

# CYBERSECURITY AND ETHICAL HACKING PROJECT

## TITLE: ENTERPRISE-LEVEL THREAT DETECTION & INCIDENT RESPONSE SYSTEM

-By KARTHIKEYA VALISETTY

Submission date – 02/04/2025

### Security Monitoring and Incident Response Report

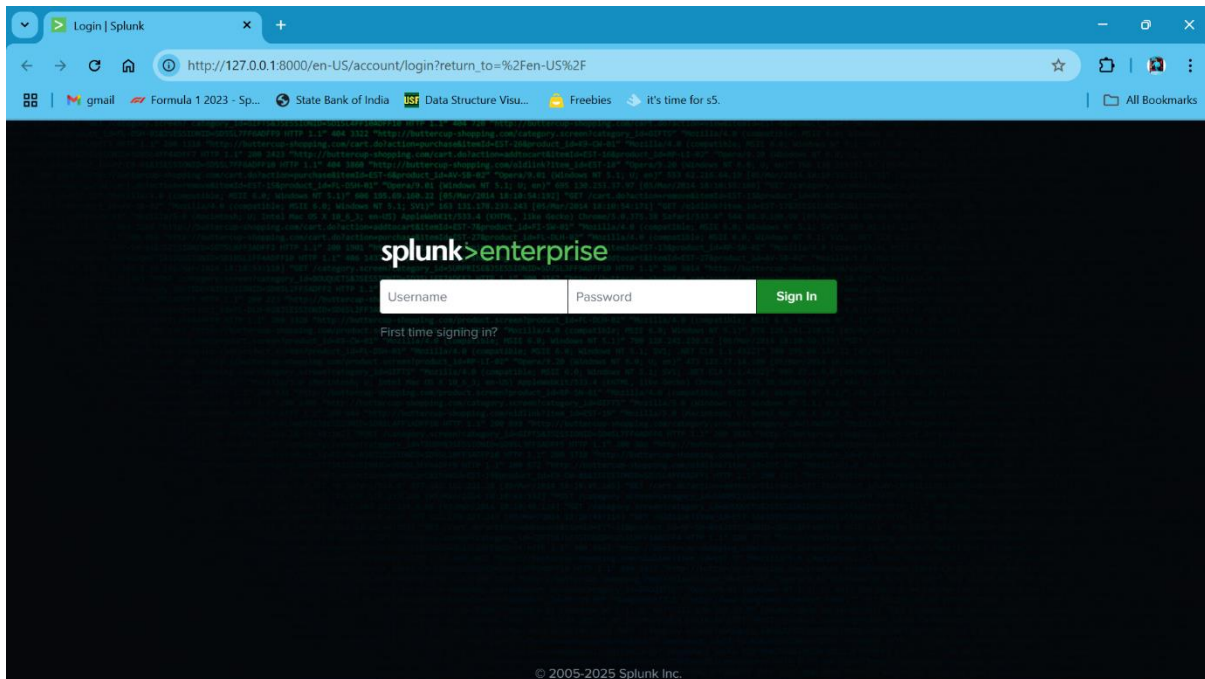
## 1. Introduction

This report provides a detailed analysis of the security monitoring architecture, log forwarding setup, automation scripts, intrusion detection system (IDS) configurations, and penetration testing activities. The goal is to document the setup, analyze logs, and provide security recommendations.

## 2. Splunk Setup Architecture

### 2.1 Overview

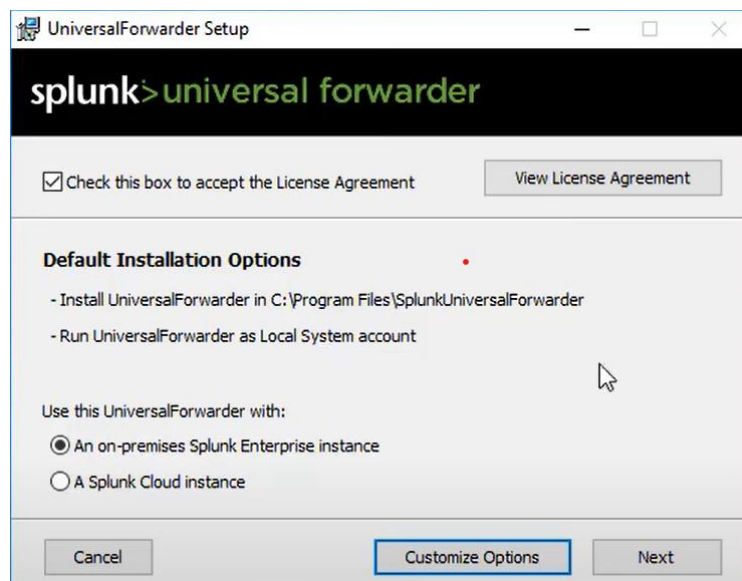
Splunk has been configured to collect and analyze logs from multiple devices using the Universal Forwarder. The logs are then indexed and monitored through Splunk's web interface.



