

A Project Report On

NETWORK SECURITY MONITORING

Submitted in partial fulfillment of the requirement for the
award of the degree

MASTER OF SCIENCE (CS & CL)
from
Marwadi University

Academic Year 2025 – 26

Sirisha Yanala	92400565059
Vendra Isaac	92400565065

Internal Guide
Prof.Akhila B. Mukundhan



Marwadi
University
Marwadi Chandarana Group



Rajkot-Morbi Road, At & PO :Gauridad, Rajkot 360 003. Gujarat. India.



Marwadi
University
Marwadi Chandarana Group



Faculty of Computer Applications (FoCA)

Certificate

This is to certify that the project work entitled
Network Security Monitoring
submitted in partial fulfillment of the requirement for
the award of the degree of
Master of Science (CS & CL)
of the

Marwadi University

is a result of the bonafide work carried out by

Sirisha Yanala 92400565059

Vendra Isaac 92400565065

during the academic year 2025 – 2026

Faculty Guide

HOD

Dean

DECLARATION

We hereby declare that this project work entitled Network Security Monitoring is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place: Marwadi University

Date: 19/08/2025

Sirisha Yanala 92400565059 Signature:_____

Vendra Isaac 92400565065 Signature:_____

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **Akhila B Mukundhan**, the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude to for his /her guidance, suggestions and expertise at every stage. A part from that his/her valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the Dean of our department **Dr. R. Sridaran** sir and HoD **Dr. Sunil Bajeja** sir for giving us an opportunity to work over this project and for their end-less and great support to all other people who directly or indirectly supported and help us to fulfil our task.

Sirisha Yanala 92400565059 Signature:_____

Vendra Isaac 92400565065 Signature:_____

CONTENTS

Chapters	Particulars	Page No.
1	Introduction 1.1. Objective of the New System 1.2. Problem Definition 1.3. Core Components 1.4. Project Profile 1.5. Assumptions and Constraints 1.6. Advantages and Limitations of the Proposed System	7-9
2	Requirement Determination & Analysis 2.1. Requirement Determination 2.2. Targeted Users 2.3. Details of tools and techniques used / implemented 2.4. Advantages and Limitations of the used security tools	10-11
3	System Design 3.1. Flow chart 3.2. Data flow Diagram 3.3. Use case Diagram 3.4. Sequential Diagram 3.5. Activity Diagram 3.6. Description of Files	12-16
4	Development 4.1. Script details / Source code 4.2. Screen Shots / UI Design of simulation (if applicable) 4.3. Test reports	17-25
5	Proposed Enhancements <ul style="list-style-type: none">• Prevention, Mitigation and Suggestion• Future Enhancement	26-28
6	Conclusion	29
7	Bibliography	30

Figure index

S.no	Fig no	Description	Page no
1	3.1	Flow Chart	12
2	3.2	Data Flow Diagram	13
3	3.3	Use Case Diagram	13
4	3.4	Sequence Diagram	14
5	3.5	Activity Diagram	15
6	3.6	Description of files	16
7	4.1	Update and Upgrade System	17
8	4.2	Snort Installation	17
9	4.3	Creating folders to save alert logs	18
10	4.4	Setting rules in local.rules folder	18
11	4.5	Logs detecting in console mode	19
12	4.6	Downloaded Splunk and Universal Forwarder Files	19
13	4.7	Starting Splunk server	20
14	4.8	Accessing Splunk server	21
15	4.9	Splunk Login	21
16	4.10	Adding default port to receive logs	22
17	4.2.1	Host connected to Splunk server	23
18	4.2.2	Events Dashboards	24
19	4.2.3	Real Time Dashboards	24

Table index

S.no	Table no	Description	Page
1	Table 1	Details of Tools and Techniques Used	11
2	Table 2	Test Reports	25

Chapter-1

Introduction

1.1 Objective of the New System

The primary objective of this project is to design and implement a Network Traffic Monitoring System that detects malicious activities, generates security alerts using Snort (Intrusion Detection System), and visualizes real-time logs and metrics in Splunk dashboards. The system has been implemented in a virtualized environment using VMware Workstation, where Kali Linux is configured as the attacker machine and Ubuntu as the victim machine to simulate real-world attack and defense scenarios.

The system ensures:

- Continuous monitoring of network traffic.
- Early detection of suspicious activities and intrusions.
- Visualization of security events for easier decision-making.
- Actionable insights to strengthen overall security posture.

1.2 Problem Definition

With the exponential growth of cyber threats, organizations face increasing challenges in monitoring large-scale network traffic and identifying malicious patterns. Traditional log monitoring methods are manual, time-consuming, and less efficient, making them unsuitable for modern dynamic networks. To address this gap, there is a need for an automated network traffic monitoring system that:

- Detects attacks such as port scanning, brute-force attempts, and denial-of-service (DoS) attacks.
- Provides real-time visibility into network traffic behavior and anomalies.
- Helps security teams respond quickly to incidents through automated alerts and visual insights.
- Simulates realistic attack-defense scenarios in a controlled environment for testing and validation.

1.3 Core Components

- Snort IDS – An open-source Intrusion Detection System used to capture, analyze, and log suspicious network traffic on the victim machine (Ubuntu).
- Splunk – A powerful log analysis and visualization platform used to index Snort logs and create interactive security dashboards.
- Splunk Universal Forwarder – Deployed on Ubuntu to forward Snort-generated logs in real time to the Splunk server.
- Network Traffic – Input data consisting of packets captured within the virtual network in VMware, including both normal and malicious traffic.

- Virtual Machines (VMware Setup):
 - Kali Linux (Attacker Machine): Used to simulate different types of attacks such as port scanning (Nmap), brute-force attempts, denial-of-service (DoS) attacks, ping-based attacks (ICMP flooding), and HTTP requests using curl. These attacks helped validate the effectiveness of Snort in detecting intrusions and forwarding alerts to Splunk.
 - Ubuntu (Victim Machine): Configured with Snort IDS and Splunk Universal Forwarder to detect incoming suspicious traffic, generate alerts, and forward logs for analysis and visualization.

1.4 Project Profile

- Project Title: Network Traffic Monitoring using Snort and Splunk
- Domain: Cybersecurity / Network Security Monitoring
- Technology Used: Snort, Splunk, Universal Forwarder, Linux/Ubuntu
- Application Area: Enterprise network monitoring, Cyber Defense, SOC (Security Operation Center)

1.5 Assumptions and Constraints

For the successful implementation of the **Network Traffic Monitoring system**, certain assumptions and constraints are considered. It is assumed that Snort rules are properly configured to effectively detect various types of network anomalies, and that the network traffic being monitored contains both normal and suspicious packets to validate detection accuracy. Additionally, the Splunk server is assumed to be accessible and capable of ingesting logs forwarded from Snort without interruptions.

However, the system also comes with specific constraints. Since network monitoring generates large volumes of log data, sufficient storage capacity is required to handle these files efficiently. In high-speed network environments, performance overhead may occur due to the continuous packet inspection and log forwarding process. Furthermore, while Splunk provides powerful indexing and visualization features, its enterprise-level deployment requires appropriate licensing, which may limit scalability for larger organizations.

1.6 Advantages and Limitations of the Proposed System

Advantages:

- Enables real-time monitoring and alerting to quickly detect suspicious activities.
- Provides intuitive dashboards in Splunk for clear visualization of network events.
- Scalable design allows deployment across small to large networks.
- Enhances incident response speed through centralized log analysis.
- Supports integration with other security tools for broader protection.

Limitations:

- Splunk enterprise licensing can be costly for large-scale use.
- Depends heavily on Snort rules, which require proper updates and tuning.
- May generate false positives, increasing analyst workload.
- High-speed traffic can cause performance and resource overheads.
- Requires skilled personnel for effective setup and management.

Chapter-2

Requirements Determination & Analysis

2. Requirement Determination & Analysis

2.1 Requirement Determination:

Hardware Requirements:

Minimum 8 GB RAM, 4-core CPU, 100 GB storage.

Stable internet/network connection.

Software Requirements:

Operating System: Ubuntu, Linux Server.

Snort IDS installed and configured.

Splunk Enterprise/Free edition.

Splunk Universal Forwarder.

2.2 Targeted Users

Network Administrators – To monitor traffic flow.

Security Analysts – To analyze logs and detect attacks.

SOC Teams – To respond to security incidents.

Students/Researchers – For learning intrusion detection.

2.3 Details of Tools and Techniques Used

Category Tool	Technique Purpose	Usage
Operating System	Linux (Ubuntu)	Provides the environment for deploying Snort, Splunk, and monitoring tools.
Intrusion Detection	Snort	Captures network packets, applies rules, and generates alerts for suspicious traffic.
Log Forwarding	Splunk Universal Forwarder	Forwards Snort logs and system logs to Splunk in real-time
SIEM & Visualization	Splunk Enterprise	Collects, indexes, and visualizes logs/dashboards for network monitoring
Alert Management	Snort Rules(Custom + Default)	Detects suspicious patterns(e.g., port scans, brute force, malicious payloads).
Visualization	Splunk Dashboards, Graphs, and Alerts	Provides security insights, anomaly detection, and trend analysis.
Security Techniques	Intrusion Detection (IDS), Log Analysis, Traffic Monitoring	Ensures proactive detection and monitoring of threats in real-time.

Table 1

2.4 Advantages and Limitations of the Used Security Tools

Snort:

Advantage: Free, customizable rules, lightweight.

Limitation: High false positives if rules are not fine-tuned.

Splunk:

Advantage: Powerful data visualization and searching.

Limitation: Free version has data indexing limits.

