Intrusion Prevention System (IPS) Report

1. Introduction

An Intrusion Prevention System (IPS) is a network security tool that monitors traffic in real time to detect and block malicious activity, preventing potential breaches. Unlike intrusion detection, which only alerts, an IPS actively stops harmful traffic based on predefined rules and patterns.

This report outlines the implementation of a lightweight IPS that detects and lists malicious IPs from user-uploaded PCAP files. The system focuses on specific attack vectors such as ICMP ping floods, TCP SYN floods, scan patterns (SYN/NULL/FIN), repeated port attempts, as well as malicious HTTP payloads and SQL injection attempts.

Although IPSs typically operate on live traffic, for simplicity and learning purposes in this project we instead extract and list the malicious IP addresses from PCAPs. These IPs can then be applied in a firewall for effective blocking.

2. Prevention Logic

The IPS prevention logic is based on detecting abnormal traffic behaviors and blocking them before they can cause harm. The logic includes:

- **ICMP Ping Floods**: Continuous ICMP echo requests (pings) can overwhelm a system. The IPS detects unusually high rates of ICMP traffic, applies rate limiting, and drops excessive requests.
- **TCP SYN Floods**: Attackers may send multiple SYN packets without completing the handshake, leaving half-open connections that consume resources. The IPS monitors for repeated incomplete handshakes and drops further malicious SYN requests.
- **Port Scan Detection**: Scans like NUL**L**, FIN, or SYN scans are used by attackers to probe open ports. The IPS identifies these scanning patterns and blocks repeated attempts from the same source.
- **Repeated port attempts:**
  This is when the same IP keeps hitting the same port (or small group of ports) again and again in a short time. It could be brute force, repeated probes, or even a misconfigured client.
- **HTTP Payload Inspection**: Attackers often use web requests to inject malicious code. The IPS scans HTTP payloads for SQL injection keywords (e.g., *"UNION SELECT"*, *"DROP TABLE"*) and path traversal attempts (e.g., *"../"*). Requests containing these are flagged and blocked to protect web applications.

3. False Positive Handling

The IPS handles false positives by using thresholds and specific pattern checks rather than blocking all suspicious activity. For ICMP ping floods, only sources exceeding a set rate are blocked, allowing normal connectivity checks. TCP SYN floods are detected by counting incomplete handshakes, so occasional failed attempts are ignored. Repeated port scans are tracked per IP, but normal retries to a few ports are allowed. HTTP payloads are inspected for exact malicious patterns (like `"UNION SELECT"` or `"../"`), avoiding blocks on harmless text with similar words. This approach ensures legitimate traffic passes while actual attacks are caught