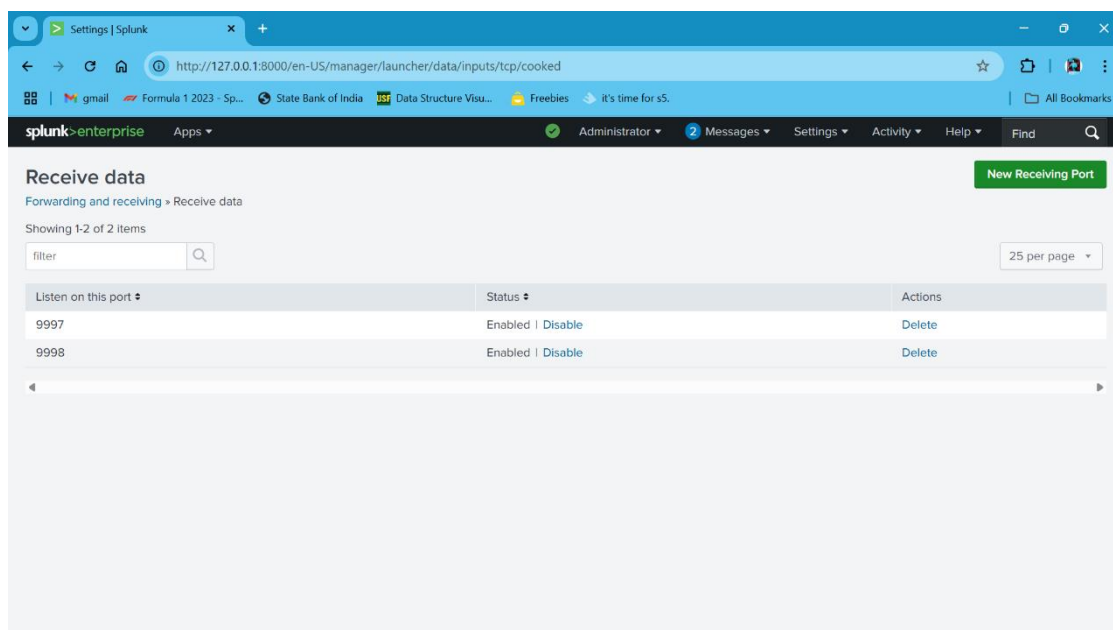
Continued....

## 2.2 Configuration Details

- **Universal Forwarder Setup:** Installed and configured on multiple devices.



- **Receiving Ports:** Configured on Splunk to accept logs from Universal Forwarders.



- **Indexing:** Separate indexes created for each system.

system1	Edit	Delete	Disable	Events	search	18 MB	2 GB	102K	3 months ago	42 minutes ago	D:\Splunk Data\splunk\system1\db	N/A
system2	Edit	Delete	Disable	Events	search	17 MB	2 GB	106K	2 months ago	3 days ago	D:\Splunk Data\splunk\system2\db	N/A

Continued....

- **Log Forwarding:** Successfully verified for each system.

**New Search**

index=system1

2,689 events (4/2/25 12:00:00 AM to 4/2/25 9:38:09 PM) No Event Sampling

Events (2,689) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect

1 hour per column

Hide Fields All Fields

SELECTED FIELDS

- host 1
- source 3
- sourcetype 3

INTERESTING FIELDS

- Account\_Domain 8
- Account\_Name 15
- ComputerName 1
- EventCode 100+

i	Time	Event
>	4/2/25 8:52:34.000 PM	04/02/2025 08:52:34 PM LogName=System EventCode=2 EventType=4 ComputerName=Karthikeya <a href="#">Show all 15 lines</a> host = KARTHIKEYA source = WinEventLog:System sourcetype = win_system
>	4/2/25 8:52:32.000 PM	04/02/2025 08:52:32 PM LogName=System

**New Search**

index=system2

105,525 events (before 4/2/25 9:39:03 PM) No Event Sampling

Events (105,525) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect

1 day per column

Hide Fields All Fields

SELECTED FIELDS

- host 1
- source 3
- sourcetype 3

INTERESTING FIELDS

- ComputerName 1
- EventCode 100+
- EventType 5
- index 1
- Keywords 12

i	Time	Event
>	3/30/25 10:41:15.000 PM	03/30/2025 10:41:15 PM LogName=Security EventCode=4798 EventType=0 ComputerName=LAPTOP-117H8462 <a href="#">Show all 27 lines</a> host = LAPTOP-117H8462 source = WinEventLog:Security sourcetype = WinEventLog:Security
>	3/30/25 10:41:15.000 PM	03/30/2025 10:41:15 PM LogName=Security EventCode=4798

Continued....

- **Alerts and Reports:** Configured periodic alerts and automated reports.

The screenshot shows the 'Searches, Reports, and Alerts' page in Splunk Enterprise. The page title is 'Searches, Reports, and Alerts' with buttons for 'New Report' and 'New Alert'. Below the title, there's a summary: '3 Searches, Reports, and Alerts'. A filter bar shows 'Type: All', 'App: Search & Reporting (search)', 'Owner: Administrator (admin)', and a search filter. A table lists the configured items:

Name	Actions	Type	Next Scheduled Time	Display View	Owner	App	Alerts	Sharing	Status
Multiple Event Occurrences	Edit Run View Recent	Report	2025-04-02 22:00:00 India Standard Time	none	admin	search	0	Private	Enabled
System Event Occurrences	Edit Run View Recent	Alert	2025-04-02 22:00:00 India Standard Time	none	admin	search	0	Private	Enabled
Windows Events Summary	Edit Run View Recent	Report	2025-04-02 22:00:00 India Standard Time	none	admin	search	0	Private	Enabled

- **Search Queries:** Queries for monitoring logs and triggering alerts.

