SOC Lab Simulation Report

© 2025 Visvakaanth

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

This project is for educational and demonstration purposes only.

Unauthorized commercial use, redistribution, or misrepresentation of this work is prohibited.

1. Lab Environment Setup

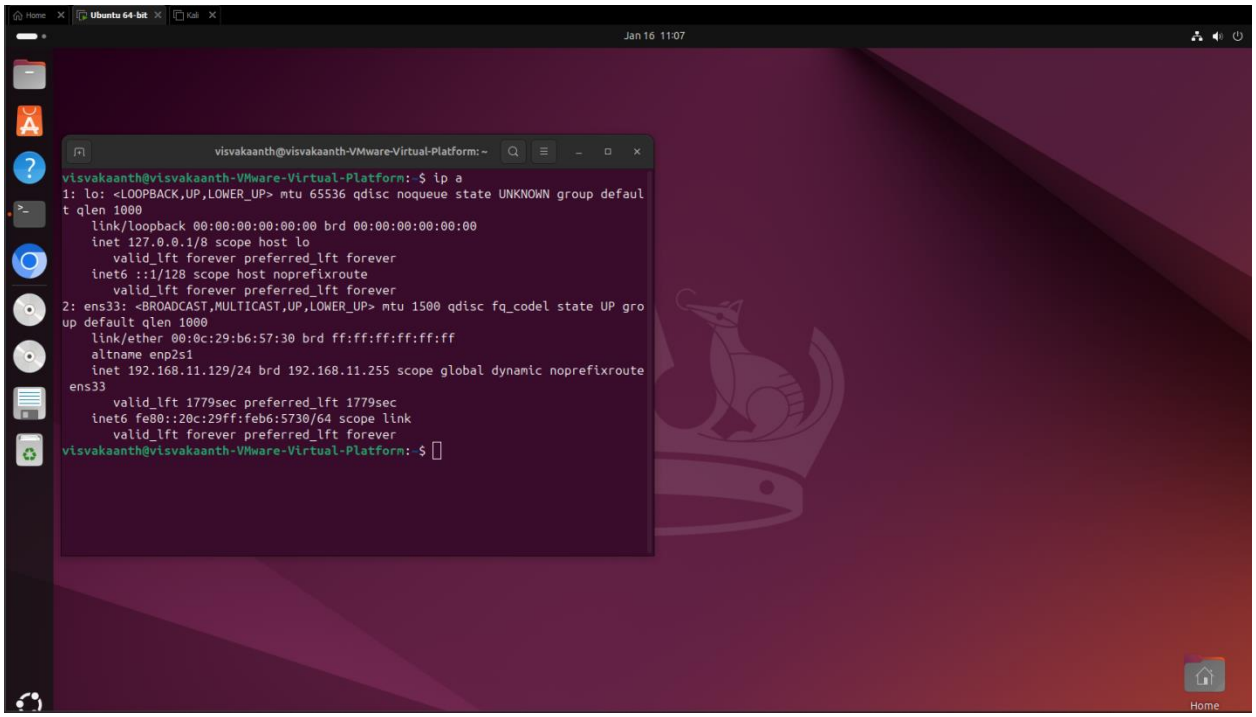
As part of this SOC lab simulation, an Ubuntu virtual machine was deployed using VMware Workstation to represent a typical employee workstation found in an enterprise environment. This machine acts as the target system for the simulated attacker activity.

The Ubuntu VM was configured with the following:

- Operating System: Ubuntu Linux
- Virtualization Platform: VMware Workstation
- Network Mode: (NAT)
- Assigned IP Address: 192.168.11.129

The purpose of this machine is to generate realistic system and authentication logs that can be forwarded to Splunk for monitoring and analysis. These logs will be used to detect and investigate malicious activity performed by the attacker during the simulation.

A screenshot of the terminal output showing the assigned IP address is provided below for reference



1.1 Firewall Configuration on Ubuntu Endpoint

The Ubuntu endpoint was secured using UFW (Uncomplicated Firewall) to simulate a baseline security posture commonly implemented on enterprise workstations.

Inbound connections were restricted by default, and only essential services were permitted. Specifically, all inbound traffic was denied except for SSH access, which was required for administrative purposes.

