Day 06 Report: SOC Fundamentals and Operations

Objective

The objective of this lab was to practice SOC fundamentals through hands-on exercises. Tools used included Nessus for vulnerability scanning, Snort for intrusion detection, Elastic SIEM for log analysis, Osquery for endpoint monitoring, and Wazuh for advanced SIEM correlation. The lab simulated core SOC workflows such as detection, triage, investigation, response, and documentation.

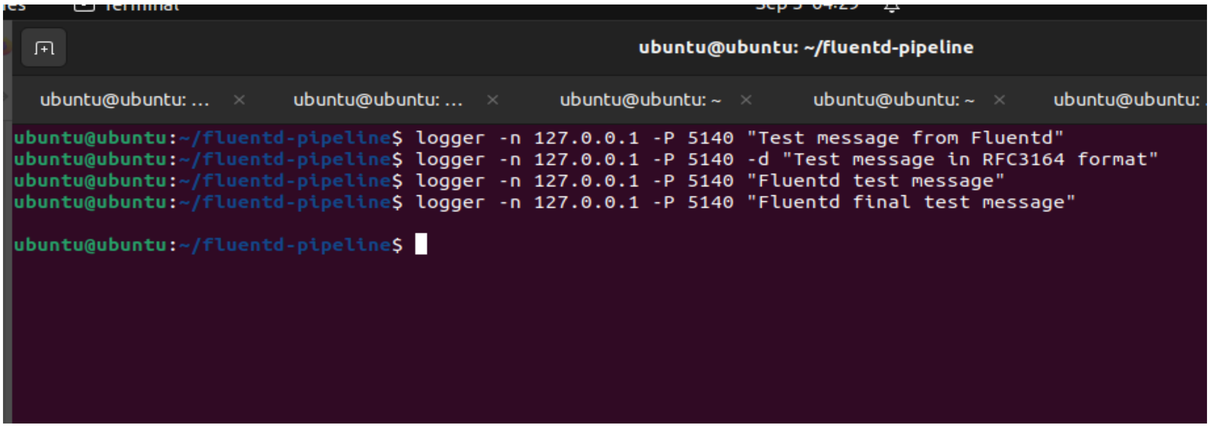
1. Log Collection Pipeline (Fluentd)

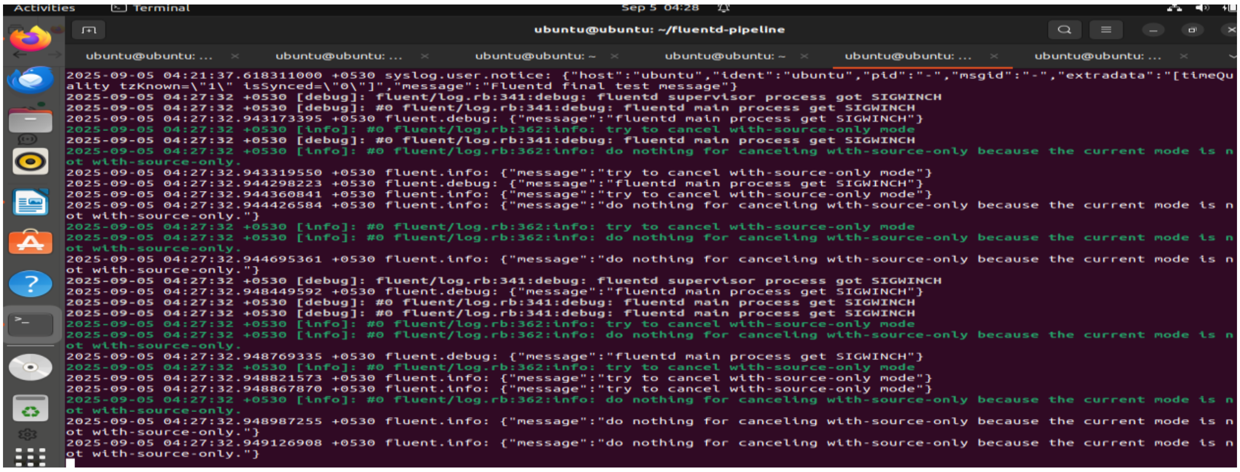
Methodology:

- Installed Ruby and required dependencies on Ubuntu.
- Installed Fluentd via RubyGems (gem install fluentd).
- Created a custom configuration file (fluent.conf).
- Configured Fluentd to listen on UDP port 5140 using the syslog input plugin.
- Set output to stdout for verification.
- Generated test logs using the logger command.

Findings:

- Fluentd successfully received and parsed syslog messages on port 5140.
- Test messages sent with logger were displayed in stdout as structured JSON.
- Validated that the pipeline works end-to-end for centralized log collection.
- Pipeline can be extended to forward logs to Elasticsearch, OpenSearch, or SIEM tools.





