



**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 Digital Forensics
National Forensic Sciences University

Table Of Contents

1. Introduction
2. Problem Statement
3. Objectives
4. Scope of Project
5. Literature Review
6. Tools and Technologies Used
7. System Architecture
8. Implementation
9. Testing & Output (Key Scenarios)
10. Limitation Of project

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

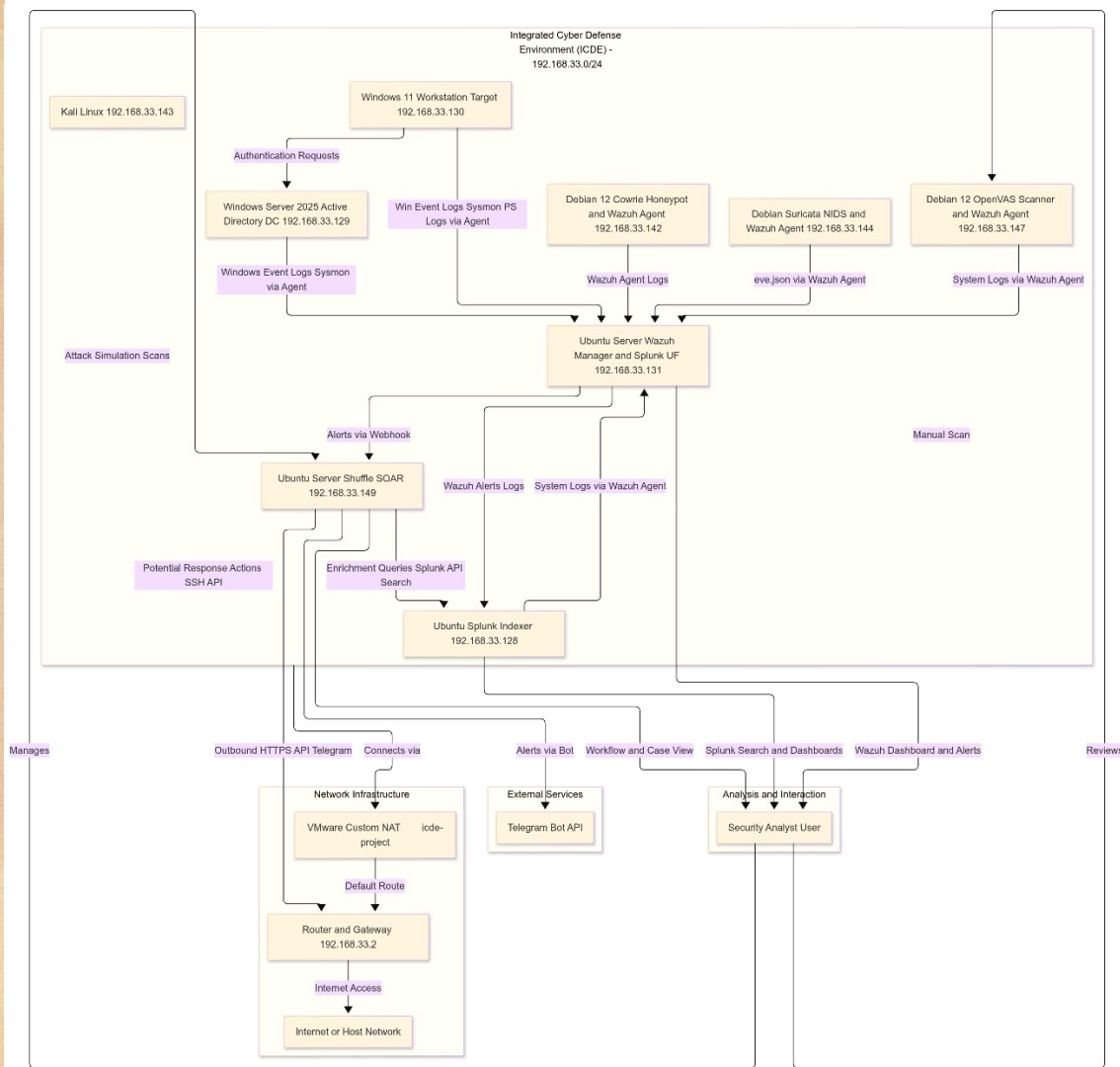
System Architecture Network Topology

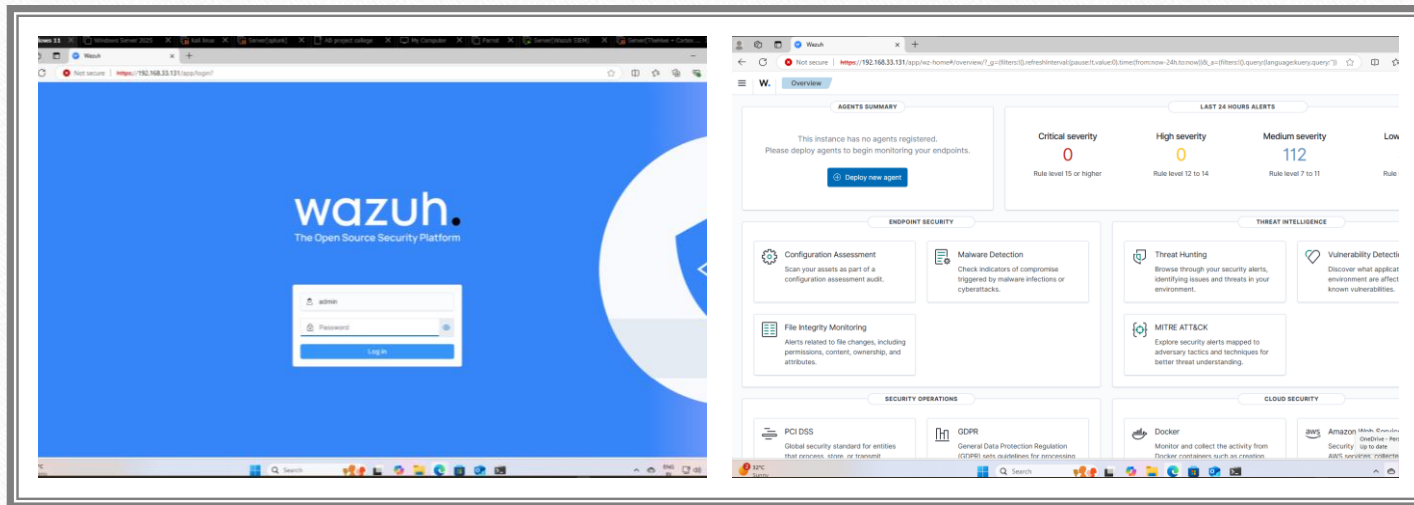
Role / Primary Tool	Operating System	IP Address	RAM	vCPUs	Hard Disk	Notes
Attacker Machine	Kali Linux	192.168.33.143/24	4gb	2	80gb	Used for launching simulated attacks.
