



Task 5: Networking and Security Operations with SIEM, Forensics, and Traffic Analysis

Submitted by:

Shashanka U N
SOC Intern – Cyber Security

Submitted to:

CyArt Tech

Date:

30th August 2025



Introduction:

This task integrated multiple domains of cybersecurity, combining networking fundamentals with security operations. The exercises focused on designing efficient subnets, analyzing network traffic, troubleshooting protocol issues, and deploying a Security Information and Event Management (SIEM) system. Additionally, the lab simulated a security incident to apply forensic analysis techniques and concluded with proactive threat hunting. The goal was to replicate real-world scenarios where organizations must not only build secure and scalable networks but also detect, investigate, and respond to threats using both network traffic data and centralized log analysis.

Objectives:

- Design and calculate a subnet for a small network.
- Analyze captured network traffic for patterns.
- Troubleshoot common network protocol issues.
- Deploy a SIEM (ELK Stack) for log collection and monitoring.
- Simulate an incident and investigate using forensics.
- Perform proactive threat hunting using traffic and log data.

Methodology

The following methodology was adopted to complete Task:

Network Setup:

- Base Network: 192.168.1.0/27 (Subnet design for 20 devices).
- Ubuntu 22.04 VM (used for SIEM setup and traffic analysis).

Tools:

- draw.io → Subnet diagram and network design.
- Wireshark 4.x → Packet capture and protocol analysis.
- tcpdump → Command-line traffic collection.
- Cisco Packet Tracer → Simulation of troubleshooting scenarios.



- ELK Stack (Elasticsearch, Logstash, Kibana) → SIEM setup, log collection, and visualization.
- hping3 / SSH brute-force attempts → Simulated attack traffic for forensic analysis.

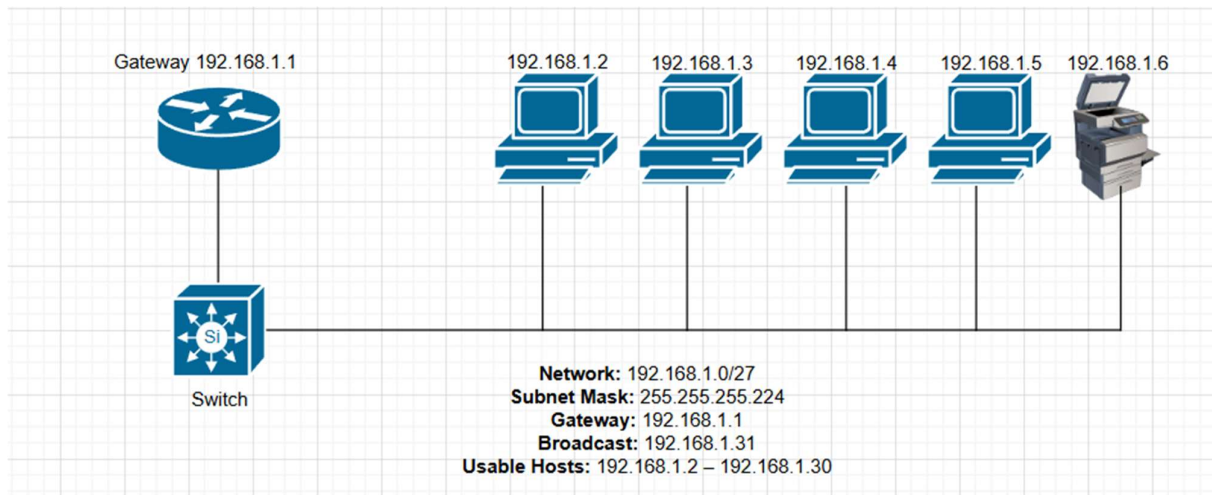
Scan Results & Observations:

Task 1 – Subnet Design

Steps:

1. Selected private IP range 192.168.1.0/24 for the office network.
2. Requirement was 20 devices → chose subnet /27 which provides 32 total IPs (30 usable).
3. Calculated subnet details:
 - **Network Address:** 192.168.1.0
 - **Subnet Mask:** 255.255.255.224 (/27)
 - **Usable Range:** 192.168.1.1 – 192.168.1.30
 - **Broadcast Address:** 192.168.1.31
4. Documented subnet math to justify CIDR selection.
5. Designed topology in **draw.io**, connecting router, switch, and sample PCs.

