National Forensic Sciences University
Knowledge | Wisdom | Fulfilment
An Institution of National Importance
(Ministry of Home Affairs, Government of India)

FINAL SEMESTER PRESENTATION

Integrated Cyber Defense Environment (ICDE)

By:
Sudarshan Rangappa
012300300008002019
Msc. Cyber Security

Under the Guidance of:
Dr. Ramya Shah
Assistant Professor

Submitted To:
School of Cybersecurity and Digitial Forensics
National Forensic Sciences University

Table Of Contents

Introduction

- The Evolving Threat Landscape: Increasing sophistication, persistence, and automation of cyber threats.
- Modern Security Operations Centers (SOCs): Central hubs for monitoring, detection, analysis, and response.
- The Need for Integration: Siloed tools lead to alert fatigue, slow response, and missed correlations.
- What is an ICDE? An integrated ecosystem where security tools (SIEM, SOAR, NIDS, HIDS, etc.) work synergistically.
- Why Build This ICDE? To create a realistic, hands-on virtual lab for training, testing, and validating defensive strategies without impacting production systems.

Problem Statement

- Challenge: Difficulty for organizations/institutions to establish effective, affordable, and comprehensive cybersecurity training/testing environments.
- Cost Barrier: Commercial solutions are often expensive.
- Integration Complexity: Integrating diverse open-source tools requires significant expertise and effort.
- The Gap: Need for a well-documented, functional, and accessible virtual lab environment based on readily available tools to bridge the skills gap.

Objectives

- Primary Goal: Build, deploy, and test a virtualized ICDE simulating key SOC functions.
- Specific Objectives:
  - Develop a scalable VMware-based framework.
  - Install & configure core security tools (Wazuh, Shuffle, Suricata, OpenVAS, Cowrie, Splunk).
  - Integrate tools for effective data exchange (logs, alerts).
  - Establish a realistic target environment (Active Directory, Endpoints).
  - Configure Shuffle (SOAR) for alert ingestion and basic automation.
  - Demonstrate detection and management of simulated attacks.
  - Validate the integrated stack using Kali Linux.

Scope of Project

- In Scope:
  - Design and implementation of the virtualized environment in VMware.
  - Installation and configuration of specified security tools.
  - Setup of Active Directory domain (home.lab), Windows Server 2025 DC, and Windows 11 endpoint.
  - Initial integration for data flow (log forwarding, alert webhooks).
  - Demonstration of core functionalities: alert generation, log collection, vulnerability scanning, honeypot interaction, alert ingestion into Shuffle.
  - Basic validation using simulated attacks from Kali Linux.

Scope of Project

- Out of Scope:
  - Exhaustive performance benchmarking under heavy load.
  - Development of complex, fully automated end-to-end response playbooks in Shuffle (designated as future work).
  - Large-scale environment simulation (limited by hardware resources).

Literature Review

Title	Key Findings	Limitation
Stallings, W. (2020). Cybersecurity Technologies for Network Defense: SIEM Solutions	SIEM plays a critical role in real-time threat detection through log aggregation and correlation.	SIEM solutions generate high volumes of alerts, often leading to alert fatigue.
Shackleford, D. (2019). SOAR: The Future of Automated Incident Response.	SOAR improves response times by automating incident workflows.	Integration challenges with legacy security infrastructure limit its effectiveness.
Spitzner, L. (2018). Honeypots: Tracking Hackers.	Honeypots are effective in gathering intelligence on attacker behavior and tactics.	They require continuous monitoring and maintenance to avoid detection by attackers.
Paxson, V. (2021). Network Monitoring with Zeek: A Deep Dive into Intrusion Detection	Zeek provides deep packet inspection and enriches threat intelligence.	High-performance overhead and storage requirements for large network environments.
Herzog, A. (2022). OpenVAS and Vulnerability Management in Modern Cybersecurity Frameworks.	OpenVAS efficiently scans for vulnerabilities and provides remediation suggestions.	High false-positive rates lead to unnecessary resource allocation.

Tools and Technologies Used

- Virtualization: VMware Workstation
- SIEM/HIDS: Wazuh (Manager, Indexer, Dashboard, Agents)
- SOAR: Shuffle (Open-Source SOAR Platform)
- NIDS: Suricata
- Vulnerability Scanner: OpenVAS (Greenbone Vulnerability Management - GVM)
- Honeypot: Cowrie (SSH/Telnet Honeypot)

Tools and Technologies Used

- Log Aggregation/Analysis: Splunk Enterprise (Free License)
- Target Environment:
  - Windows Server 2025 (Active Directory Domain Controller)
  - Windows 11 (Endpoint)
  - Sysmon (Enhanced Endpoint Monitoring)
- Attacker Machine: Kali Linux
- Operating Systems: Ubuntu Server 22.04 LTS, Debian 12, Windows

System Architecture - Network Topology

- **Environment:** VMware Workstation
- **Network:** ICDE-project [NAT network] {192.168.33.0/24}
- **VM List & IPs:**
  - Kali (Attacker): 192.168.33.143
  - Splunk (Log Aggregation): 192.168.33.128
  - Win Server 2025 (AD DC): 192.168.33.129
  - Win 11 (Endpoint): 192.168.33.130
  - Wazuh Manager: 192.168.33.131
  - Shuffle (SOAR): 192.168.33.149
  - OpenVAS (Scanner): 192.168.33.147
  - Cowrie (Honeypot): 192.168.33.142
  - Suricata (NIDS): 192.168.33.144