Log Aggregation / Analysis	Ubuntu Server 22.04 LTS	192.168.33.128/24	4gb	2	80gb	Hosts Splunk Enterprise (Free License).
Domain Controller / 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



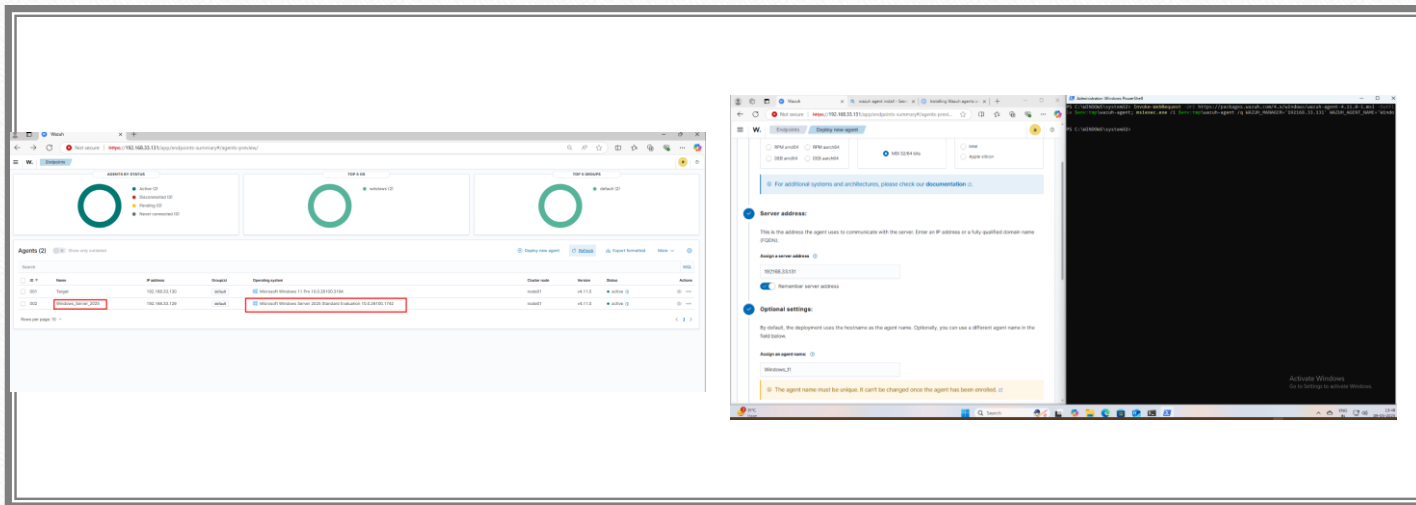


Implementation - Wazuh & Agents

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.

The screenshot displays the Suricata web interface for deploying agents. The left sidebar contains configuration steps: 'Server address' (with IP 192.168.33.131), 'Optional settings' (with agent name 'suricata'), and 'Run the following commands to download and install the agent' (with a curl command). The main panel shows a 'TS BY STATUS' donut chart and a 'TOP 5 OS' donut chart. Below these is a table listing deployed agents.