Chapter-3

System Design

3.1 Flow Chart

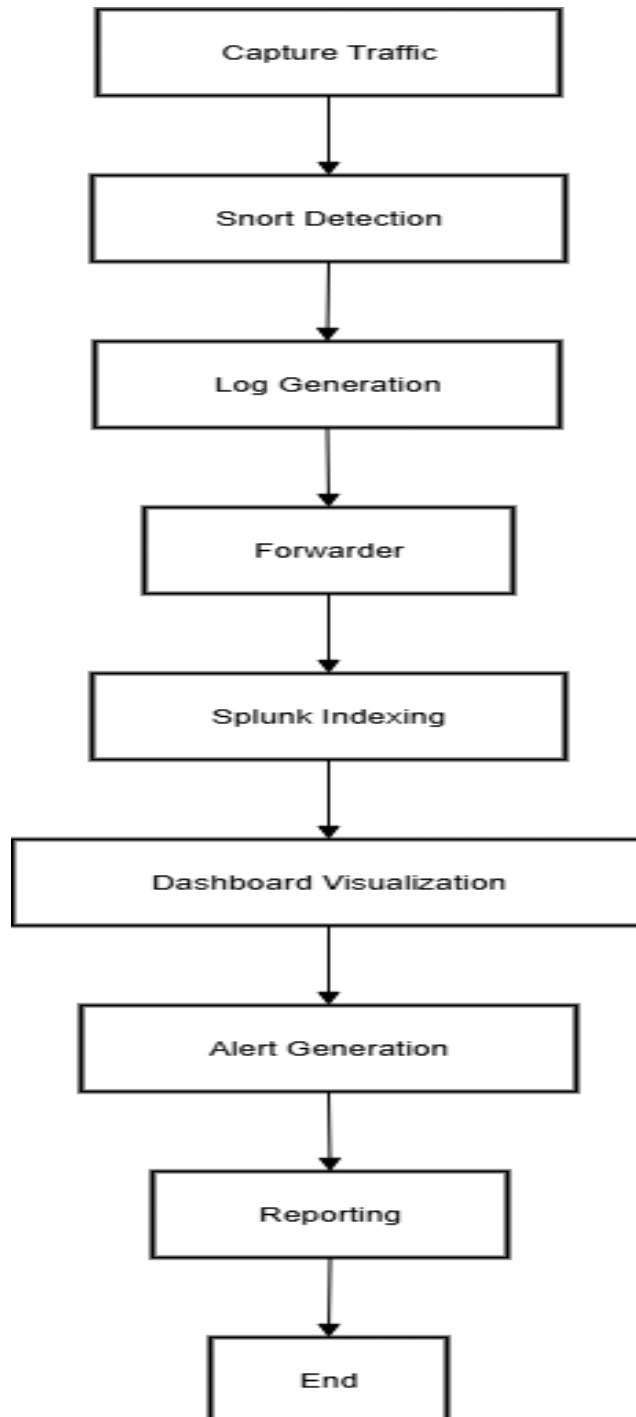


Fig : 3.1

3.2 Data Flow Diagram (DFD)

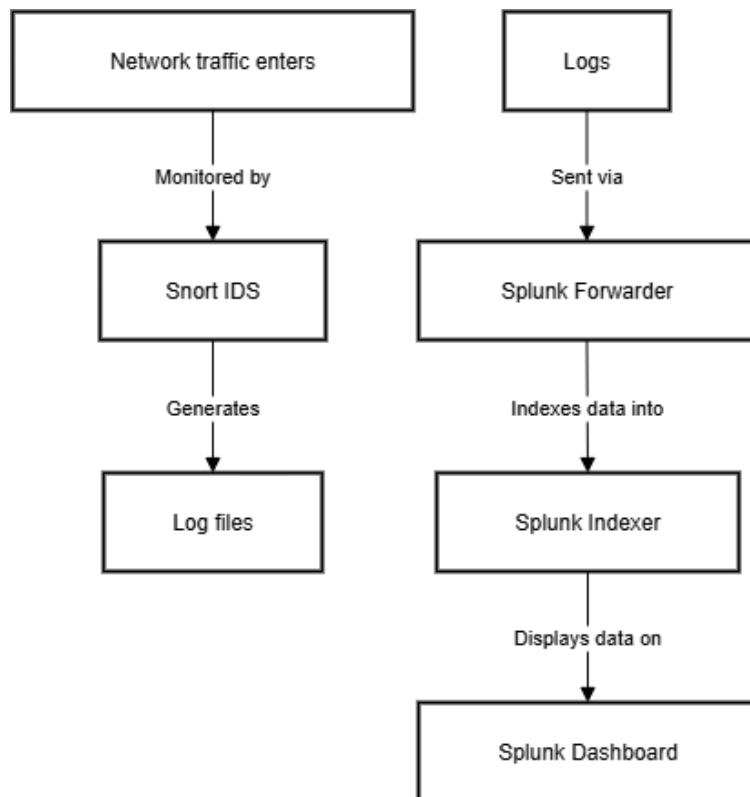


Fig : 3.2

3.3 Use Case Diagram

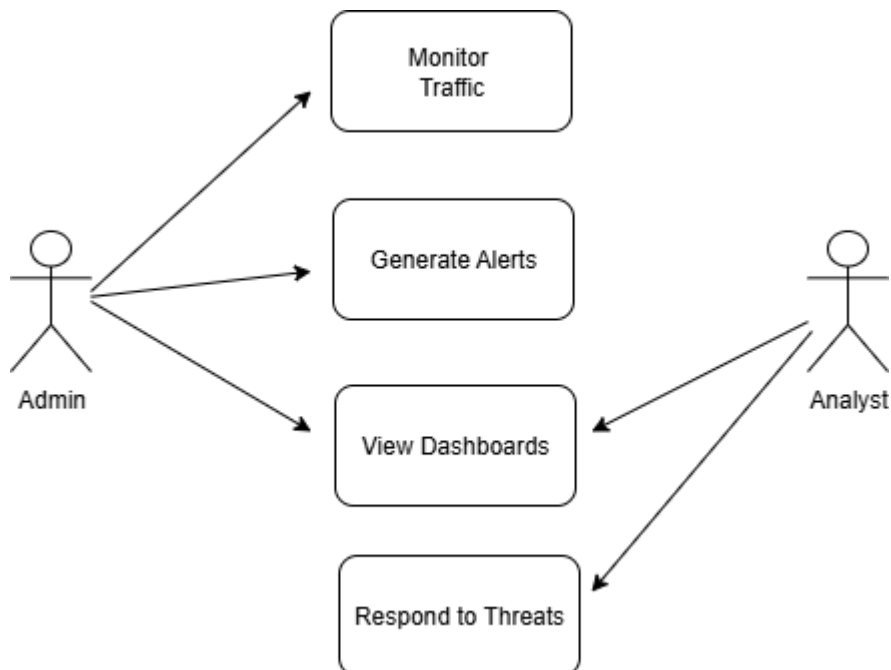


Fig : 3.3

3.4 Sequence Diagram

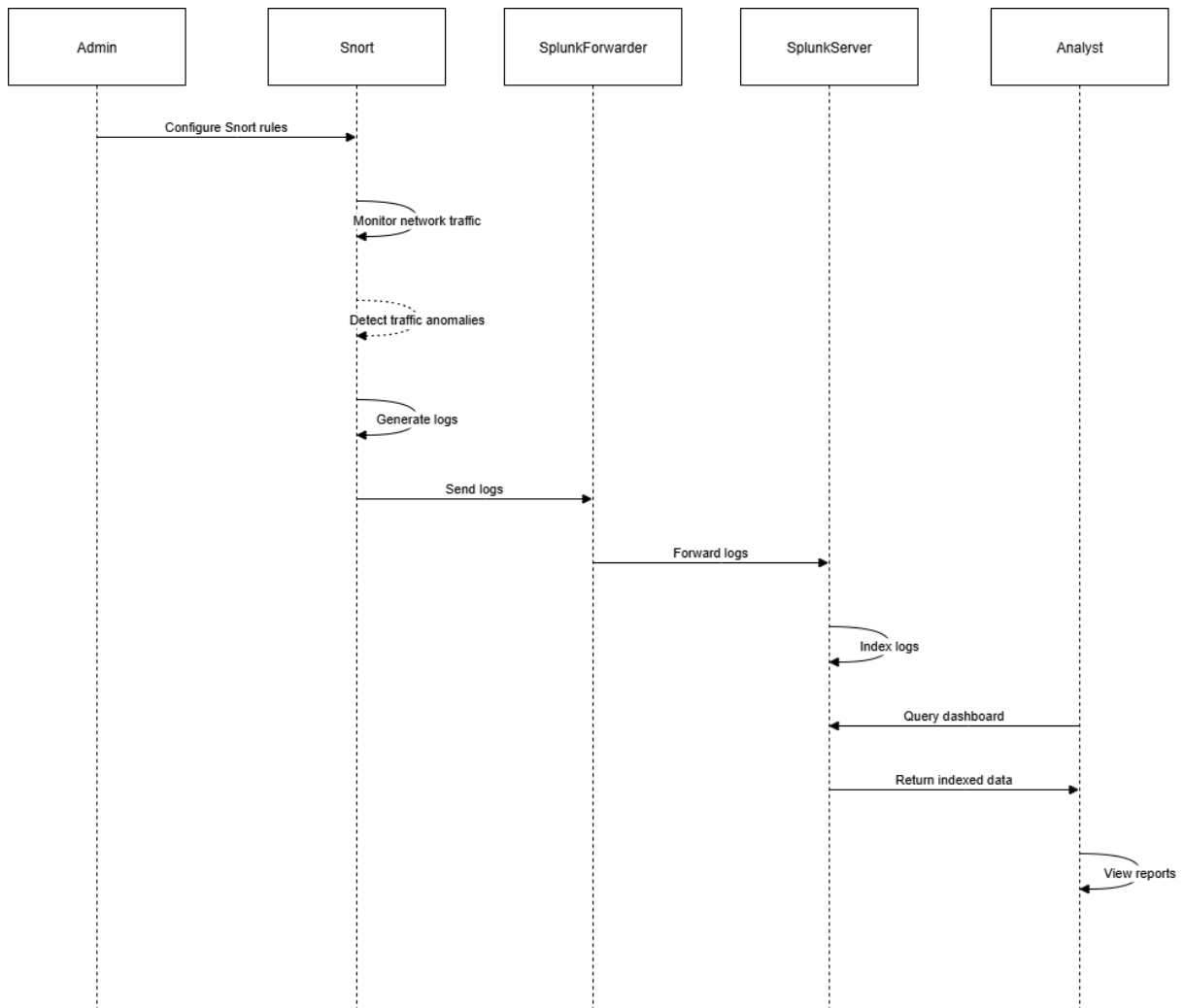


Fig : 3.4

3.5 Activity Diagram

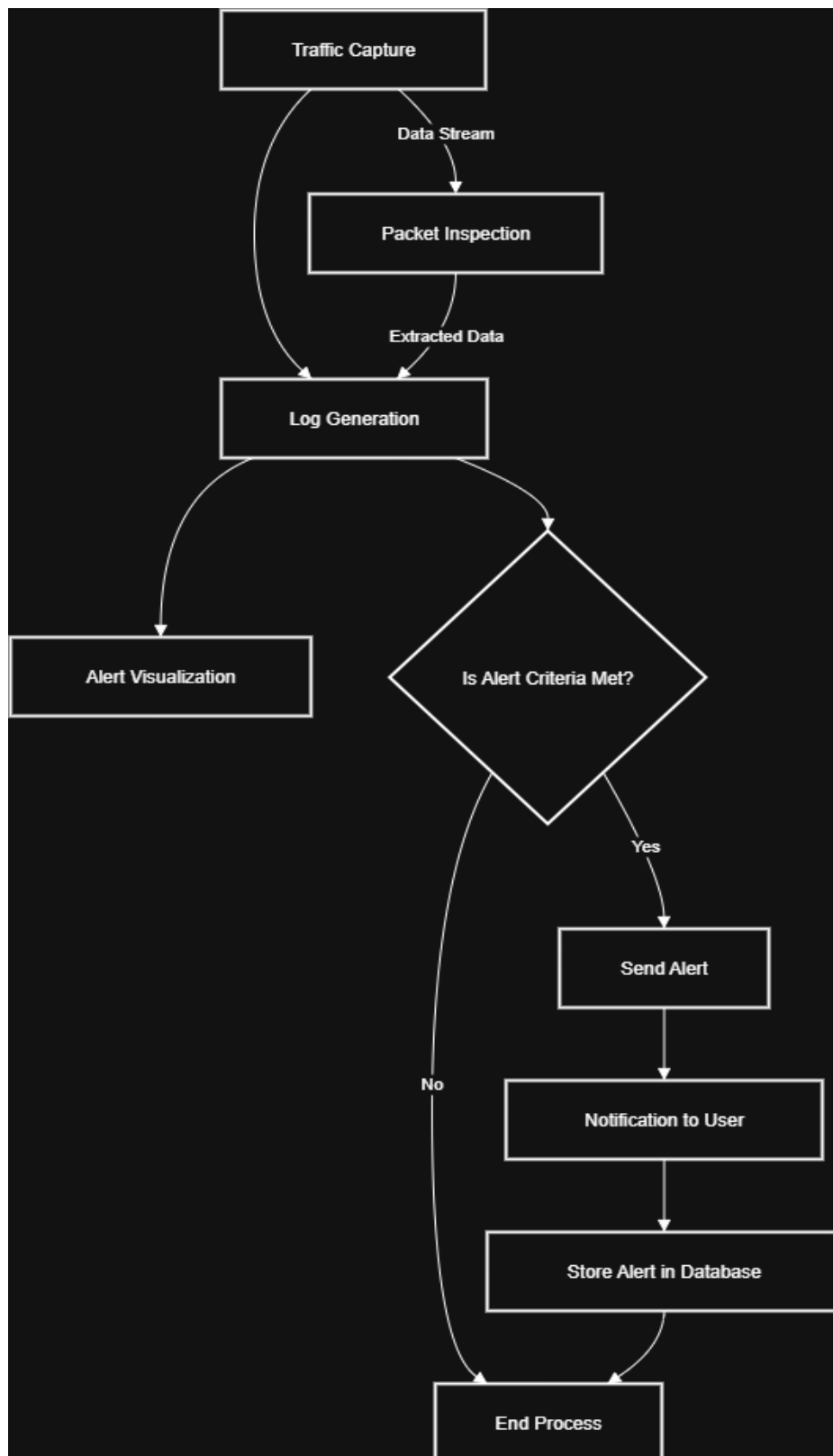


Fig : 3.5

3.6 Description of Files



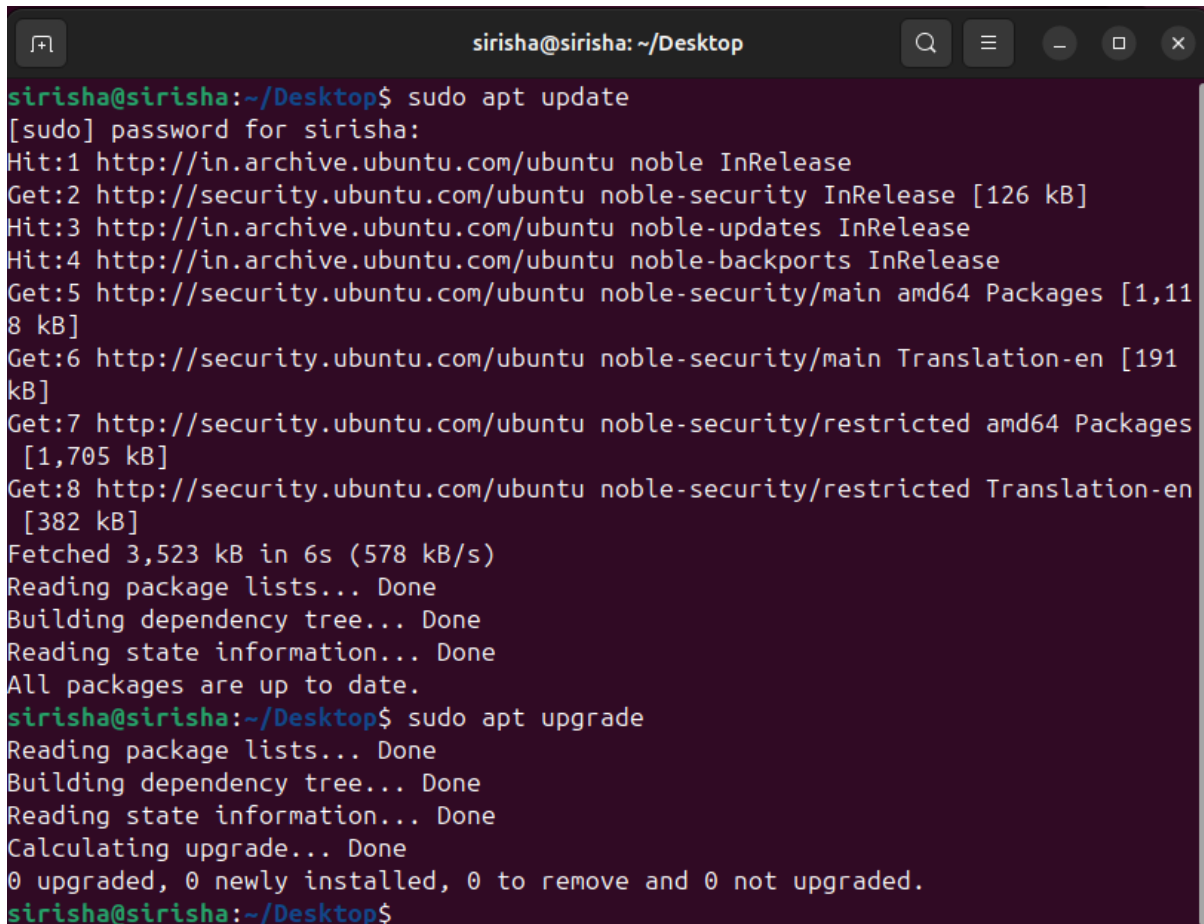
Fig : 3.6

Chapter-4

Development

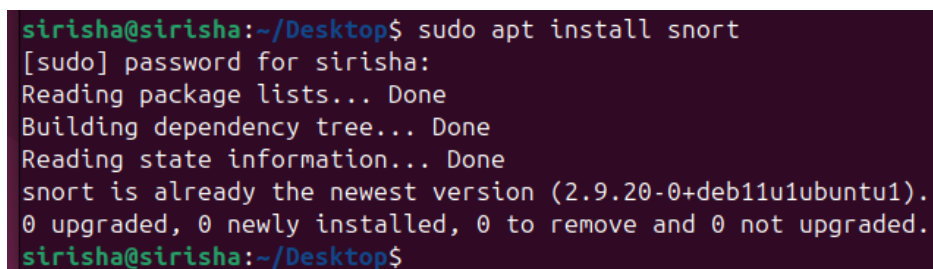
4.1 Script details/ Source code

- System Setup

A terminal window titled 'sirisha@sirisha: ~/Desktop' with standard Ubuntu window controls. The terminal shows the execution of 'sudo apt update' and 'sudo apt upgrade'. The update command lists various sources and packages to be updated, including security updates and translations. The upgrade command shows that no packages need to be upgraded.

```
sirisha@sirisha:~/Desktop$ sudo apt update
[sudo] password for sirisha:
Hit:1 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,118 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [191 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1,705 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [382 kB]
Fetched 3,523 kB in 6s (578 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
sirisha@sirisha:~/Desktop$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sirisha@sirisha:~/Desktop$
```