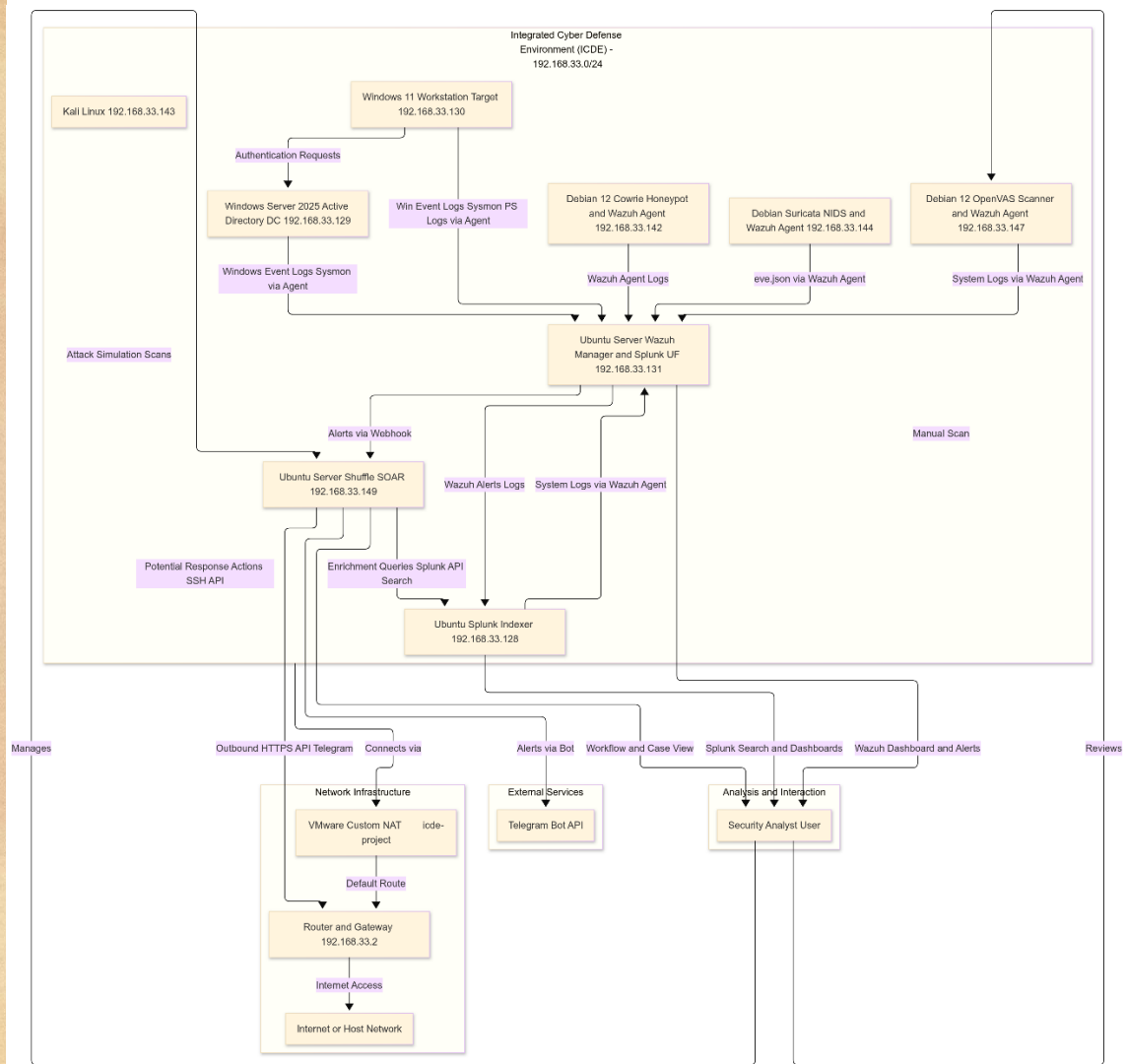
System Architecture Network Topology

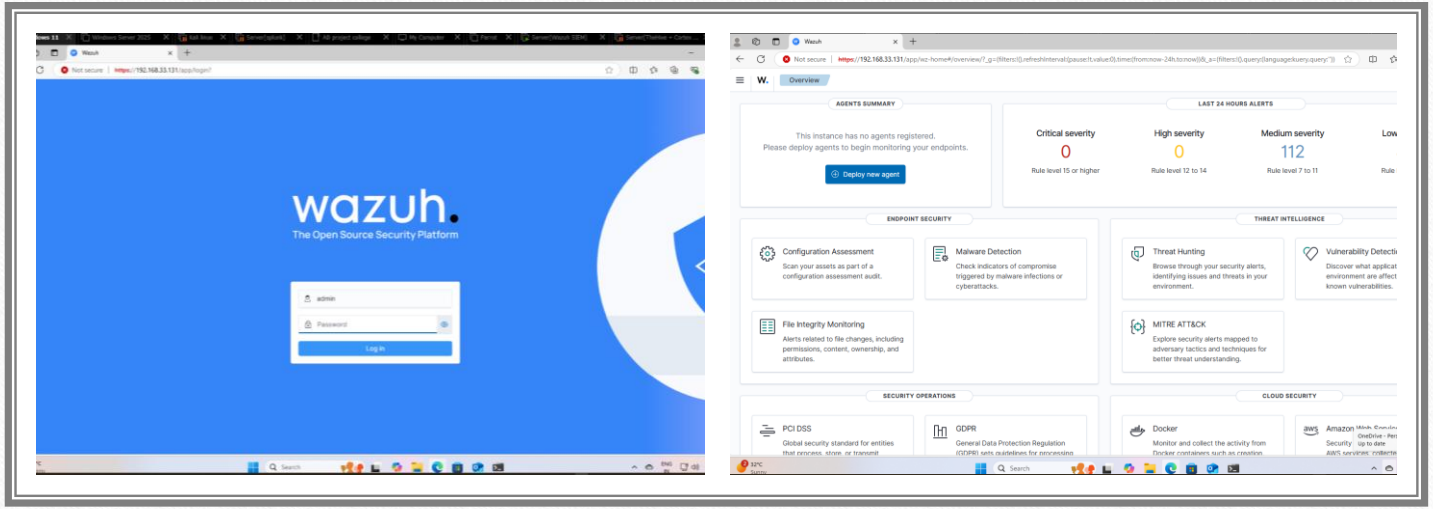
Role / Primary Tool	Operating System	IP Address	RAM	vCPUs	Hard Disk	Notes
Attacker Machine	Kali Linux	192.168.33.143/24	4gb	2	80gb	Used for launching simulated attacks.
Log Aggregation / Analysis	Ubuntu Server 22.04 LTS	192.168.33.128/24	4gb	2	80gb	Hosts Splunk Enterprise (Free License).
Domain Controller / AD	Windows Server 2025	192.168.33.129/24	4gb	4	60gb	Hosts Active Directory Domain Services.
Target Endpoint	Windows 11	192.168.33.130/24	4gb	2	80gb	Represents a typical user workstation.
SIEM / HIDS Manager	Ubuntu Server 22.04 LTS	192.168.33.131/24	4gb	2	80gb	Hosts Wazuh Manager.
SOAR Platform	Ubuntu Server 22.04 LTS	192.168.33.149/24	8gb	2	60gb	Hosts Shuffle.
Vulnerability Scanner	Debian 12	192.168.33.147/24	8gb	4	60gb	Hosts OpenVAS (GVM).
Honeypot	Debian 12	192.168.33.142/24	4gb	2	60gb	Hosts Cowrie.
NIDS Sensor	Debian 12	192.168.33.144/24	2gb	2	60gb	Hosts Suricata. (May need promiscuous mode)

System Architecture - Component Roles & Integration

- **Wazuh:** Collects endpoint/server logs, performs HIDS, correlates events, generates alerts. Forwards alerts/logs to Splunk & Shuffle.
- **Suricata:** Monitors network traffic (NIDS), generates alerts (Eve JSON). Logs collected by Wazuh agent.
- **Cowrie:** Captures SSH/Telnet interaction attempts. Logs collected by Wazuh agent.
- **OpenVAS:** Performs vulnerability scans on targets. Results viewed via GSA.
- **Splunk:** Central log repository (Wazuh, Suricata, Cowrie via Wazuh). Enables deep search and analysis.
- **Shuffle:** Ingests Wazuh alerts via webhook. Orchestrates basic notification (Telegram). Foundation for future automation.
- **AD/Endpoints:** Realistic target environment generating logs (enhanced via GPO/Sysmon). Monitored by Wazuh agents.
- **Kali:** Simulates attacks to test detection and response.
- **Integration:** Primarily via Wazuh Agents, Splunk Universal Forwarder (Wazuh -> Splunk), and Webhooks (Wazuh -> Shuffle).