IP address	Group(s)	Operating system	
192.168.33.130	default	Microsoft Windows 11 Pro 10.0.26100.3194	n
192.168.33.129	default	Microsoft Windows Server 2025 Standard Evaluation 10.0.26100.1742	n
192.168.33.136	default	Debian GNU/Linux 12	n
192.168.33.135	default	Debian GNU/Linux 12	n
192.168.33.128	default	Ubuntu 22.04.5 LTS	n
192.168.33.134	default	Ubuntu 22.04.4 LTS	n

Implementation – Agents On Linux

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

```

killmongerwazuh@wazuhh:~$ sudo cat /opt/splunkforwarder/etc/
[sudo] password for killmongerwazuh:
[tcput]
defaultGroup = default-autolb-group

[tcput:default-autolb-group]
server = 192.168.33.128:9997

[tcput-server://192.168.33.128:9997]
killmongerwazuh@wazuhh:~$

killmongerwazuh@wazuhh:~$ cat /opt/splunkforwarder/etc/syst
[monitor:///var/ossec/logs/alerts/alerts.json]
disabled = false
index = wazuh_forwarded
sourcetype = wazuh_alerts
# Add more sourcetype stanzas if needed for other files, e.
# [monitor:///var/ossec/logs/ossec.log]
# disabled = false
# index = wazuh_forwarded
# sourcetype = wazuh_ossec

```

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).

```
# Configure the type of alert (and other) logging you would like.
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
      enabled: yes
      filename: fast.log
      append: yes
      filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Extensible Event Format (nicknamed EVE) event log in JSON format
  - eve-log:
      enabled: yes
      filetype: regular #regular[syslog[unix_dgram]unix_stream]redis
      filename: eve.json
      # Enable for multi-threaded eve.json output; output files are amended with
      # with an identifier, e.g., eve-0.json
      threaded: false
      prefix: "ace: " # prefix to prepend to each log entry
      # the following are valid when type: syslog above
```

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.33.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
```

Implementation - Suricata (NIDS)

```
killmongersuricata@suricataa:~$ sudo systemctl enable suricata
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
killmongersuricata@suricataa:~$ sudo systemctl restart suricata
killmongersuricata@suricataa:~$ sudo systemctl status suricata

• suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-05-02 21:24:22 IST; 25s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata-ids.org/docs/
  Process: 37167 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
 Main PID: 37169 (Suricata-Main)
    Tasks: 8 (limit: 2241)
  Memory: 596.4M
     CPU: 21.484s
    CGroup: /system.slice/suricata.service
            └─37169 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

May 02 21:24:22 suricataa systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
May 02 21:24:22 suricataa suricata[37167]: 2/5/2025 -- 21:24:22 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
May 02 21:24:22 suricataa systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
killmongersuricata@suricataa:~$
```

Service: Enabled and started suricata.service.

Integration: Logs monitored by local Wazuh agent.

Thank you for your interest in Greenbone Free!

We are pleased to offer you free access to our vulnerability management tool.

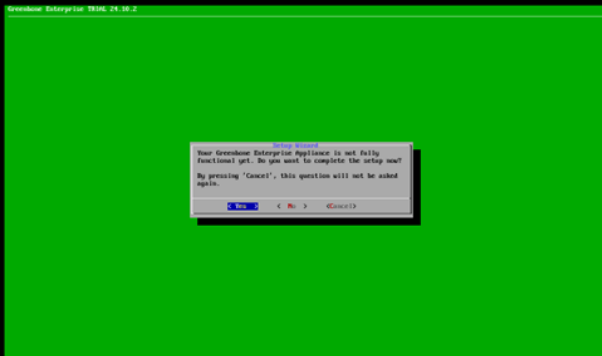
Your personal link to the product:

VMware Workstation Player/Pro:
<https://files.greenbone.net/download/delivery/website-trials-22.04-abdk-greenbone/greenbone-Enterprise-TRIAL-24.10.2-VMware-Workstation.ova>

SHA256 Checksum:
b98cd3c4fa8478094ff3bb648db171880560ca5ec1514888765ab9a5a5938d5b

Oracle VirtualBox:
<https://files.greenbone.net/download/delivery/website-trials-22.04-abdk-greenbone/greenbone-Enterprise-TRIAL-24.10.2-VirtualBox.ova>

SHA256 Checksum:
737964be627112f188534637aadb103f9851d796f19b15279def03a326c90a05



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)

New Task

Min QoS
70

Alterable Task
☐ Yes ☒ No

Auto Delete Reports
☒ Do not automatically delete reports
☐ Automatically delete oldest reports but always keep newest 5 reports

Scanner
OpenVAS Default

Scan Config
Full and fast

Order for target hosts
Sequential

Maximum concurrently executed NVTs per host
4

Maximum concurrently scanned hosts
20

Cancel Save

New Target

Given hosts was invalid

Name
windows devices

Comment

Hosts
☒ Manual 192.168.33.190, 192.168.33.129
☐ From file

Exclude hosts
☒ Manual
☐ From file

Allow simultaneous scanning via multiple IPs
☒ Yes ☐ No

Port List
All what assigned TCP

Cancel Save

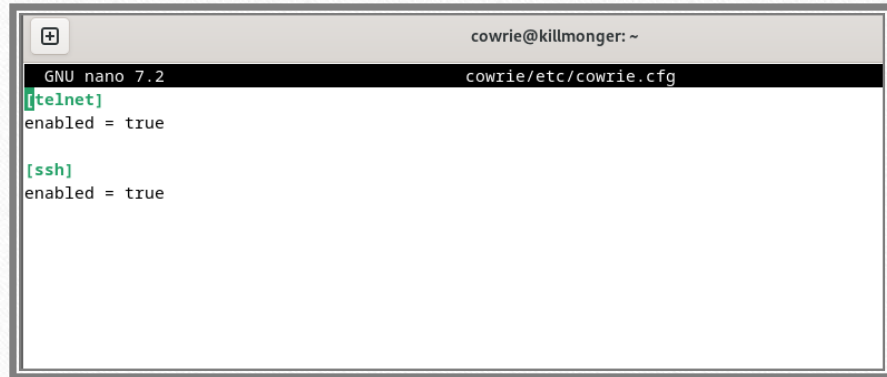
Feed Sync: Automatic synchronization of NVT/SCAP/CERT data. **Configuration:**

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.