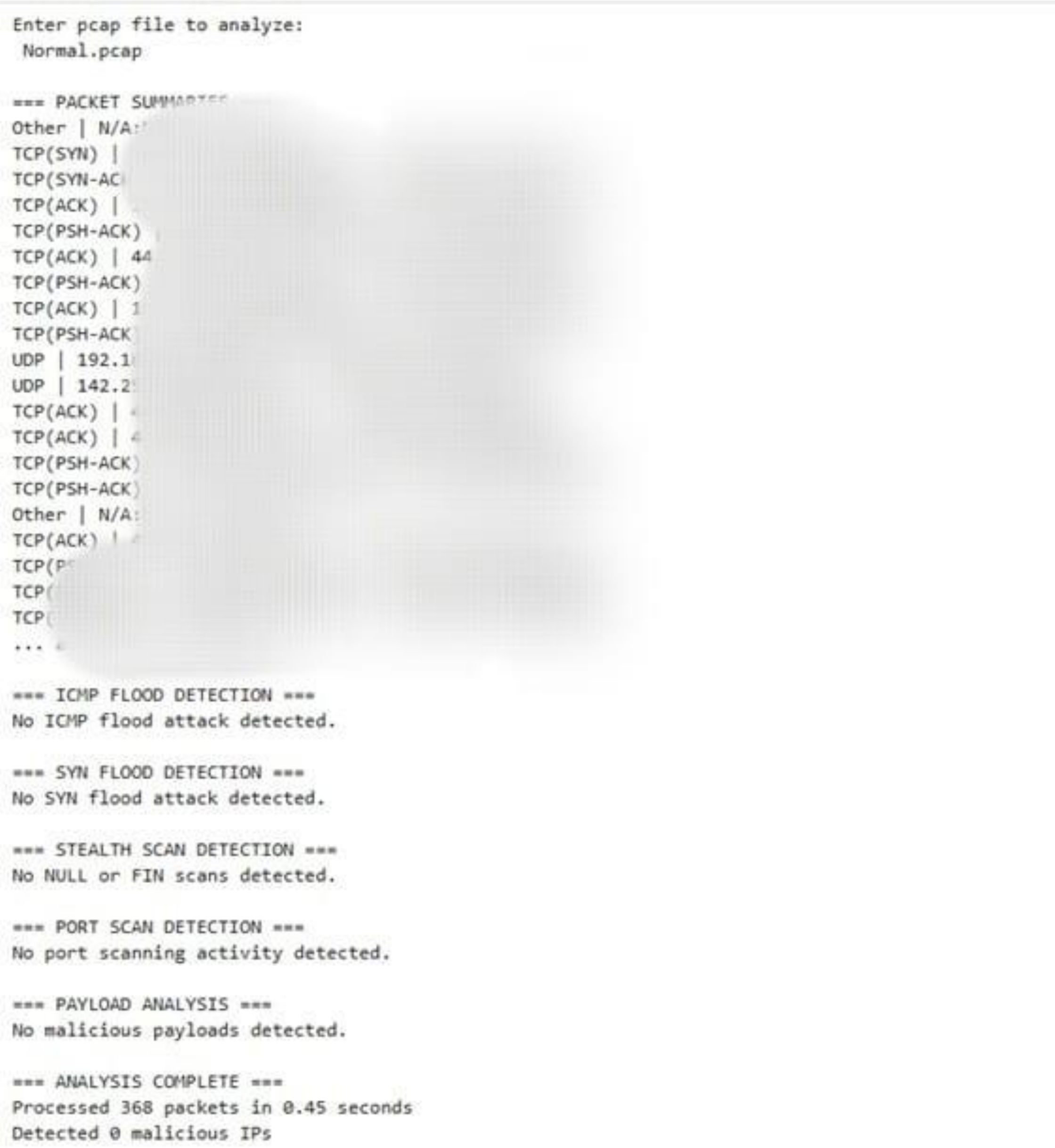
4. Demo and Results

The IPS was tested using two PCAP files:

- Normal traffic PCAP – The IPS allowed traffic without blocking.
- Malicious traffic PCAP – The IPS successfully blocked ping floods, SYN floods, and scan attempts.

Screenshots of the results are included below.