The screenshot shows the 'Windows Events Summary' search results page. The search bar contains the query: `index=system source="WinEventLog:*" | stats dc(EventCode) AS UniqueEventCodes, values(SourceName) AS SourceNames, values(Message) AS Messages BY EventCode | table EventCode SourceNames Messages`. The results show 2,739 events from 4/1/25 9:30:00.000 PM to 4/2/25 9:42:49.000 PM. The 'Statistics (136)' tab is selected, showing a table with columns: EventCode, SourceNames, and Messages. The first row shows EventCode 0 and SourceNames including EMP\_UDSA, IntelDalJhi, K7EmIPxy, K7RTScan, K7TSMngr, LMS, MiService, and OSDService. The Messages column contains the text: 'Intel(R) Dynamic Application Loader Host Interface Service started. Service started/resumed. Service stopped. The operation completed successfully.'

Continued....

Multiple Event Occurrences

index=system1 sourcetype="WinEventLog:Security" | stats dc(EventCode) AS EventCodes, values(SourceName) AS SourceNames, values(Message) AS Messages BY EventCode | table EventCode SourceNames Messages

✓ 2,040 events (4/1/25 12:00:00.000 AM to 4/2/25 12:00:00.000 AM) No Event Sampling

Events (2,040) Patterns **Statistics (24)** Visualization

20 Per Page Format Preview

EventCode	SourceNames	Messages
1100	Microsoft-Windows-Eventlog	The event logging service has shut down.
4608	Microsoft Windows security auditing.	Windows is starting up.
4624	Microsoft Windows security auditing.	An account was successfully logged on.
4634	Microsoft Windows security auditing.	An account was logged off.
4647	Microsoft Windows security auditing.	User initiated logoff:
4648	Microsoft Windows security	A logon was attempted using explicit credentials.

System Event Occurrences

index=system1 sourcetype="WinEventLog:Security"  
| bin \_time span=1d  
| stats count by \_time, EventCode, SourceName, Message  
| where count > 2  
| table EventCode, SourceName, Message

✓ 2,040 events (4/1/25 12:00:00.000 AM to 4/2/25 12:00:00.000 AM) No Event Sampling

Events (2,040) Patterns **Statistics (14)** Visualization

20 Per Page Format Preview

EventCode	SourceName	Message
4624	Microsoft Windows security auditing.	An account was successfully logged on.
4634	Microsoft Windows security auditing.	An account was logged off.
4648	Microsoft Windows security auditing.	A logon was attempted using explicit credentials.

Continued....

## 3. Python Log Analysis Automation

### 3.1 Overview

A Python script was developed for log analysis automation, executing predefined search queries and sending summarized reports via email.

### 3.2 Configuration Details

- **Script Code:** Screenshot taken of the Python script.

```
import smtplib
import pandas as pd
import json
import sys
import platform

# Ensure we are using the correct Python version
expected_version = "3.13.2"
if platform.python_version() != expected_version:
    print(f"Warning: Running on Python {platform.python_version()}, expected {expected_version}")

# Splunk alert script gets data from STDIN
alert_data = sys.stdin.read()

# Parse the incoming alert data from Splunk
events = [json.loads(line) for line in alert_data.split("\n") if line]

# Convert data to Pandas DataFrame
df = pd.DataFrame(events)

# Generate Summary
summary = {
    "Total Events": len(df),
    "Most Common Event Codes": df["EventCode"].value_counts().head(5).to_dict(),
    "Most Frequent Sources": df["SourceName"].value_counts().head(5).to_dict(),
}

# Include messages corresponding to the top 5 most common event codes
event_messages = {}

if "EventCode" in df.columns and "Message" in df.columns:
    top_events = df["EventCode"].value_counts().head(5).index # Get top 5 EventCodes

    for event_code in top_events:
        messages = df[df["EventCode"] == event_code]["Message"].dropna().unique()[:3] # Get up to 3 unique messages
        event_messages[str(event_code)] = list(messages)

summary["Event Messages"] = event_messages
```

Continued....

```

# Convert Summary to JSON
summary_text = json.dumps(summary, indent=4)

# Email Configuration
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587
EMAIL_SENDER = ""
EMAIL_PASSWORD = ""
EMAIL_RECEIVER = ""
SUBJECT = "Splunk Windows Logs Summary Alert"

# Email Body
email_body = f"Subject: {SUBJECT}\n\n{summary_text}"

# Send Email
try:
    server = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
    server.starttls()
    server.login(EMAIL_SENDER, EMAIL_PASSWORD)
    server.sendmail(EMAIL_SENDER, EMAIL_RECEIVER, email_body)
    server.quit()
    print("Summary email sent successfully.")
except Exception as e:
    print(f"Failed to send email: {e}")