```
cowrie@killmonger: ~
GNU nano 7.2 cowrie/etc/cowrie.cfg
[te]net]
enabled = true

[ssh]
enabled = true
```


Implementation - Cowrie (Honeytrap)

- **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.

```
killmonger@killmonger:~$ sudo tail -n 10 /home/cowrie/cowrie/var/log/cowrie/cowrie.json
[sudo] password for killmonger:
{"eventid": "cowrie.session.connect", "src_ip": "192.168.33.143", "src_port": 53216, "dst_ip": "192.168.33.142", "dst_port": 22, "session": "8eadcf3cf01c", "protocol": "ssh", "message": "New connection: 192.168.33.143:53216 (192.168.33.142:22) [session: 8eadcf3cf01c]", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:15.355208Z"}
{"eventid": "cowrie.client.version", "version": "SSH-2.0-OpenSSH_9.2p1 Debian-2", "message": "Remote SSH version: SSH-2.0-OpenSSH_9.2p1 Debian-2", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:15.357196Z", "src_ip": "192.168.33.143", "session": "8eadcf3cf01c"}
{"eventid": "cowrie.client.kex", "hash": "78c85d999799066a2b4554ce7b1585a6", "hashAlgorithms": "sntrup761x25519-sha512@openssh.com,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-sha256,ext-info-c,chaCha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1,none,zlib@openssh.com,zlib", "keyAlgs": "ssh-ed25519-cert-v01@openssh.com", "ecdsa-sha2-nistp256-cert-v01@openssh.com", "ecdsa-sha2-nistp384-cert-v01@openssh.com", "ecdsa-sha2-nistp521-cert-v01@openssh.com", "sk-ssh-ed25519-cert-v01@openssh.com", "sk-ecdsa-sha2-nistp256-cert-v01@openssh.com", "rsa-sha2-512-cert-v01@openssh.com", "rsa-sha2-256-cert-v01@openssh.com", "ssh-ed25519", "ecdsa-sha2-nistp256", "ecdsa-sha2-nistp384", "ecdsa-sha2-nistp521", "sk-ssh-ed25519@openssh.com", "sk-ecdsa-sha2-nistp256@openssh.com", "rsa-sha2-512", "rsa-sha2-256", "none", "zlib@openssh.com", "zlib", "langCS": "[]", "message": "SSH client hash fingerprint: 78c85d999799066a2b4554ce7b1585a6", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:15.359682Z", "src_ip": "192.168.33.143", "session": "8eadcf3cf01c"}
{"eventid": "cowrie.login.failed", "username": "killmongercowrie", "password": "\u001b[A\u001b[Asudarshan", "message": "login attempt [killmongercowrie/\u001b[A\u001b[Asudarshan] failed", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:27.871659Z", "src_ip": "192.168.33.143", "session": "8eadcf3cf01c"}
{"eventid": "cowrie.login.failed", "username": "killmongercowrie", "password": "sudarshan", "message": "login attempt [killmongercowrie/sudarshan] failed", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:35.568487Z", "src_ip": "192.168.33.143", "session": "8eadcf3cf01c"}
{"eventid": "cowrie.login.failed", "username": "killmongercowrie", "password": "sudarshan@1231", "message": "login attempt [killmongercowrie/sudarshan@1231] failed", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:45.126951Z", "src_ip": "192.168.33.143", "session": "8eadcf3cf01c"}
{"eventid": "cowrie.session.closed", "duration": "38.8", "message": "Connection lost after 38.8 seconds", "sensor": "killmonger", "timestamp": "2025-04-27T11:47:46.131172Z", "src_ip": "192.168.33.143", "session": "8eadcf3cf01c"}
killmonger@killmonger:~$
```

Implementation

- Splunk (Log Aggregation)

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

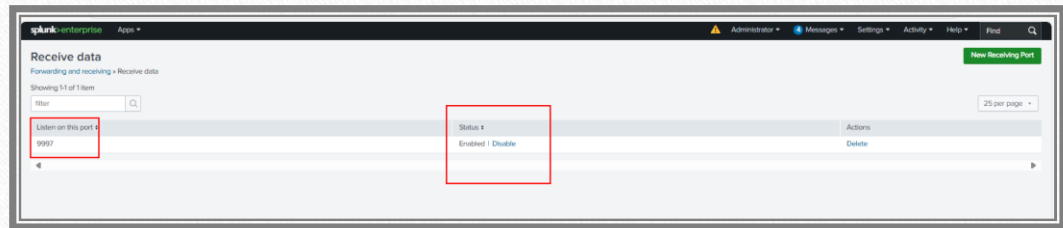
```
root@splunk:~/home/killmongersplunk# wget -O splunk.deb "https://download.splunk.com/products/splunk/releases/9.4.1/linux/splunk-9.4.1-e3bdab203ac8-linux-amd64.deb"
--2025-03-11 07:16:32-- https://download.splunk.com/products/splunk/releases/9.4.1/linux/splunk-9.4.1-e3bdab203ac8-linux-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 54.192.142.58, 54.192.142.97, 54.192.142.38, ...
Connecting to download.splunk.com (download.splunk.com)|54.192.142.58|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 921195836 (879M) [binary/octet-stream]
Saving to: 'splunk.deb'