2. Splunk Deployment & Log Receiver Configuration

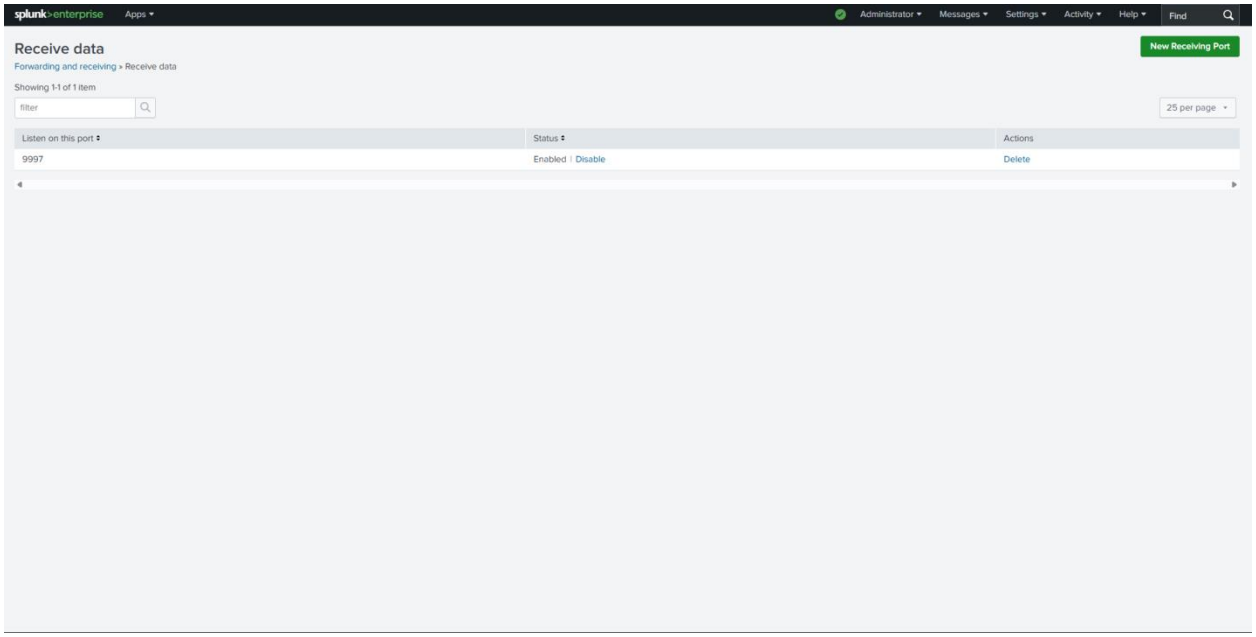
Splunk Enterprise was deployed on the Windows host system to act as the central log collection and analysis platform for this SOC lab environment.

To enable log ingestion from the Ubuntu endpoint, Splunk was configured to listen for incoming data on TCP port 9997, which is the default port used by Splunk Universal Forwarders for log forwarding.

The following steps were performed:

- Splunk Enterprise was installed and verified to be running successfully.
- A receiving port (9997) was enabled.
- The listener was activated to allow inbound log data from the Ubuntu VM.

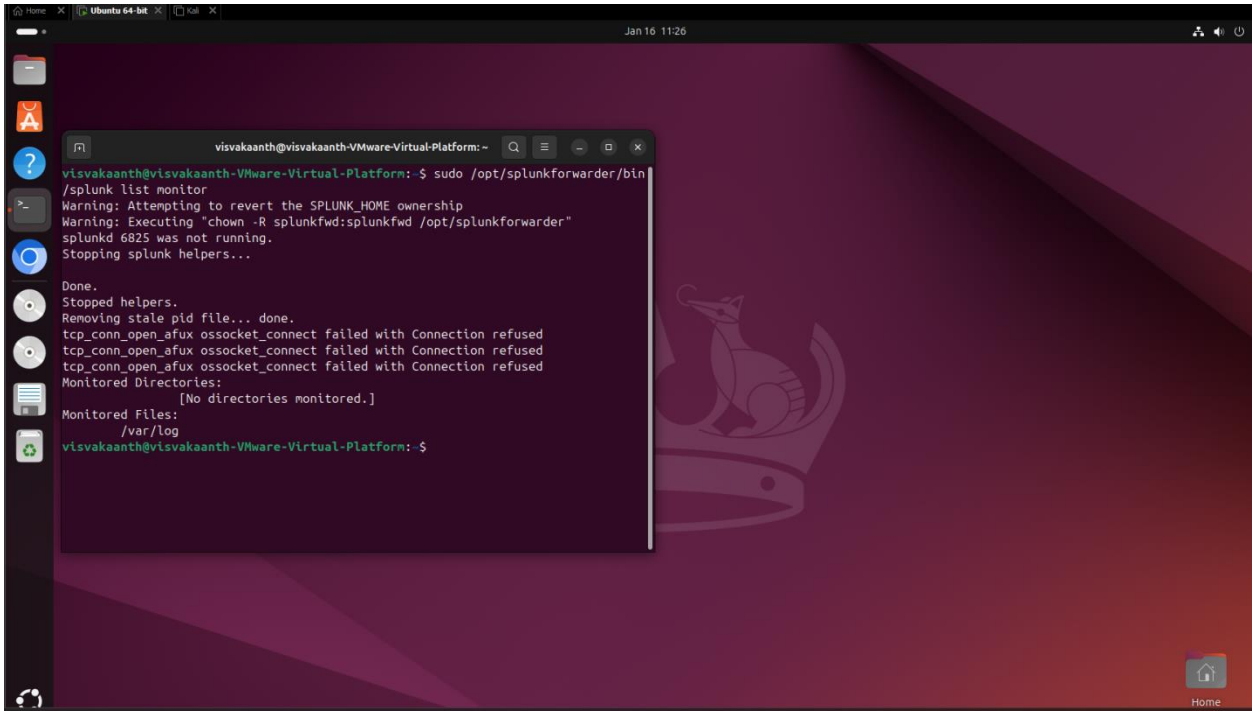
A screenshot showing Splunk listening on port 9997 is provided below for reference.



This configuration allows the Ubuntu system to forward system and authentication logs to Splunk in real time, enabling centralized monitoring and detection of suspicious activity.

