Basic Incident Response Simulation: Handler's Journal

Simulation Overview

- **Simulation Number**: [Unique Identifier]
- **Date/Time Conducted**: [10/18/2025] [16:35]
- Scenario Description: Brief overview of a network attack simulation in an isolated virtual lab. A carefully crafted network traffic (single ICMP echo request with Scapy) sent from Kali linux VM (attacker) to an Ubuntu VM (target) with UFW (uncomplicated firewall) and pfsense firewall with Suricata enabled. The goal was to validate detection, local capture, and defensive logging and to exercise containment and cleanup procedures.

Detection and Initial Response

Detection Method:

- Suricata on pfSense generated alerts (notably an ICMPv4 invalid checksum alert) indicating suspicious/malformed ICMP traffic.
- Wireshark on the Ubuntu VM recorded incoming packets for forensic review.

Initial Actions Taken:

- Observed Suricata alerts in pfSense GUI (Services → Suricata → Logs View → Alerts).
- Collected evidence files (Scapy-generated pcap from wireshak) and screenshots of Suricata alerts.
- Began containment by terminating the Scapy process on the Kali VM once sufficient evidence was collected.

Analysis and Eradication

Analysis Findings:

- Suricata logged an ICMPv4 invalid checksum alert corresponding to the single ICMP packet sent from Kali. This confirms Suricata's rule detection and visibility into malformed/edge-case ICMP packets.
- Wireshark captures on Ubuntu and pcap saved by Scapy showed the packet flow and corroborated timestamps.

 Additional Suricata messages such as QUIC failed decrypt were observed; these were background noise from normal encrypted UDP traffic and not related to the test.

Eradication Measures:

- Terminated the active Scapy/Python process on Kali (sudo kill -9 <PID>).
- Stopped live capture on Ubuntu and, where appropriate, stopped any temporary background jobs.
- Removed the test artifacts (scapy script and pcap capture) from the attacker VM to prevent accidental their re-use and regeneration of network packets

Recovery and Restoration

Recovery Steps:

- Confirmed no active Scapy or Python processes remained on Kali (ps aux | grep python).
- Verified Ubuntu networking remained stable, and normal services were reachable.
- Enabled the log and block rule in pfsense GUI(firewall → rules → LAN interface) which i had previously disabled

Verification of Recovery:

- Re-ran basic connectivity checks (ICMP/ping and ip route show) between VMs to ensure normal operation.
- Rechecked ps output on Kali to confirm the attack process ID was absent.
- Reviewed Suricata and UFW logs to ensure no ongoing alerts related to the test traffic and to verify logging continued as expected.

Post-Simulation Reflection

Challenges Encountered:

- Initial connectivity issue where the Kali VM could not reach pfSense/Ubuntu until pfSense was rebooted. This required a manual pfSense restart to ensure the virtual adapters were attached correctly.
- Some noisy Suricata alerts (e.g., QUIC decrypt failures) made quick triage slightly more time-consuming — required focusing on time window and specific alert signatures.

Eradication ie: killing the attack process was a challenge because whenever i
would kill one process another one would regenerate from it. Therefore i used
sudo pkill -f scapy to kill all the running processes associated with scapy

Efficiency of Response:

 Overall the response was effective: detection, capture, and containment were completed with minimal manual steps. The main challenges experienced were Initial connectivity issues and eradication process

Lessons Learned and Improvements

Key Takeaways:

- Such tasks should be carried out in an isolated lab environment for safely validating detection rules and response procedures.
- Always verify virtual adapter assignments and interface IPs before running tests to reduce troubleshooting time.
- Always maintain evidences for such incidents such as pcap files and screenshots, this speeds up post simulation analysis

Improvement Plan:

- **Detection capabilities:** Enable / tune relevant Suricata rule sets (ET rules for ICMP anomalies and scanning signatures); reduce false positive noise by tuning QUIC/UDP-related rules or adjusting thresholds.
- Response strategies: Create a short runbook for common lab issues (reassigning VM NICs, restarting pfSense, capture checklist) to reduce mean time to recovery.
- **Tools and resources:** Centralize logs from pfSense (Suricata) and Ubuntu (syslog/ufw) to a single repository (e.g., ELK or a syslog collector) for easier correlation in future tests.
- **Training:** Practice small multi-step scenarios (e.g., more complex but still safe Scapy probes, fragmented packets, single SYN probes) and document expected IDS signatures.