2. Nessus Vulnerability Scan (Metasploitable2)

Methodology:

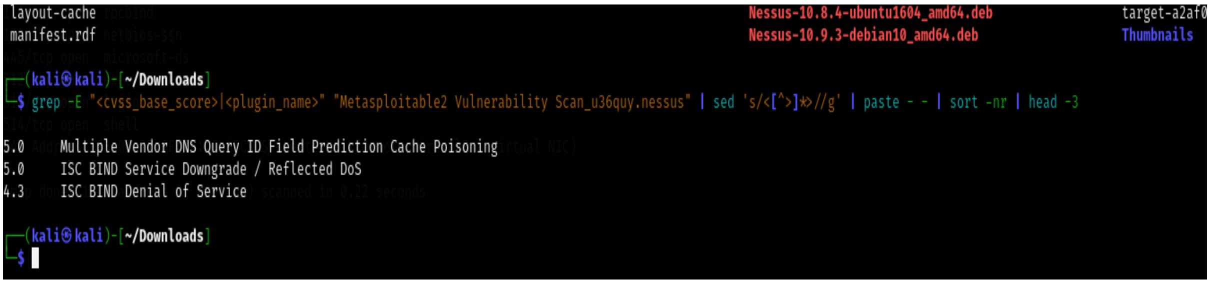
- Installed Nessus on Kali attacker VM (192.168.1.79).
- Scanned Metasploitable2 target VM (192.168.1.78).
- Exported scan results in CSV format.
- Analyzed top 3 vulnerabilities by CVSS score.

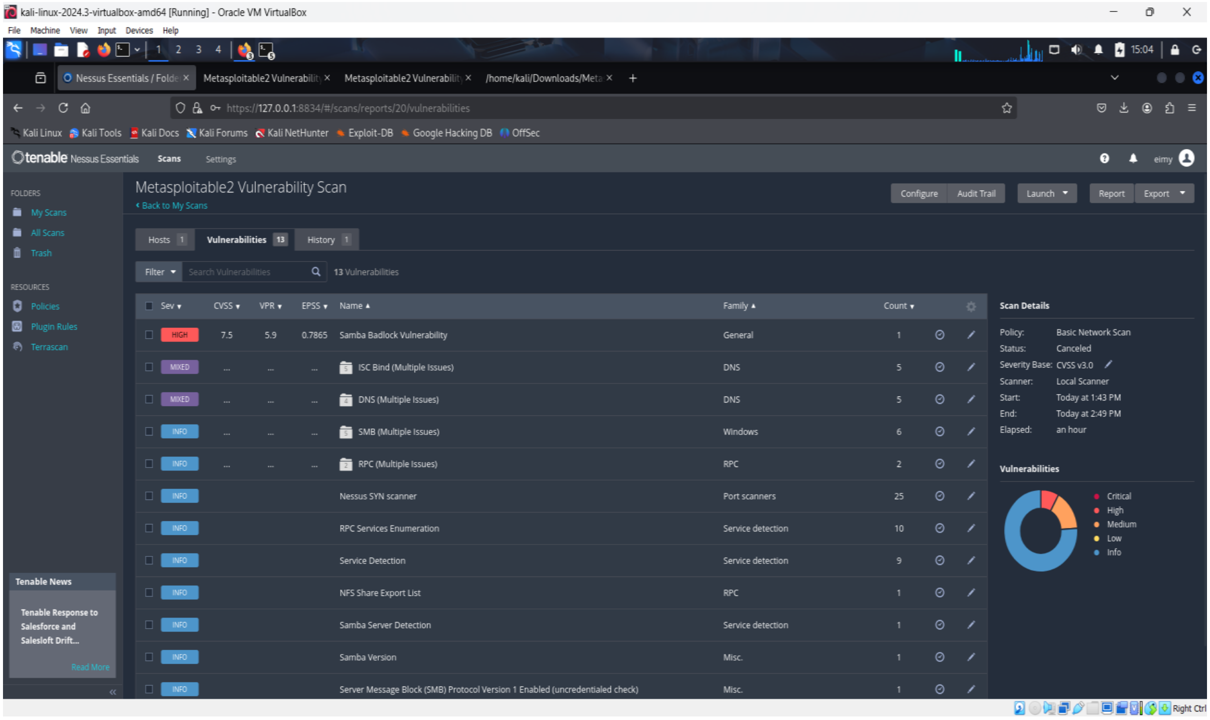
Findings:

The scan against **Metasploitable2 (192.168.1.78)** identified several vulnerabilities.