Fig 4.1 : Update and Upgrade system

A terminal window showing the command 'sudo apt install snort'. The output indicates that snort is already the newest version installed on the system.

```
sirisha@sirisha:~/Desktop$ sudo apt install snort
[sudo] password for sirisha:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snort is already the newest version (2.9.20-0+deb11u1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sirisha@sirisha:~/Desktop$
```

Fig 4.2 : Snort Installation

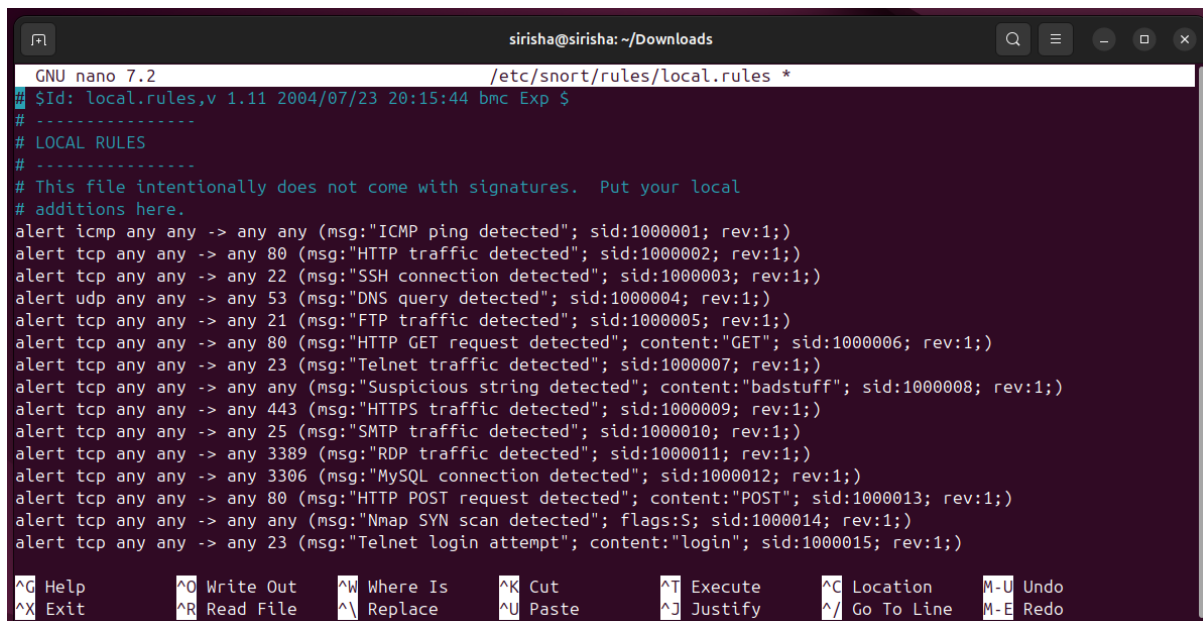
```

sirisha@sirisha:~/Downloads$ sudo mkdir /var/log/alert
mkdir: cannot create directory '/var/log/alert': File exists
sirisha@sirisha:~/Downloads$ sudo chmod 777 /var/log/alert
sirisha@sirisha:~/Downloads$

```

Fig 4.3 : Creating folders to save alert logs

- Setting Rules in /etc/snort/rules folder & configure these rules using /etc/snort/snort.conf command



```

GNU nano 7.2 /etc/snort/rules/local.rules *
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert icmp any any -> any any (msg:"ICMP ping detected"; sid:1000001; rev:1;)
alert tcp any any -> any 80 (msg:"HTTP traffic detected"; sid:1000002; rev:1;)
alert tcp any any -> any 22 (msg:"SSH connection detected"; sid:1000003; rev:1;)
alert udp any any -> any 53 (msg:"DNS query detected"; sid:1000004; rev:1;)
alert tcp any any -> any 21 (msg:"FTP traffic detected"; sid:1000005; rev:1;)
alert tcp any any -> any 80 (msg:"HTTP GET request detected"; content:"GET"; sid:1000006; rev:1;)
alert tcp any any -> any 23 (msg:"Telnet traffic detected"; sid:1000007; rev:1;)
alert tcp any any -> any any (msg:"Suspicious string detected"; content:"badstuff"; sid:1000008; rev:1;)
alert tcp any any -> any 443 (msg:"HTTPS traffic detected"; sid:1000009; rev:1;)
alert tcp any any -> any 25 (msg:"SMTP traffic detected"; sid:1000010; rev:1;)
alert tcp any any -> any 3389 (msg:"RDP traffic detected"; sid:1000011; rev:1;)
alert tcp any any -> any 3306 (msg:"MySQL connection detected"; sid:1000012; rev:1;)
alert tcp any any -> any 80 (msg:"HTTP POST request detected"; content:"POST"; sid:1000013; rev:1;)
alert tcp any any -> any any (msg:"Nmap SYN scan detected"; flags:S; sid:1000014; rev:1;)
alert tcp any any -> any 23 (msg:"Telnet login attempt"; content:"login"; sid:1000015; rev:1;)

```

Fig 4.4 : Setting rules in local.rules folder

- Running Snort in Console mode to Detect Attacks

```

sirisha@sirisha: ~/Downloads
sirisha@sirisha:~/Downloads$ sudo snort -A console -A full -q -u snort -c /etc/s
nort/snort.conf -i ens33
09/04-08:54:39.553944  /**] [1:1000004:1] DNS query detected /**] [Priority: 0]
{UDP} 192.168.79.132:44004 -> 192.168.79.2:53
09/04-08:54:39.764012  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:445
09/04-08:54:39.764018  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:143
09/04-08:54:39.764018  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:8080
09/04-08:54:39.764019  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:111
09/04-08:54:39.764020  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:1723
09/04-08:54:39.764020  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:199
09/04-08:54:39.764021  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:21
09/04-08:54:39.764021  /**] [1:1000005:1] FTP traffic detected /**] [Priority: 0
] {TCP} 192.168.79.132:37386 -> 192.168.79.131:21
09/04-08:54:39.764022  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:139
09/04-08:54:39.764481  /**] [1:1000014:1] Nmap SYN scan detected /**] [Priority:
0] {TCP} 192.168.79.132:37386 -> 192.168.79.131:1720

```

Fig 4.5 : Logs detecting in console mode

- Download Splunk and Splunk Universal Forwarder

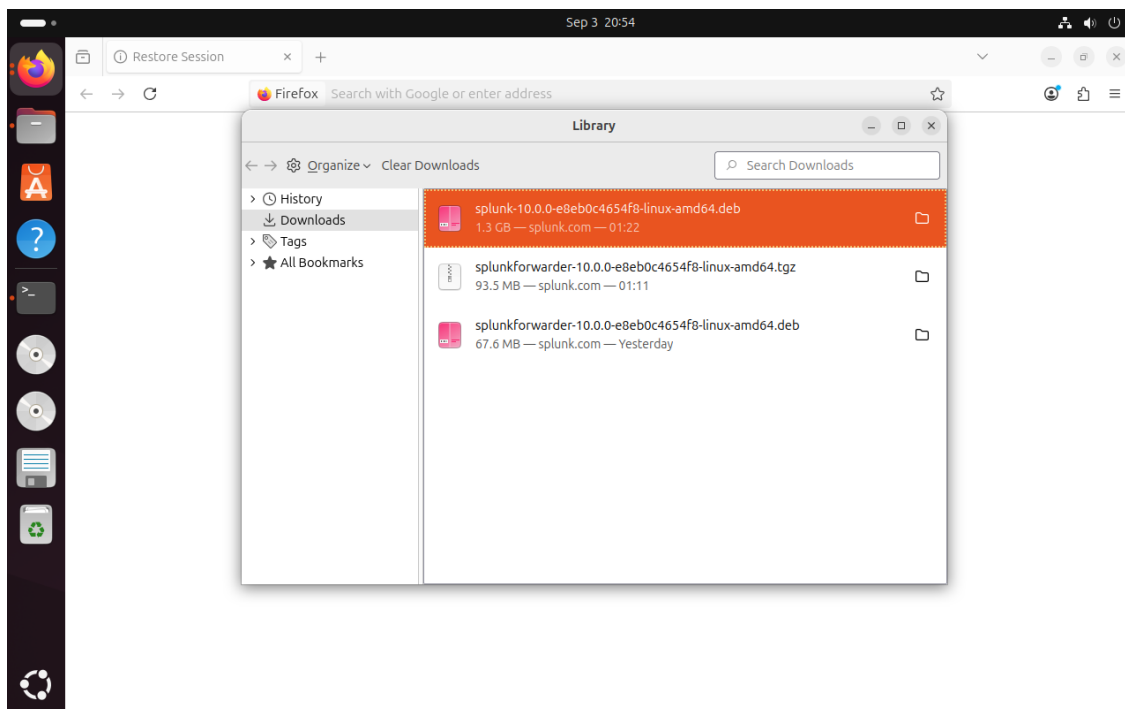
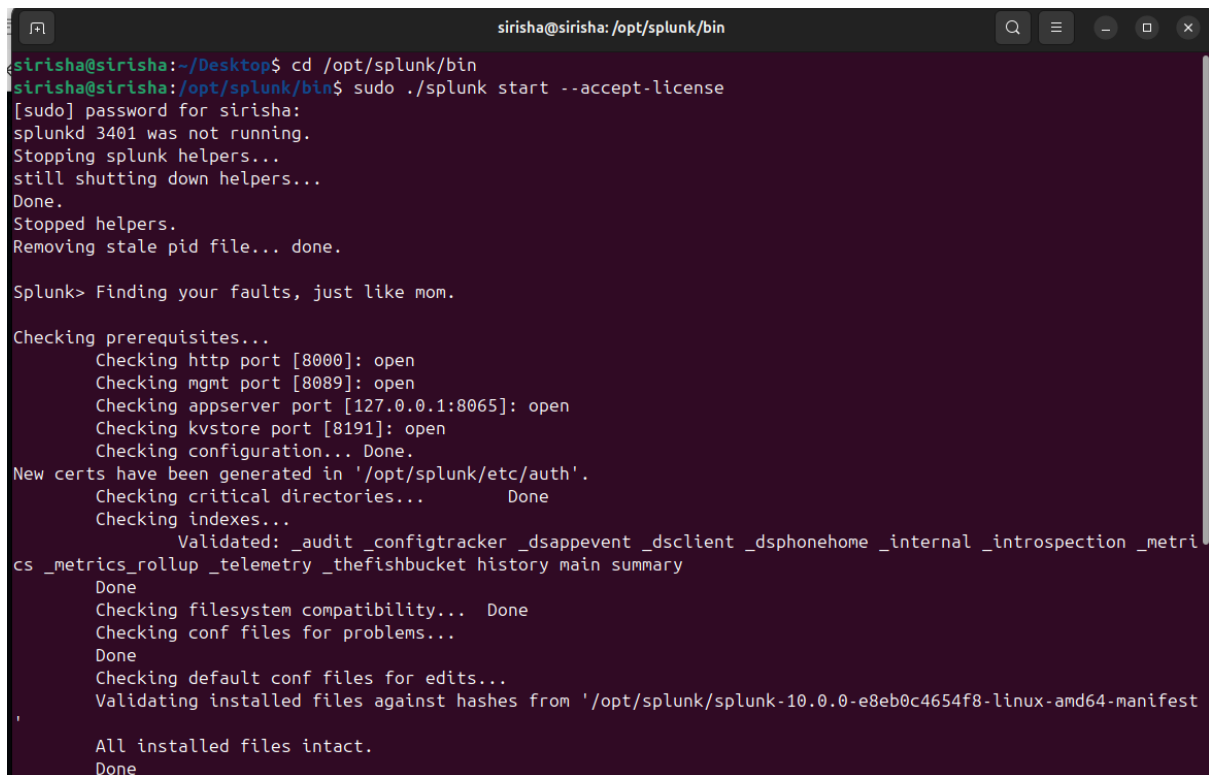