System Architecture - Component Roles & Integration



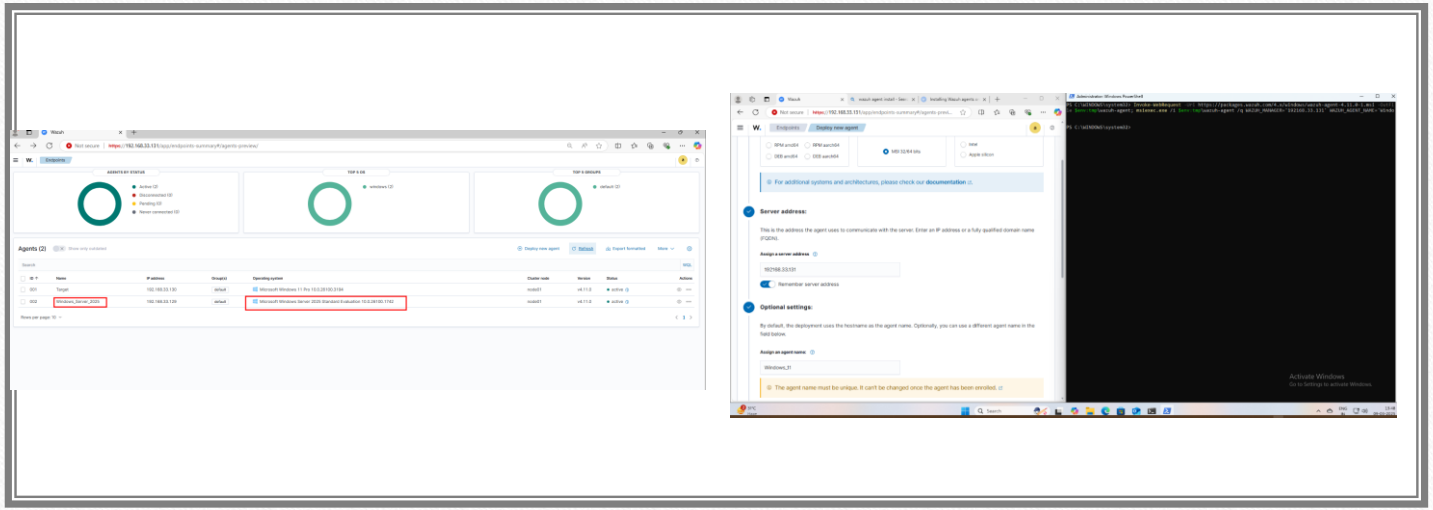


Implementation - Wazuh & Agents

Wazuh Server: Installed using assisted installation script on Ubuntu.

Implementation

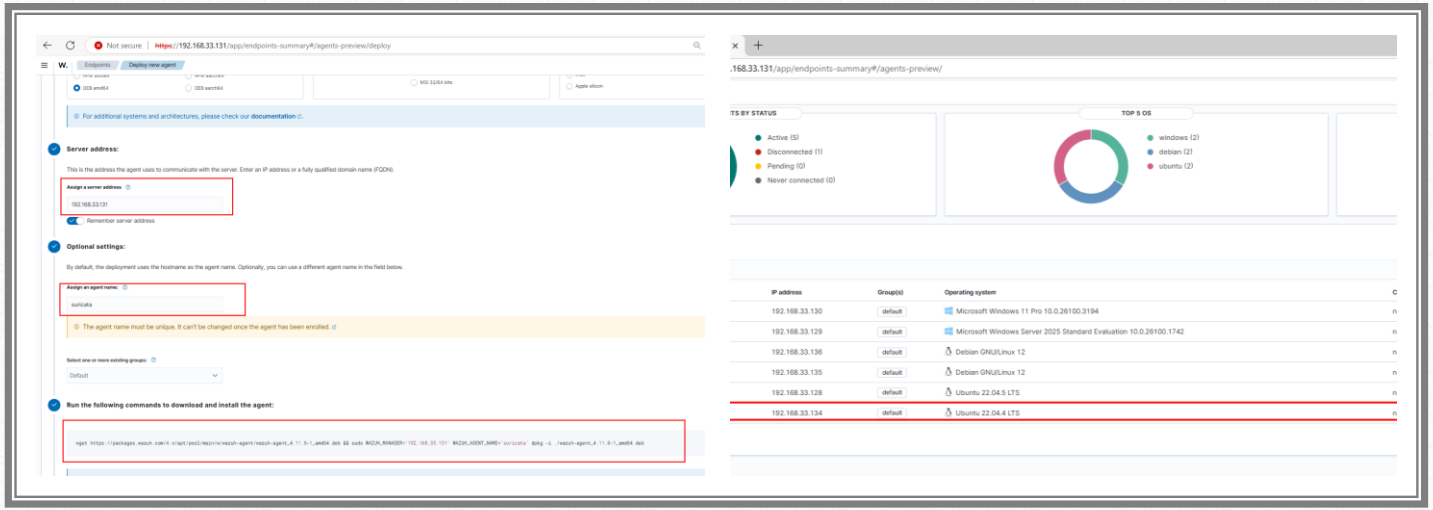
Wazuh agents on Windows machines & Linux machines ,Log forwarding to splunk, Suricata[NIDS], OpenVAS [vulnerability Assessment] , Cowrie [honeypot], Splunk[Log Aggregation], AD & Endpoints, Shuffle [SOAR]