The **Top 3 vulnerabilities by CVSS score** are:

1. **CVSS 5.0 – Multiple Vendor DNS Query ID Field Prediction Cache Poisoning**
2. **CVSS 5.0 – ISC BIND Service Downgrade / Reflected DoS**
3. **CVSS 4.3 – ISC BIND Denial of Service**





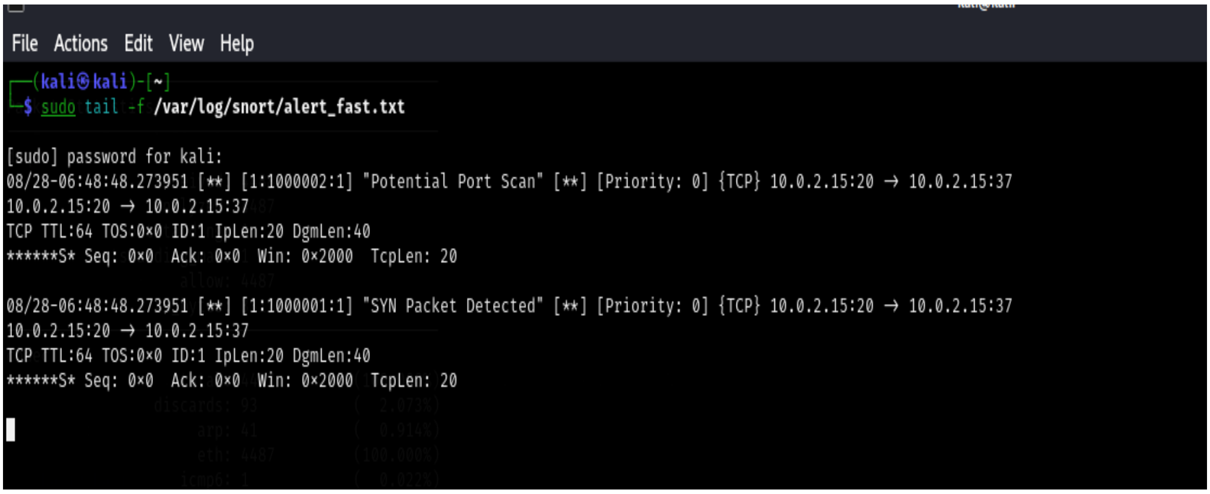
3. <u>Snort IDS Rule for Malicious Domain</u>

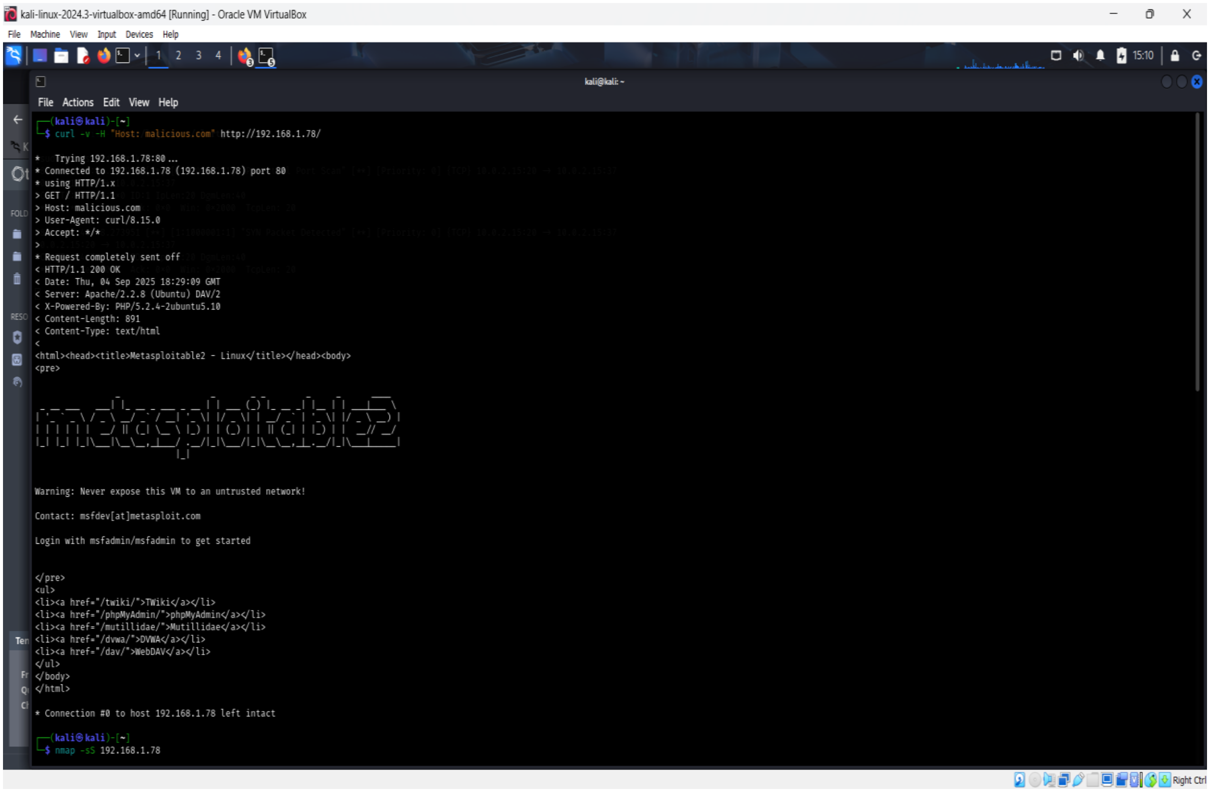
Methodology:

- Installed Snort on Kali attacker VM.
- Configured HOME_NET and added custom rule in local.rules to detect malicious.com in Host header.
- Ran Snort in alert mode listening on eth0.
- Used curl with forged Host header to trigger detection.