splunk.deb           49%[=====>] 1 439.14M  5.07MB/s  eta 98s
```

Implementation - Splunk (Log Aggregation)

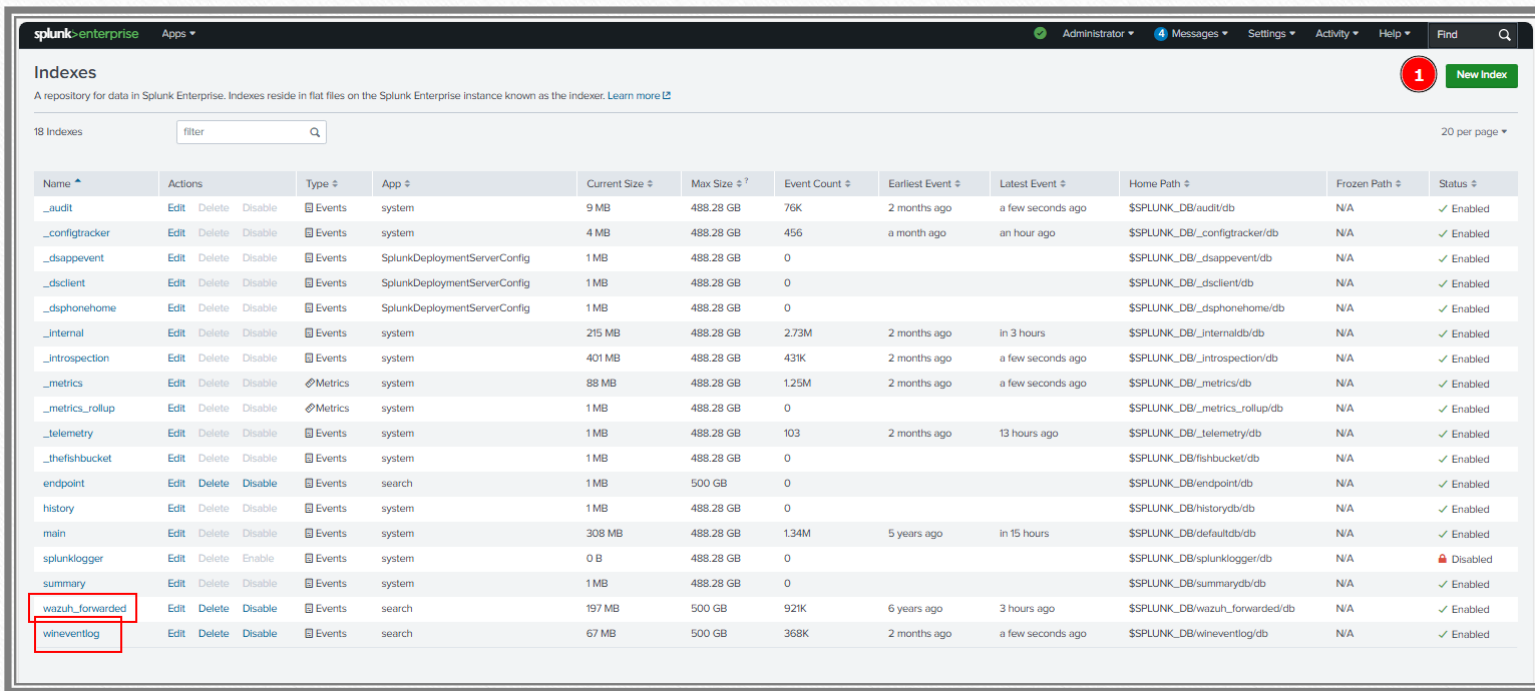
Configuration:

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



Implementation - Splunk (Log Aggregation)

Created custom indexes (wineventlog, wazuh_forwarded).



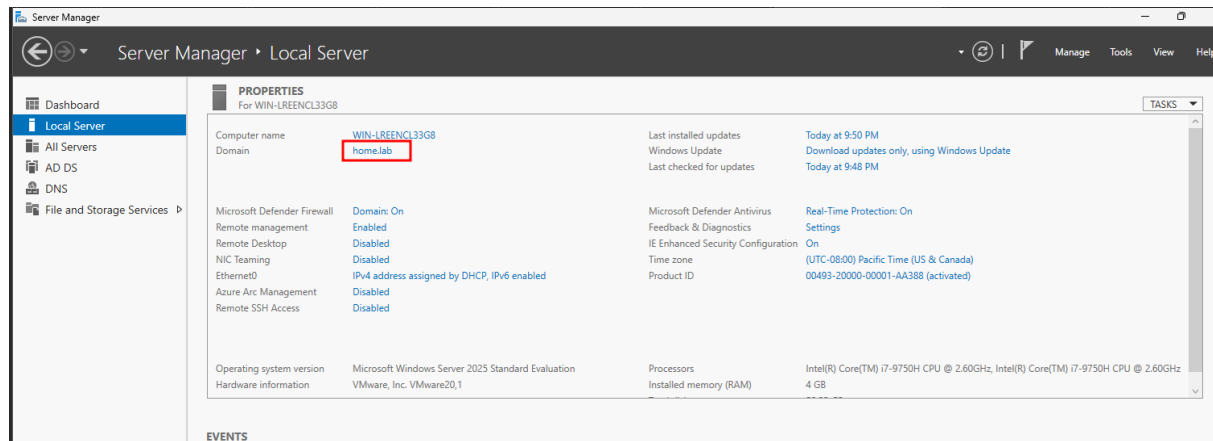
splunk enterprise Apps ▾ Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find 🔍

Indexes

A repository for data in Splunk Enterprise. Indexes reside in flat files on the Splunk Enterprise instance known as the **indexer**. [Learn more](#)

18 Indexes 20 per page ▾

Name ▴	Actions	Type ▴	App ▴	Current Size ▴	Max Size ▴?	Event Count ▴	Earliest Event ▴	Latest Event ▴	Home Path ▴	Frozen Path ▴	Status ▴
__audit	Edit Delete Disable	Events	system	9 MB	488.28 GB	76K	2 months ago	a few seconds ago	\$SPLUNK_DB/audit/db	N/A	✓ Enabled
__configtracker	Edit Delete Disable	Events	system	4 MB	488.28 GB	456	a month ago	an hour ago	\$SPLUNK_DB/__configtracker/db	N/A	✓ Enabled
__dsappevent	Edit Delete Disable	Events	SplunkDeploymentServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/__dsappevent/db	N/A	✓ Enabled
__dsclient	Edit Delete Disable	Events	SplunkDeploymentServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/__dsclient/db	N/A	✓ Enabled
__dsphonehome	Edit Delete Disable	Events	SplunkDeploymentServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/__dsphonehome/db	N/A	✓ Enabled
__internal	Edit Delete Disable	Events	system	215 MB	488.28 GB	2.73M	2 months ago	in 3 hours	\$SPLUNK_DB/__internaldb/db	N/A	✓ Enabled
__introspection	Edit Delete Disable	Events	system	401 MB	488.28 GB	431K	2 months ago	a few seconds ago	\$SPLUNK_DB/__introspection/db	N/A	✓ Enabled
__metrics	Edit