Comprehensive Report on Network Intrusion Detection System (IDS) Project

Sriker Paturi

October 24th, 2024

Contents

1 Introduction

This report documents the design, implementation, and testing of a Network Intrusion Detection System (IDS) using Snort. The primary aim is to monitor network traffic, detect potential security threats, and improve the overall security posture of the network.

—

2 Project Overview

2.1 Objective

The objective of the project is to:

- Monitor network traffic in real time.
- Detect and log suspicious activities such as unauthorized port scans or malicious payloads.
- Provide actionable insights to improve network security.

2.2 Solution

The solution leverages **Snort IDS**, an open-source intrusion detection and prevention system, to:

1. Deploy Snort for real-time packet inspection.
2. Configure custom detection rules to identify specific threats.
3. Generate alerts for suspicious activities and log them for further analysis.

2.3 Outcome

- Successfully detected 3-4 threats within 30 days.
- Blocked 3 unauthorized port scans.
- Reduced response time to security incidents by 25%.

—

3 Theoretical Background

3.1 What is an IDS?

A Network Intrusion Detection System (IDS) monitors network traffic for malicious activities or policy violations. It is a passive system that analyzes traffic and generates alerts when suspicious behavior is detected.

3.2 Snort Overview

Snort is a lightweight, open-source IDS that performs:

- **Packet Sniffing:** Captures and inspects packet data in real time.
- **Protocol Analysis:** Decodes protocols to identify abnormal usage.
- **Rule-Based Detection:** Uses customizable rules to detect and respond to threats.

3.3 Key Security Concepts

- **Packet Inspection:** Analyzing data packets to detect anomalies.
- **Detection Rules:** Predefined patterns that match malicious traffic.
- **Alerts:** Notifications generated for suspicious activity.

—

4 Tools and Platforms Used

4.1 Snort IDS

Features:

- Rule-based analysis for precise threat detection.
- Modular design allowing integration with other tools.

4.2 Linux Platform

Why Linux?

- Stability and security make it ideal for IDS deployments.
- Native compatibility with Snort and other network tools.

4.3 Wireshark

Purpose:

- Used to analyze network traffic and validate Snort alerts.

—

5 Step-by-Step Implementation

5.1 Step 1: Setting Up Snort

Installation:

```
sudo apt update
sudo apt install snort -y
```

Configuration: Edit the Snort configuration file:

```
sudo nano /etc/snort/snort.conf
# Update HOME_NET to match your network:
var HOME_NET 192.168.1.0/24
```

5.2 Step 2: Writing Custom Rules

Creating a Rule File: Create a directory for custom rules:

```
sudo mkdir /etc/snort/rules
sudo nano /etc/snort/rules/local.rules
```

Example Rules: Detect ICMP packets (ping):

```
alert icmp any any -> any any (msg:"ICMP detected"; sid:1000001; rev
    :1;)
```

Detect unauthorized SSH login attempts:

```
alert tcp any any -> any 22 (msg:"Unauthorized SSH attempt"; flags:S;
    sid:1000002; rev:1;)
```

5.3 Step 3: Testing the IDS

Run Snort in IDS Mode:

```
sudo snort -A console -q -c /etc/snort/snort.conf -i eth0
```

Validate Rules: Generate test traffic using **ping** or **Nmap**:

```
ping -c 4 192.168.1.1
nmap -p 22 192.168.1.1
```

Check if alerts are triggered in the Snort console.

5.4 Step 4: Automating Alerts and Logging

Log Rotation: Configure log rotation to prevent disk space issues:

```
sudo nano /etc/logrotate.d/snort
# Add:
/var/log/snort/*.log {
    daily
    rotate 7
    compress
    missingok
    notifempty
}
```

Email Alerts: Use **mailutils** to send alerts via email:

```
sudo apt install mailutils
echo "Intrusion detected!" | mail -s "Snort Alert" admin@example.com
```

—

6 Troubleshooting Techniques

6.1 Snort Configuration Errors

- Verify the syntax of custom rules:

```
sudo snort -T -c /etc/snort/snort.conf
```

- Ensure HOME_NET is correctly defined in `snort.conf`.

6.2 Performance Issues

- Use packet capture filters to limit traffic:

```
sudo snort -c /etc/snort/snort.conf -i eth0 'port 80'
```

- Optimize rules to avoid unnecessary processing.

—

7 Conclusion

This project successfully implemented a Network Intrusion Detection System using Snort. It demonstrated the ability to detect threats, log incidents, and generate actionable alerts. By employing automated logging and email alerts, the system ensures timely responses to network security incidents, reducing risks and enhancing overall security.