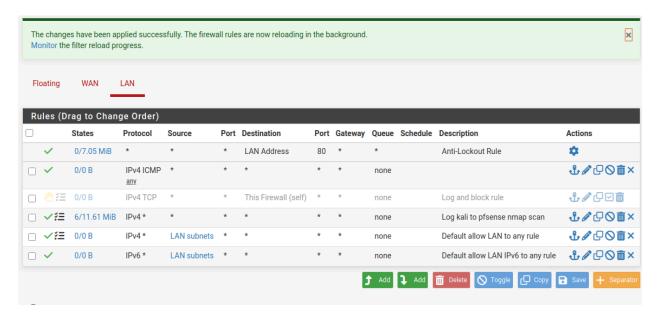
Additional Notes

Feedback on Simulation:

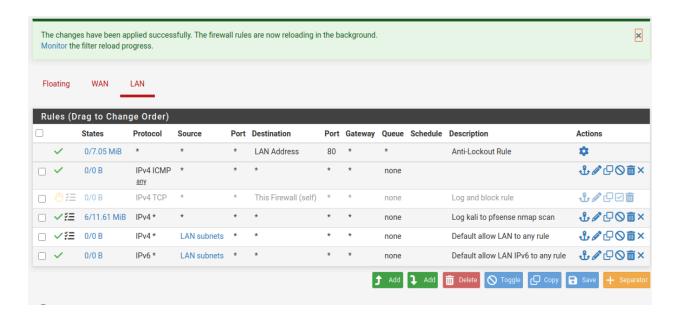
 The simulation was realistic and useful. The isolated environment allowed safe testing and produced meaningful Suricata alerts and packet captures for analysis.

Documentation

Supporting Materials:



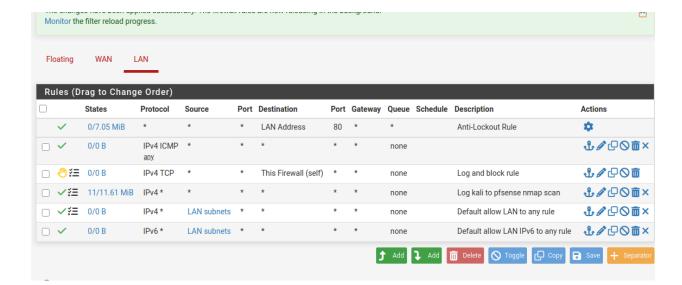
```
akshaj@akshaj-VirtualBox:~$ sudo ufw status
[sudo] password for akshaj:
Status: inactive
akshaj@akshaj-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
akshaj@akshaj-VirtualBox:~$ sudo ufw status
Status: active
                                        From
Τо
                            Action
22/tcp
                            ALLOW
                                        Anywhere
Anywhere
                            ALLOW
                                        10.0.2.4
80/tcp
                            ALLOW
                                        Anywhere
443
                            ALLOW
                                        Anywhere
22/tcp (v6)
                            ALLOW
                                        Anywhere (v6)
80/tcp (v6)
                                        Anywhere (v6)
                            ALLOW
443 (v6)
                            ALLOW
                                        Anywhere (v6)
```



```
1 from scapy.all import sr1, IP, ICMP, wrpcap
2 TARGET = "192.168.56.104"
3 IFACE = "eth0"
4
5 resp = sr1(IP(dst=TARGET)/ICMP(), timeout=2, iface=IFACE, verbose=False)
6 if resp:
7    print("reply received:")
8    resp.show()
9    wrpcap("single_icmp_reply.pcap", [resp])
9 else:
1    print("No reply (timeout or filtered)")
2
```

```
[TCP Retransmission]
                                                                                    74 [TCP Retransmission]
74 [TCP Retransmission]
                    192.168.56.104
                                             192.168.10.1
                                                                                                               3642
                                                                      TCP
264 74.435818833
                    192.168.56.104
                                             192.168.10.1
                                                                                    74 [TCP Retransmission]
                                                                                    66 [TCP Keep-Alive] 60854 →
265 75.714637421
                    192.168.56.104
                                             192.168.56.1
                                                                                    66 [TCP Keep-Alive ACK] 80 -
266 75.718640905
                    192.168.56.1
                                             192.168.56.104
                                                                      TCP
267 77.500440460
                    192.168.56.104
                                             192.168.10.1
                                                                      TCP
                                                                                    78 48442 → 53 [SYN] Seq=0 Wir
                                                                                    74 [TCP Retransmission] 4844
74 [TCP Retransmission] 4844
269 78.530568489
                    192.168.56.104
                                             192.168.10.1
270 79.555507572
                    192.168.56.104
                                             192.168.10.1
                                                                      TCP
                                                                                   74 [TCP Retransmission]
74 [TCP Retransmission]
74 [TCP Retransmission]
                                                                                                               4844
271 80.580466781
                    192.168.56.104
                                             192.168.10.1
                                                                      TCP
273 82.626727276
                    192.168.56.104
                                                                      TCP
                                                                                                               4844
274 84.674820131
                         168.56.104
                                                                                    74 [TCP Retransmission]
                                                                                    66 [TCP Keep-Alive] 60854
275 85.954728215
                                             192.168.56.1
                    192.168.56.104
                                                                                    66 [TCP Keep-Alive ACK] 80 →
                                                                      TCP
276 85.956979790 192.168.56.1
                                             192.168.56.104
```

10/18/2025-15:22:35.739538 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] 10/18/2025-15:22:52.004590 [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 10/18/2025-15:22:52.025669 [**] [1:2231000:1] SURICATA QUIC failed decrypt [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 [1:2231000:1] SURICATA QUIC failed decrypt 10/18/2025-15:22:52.090598 [Classification: 10/18/2025-15:22:52.105536 [**] [1:2231000:1] SURICATA QUIC failed decrypt [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 10/18/2025-15:23:14.242178 [**] [1:2231000:1] SURICATA QUIC failed decrypt [Classification: Generic Protocol Command Decodel [Priority: 3] {UDP} 199.6 [1:2231000:1] SURICATA QUIC failed decrypt 10/18/2025-15:23:14.242853 10/18/2025-15:23:14.243116 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 199.6 10/18/2025-15:23:14.245309 [**] [1:2231000:1] SURICATA OUIC failed decrypt [Classification: Generic Protocol Command Decodel [Priority: 3] {UDP} 192.1 10/18/2025-15:23:14.263551 [1:2231000:1] SURICATA QUIC failed decrypt [Classification: Generic Protocol Command Decode] [Priority: 3] [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 [**] [1:2210054:1] SURICATA STREAM excessive retransmissions [**] [Classification: Generic Protocol Command Decode] [Priority: 3] 10/18/2025-15:27:28.220065 10/18/2025-15:39:54.720362 10/18/2025-15:42:28.589051 [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] 10/18/2025-15:42:28.593777 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 10/18/2025-15:57:30.669467 10/18/2025-16:12:31.508240 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 10/18/2025-16:12:31.525219 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] (UDP} 192.1 [**] [1:2200076:2] SURICATA ICMPv4 invalid checksum [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {ICMP} 10/18/2025-16:34:51.737185 [**] [1:2231000:1] SURICATA QUIC failed decrypt [**] [Classification: Generic Protocol Command Decode] [Priority: 3] {UDP} 192.1 10/18/2025-16:35:16.605312



Sign-off

Prepared By: Akshaj Pathak

• Date: [10/18/2025]