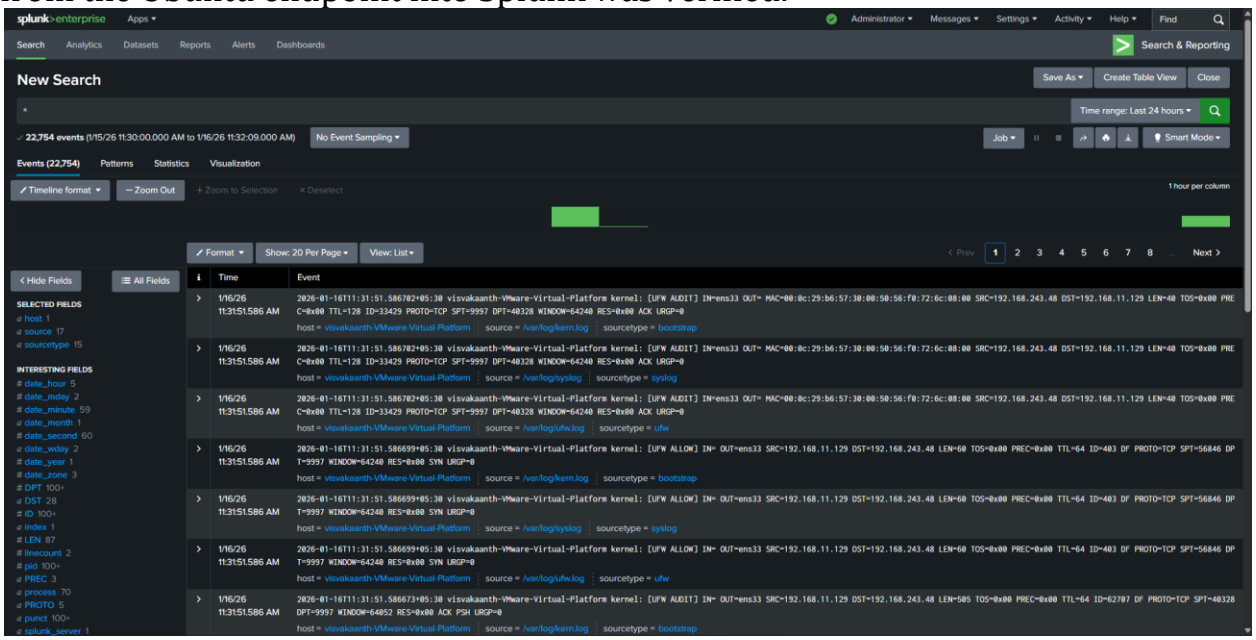
3. Ubuntu Log Forwarder Configuration

The Splunk Universal Forwarder on the Ubuntu endpoint was configured to monitor the "/var/log" directory, including authentication and system logs. These logs were forwarded in real-time to the Splunk Enterprise instance over TCP port 9997.



To validate that log forwarding was functioning correctly, real-time log ingestion

from the Ubuntu endpoint into Splunk was verified.



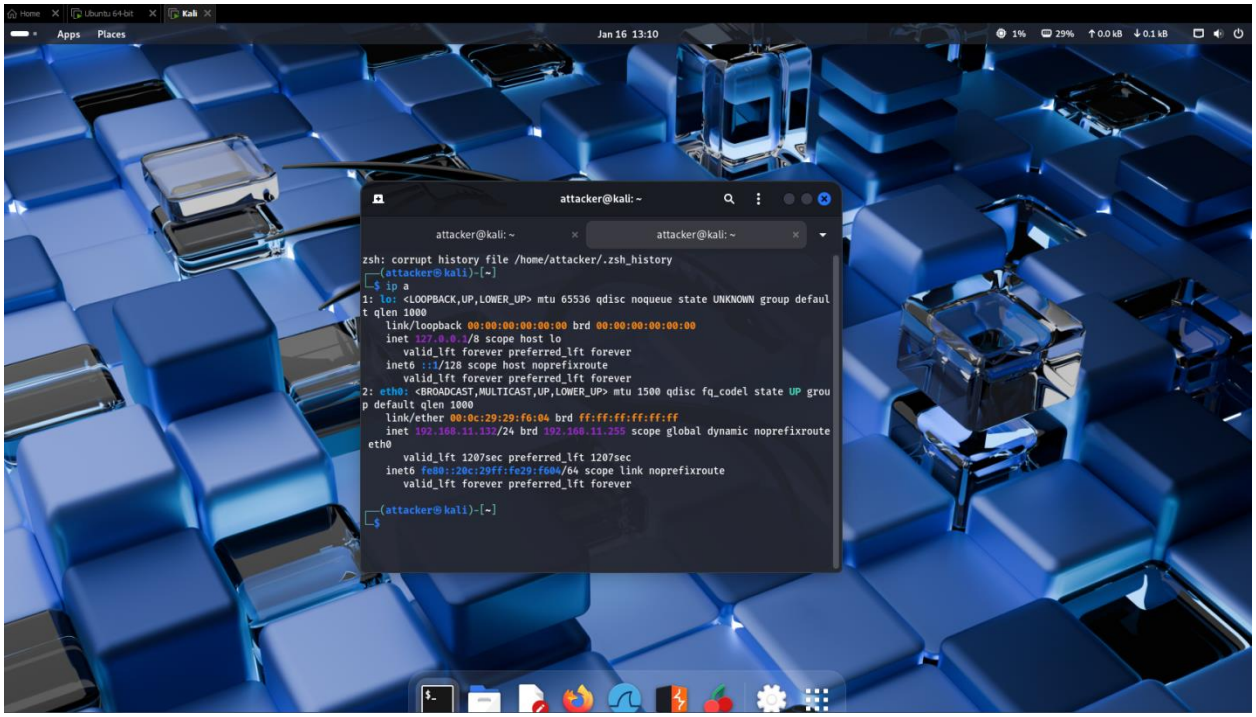
4. Attack Simulation

To simulate real-world attacker behavior, a Kali Linux virtual machine was deployed to act as the attacker system within the lab environment. Kali Linux is a penetration testing distribution commonly used by threat actors and security professionals to perform reconnaissance and exploitation activities.