Findings:

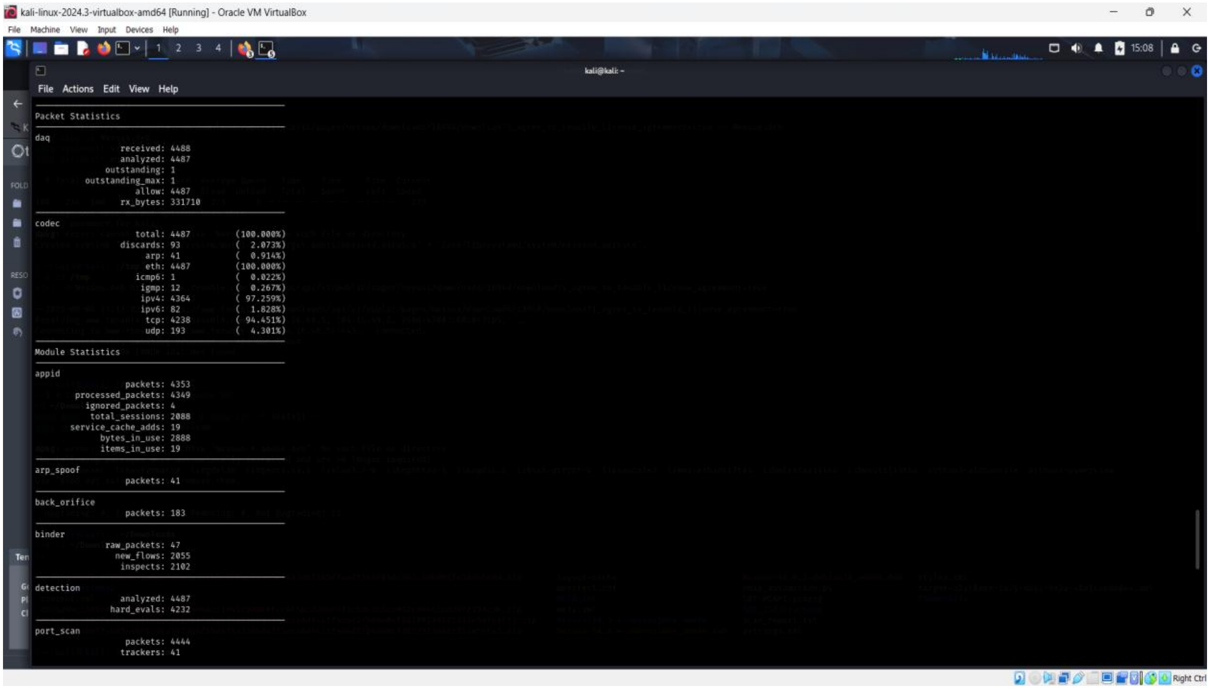
Snort successfully generated alerts when detecting HTTP traffic with Host header malicious.com.