Implementation – Wazuh Agents on windows

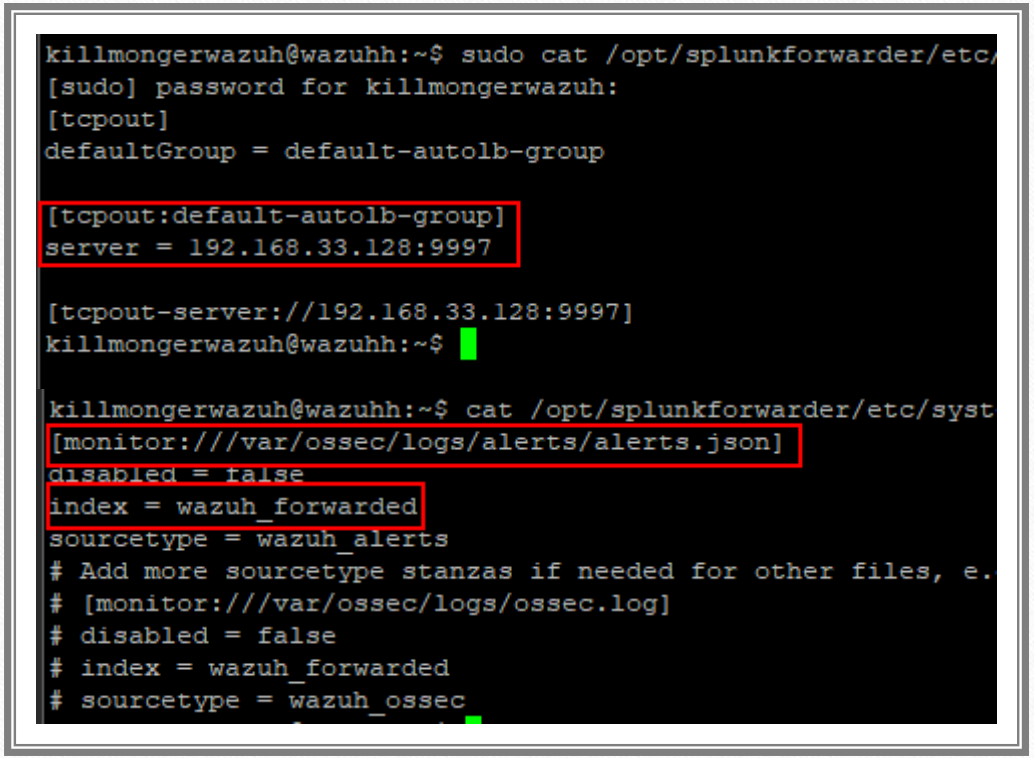
Agent Deployment: Using Wazuh UI "Deploy new agent" feature.

•Windows Agents (Server 2025, Win 11): PowerShell commands executed.



Implementation – Agents On Linux

Linux Agents (Suricata, Cowrie, Splunk, OpenVAS): Command-line deployment.

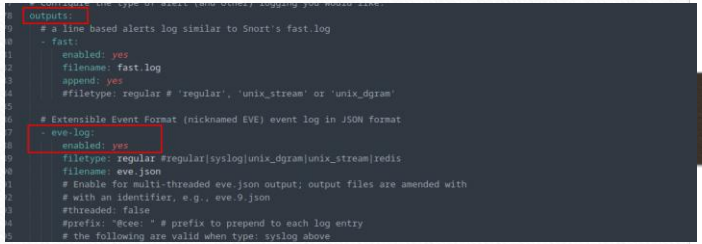


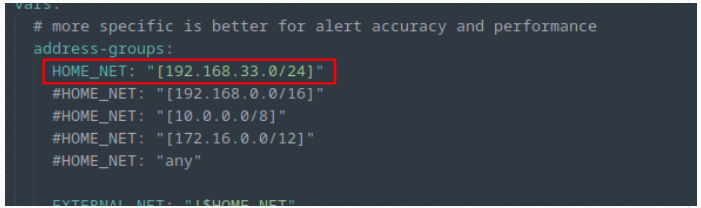
Implementation

Log Forwarding to Splunk: Splunk Universal Forwarder on Wazuh Manager monitoring alerts.json.

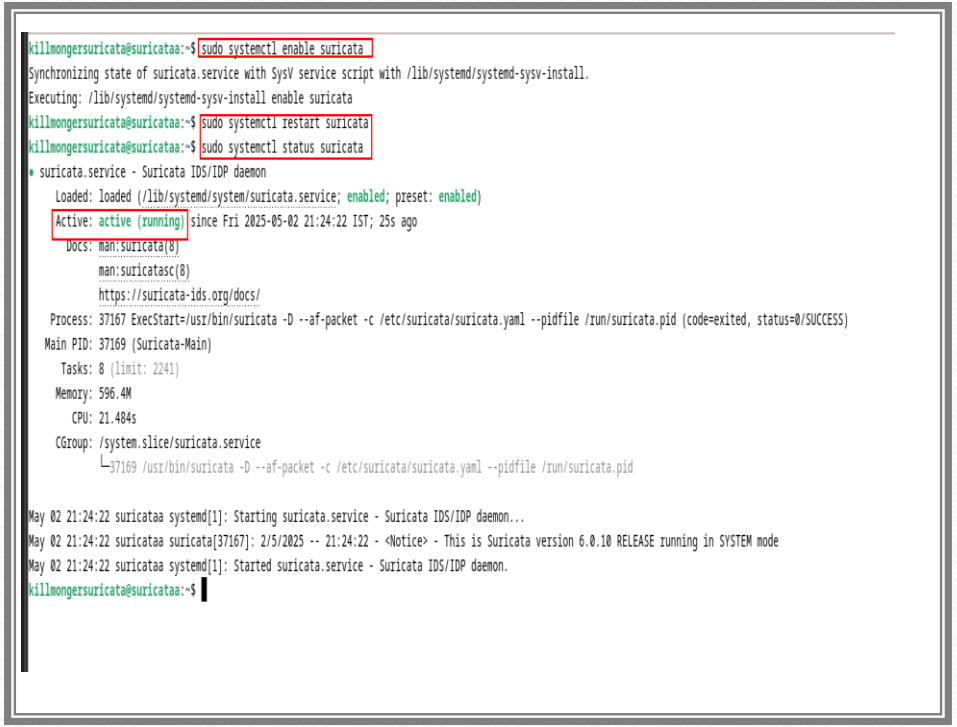
Implementation - Suricata (NIDS)

- **Installation:** Standard apt install suricata on Debian.
- **Configuration (suricata.yaml):**
- Set HOME_NET to 192.168.33.0/24.
- Configured af-packet interface (e.g., ens192).
- Enabled Eve JSON logging (eve.json).
- Updated rulesets (suricata-update).