The objective of this phase was to perform reconnaissance on the Ubuntu endpoint in order to identify open ports and running services that could potentially be targeted for further exploitation.

The Kali VM was configured with the following:

- Operating System: Kali Linux
- Virtualization Platform: VMware Workstation
- Network Mode: (NAT)
- Assigned IP Address: 192.168.11.132



5. Custom Port Scanning Attack from Kali Linux

MITRE ATT&CK:

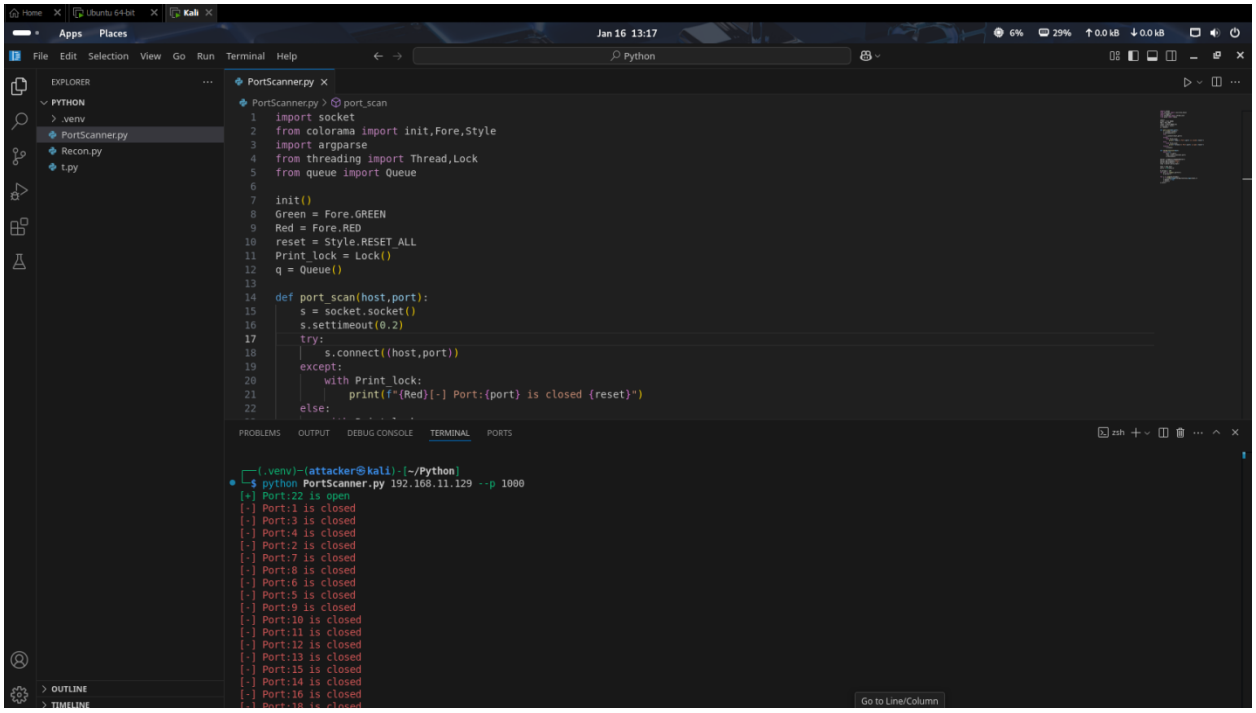
- T1046 – Network Service Discovery

To simulate attacker reconnaissance activity, a custom Python-based port scanning script was executed from the Kali Linux attacker machine against the Ubuntu target endpoint. Instead of using standard tools such as Nmap, a custom script utilizing Python's socket library was developed to mimic how threat actors may use their own tooling to evade basic detection mechanisms.

The objective of this scan was to identify open ports running on the Ubuntu system that could be targeted for further exploitation.

The script attempted TCP connections against a range of ports on the Ubuntu host to determine which services were accessible.

A screenshot of the Kali terminal executing the custom port scanning script is provided below



During the reconnaissance phase, the custom Python port scanning script did not identify any open ports on the Ubuntu endpoint, with the exception of the SSH service.

This outcome was expected due to the restrictive UFW firewall configuration in place.