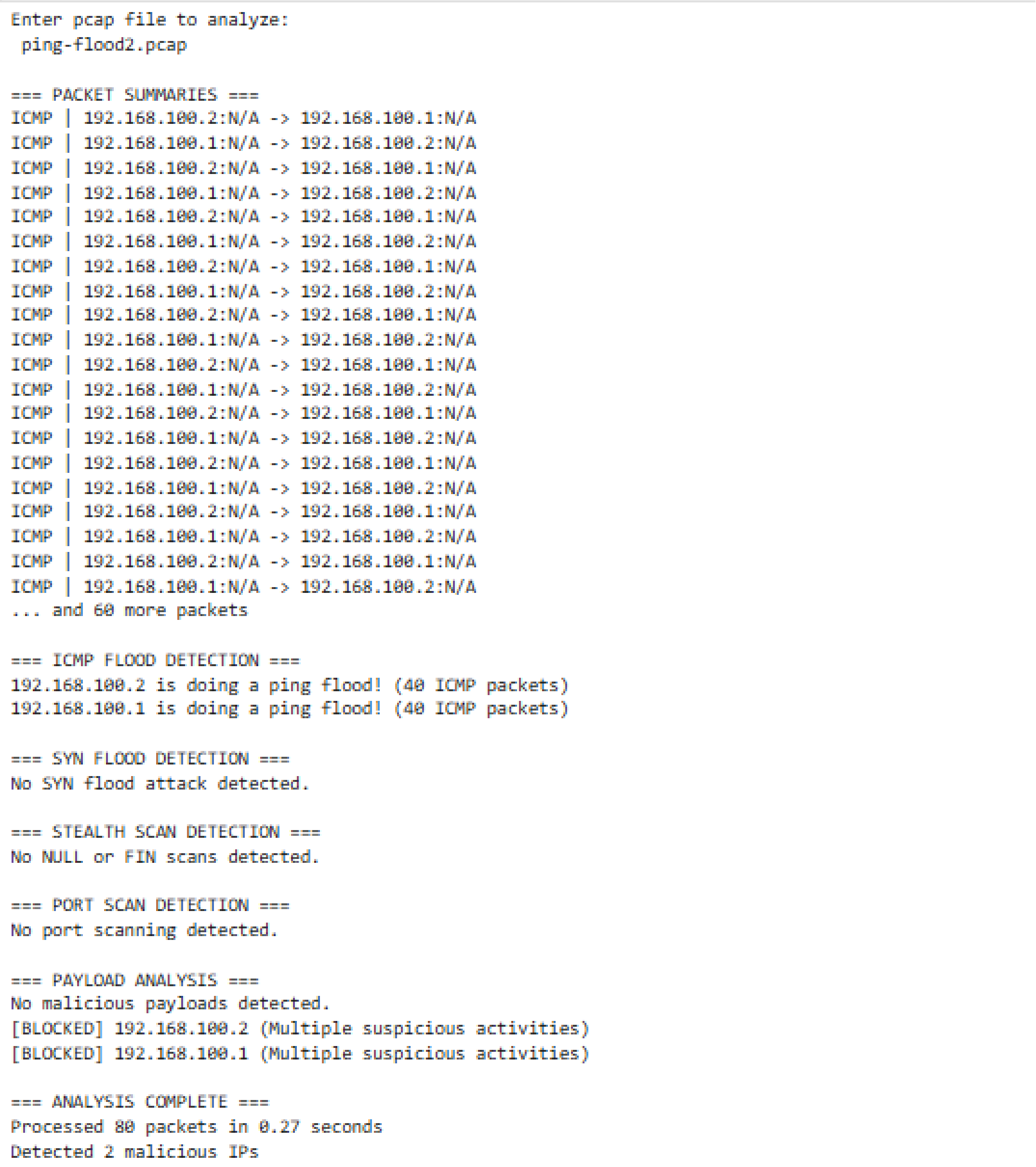
Normal Traffic Results



```
Enter pcap file to analyze:
 Normal.pcap

=== PACKET SUM [redacted]
Other | N/A:
TCP(SYN) |
TCP(SYN-AC
TCP(ACK) |
TCP(PSH-ACK)
TCP(ACK) | 44
TCP(PSH-ACK)
TCP(ACK) |
TCP(PSH-ACK
UDP | 192.1
UDP | 142.2
TCP(ACK) |
TCP(ACK) |
TCP(PSH-ACK)
TCP(PSH-ACK)
Other | N/A:
TCP(ACK)
TCP(P
TCP(
TCP(
...

=== ICMP FLOOD DETECTION ===
No ICMP flood attack detected.

=== SYN FLOOD DETECTION ===
No SYN flood attack detected.

=== STEALTH SCAN DETECTION ===
No NULL or FIN scans detected.

=== PORT SCAN DETECTION ===
No port scanning activity detected.

=== PAYLOAD ANALYSIS ===
No malicious payloads detected.

=== ANALYSIS COMPLETE ===
Processed 368 packets in 0.45 seconds
Detected 0 malicious IPs
```