```

- **Execution Commands:** Commands required to run the script documented.

```

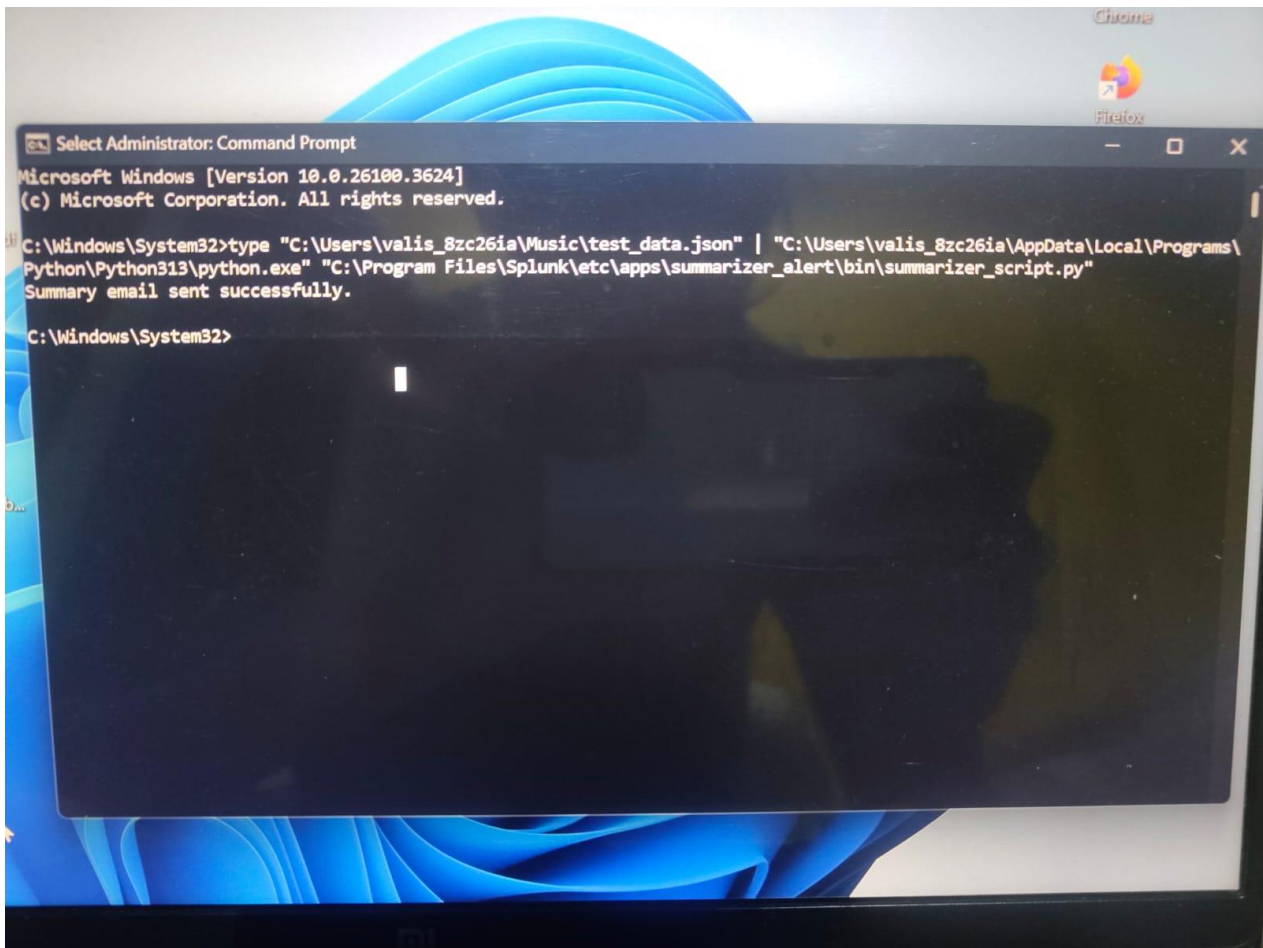
"C:\Users\valis_8zc26ia\Music\test_data.json" |
"C:\Users\valis_8zc26ia\AppData\Local\Programs\Python\Python313\python.exe" "C:\Program
Files\Splunk\etc\apps\summarizer_alert\bin\summarizer_script.py"

```

Continued....



- **Output Verification:** Screenshot of script execution results.



- **Email Notifications:** Verified successful email alerts upon execution.





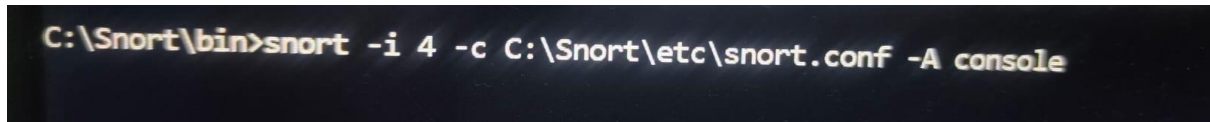
## 4. Snort Intrusion Detection System Setup