Fig 4.6 : Downloaded Splunk and Universal Forwarder Files

- Starting Splunk and accessing Splunk server



```
sirisha@sirisha:~/Desktop$ cd /opt/splunk/bin
sirisha@sirisha:/opt/splunk/bin$ sudo ./splunk start --accept-license
[sudo] password for sirisha:
splunkd 3401 was not running.
Stopping splunk helpers...
still shutting down helpers...
Done.
Stopped helpers.
Removing stale pid file... done.

Splunk> Finding your faults, just like mom.

Checking prerequisites...
  Checking http port [8000]: open
  Checking mgmt port [8089]: open
  Checking appserver port [127.0.0.1:8065]: open
  Checking kvstore port [8191]: open
  Checking configuration... Done.
New certs have been generated in '/opt/splunk/etc/auth'.
  Checking critical directories... Done
  Checking indexes...
    Validated: _audit _configtracker _dsappevent _dsclient _dsphonehome _internal _introspection _metrics
cs _metrics_rollup _telemetry _thefishbucket history main summary
    Done
  Checking filesystem compatibility... Done
  Checking conf files for problems...
    Done
  Checking default conf files for edits...
  Validating installed files against hashes from '/opt/splunk/splunk-10.0.0-e8eb0c4654f8-linux-amd64-manifest'
    Done
  All installed files intact.
    Done
```

Fig 4.7 : Starting Splunk server

```
sirisha@sirisha: /opt/splunk/bin

Checking appserver port [127.0.0.1:8065]: open
Checking kvstore port [8191]: open
Checking configuration... Done.
New certs have been generated in '/opt/splunk/etc/auth'.
Checking critical directories... Done
Checking indexes...
Validated: _audit _configtracker _dsappevent _dsclient _dsphonehome _internal _introspection _metrics
cs _metrics_rollup _telemetry _thefishbucket history main summary
Done
Checking filesystem compatibility... Done
Checking conf files for problems... Done
Checking default conf files for edits... Done
Validating installed files against hashes from '/opt/splunk/splunk-10.0.0-e8eb0c4654f8-linux-amd64-manifest'
All installed files intact.
Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Done

Waiting for web server at http://127.0.0.1:8000 to be available..... Done

If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://sirisha:8000

sirisha@sirisha: /opt/splunk/bin$
```