Delete Disable	Metrics	system	88 MB	488.28 GB	1.25M	2 months ago	a few seconds ago	\$SPLUNK_DB/__metrics/db	N/A	✓ Enabled
__metrics_rollup	Edit Delete Disable	Metrics	system	1 MB	488.28 GB	0			\$SPLUNK_DB/__metrics_rollup/db	N/A	✓ Enabled
__telemetry	Edit Delete Disable	Events	system	1 MB	488.28 GB	103	2 months ago	13 hours ago	\$SPLUNK_DB/__telemetry/db	N/A	✓ Enabled
__thefishbucket	Edit Delete Disable	Events	system	1 MB	488.28 GB	0			\$SPLUNK_DB/__thefishbucket/db	N/A	✓ Enabled
endpoint	Edit Delete Disable	Events	search	1 MB	500 GB	0			\$SPLUNK_DB/endpoint/db	N/A	✓ Enabled
history	Edit Delete Disable	Events	system	1 MB	488.28 GB	0			\$SPLUNK_DB/historydb/db	N/A	✓ Enabled
main	Edit Delete Disable	Events	system	308 MB	488.28 GB	1.34M	5 years ago	in 15 hours	\$SPLUNK_DB/defaultdb/db	N/A	✓ Enabled
splunklogger	Edit Delete Enable	Events	system	0 B	488.28 GB	0			\$SPLUNK_DB/splunklogger/db	N/A	⛔ Disabled
summary	Edit Delete Disable	Events	system	1 MB	488.28 GB	0			\$SPLUNK_DB/summarydb/db	N/A	✓ Enabled
wazuh_forwarded	Edit Delete Disable	Events	search	197 MB	500 GB	921K	6 years ago	3 hours ago	\$SPLUNK_DB/wazuh_forwarded/db	N/A	✓ Enabled
wineventlog	Edit Delete Disable	Events	search	67 MB	500 GB	368K	2 months ago	a few seconds ago	\$SPLUNK_DB/wineventlog/db	N/A	✓ Enabled



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)

Device specifications

Copy

Device name

Target

Full device name

Target.home.lab

Processor

Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz (2 processors)

Installed RAM

4.00 GB

Device ID

BEF107F0-147D-4AF1-972D-F1C0851C55B8

Product ID

00330-80000-00000-AA754

System type

64-bit operating system, x64-based processor

Pen and touch

No pen or touch input is available for this display

Related links

