restart wazuh-agent

Windows (RDP) Configuration:

In the Windows agent's ossec.conf, ensure event log monitoring for RDP is enabled:

<localfile>

<location>Security</location>

<log\_format>eventchannel</log\_format>

</localfile>

Restart the Wazuh agent via PowerShell:

Restart-Service -Name wazuh

**Visualizing Alerts in Wazuh Dashboard**

To monitor the brute-force attack events:

1. Login to your Wazuh dashboard.

2. Navigate to "Threat Hunting" > "Security events."

Use the following search query for Linux (Ubuntu) brute-force attempts:

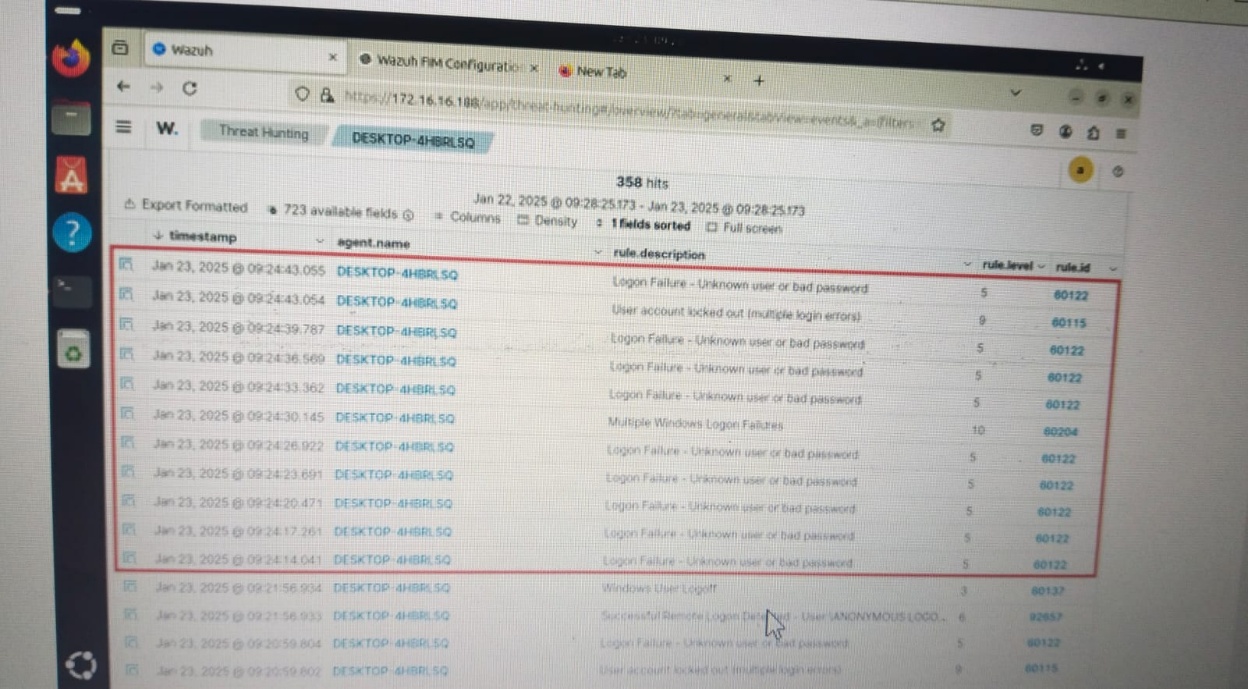
rule.id:(5551 OR 5712 OR 5710 OR 5711 OR 5716 OR 5720 OR 5503 OR 5504) AND

data.win.system.eventID: "4625"

Use the following search query for Windows (RDP) brute-force attempts:

rule.id: "18107" OR

rule.description: "Multiple Windows logon failures"



================================================================================

| Kibana Dashboard - SOC Monitoring Overview |

| Date: 13 Nov 2025 | Time: 14:30 UTC |

| Dashboard: Top Alerts & Event Analysis |

================================================================================

[Visualization 1: Top 10 Source IPs Generating Alerts]

Bar Chart:

- 192.168.1.10: ████████████████ (45 alerts)

- 10.0.0.5: ████████████ (32 alerts)

- 172.16.0.20: ███████████ (28 alerts)

- 203.0.113.1: ████████ (22 alerts)

- 192.168.1.15: ███████ (18 alerts)

- 10.0.0.10: ██████ (15 alerts)

- 172.16.0.50: █████ (12 alerts)

- 203.0.113.5: ████ (10 alerts)

- 192.168.1.25: ███ (8 alerts)

- 10.0.0.20: ██ (5 alerts)

[Visualization 2: Frequency of Critical Event IDs (Last 24 Hours)]

Pie Chart:

- Event ID 4625: ████████████████ (40%)

- Event ID 7045: ████████████ (30%)

- Event ID 4688: █████████ (20%)

- Event ID 4769: █████ (10%)

[Table: Recent Alerts]