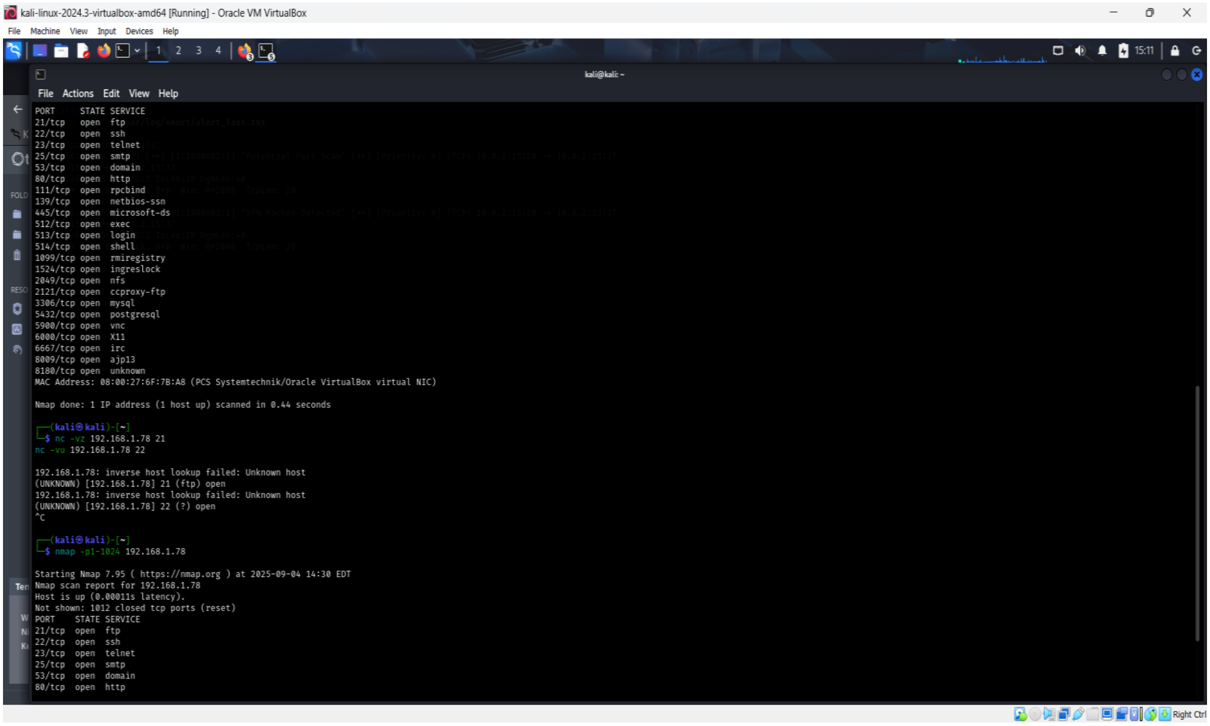

File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo tail -f /var/log/snort/alert_fast.txt
[sudo] password for kali:
08/28-06:48:48.273951 [**] [1:1000002:1] "Potential Port Scan" [**] [Priority: 0] {TCP} 10.0.2.15:20 → 10.0.2.15:37
10.0.2.15:20 → 10.0.2.15:37
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0 Ack: 0x0 Win: 0x2000 TcpLen: 20
08/28-06:48:48.273951 [**] [1:1000001:1] "SYN Packet Detected" [**] [Priority: 0] {TCP} 10.0.2.15:20 → 10.0.2.15:37
10.0.2.15:20 → 10.0.2.15:37
TCP TTL:64 TOS:0x0 ID:1 IpLen:20 DgmLen:40
******S* Seq: 0x0 Ack: 0x0 Win: 0x2000 TcpLen: 20


kali-linux-2024.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ curl -v -H "Host: malicious.com" http://192.168.1.78/
* Trying 192.168.1.78:80 ...
* Connected to 192.168.1.78 (192.168.1.78) port 80
* using HTTP/1.x
> GET / HTTP/1.1
> Host: malicious.com
> User-Agent: curl/8.15.0
> Accept: */*
* Request completely sent off
< HTTP/1.1 200 OK
< Date: Thu, 04 Sep 2025 18:29:09 GMT
< Server: Apache/2.2.8 (Ubuntu) DAV/2
< X-Powered-By: PHP/5.2.4-2ubuntu5.10
< Content-Length: 891
< Content-Type: text/html
<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started
</pre>

TWiki
phpMyAdmin
Mutillidae
DVWA
WebDAV

</body>
</html>
* Connection #0 to host 192.168.1.78 left intact
(kali㉿kali)-[~]
$ nmap -sS 192.168.1.78


kali-linux-2024.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
Packet Statistics
daq
received: 4488
analyzed: 4487
outstanding: 1
outstanding_max: 1
allow: 4487
rx_bytes: 331710
codec
total: 4487 (100.000%)
discards: 93 (2.073%)
arp: 41 (0.914%)
eth: 4487 (100.000%)
icmp6: 1 (0.022%)
igmp: 12 (0.267%)
ipv4: 4364 (97.259%)
ipv6: 82 (1.828%)
tcp: 4238 (94.451%)
udp: 193 (4.301%)
Module Statistics
appid
packets: 4353
processed_packets: 4349
ignored_packets: 4
total_sessions: 2088
service_cache_adds: 19
bytes_in_use: 2888
items_in_use: 19
arp_spoof
packets: 41
back_orifice
packets: 183
binder
raw_packets: 47
new_flows: 2055
inspects: 2102
detection
analyzed: 4487
hard_evals: 4232
port_scan
packets: 4444
trackers: 41
15:08
Right Ctrl


kali-linux-2024.3-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
23/tcp open telnet
25/tcp open smtp
53/tcp open domain
80/tcp open http
111/tcp open rpcbind
139/tcp open netbios-ssn
445/tcp open microsoft-ds
512/tcp open exec
513/tcp open login
514/tcp open shell
1099/tcp open rmiregistry
1524/tcp open ingreslock
2049/tcp open nfs
2121/tcp open ccproxy-ftp
3306/tcp open mysql
5432/tcp open postgresql
5900/tcp open vnc
6000/tcp open X11
6667/tcp open irc
8009/tcp open ajp13
8180/tcp open unknown
MAC Address: 08:00:27:6F:7B:A8 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds
(kali㉿kali)-[~]
$ nc -vz 192.168.1.78 21
nc -vu 192.168.1.78 22
192.168.1.78: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.78] 21 (ftp) open
192.168.1.78: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.78] 22 (?) open
^C
(kali㉿kali)-[~]
$ nmap -p1-1024 192.168.1.78
Starting Nmap 7.95 (https://nmap.org) at 2025-09-04 14:30 EDT
Nmap scan report for 192.168.1.78
Host is up (0.00011s latency).
Not shown: 1012 closed tcp ports (reset)
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
23/tcp open telnet
25/tcp open smtp
53/tcp open domain
80/tcp open http
15:11
Right Ctrl