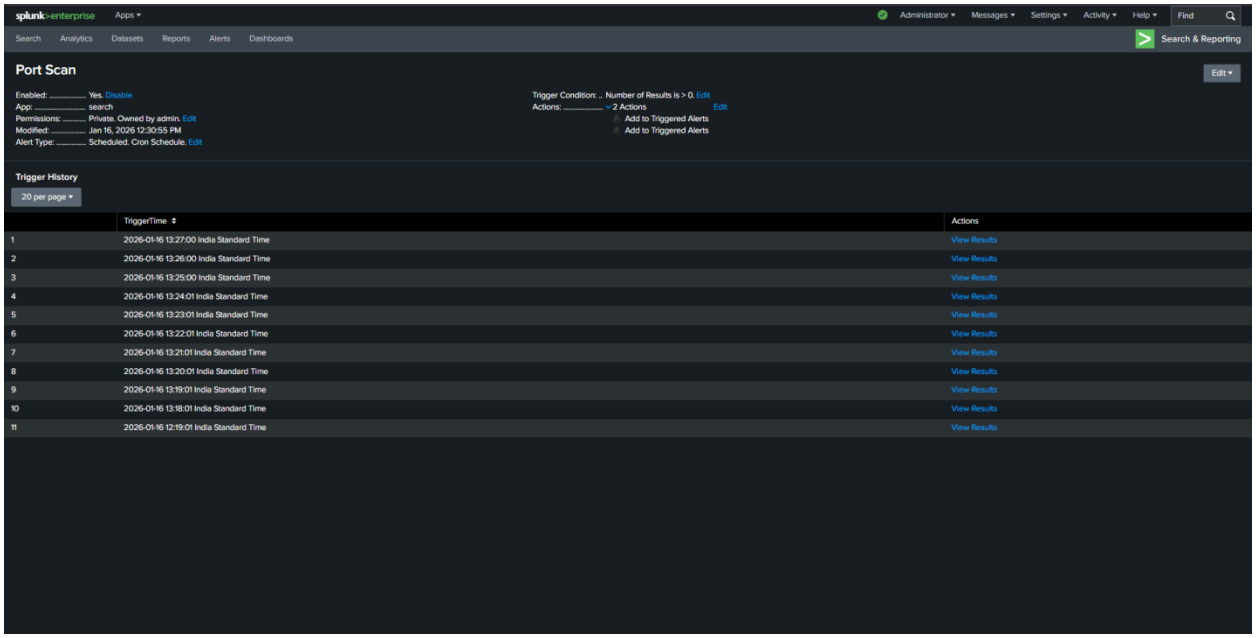
6. Detection & Alerting – Port Scanning Activity

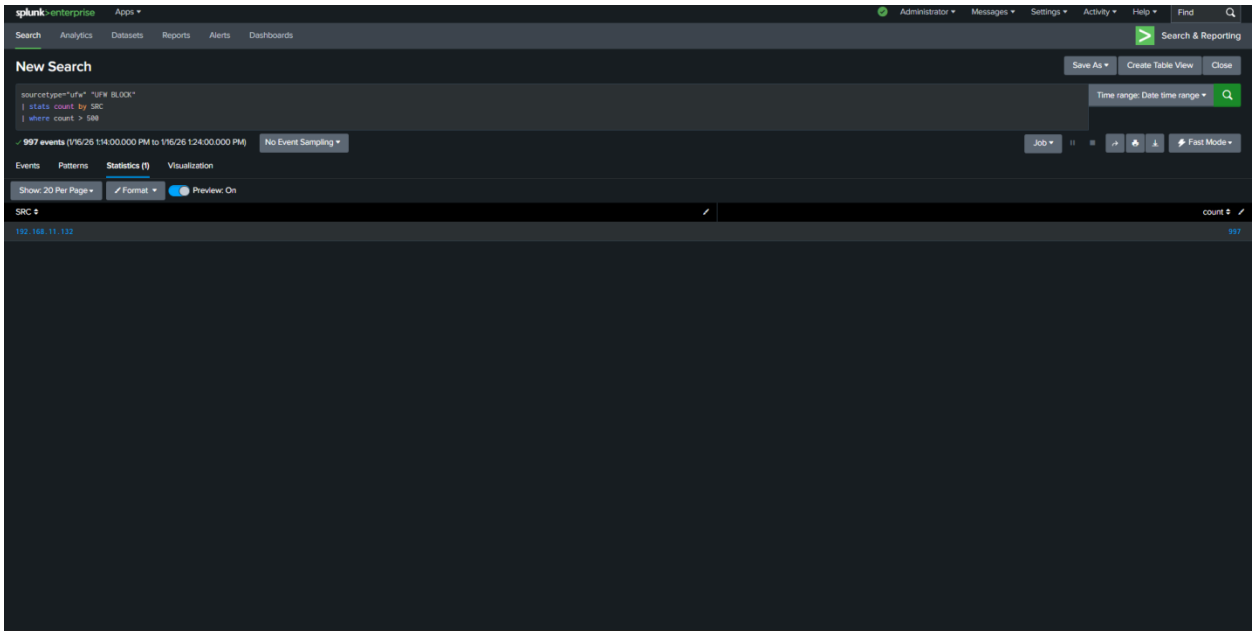
To detect reconnaissance activity targeting the Ubuntu endpoint, Splunk Enterprise was configured with a custom alert designed to identify port scanning behavior.

The alert is based on a search query that monitors repeated connection attempts from a single source IP address to multiple ports within a defined time window. This pattern is commonly associated with automated scanning activity performed during the reconnaissance phase of an attack.

During the execution of the custom Python port scanning script from the Kali Linux attacker machine, the alert was successfully triggered. This confirms that the detection logic is effective in identifying suspicious scanning behavior in real time.

A screenshot of the triggered Splunk alert is provided below for reference.





7. Attack Simulation – SSH Brute Force Attack

MITRE ATT&CK:

- **T1110 – Brute Force**
- **T1021.004 – Remote Services: SSH**

Following the reconnaissance phase, the attacker targeted the exposed SSH service on the Ubuntu endpoint in an attempt to gain unauthorized access. Since SSH was the only service permitted through the firewall, it became the primary attack vector.