### 4.1 Overview

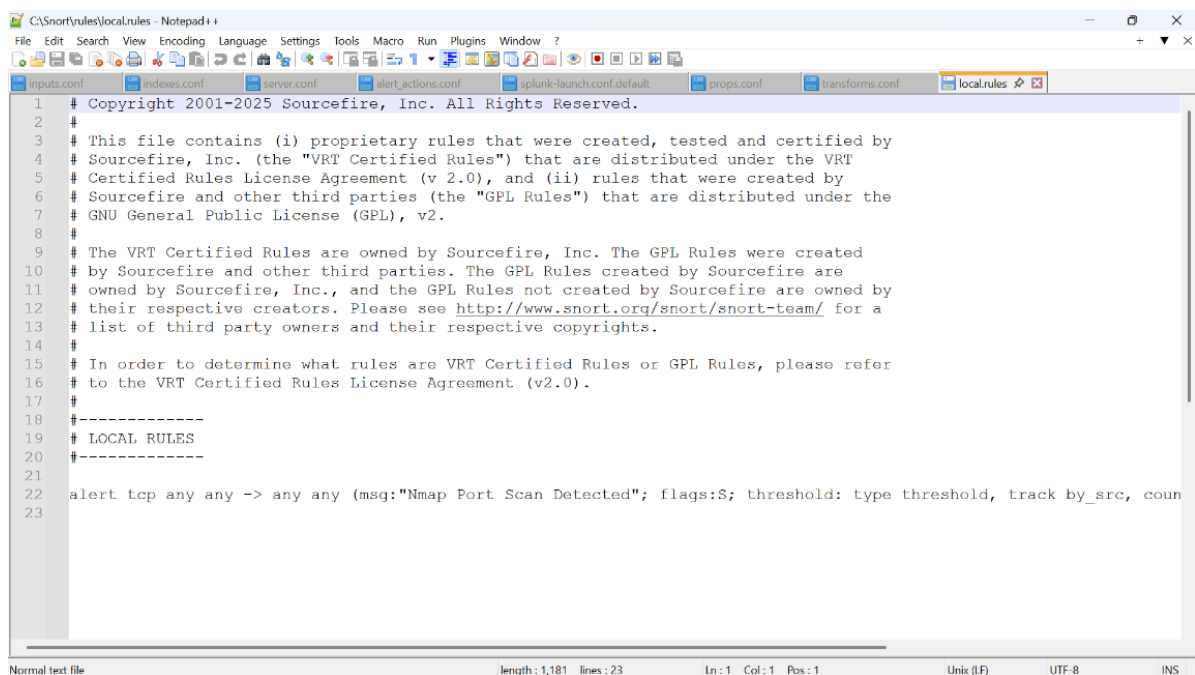
Snort has been installed and configured to detect network intrusions and generate logs for security analysis.

### 4.2 Configuration Details

- **Snort Configuration:** Configured snort.conf for IDS mode.



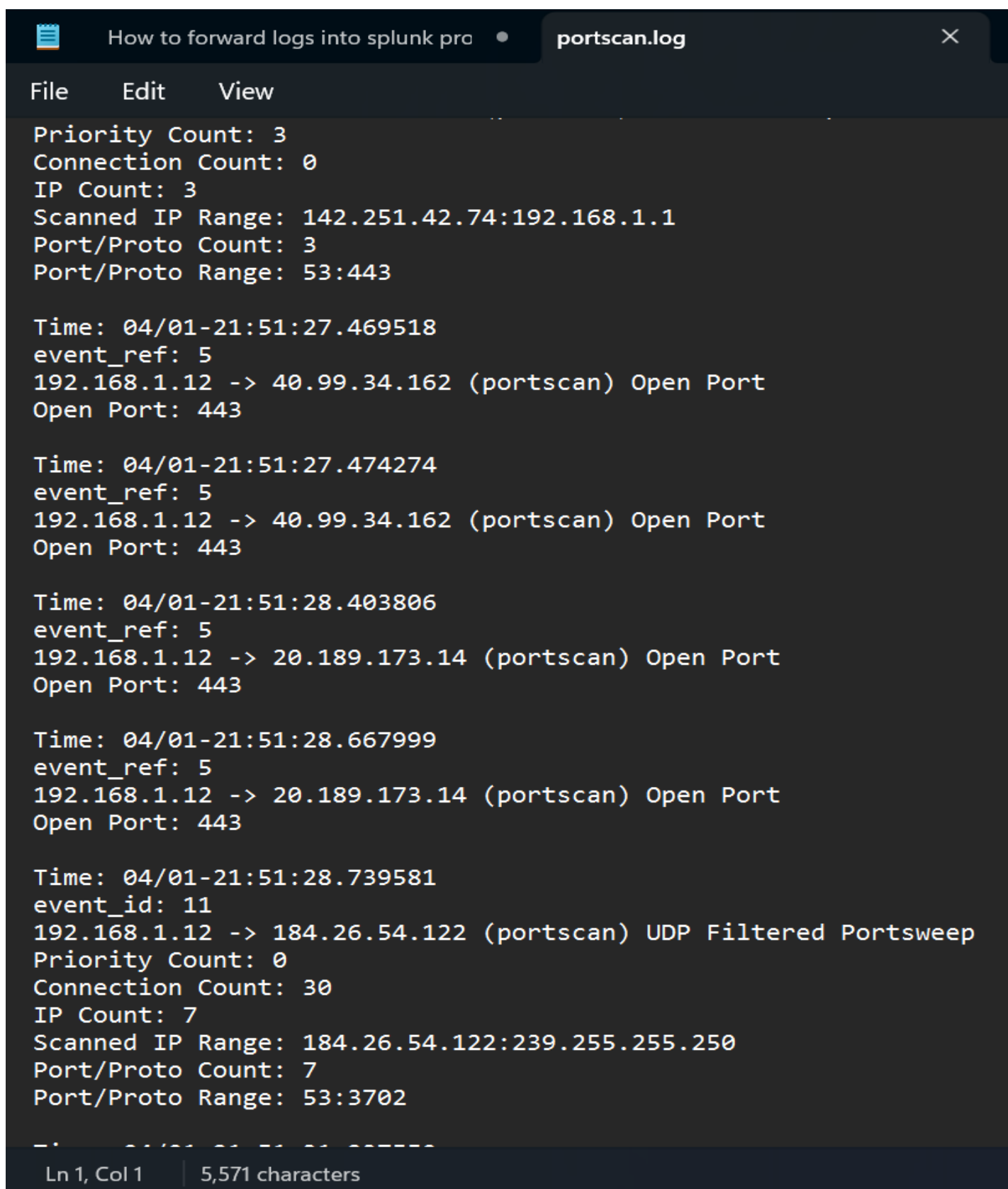
- **Custom Rules:** Added local rules in local.rules file to detect suspicious activities.