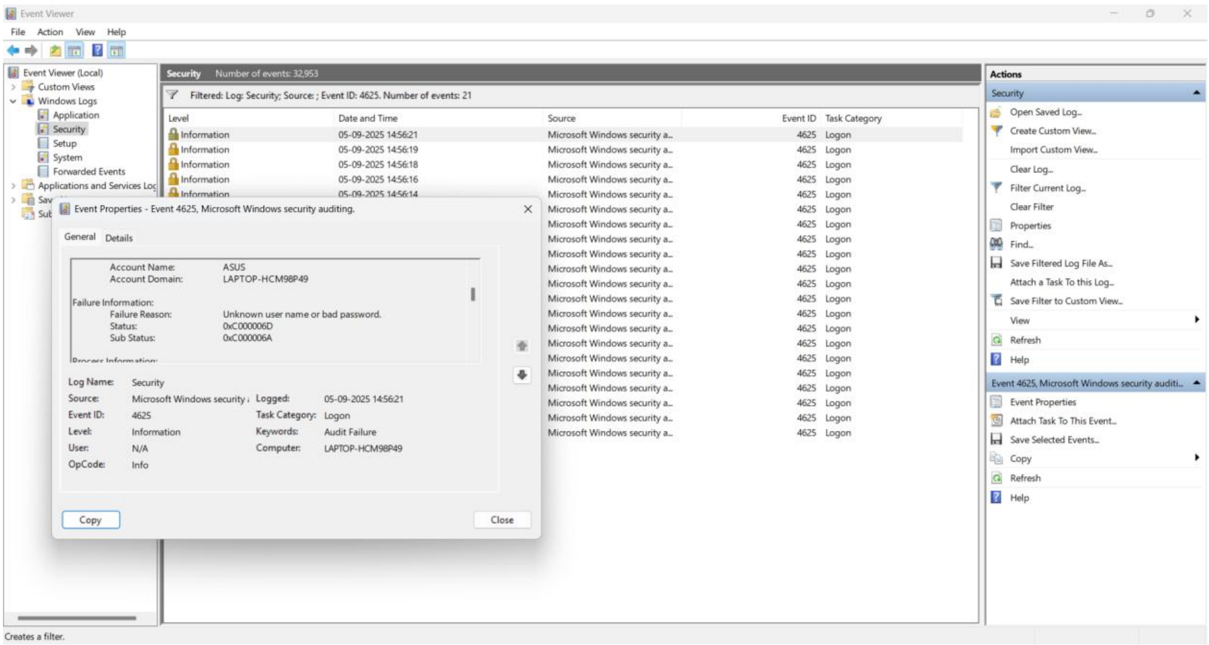
4. Brute-Force Detection (Windows Event Logs)

Methodology:

- Attempted multiple failed logins on the Windows VM using incorrect credentials.
- Opened Event Viewer → Windows Logs → Security.
- Filtered logs for Event ID 4625 (failed logon attempts).
- Analyzed event details: target account, logon type, failure reason, and source IP.
- Exported filtered results in CSV format for further analysis.

Findings:

- Event ID 4625 entries confirmed failed login attempts.
- Targeted account: ASUS (non-existent / wrong password).
- Logon Type: 2 → Interactive (local login attempt).
- Failure Reason: Unknown user name or bad password.
- Source Network Address: 127.0.0.1 (local machine).
- Demonstrated how repeated 4625 events can reveal brute-force behavior and provide forensic evidence for SIEM monitoring.