Fig 4.8 : Accessing Splunk server

- Access Splunk using credentials given above

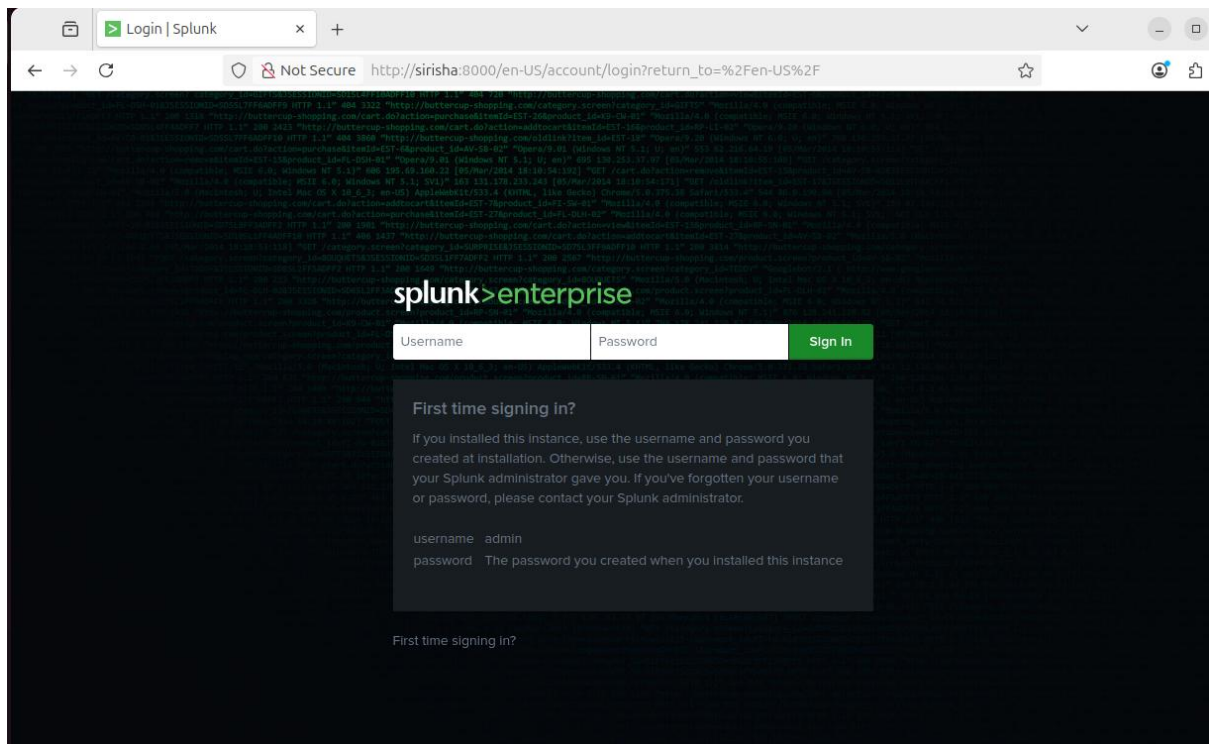


Fig 4.9: Splunk Login

- Giving default port number to receive logs using Universal Forwarder

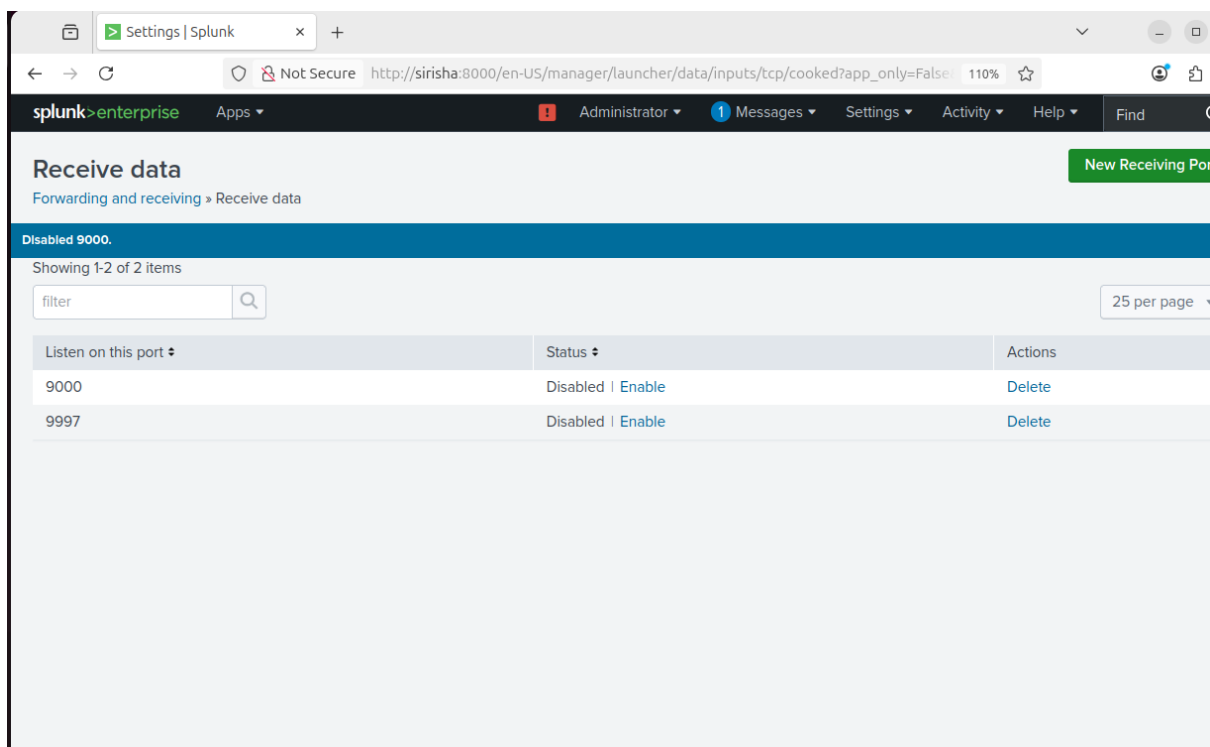
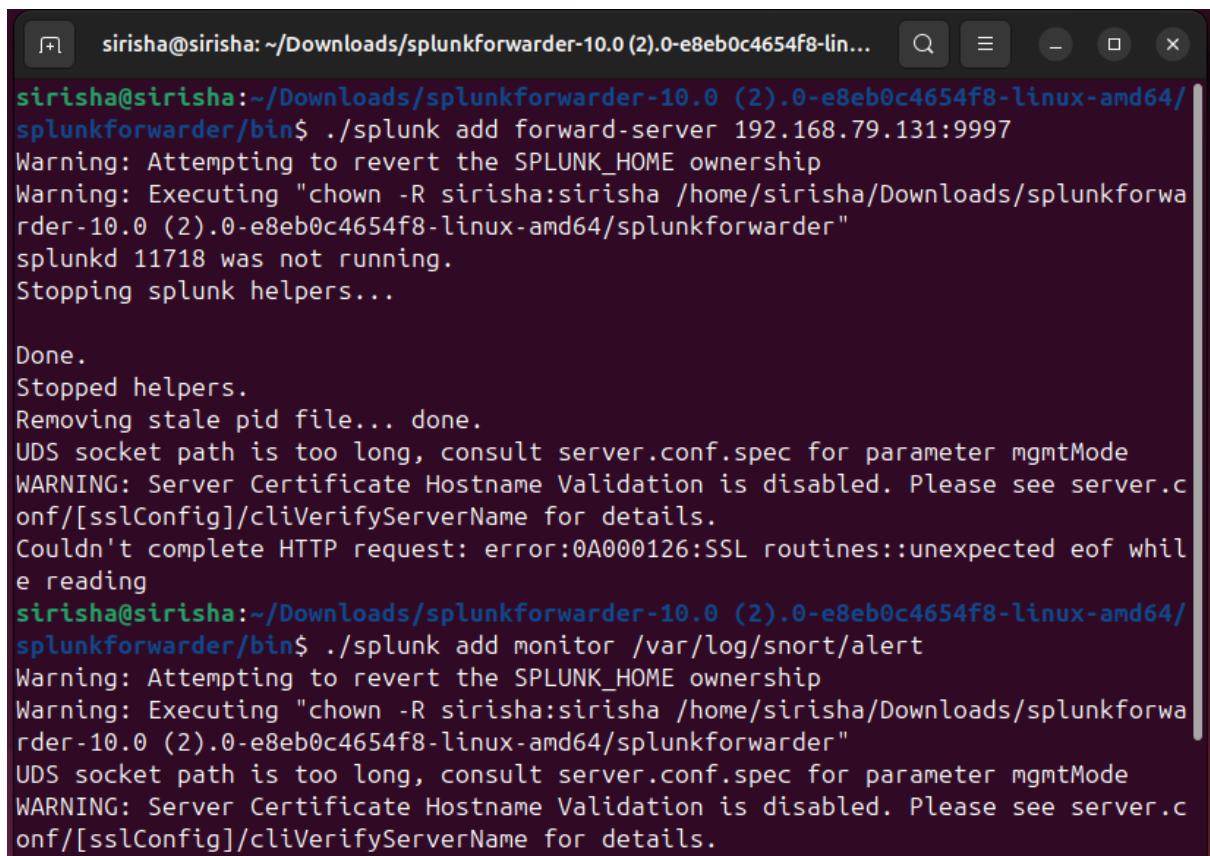


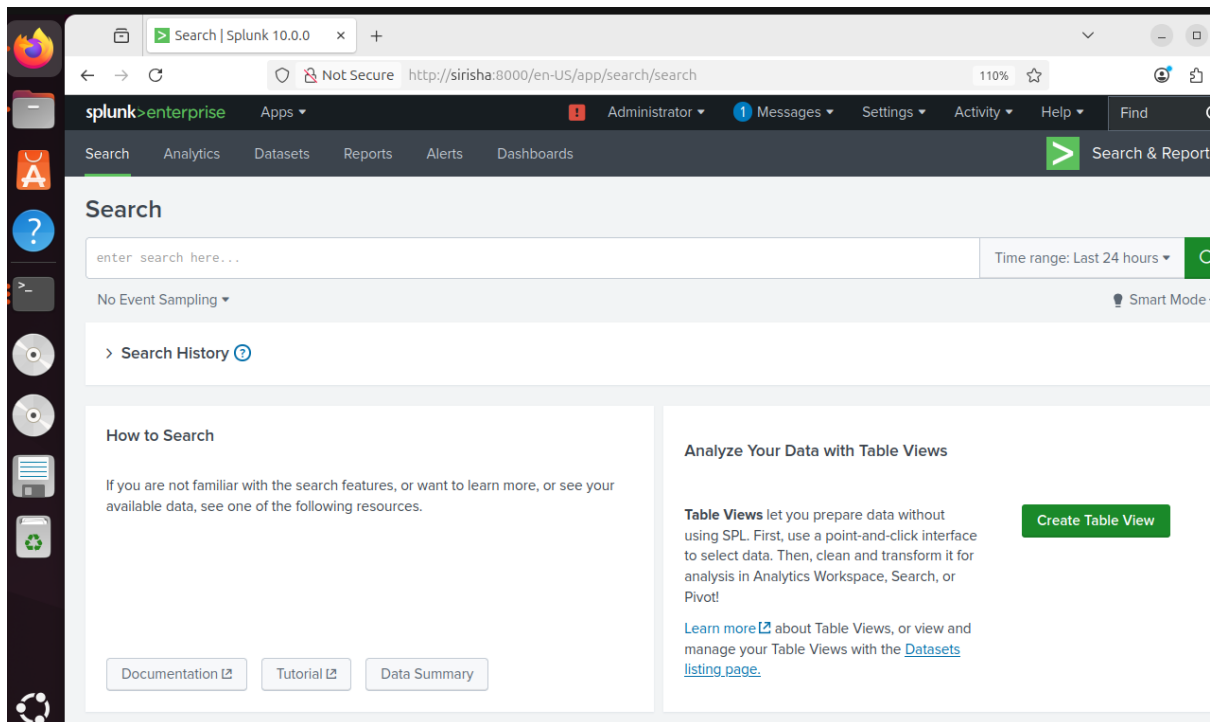
Fig 4.10 : Adding default port to receive logs

- Adding forward-server and monitoring the alerts folder to receive logs from snort.



4.2 Screen Shots / UI Design simulation

- By clicking Data Summary in this window



- If port is enabled, forward-server added and monitoring is added it shows the logs receiving here by changing time range to 30 sec the count will increase.