- **Version Check:** Verified Snort version using terminal commands.



- **Running Snort in IDS Mode:** Documented commands and execution results.
- **Log Analysis:** Verified logs generated in portscan.log, including detection of an Nmap scan.



The screenshot shows a code editor window with a dark theme. The title bar at the top contains a file icon, the text "How to forward logs into splunk pro", a separator, and the filename "portscan.log" with a close button (X) on the right. Below the title bar is a menu bar with "File", "Edit", and "View". The main area displays the contents of "portscan.log" in a monospaced font. The log contains several entries, including summary statistics and detailed scan results for different IP addresses and ports. At the bottom of the editor, a status bar shows "Ln 1, Col 1" and "5,571 characters".

```
Priority Count: 3
Connection Count: 0
IP Count: 3
Scanned IP Range: 142.251.42.74:192.168.1.1
Port/Proto Count: 3
Port/Proto Range: 53:443

Time: 04/01-21:51:27.469518
event_ref: 5
192.168.1.12 -> 40.99.34.162 (portscan) Open Port
Open Port: 443

Time: 04/01-21:51:27.474274
event_ref: 5
192.168.1.12 -> 40.99.34.162 (portscan) Open Port
Open Port: 443

Time: 04/01-21:51:28.403806
event_ref: 5
192.168.1.12 -> 20.189.173.14 (portscan) Open Port
Open Port: 443

Time: 04/01-21:51:28.667999
event_ref: 5
192.168.1.12 -> 20.189.173.14 (portscan) Open Port
Open Port: 443

Time: 04/01-21:51:28.739581
event_id: 11
192.168.1.12 -> 184.26.54.122 (portscan) UDP Filtered Portsweep
Priority Count: 0
Connection Count: 30
IP Count: 7
Scanned IP Range: 184.26.54.122:239.255.255.250
Port/Proto Count: 7
Port/Proto Range: 53:3702
```