5. Osquery Endpoint Monitoring

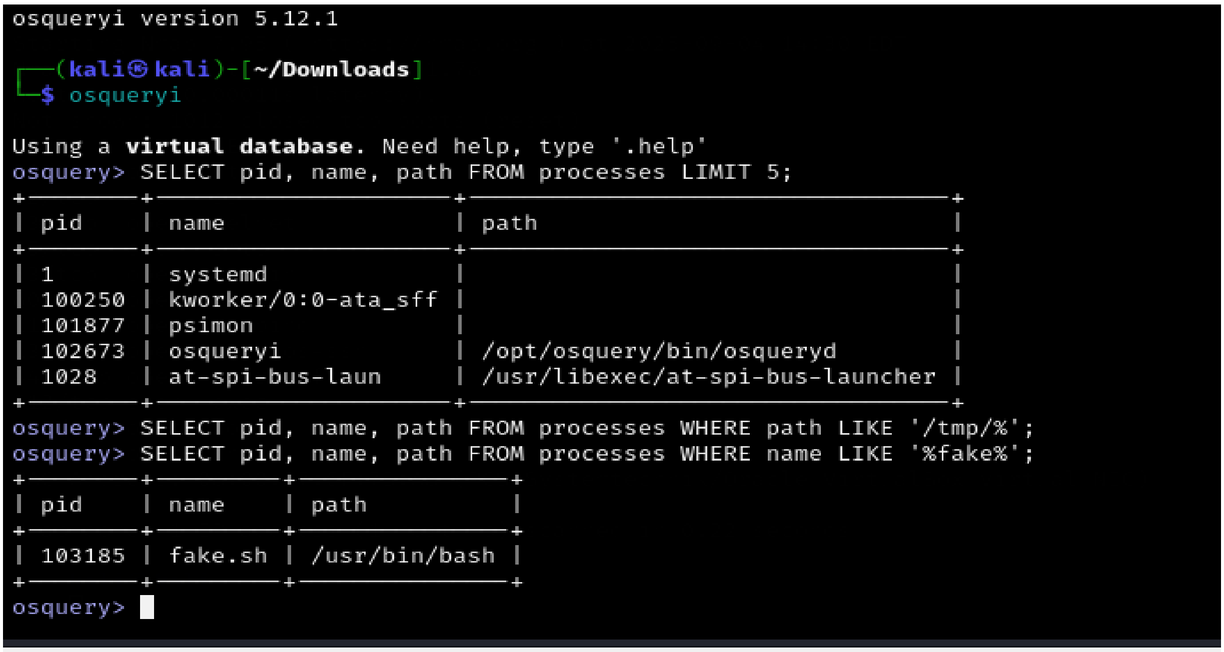
Methodology

1. Installed **Osquery** on Kali attacker VM.
2. Queried running processes using SQL-like syntax (SELECT * FROM processes;).
3. Created a fake process (/tmp/fake.sh) to simulate malicious activity.
4. Verified fake process visibility with:
5. SELECT pid, name, path FROM processes WHERE name LIKE '%fake%';

Findings

- Osquery successfully detected the suspicious process.
- Endpoint monitoring allows analysts to identify unusual or malicious binaries running on the system.

```
—(kali㉿kali)-[~]
—$ echo -e '#!/bin/bash\nsleep 1000' > /tmp/fake.sh
hmod +x /tmp/fake.sh
tmp/fake.sh &
1] 103185
—(kali㉿kali)-[~]
—$ ps aux | grep fake.sh
ali      103185  0.0  0.1   7088  3124 pts/6    SN   15:56   0:00 /bin/bash /tmp/fake.sh
ali      103400  0.0  0.1   6660  2248 pts/6    S+   15:57   0:00 grep --color=auto fake.sh
```



```
osqueryi version 5.12.1

┌──(kali㉿kali)-[~/Downloads]
└─$ osqueryi

Using a virtual database. Need help, type '.help'
osquery> SELECT pid, name, path FROM processes LIMIT 5;
+--------+-------------------+-----------------------------------+
| pid    | name              | path                              |
+--------+-------------------+-----------------------------------+
| 1      | systemd           |                                   |
| 100250 | kworker/0:0-ata_sff |                                 |
| 101877 | psimon            |                                   |
| 102673 | osqueryi          | /opt/osquery/bin/osqueryd         |
| 1028   | at-spi-bus-laun   | /usr/libexec/at-spi-bus-launcher  |
+--------+-------------------+-----------------------------------+
osquery> SELECT pid, name, path FROM processes WHERE path LIKE '/tmp/%';
osquery> SELECT pid, name, path FROM processes WHERE name LIKE '%fake%';
+--------+---------+---------------+
| pid    | name    | path          |
+--------+---------+---------------+
| 103185 | fake.sh | /usr/bin/bash |
+--------+---------+---------------+
osquery>
```

6. Zimmerman Tools Practice (Chrome History Analysis)

Methodology:

- Located Chrome history database at: C:\Users\<User>\AppData\Local\Google\Chrome\User Data\Default\History.
- Downloaded and extracted **SQLECmd** from Eric Zimmerman's GitHub.
- Executed SQLECmd on the Chrome History SQLite file:
- SQLECmd.exe -f "C:\Users\<User>\AppData\Local\Google\Chrome\User Data\Default\History" --csv "C:\Users\<User>\Desktop\ChromeHistory"
- Exported parsed browsing data into a CSV file.
- Opened CSV in Excel and searched for the test URL (http://test.com).

Findings:

- SQLECmd successfully parsed the Chrome history database.
- The test URL (http://test.com) was found in the browsing history.
- Exported data included details such as URL, timestamp, and visit count.
- Demonstrated that browser artifacts can be extracted and analyzed for forensic evidence.

7. Wazuh SIEM Integration

Methodology

- Installed Wazuh Manager and deployed agents on test VMs.
- Configured log forwarding (Windows Event Logs + syslog).
- Created custom rule sets for:
  - **User account changes (Rule ID: 60110)**
  - **Logon failures – unknown user or bad password (Rule ID: 60122)**
- Monitored events via Wazuh dashboard.