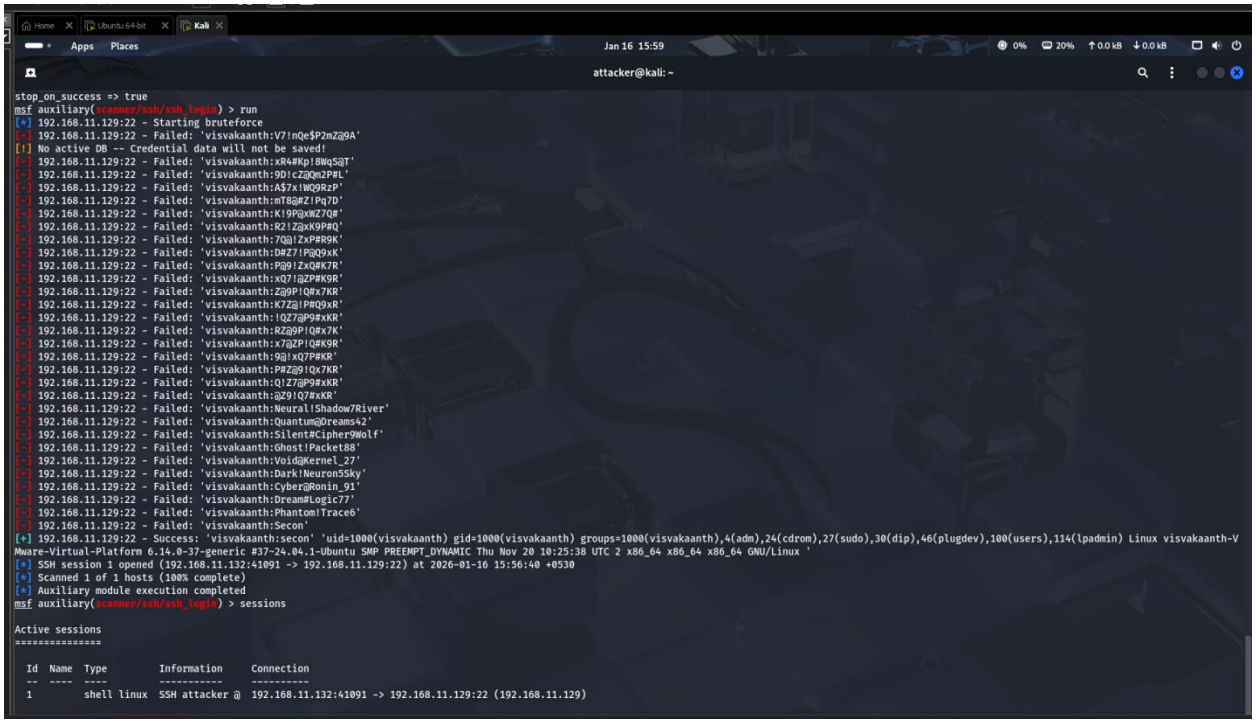
A credential-based brute force attack was executed from the Kali Linux attacker machine against the Ubuntu system using repeated authentication attempts with different username and password combinations. This technique is commonly used by attackers to gain initial access when weak or reused credentials are present.

After multiple failed login attempts, the attacker successfully authenticated to the system, resulting in unauthorized access to the Ubuntu endpoint and the establishment of an interactive SSH session.

The sequence of events observed was:

1. Multiple failed SSH authentication attempts
2. Successful SSH login from the attacker system
3. Establishment of an interactive session on the target host

A screenshot showing the brute force activity and successful login is provided below.



8. Detection & Analysis – SSH Brute Force Activity

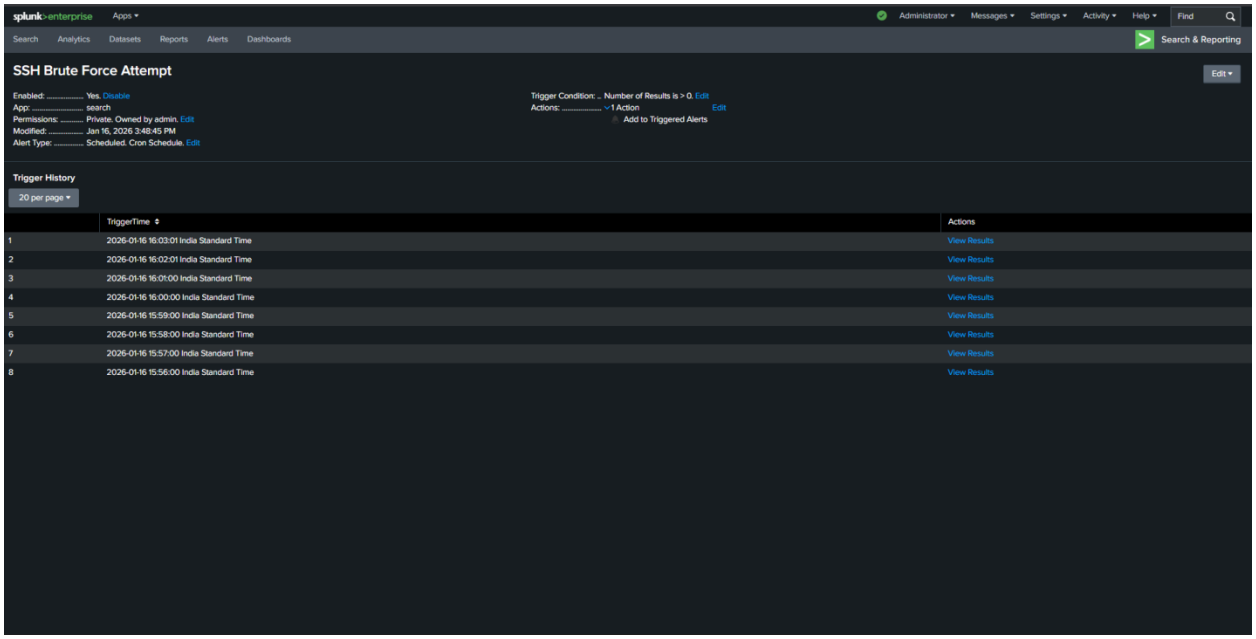
To detect credential-based attacks targeting the Ubuntu endpoint, a custom alert was configured in Splunk to identify excessive failed SSH authentication attempts from a single source IP address within a defined time window.