Results:



Calculation:

Requirement = 20 devices.

Choose /27 since usable hosts = $2^{(32-27)} - 2 = 2^5 - 2 = 32 - 2 = 30$ (≥ 20).



Therefore subnet chosen: 192.168.1.0/27 (mask 255.255.255.224), usable range 192.168.1.1–192.168.1.30, broadcast 192.168.1.31.

- Network: 192.168.1.0/27
- Subnet Mask: 255.255.255.224
- Gateway: 192.168.1.1
- Broadcast: 192.168.1.31
- Usable Hosts: 192.168.1.2 – 192.168.1.30

Task 2 – Network Traffic Analysis

Steps:

1. Configured **Wireshark** on Ubuntu VM to capture traffic from the active network interface.
2. Collected traffic for ~10 minutes while performing normal activities (web browsing, DNS lookups, SSH connections).
3. Applied Wireshark filters to identify specific protocols:
 - a. **tcp** → TCP traffic
 - b. **udp** → UDP traffic
 - c. **dns** → DNS queries/responses
4. Used **Statistics** → **Protocol Hierarchy** to observe protocol distribution.
5. Used **Statistics** → **Conversations** to find **Top Talkers** (hosts generating the most traffic).
6. Checked for anomalies such as spikes in DNS queries or abnormal port usage.

Results

The screenshot displays a Wireshark packet capture interface with a list of captured packets. The packets are filtered by 'eth0' and show a variety of protocols including HTTP, DNS, and ICMP. The packet list includes details such as time, source, destination, protocol, and length. The packet details pane on the right shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Hypertext Transfer Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
2	0.36570017	91.189.91.157	172.26.57.3	HTTP	90	HTTP Version 4, server
3	1.46521791	Microsoft 75:36:c2	Microsoft eb-b5:2f	ARP	42	Who has 172.26.57.3? Tell 172.26.48.1
4	1.46534853	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	172.26.57.3 is at 08:15:5d:eb:b5:2f
5	1.14447675	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	Who has 172.26.48.1? Tell 172.26.57.3
6	1.14492036	Microsoft 75:36:c2	Microsoft eb-b5:2f	ARP	42	172.26.48.1 is at 08:15:5d:eb:b5:2f
7	5.42044164	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
8	5.73268063	91.189.91.157	172.26.57.3	HTTP	90	HTTP Version 4, server
9	9.53531667	fe8b::1215:5dff:feeb::	ff02::2	ICMPv6	78	Router Solicitation from 08:15:5d:eb:b5:2f
10	42.96823023	Microsoft 75:36:c2	Microsoft eb-b5:2f	ARP	42	Who has 172.26.57.3? Tell 172.26.48.1
11	42.96828259	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	172.26.57.3 is at 08:15:5d:eb:b5:2f
12	43.02649462	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	Who has 172.26.48.1? Tell 172.26.57.3
13	43.02685712	Microsoft 75:36:c2	Microsoft eb-b5:2f	ARP	42	172.26.48.1 is at 08:15:5d:eb:b5:2f
14	76.89951264	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
15	77.22921072	91.189.91.157	172.26.57.3	HTTP	90	HTTP Version 4, server
16	77.22778648	172.26.48.1	224.0.0.251	MNMS	442	Standard query response 0x0000 PTR shashank.04.dosvc.tcp.local SRV 0 0 7680 shashank.04.local TXT
17	77.22952458	fe8b::dc9:91d:1c3b::	ff02::f	MNMS	465	Standard query response 0x0000 PTR shashank.04.dosvc.tcp.local SRV 0 0 7680 shashank.04.local TXT
18	77.23236384	172.26.48.1	224.0.0.251	MNMS	89	Standard query 0x0000 ANY shashank.04.dosvc.tcp.local, "QM" question
19	77.23343560	fe8b::dc9:91d:1c3b::	ff02::f	MNMS	109	Standard query 0x0000 ANY shashank.04.dosvc.tcp.local, "QM" question
20	77.23627562	172.26.48.1	224.0.0.251	MNMS	89	Standard query 0x0000 ANY shashank.04.dosvc.tcp.local, "QM" question
21	77.23789484	fe8b::dc9:91d:1c3b::	ff02::f	MNMS	109	Standard query 0x0000 ANY shashank.04.dosvc.tcp.local, "QM" question
22	78.21564895	172.26.48.1	224.0.0.251	MNMS	89	Standard query 0x0000 ANY shashank.04.dosvc.tcp.local, "QM" question
23	78.22832411	fe8b::dc9:91d:1c3b::	ff02::f	MNMS	109	Standard query 0x0000 ANY shashank.04.dosvc.tcp.local, "QM" question
24	78.46911435	172.26.48.1	224.0.0.251	MNMS	535	Standard query response 0x0000 PTR, cache flush shashank.04.dosvc.tcp.local SRV, cache flush 0 0 7680 shashank.04.local TXT, cache flush A, cache flush 172.26.48.1 AAAA, cache flush
25	78.46226976	fe8b::dc9:91d:1c3b::	ff02::f	MNMS	545	Standard query response 0x0000 PTR, cache flush shashank.04.dosvc.tcp.local SRV, cache flush 0 0 7680 shashank.04.local TXT, cache flush A, cache flush 172.26.48.1 AAAA, cache flush
26	78.51383062	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	Who has 172.26.57.3? Tell 172.26.48.1
27	78.51383062	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	Who has 172.26.48.1? Tell 172.26.57.3
28	79.13900997	Microsoft eb-b5:2f	Microsoft 75:36:c2	ARP	42	Who has 172.26.48.1? Tell 172.26.57.3
29	79.14064984	Microsoft eb-b5:2f	Microsoft eb-b5:2f	ARP	42	172.26.48.1 is at 08:15:5d:eb:b5:2f
30	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
31	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
32	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
33	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
34	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
35	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
36	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
37	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
38	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
39	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
40	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
41	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
42	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
43	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
44	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
45	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
46	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
47	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
48	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
49	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
50	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
51	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
52	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
53	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
54	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
55	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
56	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
57	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
58	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client
59	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, server
60	32.86227530	172.26.57.3	91.189.91.157	HTTP	90	HTTP Version 4, client