[Domain or workgroup](#)

[System protection](#)

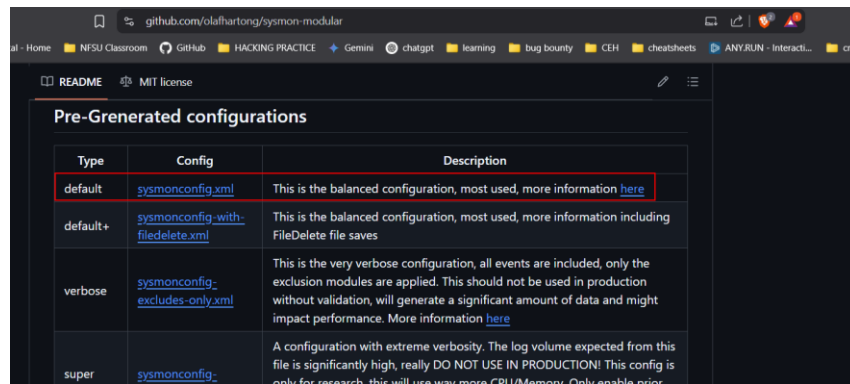
[Advanced system settings](#)

Implementation - Active Directory & Endpoints

Endpoint: Windows 11 joined to home.lab domain.

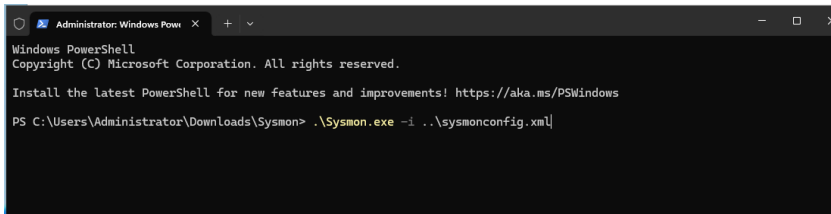
Implementation - Active Directory & Endpoints

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



The screenshot shows the GitHub repository for `sysmon-modular` by `olafhartong`. The page displays a table titled "Pre-Generated configurations" with the following data:

Type	Config	Description
default	sysmonconfig.xml	This is the balanced configuration, most used, more information here
default+	sysmonconfig-with-filedelete.xml	This is the balanced configuration, most used, more information including FileDelete file saves
verbose	sysmonconfig-excludes-only.xml	This is the very verbose configuration, all events are included, only the exclusion modules are applied. This should not be used in production without validation, will generate a significant amount of data and might impact performance. More information here
super	sysmonconfig-	A configuration with extreme verbosity. The log volume expected from this file is significantly high, really DO NOT USE IN PRODUCTION! This config is only for research, this will use way more CPU/Memory. Only enable prior



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Administrator\Downloads\Sysmon> .\Sysmon.exe -i ..\sysmonconfig.xml
```



```
killmongershuffle@shufflee:~/Shuffle$ docker compose up -d
[+] Running 4/4
  ✓ Container shuffle-opensearch Running 0.0s
  ✓ Container shuffle-backend Running 0.0s
  ✓ Container shuffle-frontend Running 0.0s
  ✓ Container shuffle-orborus Running 0.0s
killmongershuffle@shufflee:~/Shuffle$
```