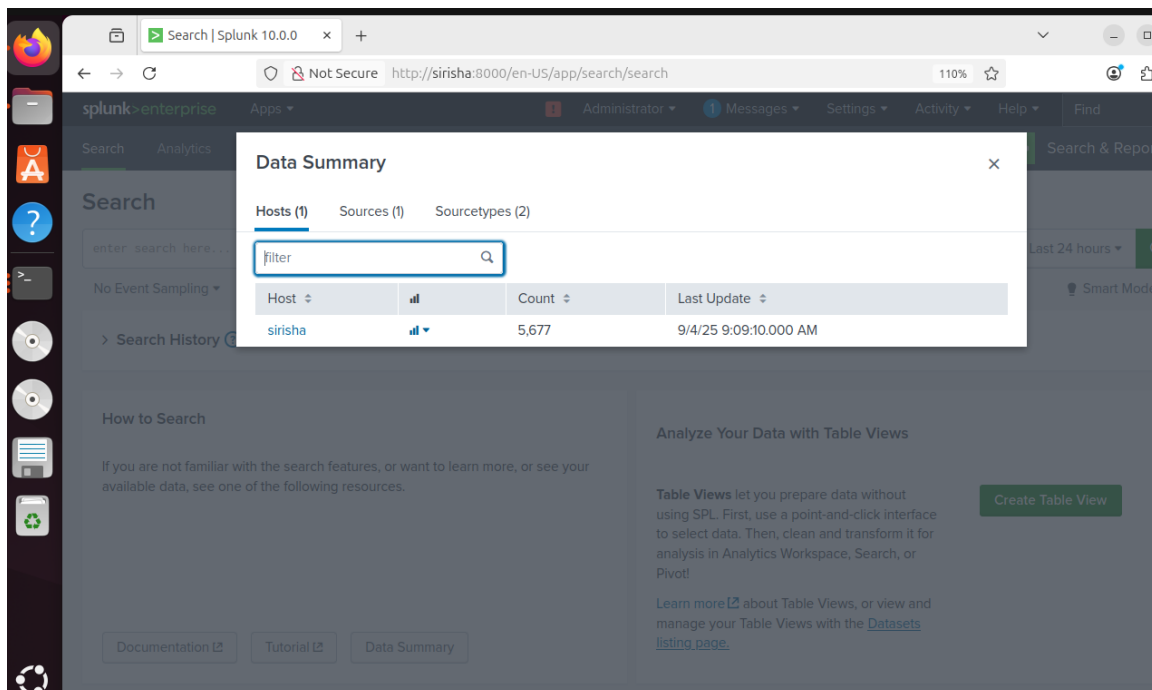
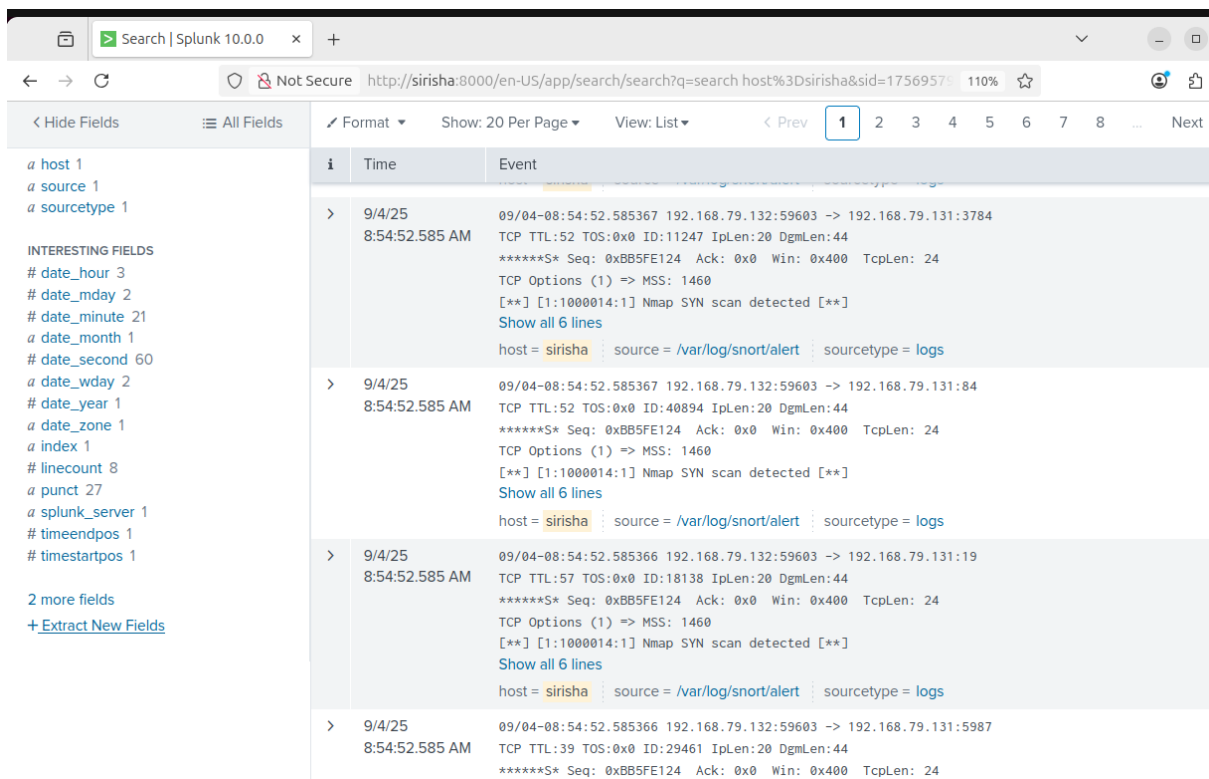


Fig 4.2.1 : Host connected to Splunk server

Dashboards

- These are the event logs dashboard, It shows the raw data which is directly receiving from snort alerts folder.



The screenshot shows the Splunk Events Dashboard interface. On the left, there's a sidebar with 'All Fields' and 'INTERESTING FIELDS' including date and time fields. The main panel displays a table of events with columns for Time and Event. The events are filtered by 'host=sirisha' and show network traffic logs with details like IP addresses, ports, and protocols.

Time	Event
9/4/25 8:54:52.585 AM	09/04-08:54:52.585367 192.168.79.132:59603 -> 192.168.79.131:3784 TCP TTL:52 TOS:0x0 ID:11247 IpLen:20 DgmLen:44 *****S* Seq: 0xBB5FE124 Ack: 0x0 Win: 0x400 TcpLen: 24 TCP Options (1) => MSS: 1460 [**] [1:1000014:1] Nmap SYN scan detected [**] Show all 6 lines host = sirisha source = /var/log/snort/alert sourcetype = logs
9/4/25 8:54:52.585 AM	09/04-08:54:52.585367 192.168.79.132:59603 -> 192.168.79.131:84 TCP TTL:52 TOS:0x0 ID:40894 IpLen:20 DgmLen:44 *****S* Seq: 0xBB5FE124 Ack: 0x0 Win: 0x400 TcpLen: 24 TCP Options (1) => MSS: 1460 [**] [1:1000014:1] Nmap SYN scan detected [**] Show all 6 lines host = sirisha source = /var/log/snort/alert sourcetype = logs
9/4/25 8:54:52.585 AM	09/04-08:54:52.585366 192.168.79.132:59603 -> 192.168.79.131:19 TCP TTL:57 TOS:0x0 ID:18138 IpLen:20 DgmLen:44 *****S* Seq: 0xBB5FE124 Ack: 0x0 Win: 0x400 TcpLen: 24 TCP Options (1) => MSS: 1460 [**] [1:1000014:1] Nmap SYN scan detected [**] Show all 6 lines host = sirisha source = /var/log/snort/alert sourcetype = logs
9/4/25 8:54:52.585 AM	09/04-08:54:52.585366 192.168.79.132:59603 -> 192.168.79.131:5987 TCP TTL:39 TOS:0x0 ID:29461 IpLen:20 DgmLen:44 *****S* Seq: 0xBB5FE124 Ack: 0x0 Win: 0x400 TcpLen: 24

Fig 4.2.2 : Events Dashboard

Here the graph will move while receiving the logs

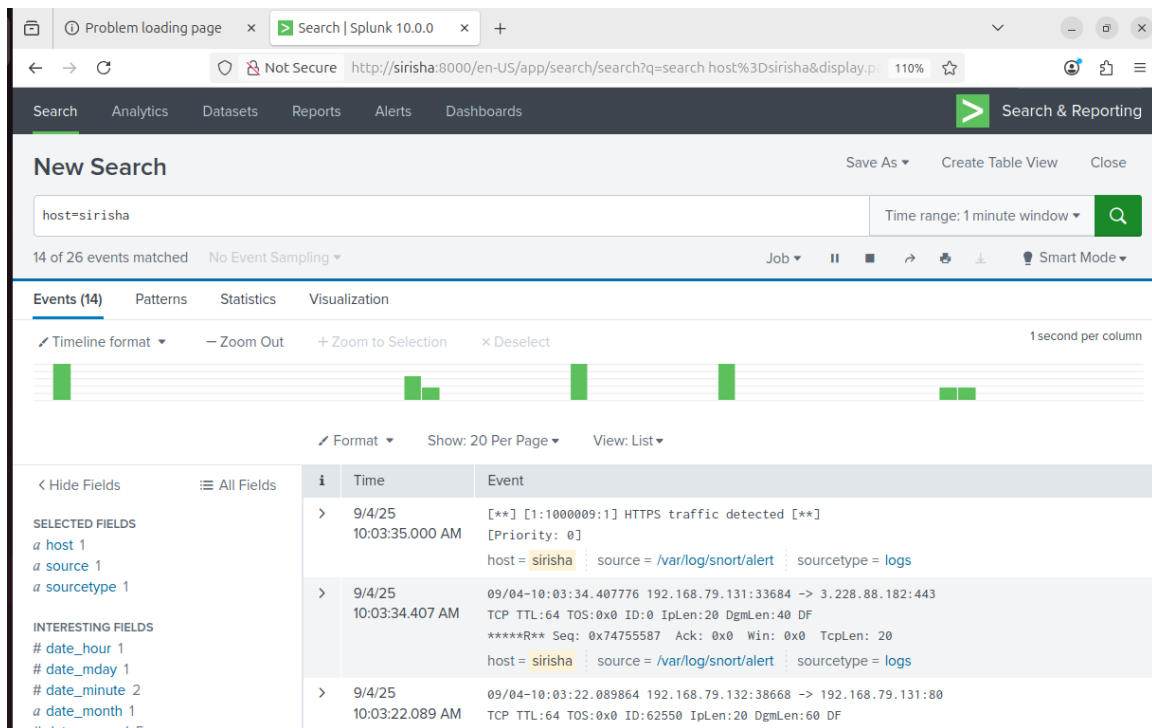


Fig 4.2.3 : Realtime Dashboard

4.3 Test Reports

Test Scenarios & Test Cases

Test Case ID	Scenario	Expected Result	Status
TC-01	Traffic Capture	Captured successfully	Pass
TC-02	Snort Rule Detection	Alerts logged correctly	Pass
TC-03	Log Forwarding	Logs received in Splunk	Pass
TC-04	Splunk Indexing	Indexed successfully	Pass
TC-05	Dashboard Visualization	Alerts displayed	Pass
TC-06	Malicious Scan Detection	Alerts generated & logged	Pass
TC-07	False Positive Handling	No false alerts observed	Pass

Table 2

Chapter-5

Proposed Enhancements

Proposed Enhancements

1. Integration with Machine Learning Models

- Enhance Snort detection by applying ML algorithms to detect unknown (zero-day) attacks.
- Use anomaly-based detection instead of only rule/signature-based detection.