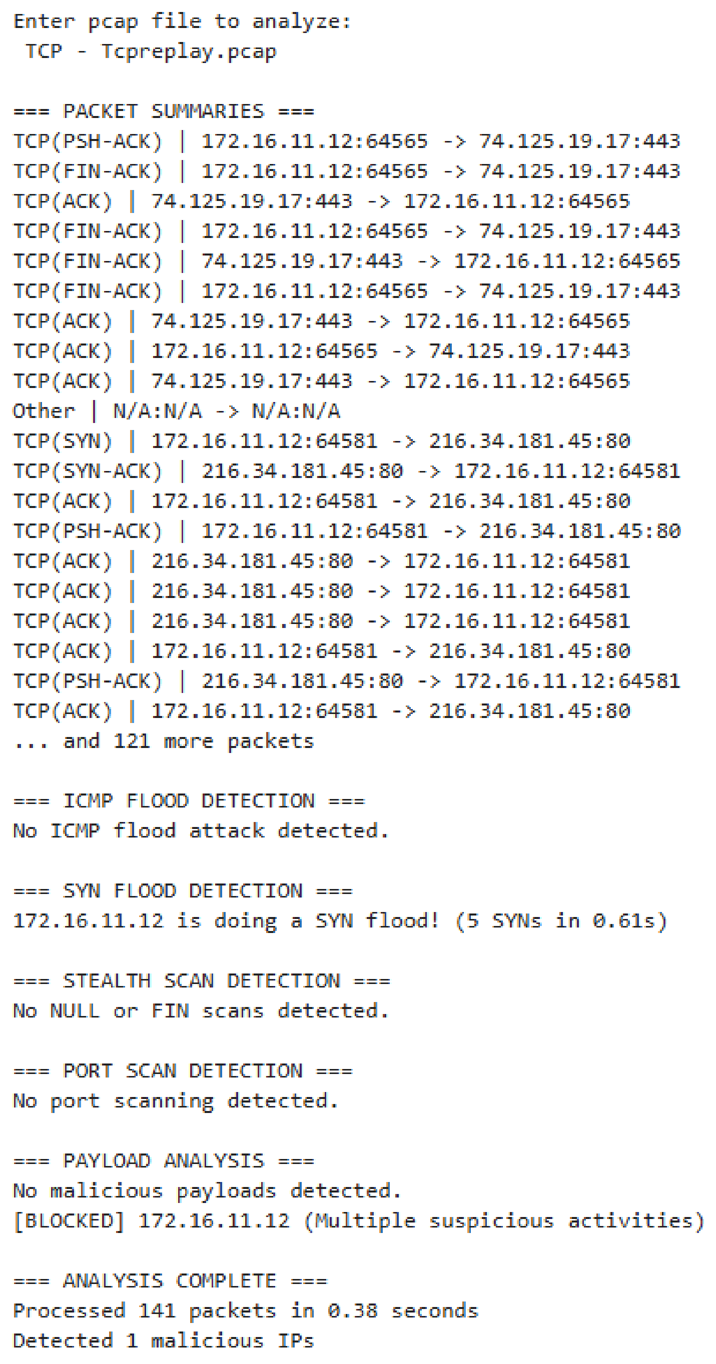
Malicious Traffic Results

1. **ICMP flood attack**



```
Enter pcap file to analyze:
 ping-flood2.pcap

=== PACKET SUMMARIES ===
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
ICMP | 192.168.100.2:N/A -> 192.168.100.1:N/A
ICMP | 192.168.100.1:N/A -> 192.168.100.2:N/A
... and 60 more packets

=== ICMP FLOOD DETECTION ===
192.168.100.2 is doing a ping flood! (40 ICMP packets)
192.168.100.1 is doing a ping flood! (40 ICMP packets)

=== SYN FLOOD DETECTION ===
No SYN flood attack detected.

=== STEALTH SCAN DETECTION ===
No NULL or FIN scans detected.

=== PORT SCAN DETECTION ===
No port scanning detected.

=== PAYLOAD ANALYSIS ===
No malicious payloads detected.
[BLOCKED] 192.168.100.2 (Multiple suspicious activities)
[BLOCKED] 192.168.100.1 (Multiple suspicious activities)

=== ANALYSIS COMPLETE ===
Processed 80 packets in 0.27 seconds
Detected 2 malicious IPs
```