Task 3 – Troubleshoot Network Protocol Issues

1. Designed a small test network with **1 Router, 1 Client (PC), and 1 Server** using Packet Tracer / VM setup.

Router: 192.168.1.1/24

Client: 192.168.1.10/24, gateway 192.168.1.1

Server: 192.168.1.20/24, gateway 192.168.1.1

3. Introduced a deliberate issue → configured **incorrect DNS IP** on the client.

4. Tested connectivity:

Ping to Router IP → successful (local connectivity OK).

Traceroute to Server IP → successful (routing OK).

DNS resolution (pinging hostname) → failed.

5. Captured traffic in **Wireshark** to confirm repeated DNS queries without valid responses.

6. Corrected DNS configuration (set valid DNS IP).



Task 4: SOC and Networking Integration

1. Set Up ELK Stack for SIEM

Steps Performed:

1. Installed ELK Stack on a Linux VM (Ubuntu).
 - Installed Elasticsearch, Logstash, and Kibana.
2. Configured Logstash to collect logs:
 - System logs (syslog)
 - Network device logs (router, firewall)
3. Created Kibana dashboards to visualize key metrics:
 - Login attempts
 - Traffic volume
 - Security alerts

2. Simulate an Incident and Perform Network Forensics

Steps Performed:

1. Simulated a security incident:
 - Example: Unauthorized SSH access attempt / DoS attack
2. Captured network traffic using Wireshark during the incident.
3. Analyzed captured traffic to:
 - Identify attack vector (source IP, destination IP, port, protocol)
 - Document malicious packets (e.g., SYN flood pattern, repeated login attempts)
4. Correlated network data with SIEM logs in Kibana to build a timeline of the incident.

3. Conduct Threat Hunting

Steps Performed:

1. Reviewed SIEM logs in Kibana for suspicious activities.
2. Monitored network traffic in Wireshark for anomalies:
 - Unusual outbound connections
 - Repeated failed login attempts
 - Suspicious protocols or ports
3. Correlated findings with historical data to detect potential threats.