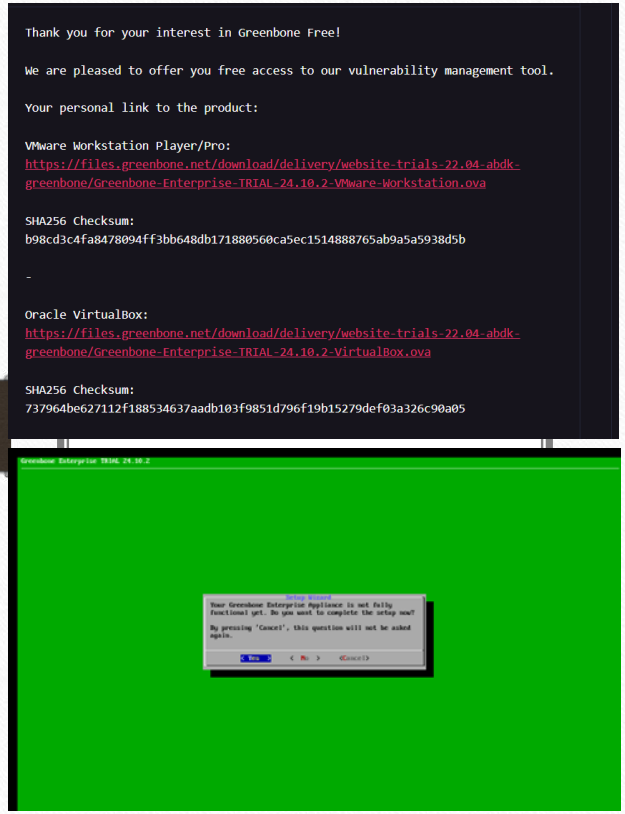


Implementation - Suricata (NIDS)



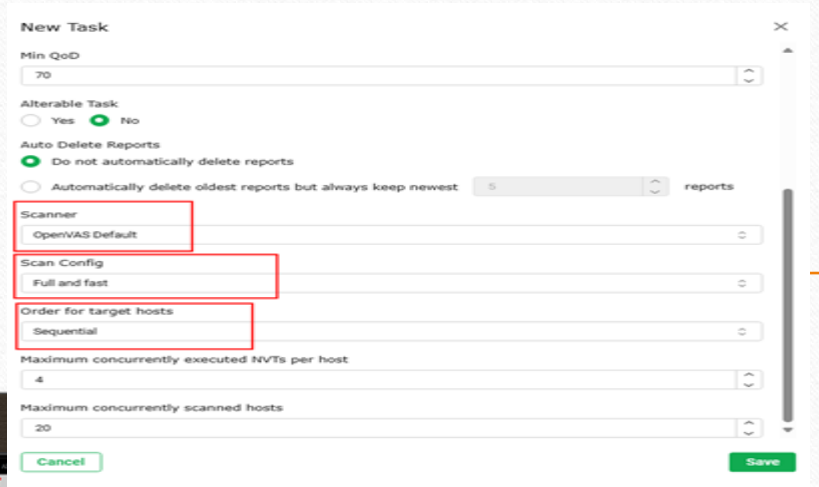
Service: Enabled and started suricata.service.

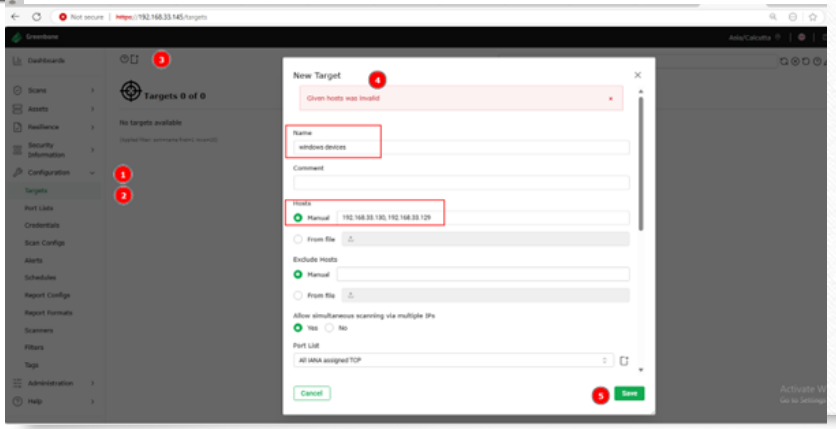
Integration: Logs monitored by local Wazuh agent.



Implementation - OpenVAS (Vulnerability Scanner)

- **Deployment:** Downloaded OVA file from Official Greenbone Site & Imported Greenbone Community Edition OVA into VMware.
- **Setup:** Initial wizard for admin user creation.





Implementation - OpenVAS (Vulnerability Scanner)

Feed Sync: Automatic synchronization of NVT/SCAP/CERT data. **Configuration:**

Defined target IPs (Win Server, Win 11) in GSA.

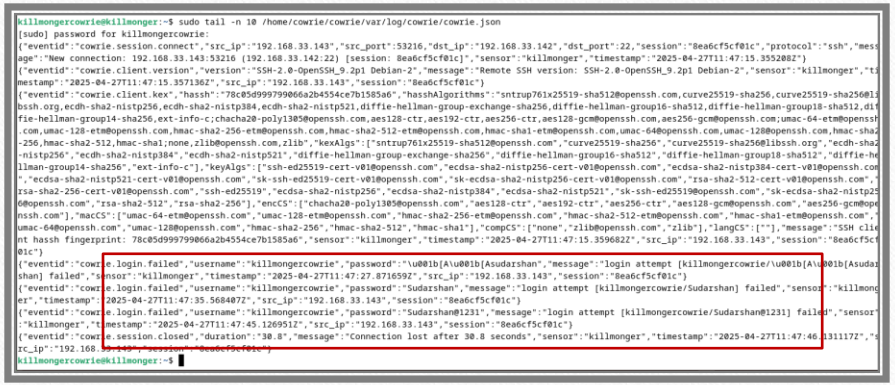
Created and ran scan tasks ("Full and fast").

Implementation - Cowrie (Honeypot)

- **Installation:** Followed official documentation (dependencies, git clone, venv).
- **Configuration (cowrie.cfg):**
  - Set realistic hostname.
  - Enabled SSH (port 2222) and Telnet listeners.

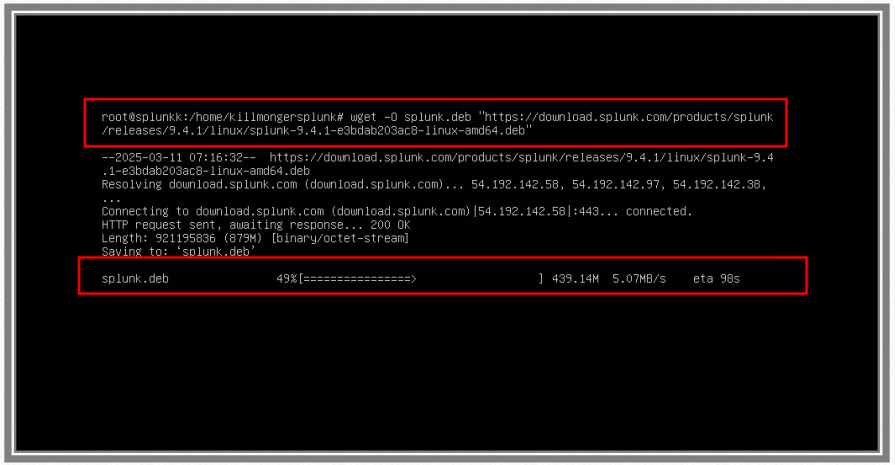
Implementation - Cowrie (Honeypot)

- **Service:** Started using cowrie start (or configured as a service).
- **Integration:** Logs (cowrie.json) monitored by local Wazuh agent.
- **Testing:** SSH attempts from Kali captured.