2. SYN flood

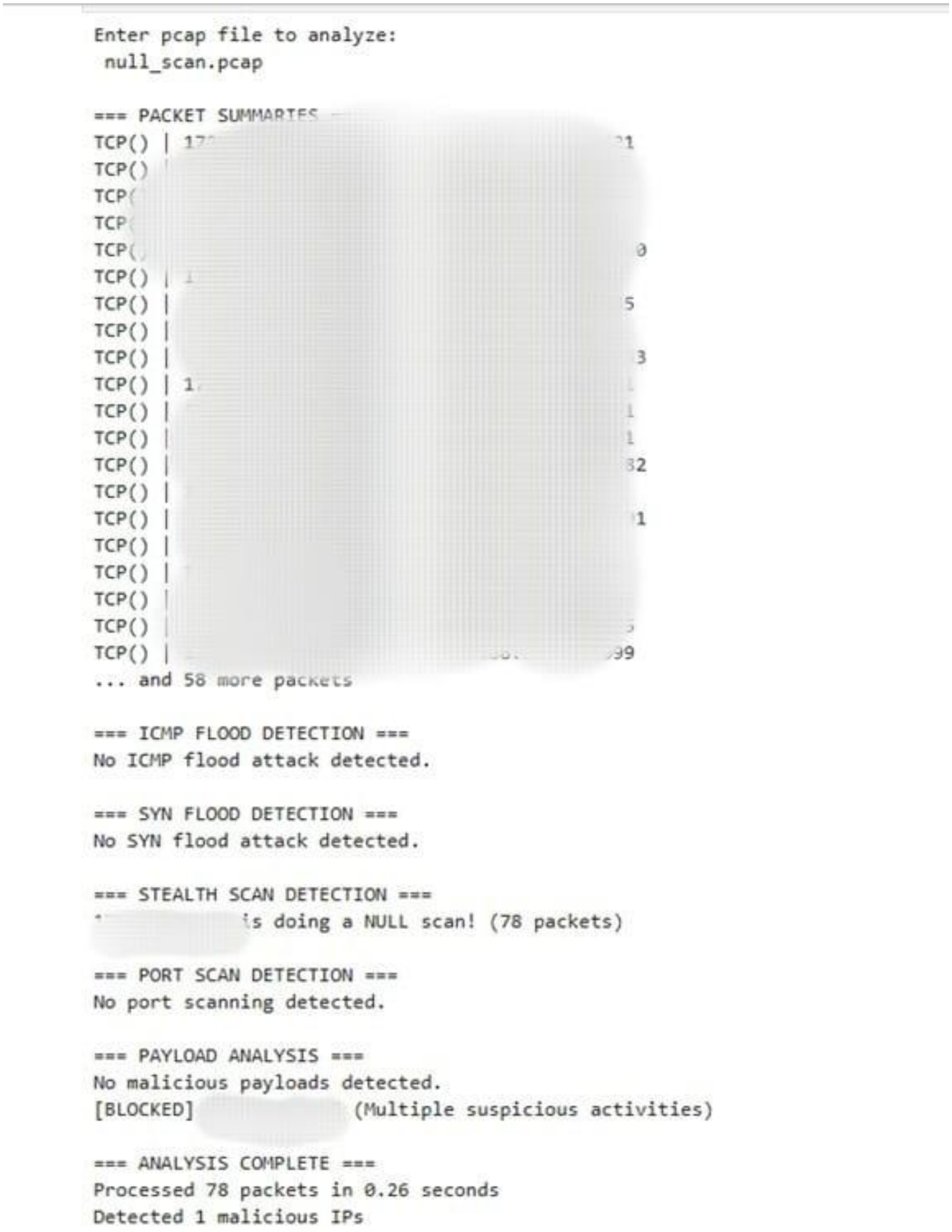


```
Enter pcap file to analyze:
 TCP - Tcpreplay.pcap

=== PACKET SUMMARIES ===
TCP(PSH-ACK) | 172.16.11.12:64565 -> 74.125.19.17:443
TCP(FIN-ACK) | 172.16.11.12:64565 -> 74.125.19.17:443
TCP(ACK) | 74.125.19.17:443 -> 172.16.11.12:64565
TCP(FIN-ACK) | 172.16.11.12:64565 -> 74.125.19.17:443
TCP(FIN-ACK) | 74.125.19.17:443 -> 172.16.11.12:64565
TCP(FIN-ACK) | 172.16.11.12:64565 -> 74.125.19.17:443
TCP(ACK) | 74.125.19.17:443 -> 172.16.11.12:64565
TCP(ACK) | 172.16.11.12:64565 -> 74.125.19.17:443
TCP(ACK) | 74.125.19.17:443 -> 172.16.11.12:64565
Other | N/A:N/A -> N/A:N/A
TCP(SYN) | 172.16.11.12:64581 -> 216.34.181.45:80
TCP(SYN-ACK) | 216.34.181.45:80 -> 172.16.11.12:64581
TCP(ACK) | 172.16.11.12:64581 -> 216.34.181.45:80
TCP(PSH-ACK) | 172.16.11.12:64581 -> 216.34.181.45:80
TCP(ACK) | 216.34.181.45:80 -> 172.16.11.12:64581
TCP(ACK) | 216.34.181.45:80 -> 172.16.11.12:64581
TCP(ACK) | 216.34.181.45:80 -> 172.16.11.12:64581
TCP(ACK) | 172.16.11.12:64581 -> 216.34.181.45:80
TCP(PSH-ACK) | 216.34.181.45:80 -> 172.16.11.12:64581
TCP(ACK) | 172.16.11.12:64581 -> 216.34.181.45:80
... and 121 more packets

=== ICMP FLOOD DETECTION ===
No ICMP flood attack detected.

=== SYN FLOOD DETECTION ===
172.16.11.12 is doing a SYN flood! (5 SYNs in 0.61s)

=== STEALTH SCAN DETECTION ===
No NULL or FIN scans detected.

=== PORT SCAN DETECTION ===
No port scanning detected.

=== PAYLOAD ANALYSIS ===
No malicious payloads detected.
[BLOCKED] 172.16.11.12 (Multiple suspicious activities)

=== ANALYSIS COMPLETE ===
Processed 141 packets in 0.38 seconds
Detected 1 malicious IPs
```