2. Automated Incident Response

- Configure Splunk to trigger automated scripts (e.g., block IP, disable port) when critical alerts are detected.
- Reduce manual intervention for faster mitigation.

3. Threat Intelligence Integration

- Integrate external threat intelligence feeds with Splunk to enrich alerts with context (e.g., malicious IP reputation, blacklists).
- Improves detection accuracy.

4. Scalability Improvements

- Deploy the system in a distributed architecture to handle larger enterprise traffic volumes.
- Use load balancing for Snort sensors across multiple network segments.

5. Advanced Dashboard & Reporting

- Add visual analytics like geolocation maps of attacks, heatmaps, and trend analysis in Splunk.
- Automated weekly/monthly reports for management.

6. Log Correlation with Other Sources

- Extend monitoring to include firewall logs, IDS/IPS logs, system logs, and endpoint security data.
- Helps build a complete SIEM ecosystem.

7. Alert Prioritization

- Implement a severity-based classification system for alerts.
- Helps analysts focus on critical issues first.

Preventions, Mitigations, and Suggestions

Preventions

- **Network Segmentation** – Isolate critical systems from general network traffic to minimize attack spread.
- **Access Control** – Enforce least-privilege access for users and administrators.
- **Patch Management** – Regularly update operating systems, Snort, Splunk, and dependent libraries to close known vulnerabilities.
- **Encryption** – Use TLS for log forwarding between Splunk Forwarders and Indexers to prevent interception.
- **Strong Authentication** – Apply multi-factor authentication (MFA) for accessing Splunk dashboards and administrative consoles.

Mitigations

- **Rule Optimization** – Regularly review and tune Snort signatures to reduce false positives and improve detection accuracy.
- **Real-time Alerts** – Configure Splunk to trigger alerts for high-severity events (e.g., SQL injection, brute-force attempts).
- **Incident Response** – Integrate Splunk alerts with automated response systems (SOAR) to contain threats quickly.
- **Threat Intelligence Feeds** – Enrich Snort and Splunk with external intelligence (blocklists, IP reputation databases) for faster detection.
- **Backup & Recovery** – Maintain periodic Splunk configuration and index backups to recover monitoring data in case of compromise.

Suggestions

- **Awareness Training** – Train staff to understand dashboards, alerts, and common attack patterns.
- **Continuous Monitoring** – Establish 24/7 monitoring using Splunk dashboards and automated reporting.
- **Integration with SIEM** – Enhance Splunk with enterprise SIEM capabilities (Splunk ES, QRadar, or OSSIM) for broader correlation.
- **Performance Monitoring** – Track Splunk Forwarder and Indexer performance to ensure log ingestion is not delayed.
- **Future Expansion** – Extend the monitoring framework to cover **cloud logs, endpoint logs, and IoT devices** for holistic coverage.

Future Enhancements

- **Distributed Monitoring**
 - Deploying multiple Snort sensors across different network zones such as DMZ, internal LAN, and data centers can provide comprehensive coverage of network traffic.
 - Aggregating logs from these distributed sensors into Splunk ensures centralized analysis and correlation, helping detect attacks that may span multiple network segments.
 - This approach improves visibility, reduces blind spots, and strengthens overall network security posture.
- **Machine Learning Integration**
 - Leveraging Splunk's Machine Learning Toolkit (MLTK) allows for the detection of anomalous network behavior that may not be captured by traditional signature-based detection.
 - Examples include detecting unusual login patterns, traffic spikes, or slow-moving stealth attacks.
 - Integrating machine learning enhances predictive capabilities, enabling proactive detection and early warning of potential threats.
- **Advanced Correlation**
 - Integrating Splunk with Enterprise Security (ES) or open-source SIEM solutions like OSSIM enables advanced event correlation across multiple data sources.
 - This allows security teams to detect complex attack patterns, reduce false positives, and prioritize alerts effectively.
 - Correlation of firewall logs, IDS/IPS events, endpoint data, and application logs provides a holistic view of the network threat landscape.
- **Cloud Environment Monitoring**
 - Modern organizations increasingly use cloud platforms such as AWS, Azure, and GCP. Extending monitoring to these environments ensures complete visibility.
 - Splunk forwarders or cloud-native connectors can ingest logs from cloud services, virtual networks, and cloud-hosted applications.
 - This enables centralized monitoring, anomaly detection, and compliance auditing across both on-premises and cloud infrastructures.
- **Automation & SOAR (Security Orchestration, Automation, and Response)**
 - Implementing SOAR solutions in conjunction with Splunk allows for automated response to detected threats.
 - Example: Automatically blocking malicious IPs, isolating compromised hosts, or notifying security teams in real-time.
 - Automation reduces response time, mitigates the impact of attacks, and allows security teams to focus on strategic threat analysis rather than repetitive tasks.

Chapter-6

Conclusion

The successful implementation of the **Network Traffic Monitoring System using Snort and Splunk** demonstrates the potential of integrating an open-source Intrusion Detection System (IDS) with a powerful Security Information and Event Management (SIEM) platform to provide a robust and reliable network defense mechanism. In today's rapidly evolving cyber landscape, where organizations are exposed to increasingly complex threats, the ability to monitor, detect, and respond to malicious activity in real time has become a critical necessity. This project has addressed that need by combining Snort's capability to detect abnormal or suspicious network traffic with Splunk's advanced log management, indexing, and visualization functionalities.

Throughout the development and testing phases, the system has proven effective in capturing raw network packets, analyzing them against predefined Snort rules, and generating timely alerts whenever suspicious traffic patterns were detected. These alerts were successfully forwarded to Splunk using the Universal Forwarder, enabling seamless log integration. Splunk's indexing capability ensured that the logs were not only stored efficiently but also made available for fast searching and correlation. Furthermore, the interactive dashboards created in Splunk provided a clear, user-friendly interface for visualizing detected anomalies, monitoring overall traffic patterns, and assisting security analysts in identifying potential threats at a glance.

The testing phase further validated the system's reliability. ICMP and port scan traffic were correctly detected by Snort, while false positives were minimal during legitimate traffic flows. The smooth transition of alerts from Snort to Splunk reinforced the practicality of the integration. The dashboards, which formed the final visualization layer of the system, proved effective in turning complex log data into actionable insights for administrators and analysts. Overall, the project successfully met its objectives of real-time detection, alert generation, log forwarding, indexing, and visualization.

Beyond its immediate success, the project also highlights the broader importance of adopting hybrid security models that combine traditional IDS solutions with advanced SIEM platforms. While Snort provides robust signature-based detection, Splunk complements it with advanced analytics, centralized visibility, and customizable dashboards, creating a holistic monitoring environment. This synergy is particularly valuable in organizational settings where vast amounts of data need to be analyzed continuously to ensure security compliance and resilience against cyberattacks.

In conclusion, this project stands as a **practical, scalable, and future-ready solution for network traffic monitoring and intrusion detection**. It establishes a solid foundation for enterprises and academic institutions seeking to strengthen their cybersecurity posture. The project can be further enhanced through integration with machine learning-based anomaly detection, automated incident response systems, external threat intelligence feeds, and large-scale distributed monitoring architectures. By continuing to build upon the framework established in this project, organizations can develop a comprehensive and proactive security monitoring system capable of addressing the challenges of modern cyber threats.

Bibliography

Offline References

- [1] M. Roesch, *Snort – Lightweight Intrusion Detection for Networks*, Proc. 13th USENIX Conf. on System Administration (LISA), USENIX Association, 1999.
- [2] K. Scarfone and P. Mell, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, NIST Special Publication 800-94, National Institute of Standards and Technology, 2007.
- [3] K. Scarfone and P. Mell, *Guide to Computer Security Log Management*, NIST Special Publication 800-92, National Institute of Standards and Technology, 2009.
- [4] W. Stallings, *Network Security Essentials: Applications and Standards*, 6th ed. Boston, MA, USA: Pearson Education, 2020.
- [5] S. Northcutt and J. Novak, *Network Intrusion Detection*, 3rd ed. Indianapolis, IN, USA: New Riders Publishing, 2002.
- [6] R. Bejtlich, *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. San Francisco, CA, USA: No Starch Press, 2014.
- [7] A. Chuvakin, K. Schmidt, and C. Phillips, *Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management*. Waltham, MA, USA: Syngress Publishing, 2013.
- [8] V. Kumar and R. Singh, “A Review on Intrusion Detection System Using Snort and SIEM Tools,” *Int. J. Comput. Appl.*, vol. 183, no. 27, pp. 15–20, 2021.

Online References

- [9] Cisco Systems, Inc., “Snort Official Documentation,” 2023. [Online]. Available: <https://www.snort.org/>. [Accessed: Sep. 4, 2025].
- [10] Splunk Inc., “Splunk Enterprise Documentation,” 2023. [Online]. Available: <https://docs.splunk.com/>. [Accessed: Sep. 4, 2025].
- [11] Canonical Ltd., “Ubuntu Official Documentation,” 2023. [Online]. Available: <https://ubuntu.com/server/docs>. [Accessed: Sep. 4, 2025].
- [12] OWASP Foundation, “Intrusion Detection Systems,” 2023. [Online]. Available: <https://owasp.org/>. [Accessed: Sep. 4, 2025].
- [13] MITRE, “ATT&CK – Adversarial Tactics and Techniques Framework,” 2023. [Online]. Available: <https://attack.mitre.org/>. [Accessed: Sep. 4, 2025].
- [14] Tutorialspoint, “Intrusion Detection Systems (IDS) Basics,” 2022. [Online]. Available: https://www.tutorialspoint.com/intrusion_detection_system/. [Accessed: Sep. 4, 2025].
- [15] Medium, “Using Splunk for Security Monitoring,” 2022. [Online]. Available: <https://medium.com/>. [Accessed: Sep. 4, 2025].