Ln 1, Col 1 | 5,571 characters

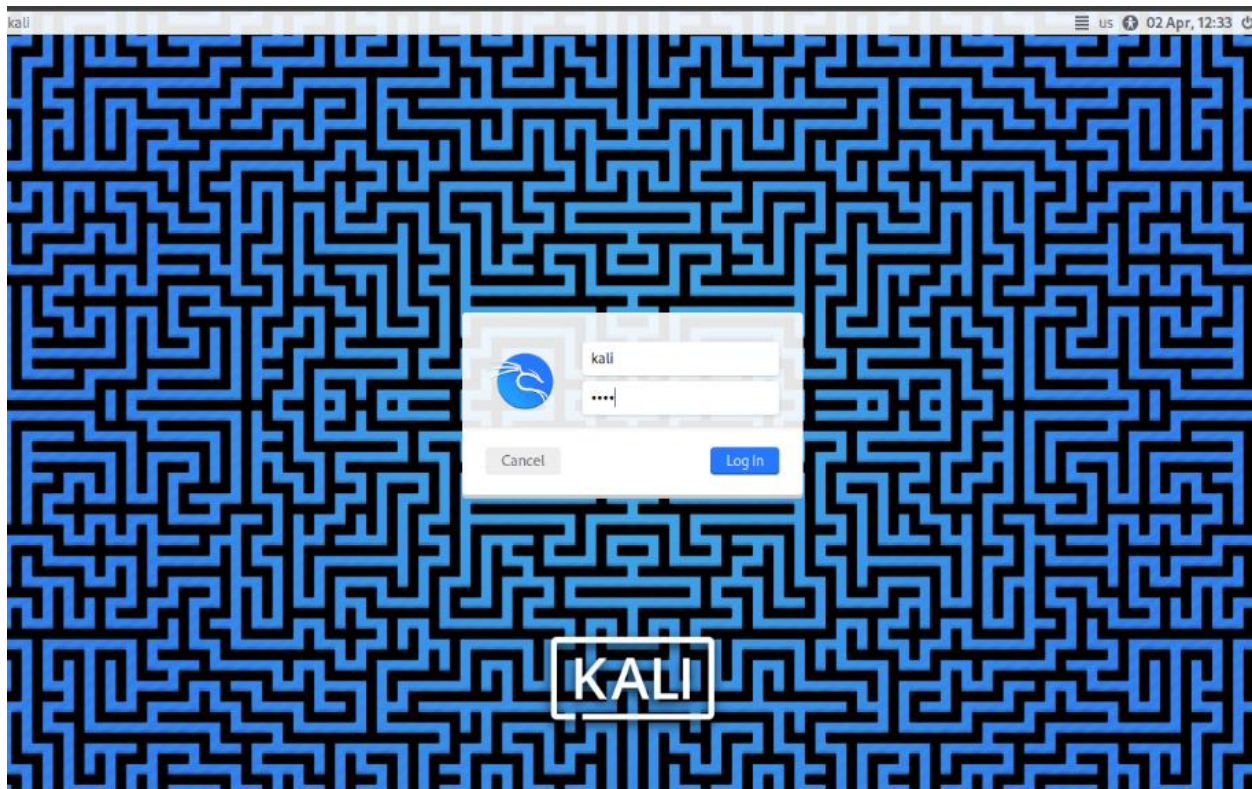
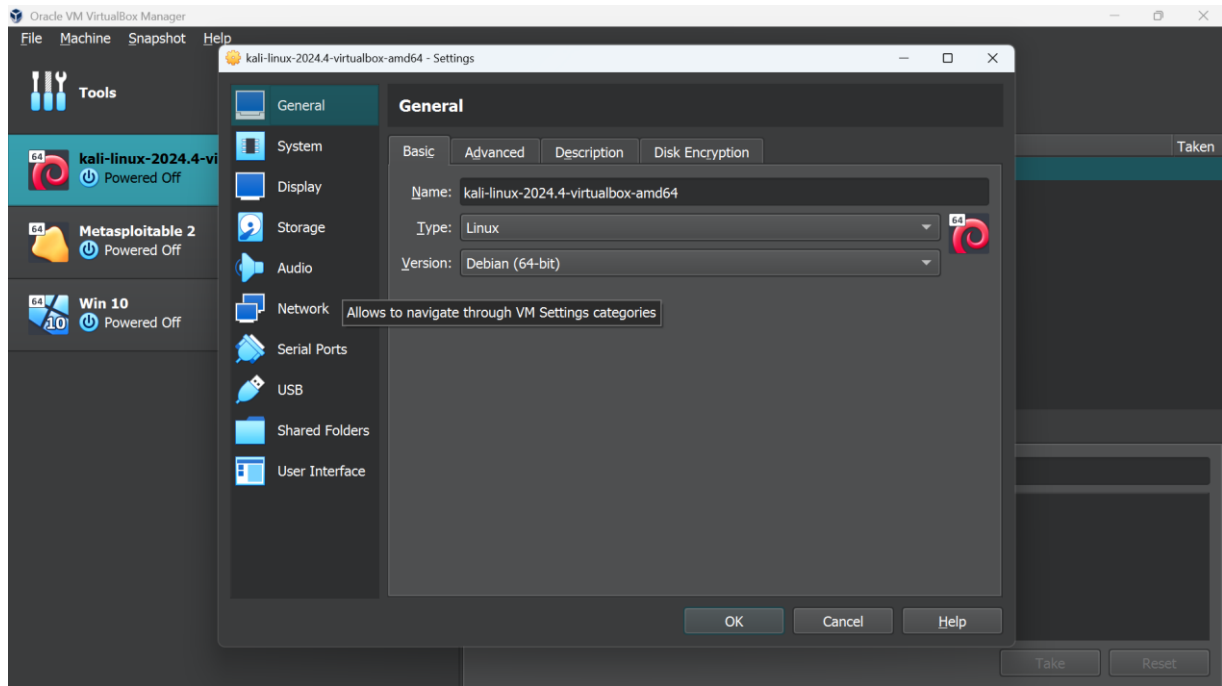
## 5. Kali Linux Setup and Penetration Testing