3. NULL SCAN

Performing NULL Scan using nmap and capturing and saving the traffic using wireshark

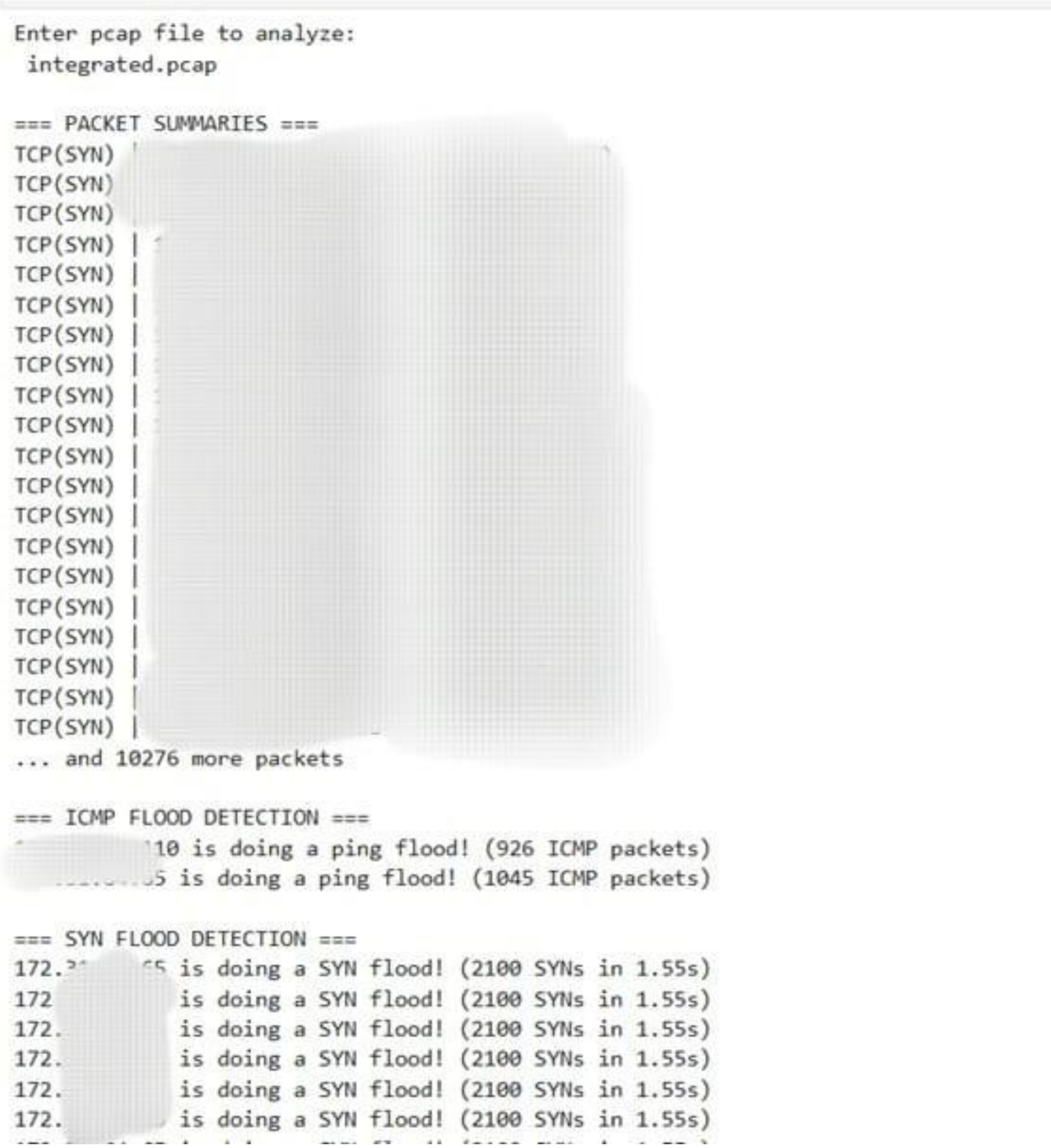
nmap -sN -Pn <target-ip>



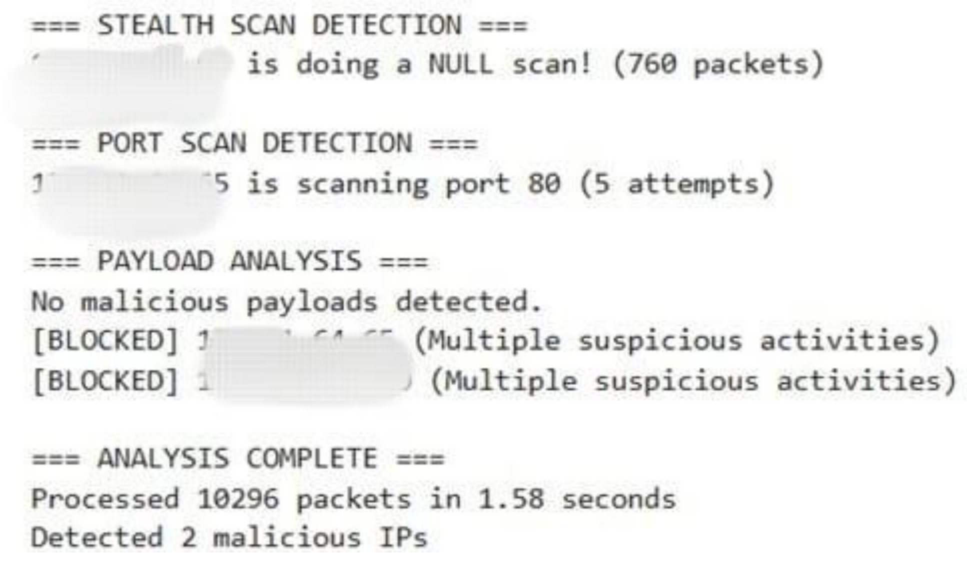
```
Enter pcap file to analyze:
 null_scan.pcap

=== PACKET SUMMARIES [redacted]
TCP() |
TCP()
TCP
TCP
TCP
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
TCP() |
... and 58 more packets

=== ICMP FLOOD DETECTION ===
No ICMP flood attack detected.

=== SYN FLOOD DETECTION ===
No SYN flood attack detected.

=== STEALTH SCAN DETECTION ===
[redacted] is doing a NULL scan! (78 packets)

=== PORT SCAN DETECTION ===
No port scanning detected.

=== PAYLOAD ANALYSIS ===
No malicious payloads detected.
[BLOCKED] [redacted] (Multiple suspicious activities)

=== ANALYSIS COMPLETE ===
Processed 78 packets in 0.26 seconds
Detected 1 malicious IPs
```