Implementation - Shuffle (SOAR)

Installation: Deployed using Docker Compose on Ubuntu.

Implementation - Shuffle (SOAR)

Integration with Wazuh:

- Created Webhook listener in Shuffle.

The screenshot displays the Shuffle (SOAR) interface. On the left, a workflow diagram shows a 'Telegram alert' node (marked with a red circle 1) connected to a 'Webhook alert to shuffle' node (marked with a red circle 2). The 'Webhook alert to shuffle' node is highlighted with a red box. On the right, the 'Webhook: SUCCESS' configuration panel is shown. It includes fields for 'Name' (wazuh alert to shuffle, marked with a red circle 3), 'Associated App (optional)', 'Environment' (onprem), and 'Parameters'. The 'Parameters' section has a 'Webhook URI' field (marked with a red circle 4) containing the URL 'https://192.168.33.149:3443/api/'. Below this is a 'START' button and a 'STOP' button. The 'Authentication headers' section is also visible, showing 'AUTH_HEADER=AUTH_VALUE1'. At the bottom, there is a message about activating Windows.

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


```
(killmonger@kalii)-[~]  
$ ssh testforproject@192.168.33.142 -p 2222  
testforproject@192.168.33.142's password:  
Permission denied, please try again.  
testforproject@192.168.33.142's password:  
Permission denied, please try again.  
testforproject@192.168.33.142's password:  
testforproject@192.168.33.142: Permission denied (publickey,password).
```

Testing & Output

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

Testing & Output

```
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'
2025-05-03T05:19:33.351504Z [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'2ndattemptmade'] 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'cowrieisworkingthink'] 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
```

Detection:

- Cowrie logged the attempt.

Testing & Output

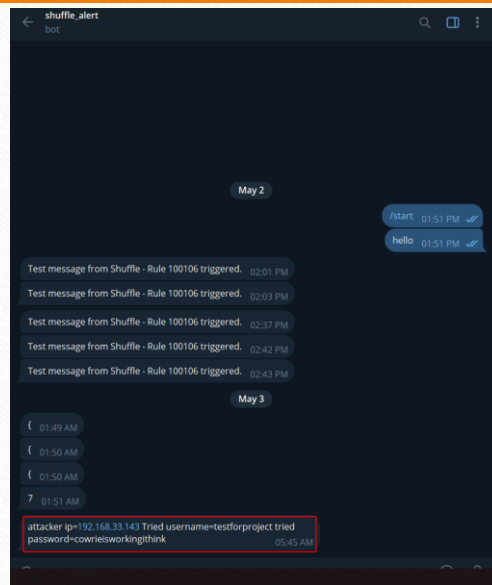
- Wazuh agent sent log to Manager.
- Wazuh generated high-severity alert (Rule 120003).

```
< local_rules.xml
Ruleset Test Save
0 <!-- example -->
7 <group name="local,syslog,sshd,">
8
9 <!--
10 Dec 10 01:02:02 host sshd[1234]: Failed none for root from 1.1.1.1 port 1066 ssh2
11 -->
12 <rule id="100001" level="5">
13 <if_sid>5716</if_sid>
14 <srcip>1.1.1.1</srcip>
15 <description>sshd: authentication failed from IP 1.1.1.1.</description>
16 <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
17 </rule>
18 <rule id="120001" level="3">
19 <decoded_as>json</decoded_as>
20 <field name="eventId">cowrie.session.connect</field>
21 <description>Cowrie Honeypot: New Connection from $(src_ip)</description>
22 <group>connection_attempt,</group>
23 </rule>
24
25 <rule id="120003" level="15">
26 <decoded_as>json</decoded_as>
27 <field name="eventId">cowrie.login.failed</field>
28 <description>Cowrie Honeypot: Failed Login from $(src_ip) using user [$(username)] and password [$(password)]</description>
29 <mitre id="T1110">id</mitre>
30 <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
31 </rule>
32
33 <rule id="100101" level="5">
34 <if_sid>60122</if_sid> <field name="win.eventdata.status">0xc0000064</field>
35 <description>Windows: Logon attempt with non-existent or misspelled username $(win.eventdata.targetUserName) from $(win.eventdata.ipAddress).</description>
36 <group>authentication_failure,pci_dss_10.2.5,gdpr_IV_32.2,</group>
37 </rule>
38
39 <rule id="100102" level="4">
40 <if_sid>60107</if_sid> <field name="win.eventdata.image" type="pore2">{?
41 <description>Windows: Test command "hostname.exe" executed on $(agent.name)
42 <group>test_rule,execution,</group>
43 </rule>
```


Testing & Output

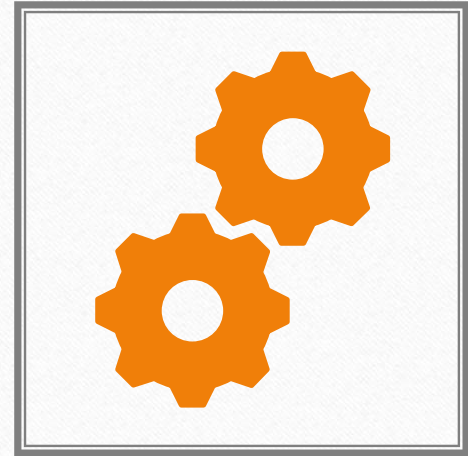
Automation:

- 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