Implementation - Splunk (Log Aggregation)

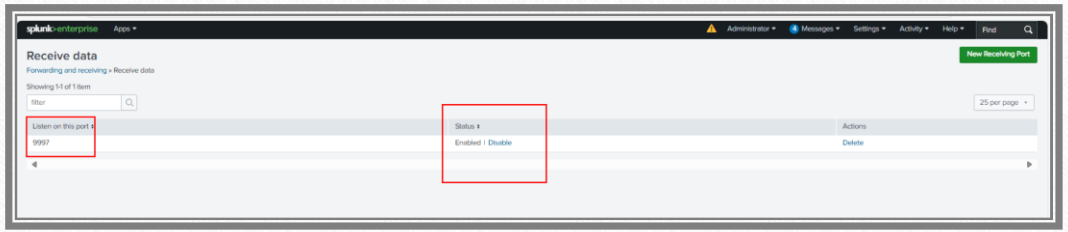
Installation: dpkg -i splunk*.deb on Ubuntu. Initial setup via CLI.



Implementation - Splunk (Log Aggregation)

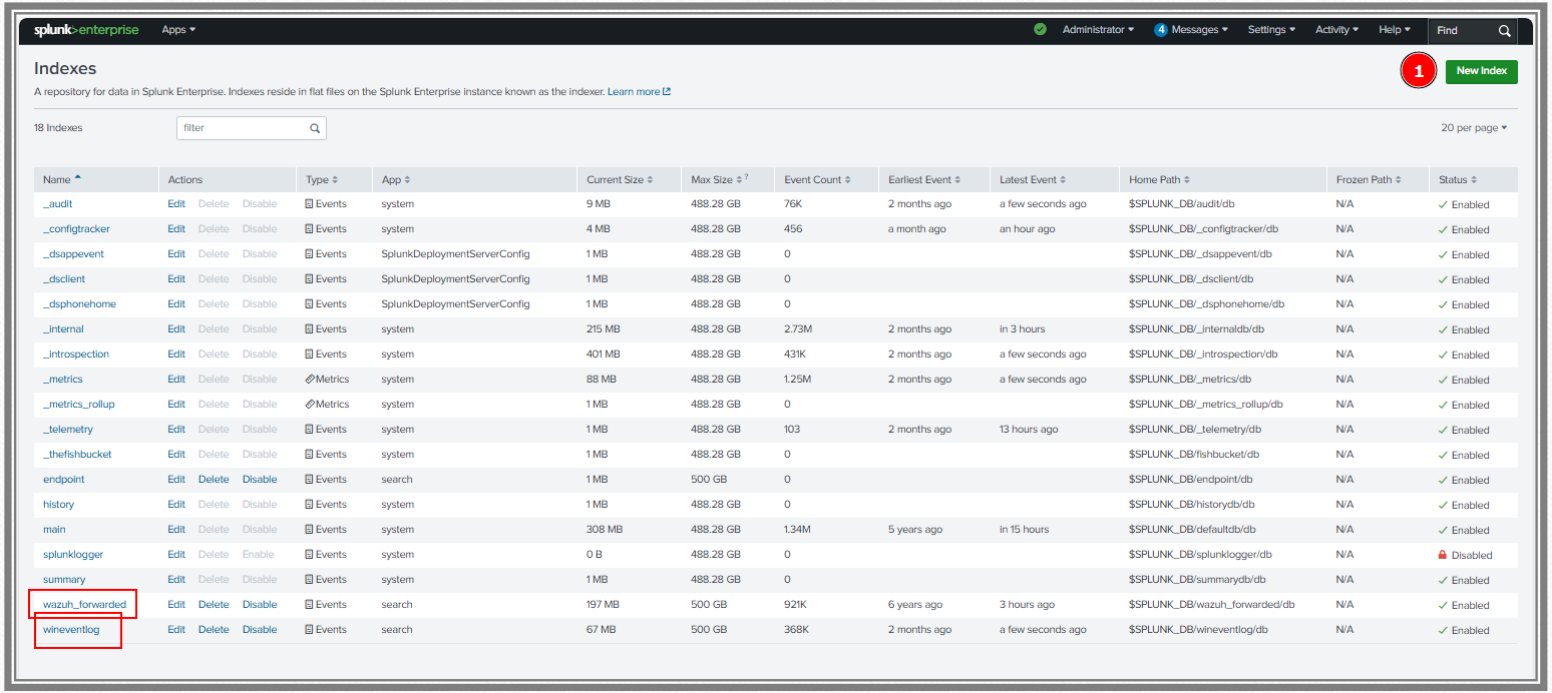
Configuration:

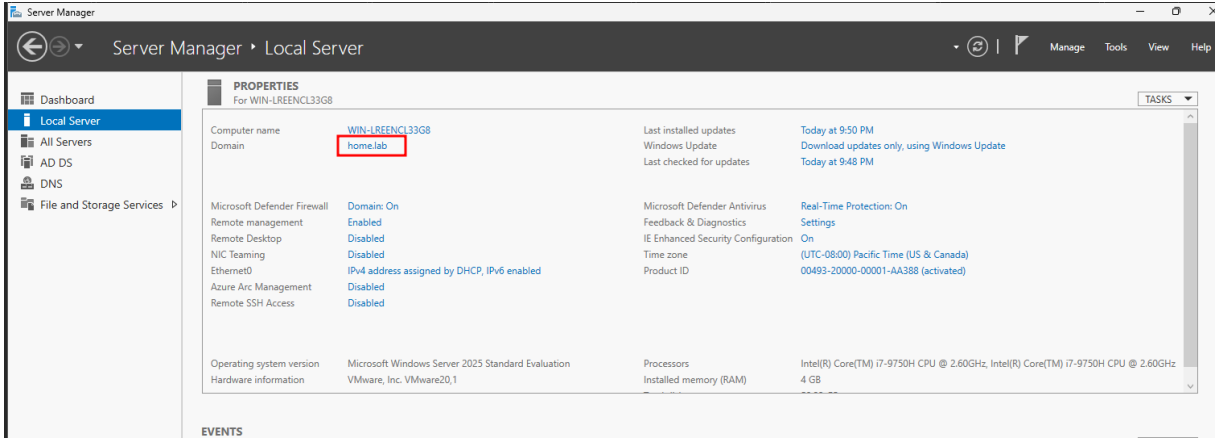
- Enabled TCP input on port 9997 for Universal Forwarder data.



Implementation - Splunk (Log Aggregation)

Created custom indexes (wineventlog, wazuh_forwarded).