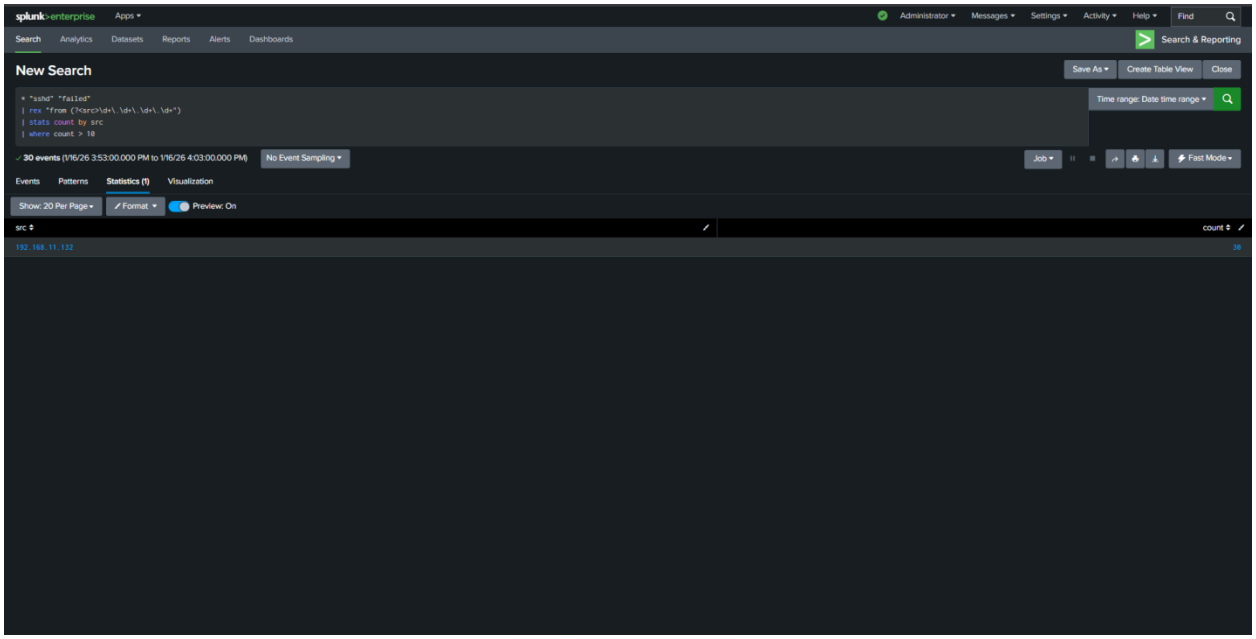
This detection logic is designed to identify brute force behavior, which is commonly characterized by repeated login failures followed by a potential successful authentication.

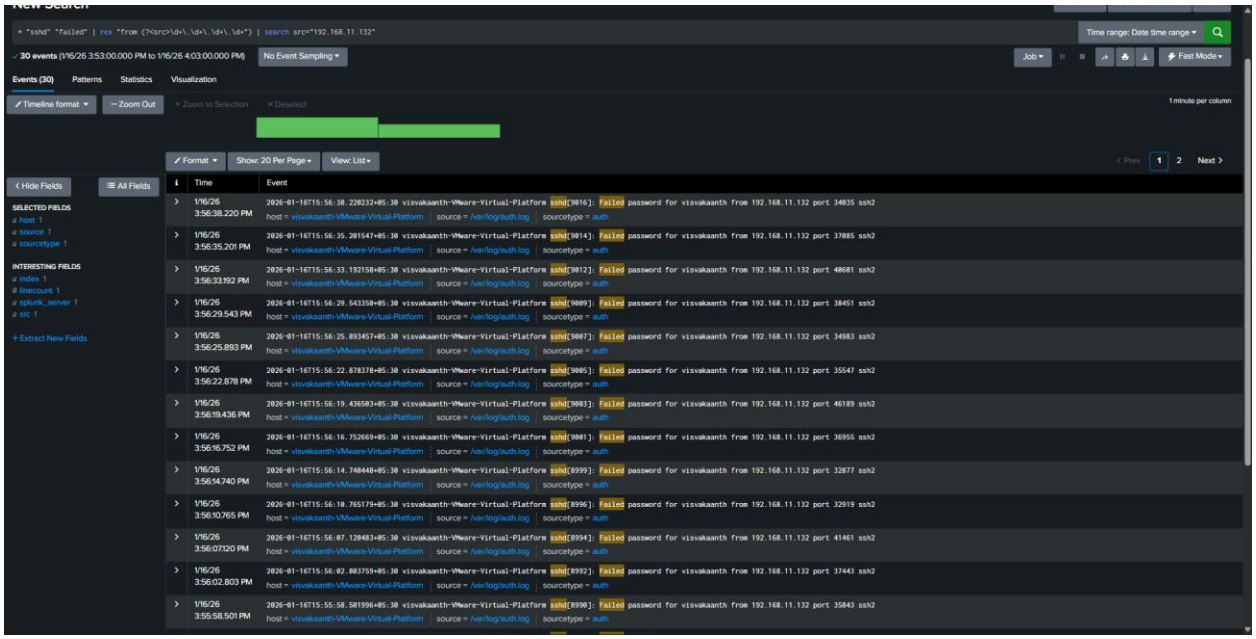
During the execution of the SSH brute force attack from the Kali Linux attacker machine, the alert was successfully triggered. The alert captured multiple failed login attempts originating from the same source, followed by a successful authentication event, confirming unauthorized access.