5. Unit and Integration Testing

Basic unit tests were developed to validate core prevention functions such as ICMP flood blocking, SYN flood detection, and SQL injection pattern matching. Integration testing was performed with PCAP replay to ensure the system works under realistic conditions.



```
Enter pcap file to analyze:
 integrated.pcap

=== PACKET SUMMARIES ===
TCP(SYN) [redacted]
TCP(SYN)
TCP(SYN)
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
TCP(SYN) |
... and 10276 more packets

=== ICMP FLOOD DETECTION ===
[redacted]10 is doing a ping flood! (926 ICMP packets)
 is doing a ping flood! (1045 ICMP packets)

=== SYN FLOOD DETECTION ===
172.[redacted] is doing a SYN flood! (2100 SYNs in 1.55s)
172          is doing a SYN flood! (2100 SYNs in 1.55s)
172.         is doing a SYN flood! (2100 SYNs in 1.55s)
172.         is doing a SYN flood! (2100 SYNs in 1.55s)
172.         is doing a SYN flood! (2100 SYNs in 1.55s)
172.         is doing a SYN flood! (2100 SYNs in 1.55s)
```



```
=== STEALTH SCAN DETECTION ===
[redacted] is doing a NULL scan! (760 packets)

=== PORT SCAN DETECTION ===
[redacted] is scanning port 80 (5 attempts)

=== PAYLOAD ANALYSIS ===
No malicious payloads detected.
[BLOCKED] [redacted] (Multiple suspicious activities)
[BLOCKED] [redacted] (Multiple suspicious activities)

=== ANALYSIS COMPLETE ===
Processed 10296 packets in 1.58 seconds
Detected 2 malicious IPs
```

6. Limitations

- Reliance on static thresholds may miss attacks or trigger false positives.
- Simple string matching for HTTP/SQL detection may not catch complex or obfuscated attacks.
- No real-time integration with firewalls or SIEM systems for automated response.

6. Ideas for Improvement

Future improvements to the IPS could include:

- Adaptive thresholds based on normal traffic behavior.
- Whitelisting trusted IPs or hosts.
- Context-aware HTTP filtering to better distinguish legitimate requests from attacks.
- Progressive blocking (gradually limiting suspicious IPs) instead of instant blocking.
- Multi-signature correlation for more accurate payload detection.
- Machine learning-based anomaly detection
- More advanced payload analysis
- Improved logging and alerting mechanisms
- Integration with firewalls and SIEM systems.

7. Conclusion

The lightweight IPS successfully demonstrated the ability to detect and block common attacks such as ICMP floods, SYN floods, and scan attempts, while minimizing false positives. This system provides a foundation for further development and can be extended with advanced detection and prevention features.