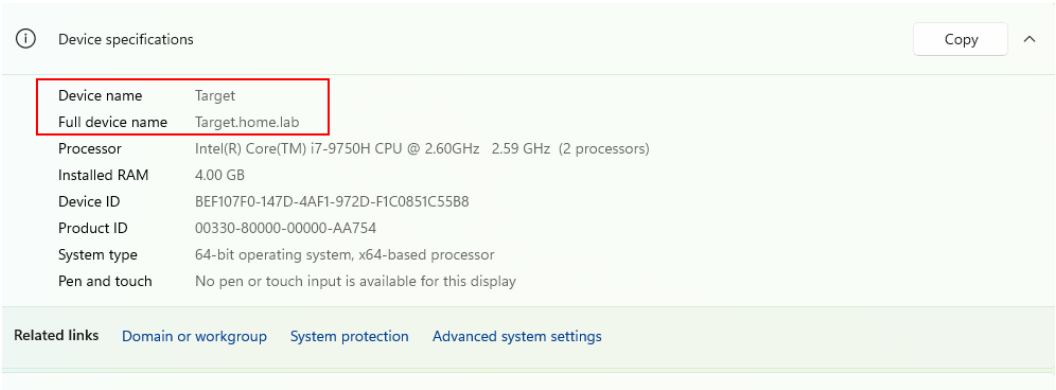
Implementation - Active Directory & Endpoints

AD Setup: Windows Server 2025 promoted to DC for home.lab domain.

Implementation - Active Directory & Endpoints

Structure: Created OUs (IT, Finance, Security, etc.), Users (AdminUser, Bob, etc.), and Groups.

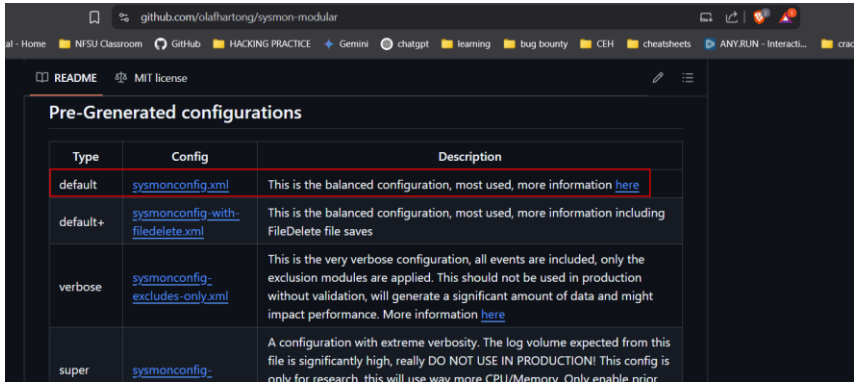
OU Name	Groups Inside OU	Users Inside OU	Group Memberships
IT	IT Admins	AdminUser	AdminUser → IT Admins, Domain Users
Finance	Finance Team	FinanceUser	FinanceUser → Finance Team, Domain Users
Security	Security Analysts, Log Readers	SecurityAnalyst	SecurityAnalyst → Security Analysts, Log Readers, Domain Users
GeneralUsers	Remote Access Users	Bob, Alice	Bob, Alice → Remote Access Users, Domain Users
Workstations	(No groups, just stores computers)	Windows 11 client	(N/A)
Servers	(No groups, just stores servers)	Windows Server, Wazuh, TheHive, Splunk, OpenVAS, Cowrie, Zeek	(N/A)

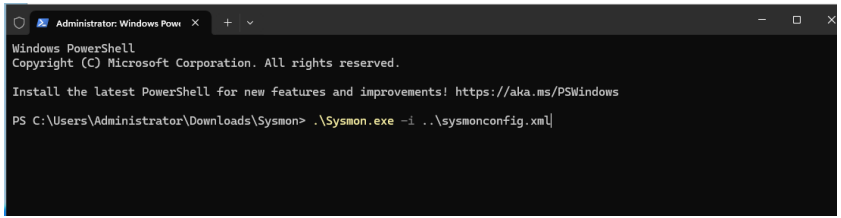


Implementation - Active Directory & Endpoints

Endpoint: Windows 11 joined to home.lab domain.

Implementation - Active Directory & Endpoints

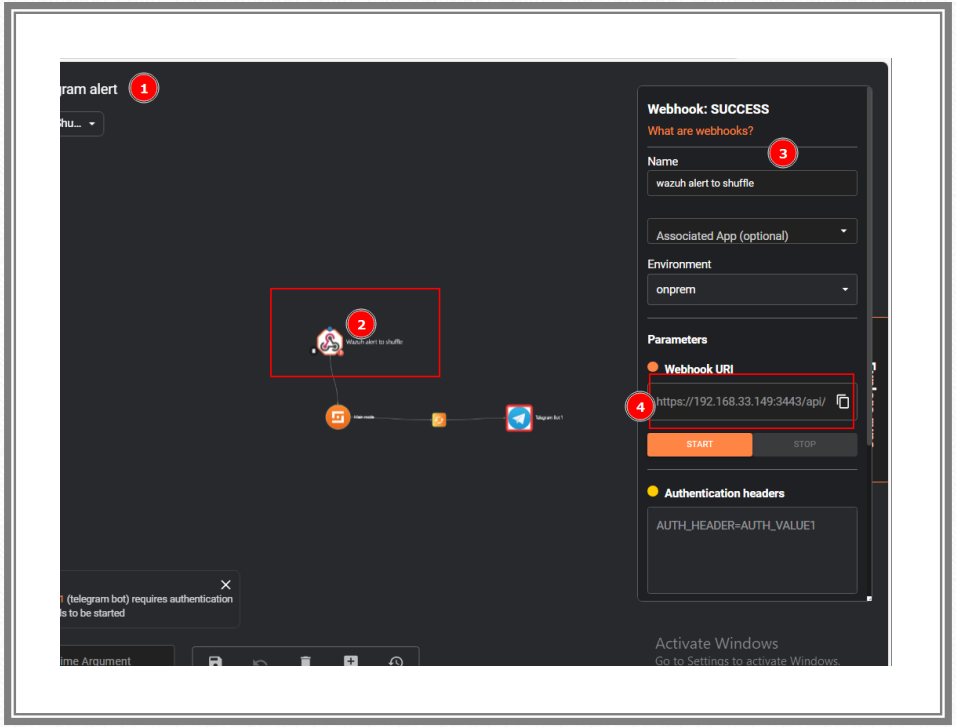




Sysmon: Installed with Olaf Hartong's config (sysmonconfig.xml).

Implementation - Shuffle (SOAR)

Installation: Deployed using Docker Compose on Ubuntu.



Implementation - Shuffle (SOAR)

Integration with Wazuh:

- Created Webhook listener in Shuffle.