### 5.1 Overview

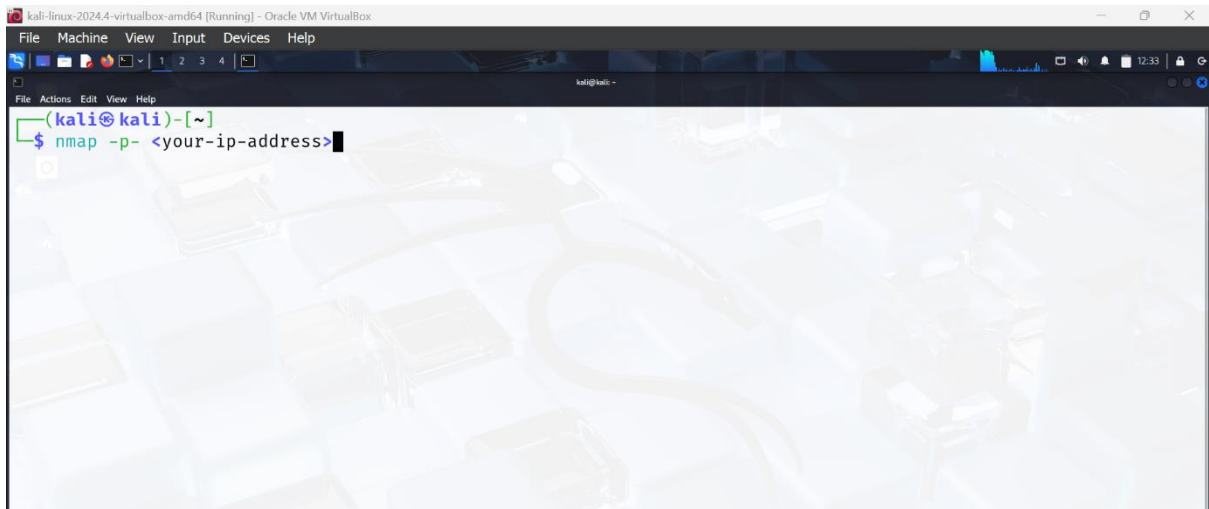
Kali Linux was used for penetration testing to assess network security.

### 5.2 Configuration Details

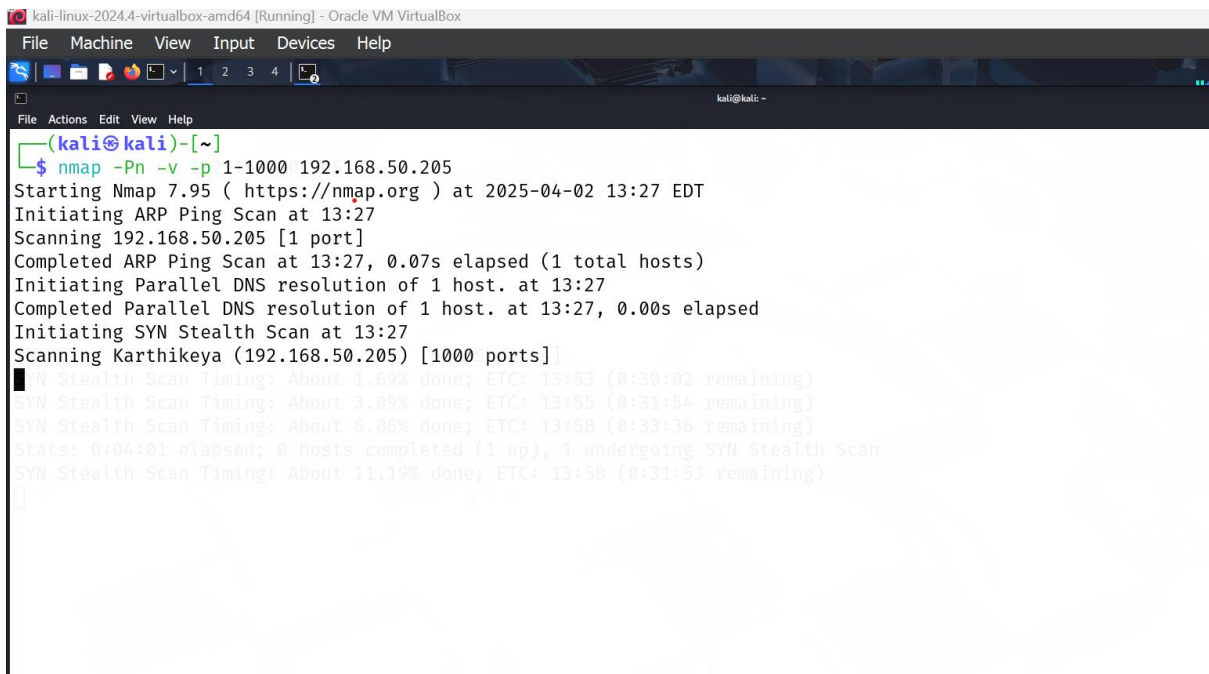
- **Kali Linux Installation:** Successfully set up and configured.



- **Nmap Scan Execution:** Conducted an Nmap scan on a target system.



- **Command Documentation:** Screenshots of commands used.



- **Scan Results:** Analyzed scan output to identify open ports and vulnerabilities.

## 6. Incident Response and Findings

### 6.1 Security Event Detection

- Splunk alerts successfully triggered based on predefined queries.
- Snort detected unauthorized network scanning activity (Nmap scan).
- Python automation script provided timely log summaries via email.

### 6.2 Security Recommendations

- **Splunk Optimization:** Implement advanced correlation rules and dashboards.
- **Snort Enhancement:** Expand rule set to detect a wider range of attacks.
- **Network Hardening:** Limit open ports and monitor suspicious activities.
- **Incident Response Plan:** Establish a standardized incident response framework.

## 7. Conclusion

This report documents the security setup and its effectiveness in detecting security events. Based on the findings, recommendations have been provided to improve overall cybersecurity resilience. Continuous monitoring, log analysis, and proactive security measures are essential for maintaining a robust security infrastructure.