**INFORMATION SECURITY IA 1 REPORT**

**16010122012 RISHIKA BANERJEE**

**16010122016 SANIKA BHALBAR**

**16010122025 PARTH CHAMPANERKAR**

## 1. Introduction

Snort is an open-source Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) widely used for network security. Developed by Martin Roesch in 1998, Snort enables real-time traffic analysis and packet logging to detect network intrusions and malicious activities. It is a powerful tool for monitoring network behavior and mitigating threats, making it an essential component in cybersecurity infrastructure. With its flexibility, Snort can operate in various modes, including packet sniffer, packet logger, and a fully functional network intrusion detection system.

Network security is crucial in today’s digital era due to the increasing number of cyber threats, including malware, ransomware, and unauthorized access. Snort provides a robust framework for detecting these threats by leveraging a signature-based approach alongside anomaly detection capabilities. By analyzing packet headers and payloads, Snort helps organizations and security professionals safeguard their network environments effectively.

## 2. Features/Characteristics

Snort has several key features that make it a widely adopted tool for network security monitoring:

* Real-Time Packet Capture: Snort captures and inspects live network traffic to detect suspicious patterns.
* Signature-Based Detection: It uses a vast library of predefined signatures to identify known threats.
* Custom Rule Creation: Users can create and modify Snort rules to tailor detection to specific security needs.
* Multi-Mode Operation: Snort can function as a passive network sniffer, a packet logger for forensic analysis, or a full intrusion detection system.
* Protocol Analysis: Examines traffic across multiple protocols, including TCP, UDP, and ICMP, for anomalies.
* Logging and Alerting: Generates alerts for detected threats and logs relevant packet information for further investigation.
* Scalability and Integration: Can be integrated with SIEM (Security Information and Event Management) solutions for centralized monitoring and analysis.

## 3. Methodology

To implement and evaluate Snort effectively, a structured approach was followed, consisting of installation, configuration, rule creation, testing, and integration with additional security tools.

### 3.1 Installation

The installation process involved the following steps:

* Downloaded and installed Snort on a WINDOWS-based system.
* Installed necessary dependencies such as packet capture libraries (NPCAP) and additional rule sets.
* Verified installation by running test commands to ensure Snort was operational.

### 3.2 Configuration

* Configured the snort.conf file to define the network environment and specify rule paths.
* Enabled logging settings to capture packet data and generate alerts based on rule matches.
* Adjusted performance settings to optimize Snort’s efficiency for real-time monitoring.

### 3.3 Rule Creation and Testing

To enhance Snort’s detection capabilities, multiple custom rules were created and tested:

* Basic Rules: Configured rules to detect common network activities such as ICMP pings and HTTP requests.
* Content-Based Rules: Created rules to inspect packet payloads for specific keywords and malware signatures.
* Flow-Based Rules: Developed session-based rules to monitor and analyze the behavior of network connections.
* Threshold Rules: Implemented rules to minimize alert flooding by setting frequency limits on repeated detections.

Testing involved:

* Using tools like Tcpdump to capture and analyze network traffic.
* Simulating attacks such as port scanning and brute-force login attempts to validate Snort's effectiveness.
* Reviewing log files and alert data to assess the accuracy of Snort’s detections.

### 3.4 Implementation in a Live Network

* Deployed Snort on a monitored network to observe real-time packet analysis.
* Analyzed network traffic to detect unauthorized access attempts and suspicious data transfers.
* Integrated Snort logs with a centralized logging system for improved visibility and alert correlation.

## 4. Results

Following the implementation and testing of Snort, key observations were recorded:

* Successfully identified and logged multiple intrusion attempts, including port scans and unauthorized SSH access.
* Detected malware signatures and suspicious payload patterns within monitored traffic.
* Improved rule effectiveness by refining detection parameters and reducing false positives.
* Enhanced network security posture by integrating Snort with SIEM tools for streamlined threat analysis.
* Observed performance optimizations that ensured minimal impact on network latency.

## 5. Conclusion

The implementation of Snort provided valuable insights into network security monitoring and intrusion detection. Through its rule-based approach and real-time traffic analysis capabilities, Snort effectively identified and reported suspicious network activities. The ability to create custom rules allowed for enhanced threat detection, while integration with external monitoring systems improved overall security visibility.

Future improvements include:

* Deploying Snort in inline mode to actively block malicious traffic rather than just detect it.
* Enhancing integration with machine learning-based anomaly detection for better threat identification.
* Expanding rule sets to cover emerging cyber threats and sophisticated attack vectors.

Snort remains a critical tool in cybersecurity, providing organizations with a powerful defense mechanism against evolving network threats. Our work with Snort has strengthened our understanding of intrusion detection systems and their role in maintaining a secure digital environment.

### Summary of Snort Demonstration

This demonstration showcases the practical implementation of Snort, an open-source Intrusion Detection and Prevention System (IDPS). The following operational modes were tested:

 Sniffer Mode: Captures and displays live network traffic. Verified by running ping [google.com](http://google.com/), observing packet flow in real-time.

 Packet Logger Mode: Stores network traffic for analysis. Logs were generated via curl [http://example.com](http://example.com/), then accessed in /var/log/snort/.

 Intrusion Detection Mode (IDS): Detects suspicious activities based on predefined rules. Snort was run with a configuration file, and an aggressive ping -f triggered alerts logged in /var/log/snort/alert.

 Custom Rule Implementation: A Snort rule (alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:1000002;)) was created to detect ICMP traffic. It successfully generated alerts upon execution of ping -c 4 [google.com](http://google.com/).

 Intrusion Prevention Mode (IPS): Configured Snort in inline mode (-Q flag) with iptables (iptables -A FORWARD -j QUEUE). A simulated Nmap scan was successfully detected and blocked, demonstrating Snort’s active threat mitigation capabilities.

### Conclusion

The demonstration effectively highlights Snort’s role in network security through its sniffer, logging, IDS, and IPS capabilities. Key takeaways include custom rule creation, alert tuning, and real-time prevention of threats.