Implementation - Shuffle (SOAR)

Configured Wazuh **ossec.conf** to send alerts to Shuffle webhook URL.

```
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <integration>
    <name>wazuh-cowrie-alert</name>
    <hook_url>https://192.168.33.149:3443/api/v1/hooks/webhook_dd0acdc7-7804-4d6f-884e-eb12ac72894a </hook_url>
    <rule_id>120003</rule_id>
    <alert_format>json</alert_format>
  </integration>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>

  <!-- Policy monitoring -->
```

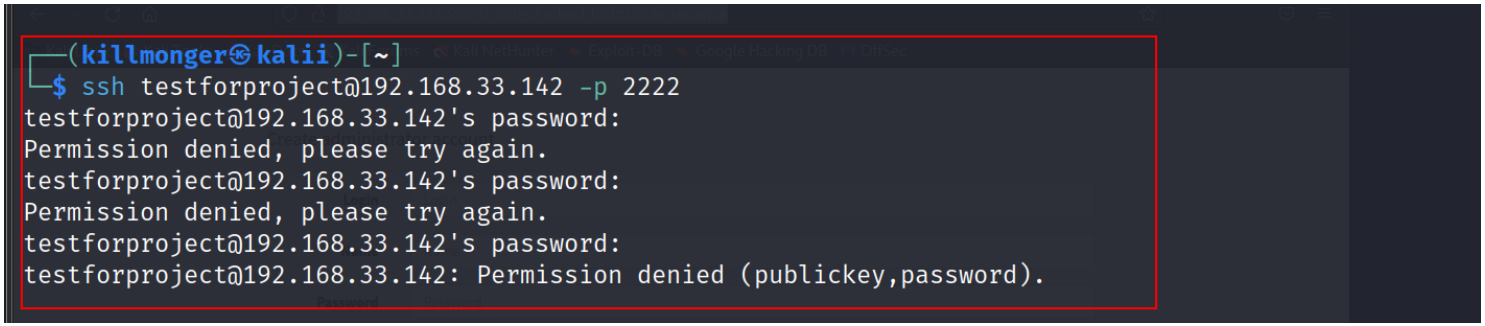
Implementation - Shuffle (SOAR)

Basic Workflow:

- Configured Telegram App for notifications.
- Simple workflow: Webhook Trigger -> Telegram Send Message.

Testing & Output

Honeypot Interaction & SOAR



Testing & Output

Scenario: SSH brute-force attempt from Kali to Cowrie (port 2222).

```
2025-05-03T05:19:23.101012Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha2-256' b'none'
2025-05-03T05:19:23.107387Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2025-05-03T05:19:23.109421Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2025-05-03T05:19:23.110071Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' trying auth b'none'
2025-05-03T05:19:33.350902Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' trying auth b'password'
2025-05-03T05:19:33.351584Z [HoneyPotSSHTransport,15,192.168.33.143] Could not read etc/userdb.txt, default database activated
2025-05-03T05:19:33.351682Z [HoneyPotSSHTransport,15,192.168.33.143] login attempt [b'testforproject'/b'this_is_a_test'] failed
2025-05-03T05:19:34.353818Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' failed auth b'password'
2025-05-03T05:19:34.353969Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2025-05-03T05:19:41.447325Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' trying auth b'password'
2025-05-03T05:19:41.447617Z [HoneyPotSSHTransport,15,192.168.33.143] Could not read etc/userdb.txt, default database activated
2025-05-03T05:19:41.447732Z [HoneyPotSSHTransport,15,192.168.33.143] login attempt [b'testforproject'/b'2ndatteptimade'] failed
2025-05-03T05:19:42.450031Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' failed auth b'password'
2025-05-03T05:19:42.450200Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2025-05-03T05:19:51.963492Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' trying auth b'password'
2025-05-03T05:19:51.963694Z [HoneyPotSSHTransport,15,192.168.33.143] Could not read etc/userdb.txt, default database activated
2025-05-03T05:19:51.963777Z [HoneyPotSSHTransport,15,192.168.33.143] login attempt [b'testforproject'/b'cowrieisworkingithink'] failed
2025-05-03T05:19:52.965516Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'testforproject' failed auth b'password'
2025-05-03T05:19:52.965675Z [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] unauthorized login: ()
2025-05-03T05:19:52.966947Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-05-03T05:19:52.967094Z [HoneyPotSSHTransport,15,192.168.33.143] Connection lost after 29.9 seconds
```

Testing & Output

Detection:

- Cowrie logged the attempt.

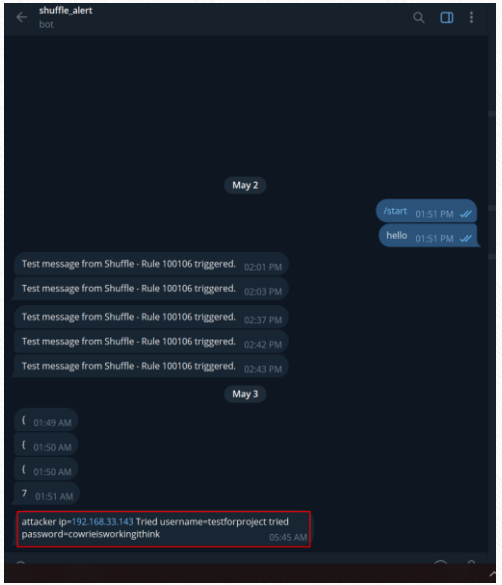
Testing & Output

- Wazuh agent sent log to Manager.
- Wazuh generated high-severity alert (Rule 120003).

Testing & Output

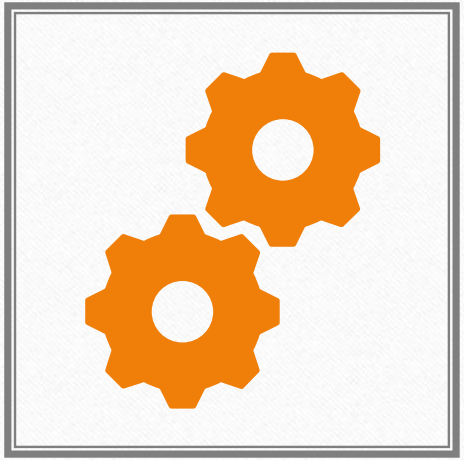
Automation:

- Wazuh sent alert via webhook to Shuffle.
- Shuffle workflow triggered.
- Telegram notification sent to analyst.



Limitations of the Project

- **Focus on Integration & Configuration:** The primary focus was setting up and connecting the tools, not deep performance optimization or advanced feature utilization.
- **Limited SOAR Automation:** Development of complex, automated response playbooks in Shuffle was outside the defined scope (future work). Only basic alert ingestion and notification were implemented.
- **Hardware Constraints:** The scale of the environment (number of endpoints, AD complexity, simulation intensity) was limited by the physical resources of the host machine.
- **No Heavy Load Testing:** Performance under sustained, high-volume attack scenarios was not formally benchmarked.



thank you