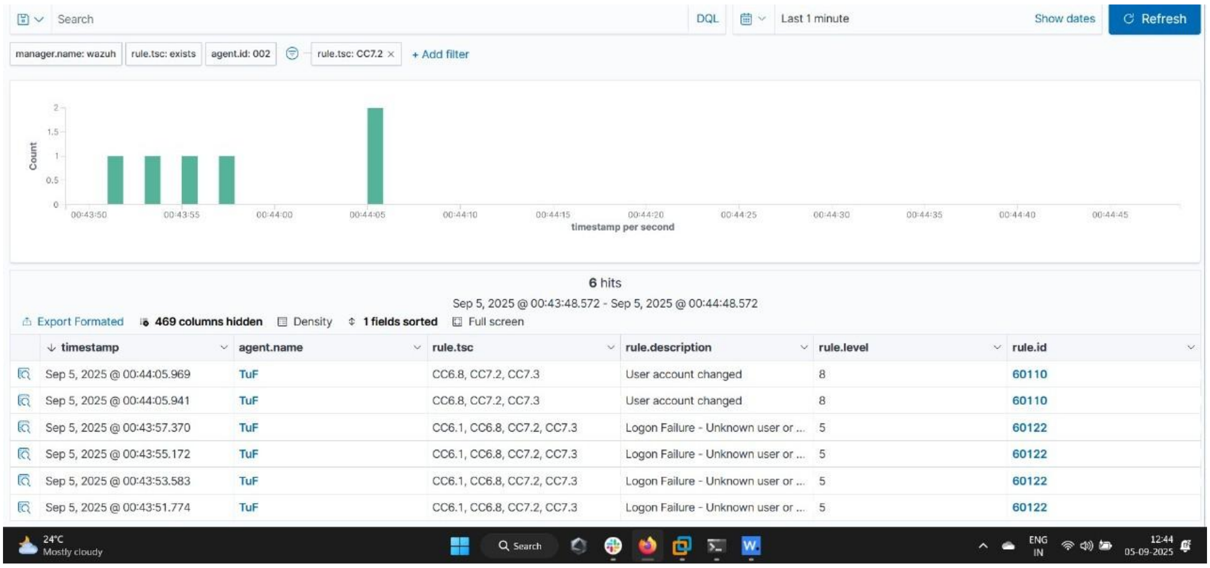
Findings

- Wazuh successfully detected both **user account modification events** and **failed logon attempts**.
- Events were categorized with different **rule levels** (5 for failed logins, 8 for account changes).

- The dashboard visualization confirmed multiple logon failures within seconds, correlating to brute-force simulation.

Sample Detection (from dashboard):

Timestamp	Agent Name	Rule ID	Rule Description	Rule Level	Action Taken
2025-09-05 00:44:05	TuF	60110	User account changed	8	Alert logged, verified
2025-09-05 00:43:57	TuF	60122	Logon Failure – Unknown user or bad password	5	Multiple failures detected, source monitored



8. Security Concepts Application

- **CIA Triad Examples from Lab:**
  - **Confidentiality:** Protecting Nessus scan reports from unauthorized access.
  - **Integrity:** Ensuring logs are not altered.
  - **Availability:** Target services (Metasploitable) remained online during testing.
- **Threat vs Vulnerability vs Risk:**
  - **Threat:** Attacker leveraging malicious.com domain.
  - **Vulnerability:** VSFTPD backdoor in Metasploitable2.
  - **Risk:** Unauthorized remote compromise of target VM.

9. Incident Response Workflow Simulation

Scenario: Brute-force SSH attempts + Snort alert for malicious.com.

Steps:

1. **Detection:** Alerts triggered in Snort + Kibana (failed SSH logins).
2. **Triage:** Classified as **high severity** due to repeated failed logins and suspicious domain detection.
3. **Investigation:** Correlated Snort alert and Kibana event logs by IP and timestamp.
4. **Response:** Blocked offending IP and terminated suspicious process.
5. **Recovery:** Reset credentials, applied system patches.
6. **Lessons Learned:** Enhanced Snort rules, enforced stronger password policies, and tuned Kibana alerts to reduce noise.

10. Documentation Standards

Date/Time	**Source IP**	**Event ID**	**Description**	**Action Taken**
2025-09-04 10:12	192.168.1.79	4625	Multiple failed SSH logins detected	Blocked offending IP
2025-09-04 10:20	192.168.1.79	Snort	Malicious.com Host header detected	Verified rule detection

Date/Time	Source IP	Event ID	Description	Action Taken
2025-09-04 10:25	192.168.1.79	Kibana	Spike in failed login events	Correlated with Snort

Conclusion

This lab demonstrated end-to-end SOC operations using Nessus, Snort, Elastic SIEM, Osquery, and Wazuh. Together, these tools enabled detection of vulnerabilities, malicious domains, brute-force attempts, suspicious processes, and account changes. The exercise highlighted the importance of monitoring, alerting, and documentation in supporting effective incident response.

Report By
ANGEL MF