This sequence of events is a strong indicator of malicious activity and would require immediate investigation and response in a production environment.

A screenshot of the triggered Splunk alert is provided below for reference.




splunk>enterprise
Apps
Administrator
Messages
Settings
Activity
Help
Find
Search
Analytics
Datasets
Reports
Alerts
Dashboards
Search & Reporting
New Search
Save As
Create Table View
Close
* "sshd" "failed"
| rex "from (?<src>\d+\.\d+\.\d+\.\d+)"
| stats count by src
| where count > 10
Time range: Date time range
30 events (1/16/26 3:53:00.000 PM to 1/16/26 4:03:00.000 PM)
No Event Sampling
Job
Fast Mode
Events
Patterns
Statistics (1)
Visualization
Show: 20 Per Page
Format
Preview: On
src
count
192.168.11.132
30


* "sshd" "failed" | rex "from (?<src>\d+\.\d+\.\d+\.\d+)" | search src="192.168.11.132"
Time range: Date time range
30 events (1/16/26 3:53:00.000 PM to 1/16/26 4:03:00.000 PM)
No Event Sampling
Job
Fast Mode
Events (30)
Patterns
Statistics
Visualization
Timeline format
Zoom Out
Zoom to Selection
Deselect
1 minute per column
Format
Show: 20 Per Page
View: List
Prev
1
2
Next
Hide Fields
All Fields
SELECTED FIELDS
host 1
source 1
sourcetype 1
INTERESTING FIELDS
index 1
linecount 1
splunk_server 1
src 1
Extract New Fields
Time
Event
1/16/26 3:56:38.220 PM
2026-01-16T15:56:38.220232+05:30 visvakaanth-VMware-Virtual-Platform sshd[9816]: Failed password for visvakaanth from 192.168.11.132 port 34835 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:35.201 PM
2026-01-16T15:56:35.201547+05:30 visvakaanth-VMware-Virtual-Platform sshd[9814]: Failed password for visvakaanth from 192.168.11.132 port 37085 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:33.192 PM
2026-01-16T15:56:33.192158+05:30 visvakaanth-VMware-Virtual-Platform sshd[9812]: Failed password for visvakaanth from 192.168.11.132 port 40681 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:29.543 PM
2026-01-16T15:56:29.543358+05:30 visvakaanth-VMware-Virtual-Platform sshd[9809]: Failed password for visvakaanth from 192.168.11.132 port 38451 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:25.893 PM
2026-01-16T15:56:25.893457+05:30 visvakaanth-VMware-Virtual-Platform sshd[9807]: Failed password for visvakaanth from 192.168.11.132 port 34983 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:22.878 PM
2026-01-16T15:56:22.878378+05:30 visvakaanth-VMware-Virtual-Platform sshd[9805]: Failed password for visvakaanth from 192.168.11.132 port 35547 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:19.436 PM
2026-01-16T15:56:19.436503+05:30 visvakaanth-VMware-Virtual-Platform sshd[9803]: Failed password for visvakaanth from 192.168.11.132 port 46189 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:16.752 PM
2026-01-16T15:56:16.752669+05:30 visvakaanth-VMware-Virtual-Platform sshd[9801]: Failed password for visvakaanth from 192.168.11.132 port 36955 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:14.740 PM
2026-01-16T15:56:14.740448+05:30 visvakaanth-VMware-Virtual-Platform sshd[8999]: Failed password for visvakaanth from 192.168.11.132 port 32877 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:10.765 PM
2026-01-16T15:56:10.765179+05:30 visvakaanth-VMware-Virtual-Platform sshd[8996]: Failed password for visvakaanth from 192.168.11.132 port 32919 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:07.120 PM
2026-01-16T15:56:07.120483+05:30 visvakaanth-VMware-Virtual-Platform sshd[8994]: Failed password for visvakaanth from 192.168.11.132 port 41461 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:56:02.803 PM
2026-01-16T15:56:02.803759+05:30 visvakaanth-VMware-Virtual-Platform sshd[8992]: Failed password for visvakaanth from 192.168.11.132 port 37443 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/16/26 3:55:58.501 PM
2026-01-16T15:55:58.501996+05:30 visvakaanth-VMware-Virtual-Platform sshd[8990]: Failed password for visvakaanth from 192.168.11.132 port 35843 ssh2
host = visvakaanth-VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth

9. Post-Exploitation Activity – Data Staging

MITRE ATT&CK:

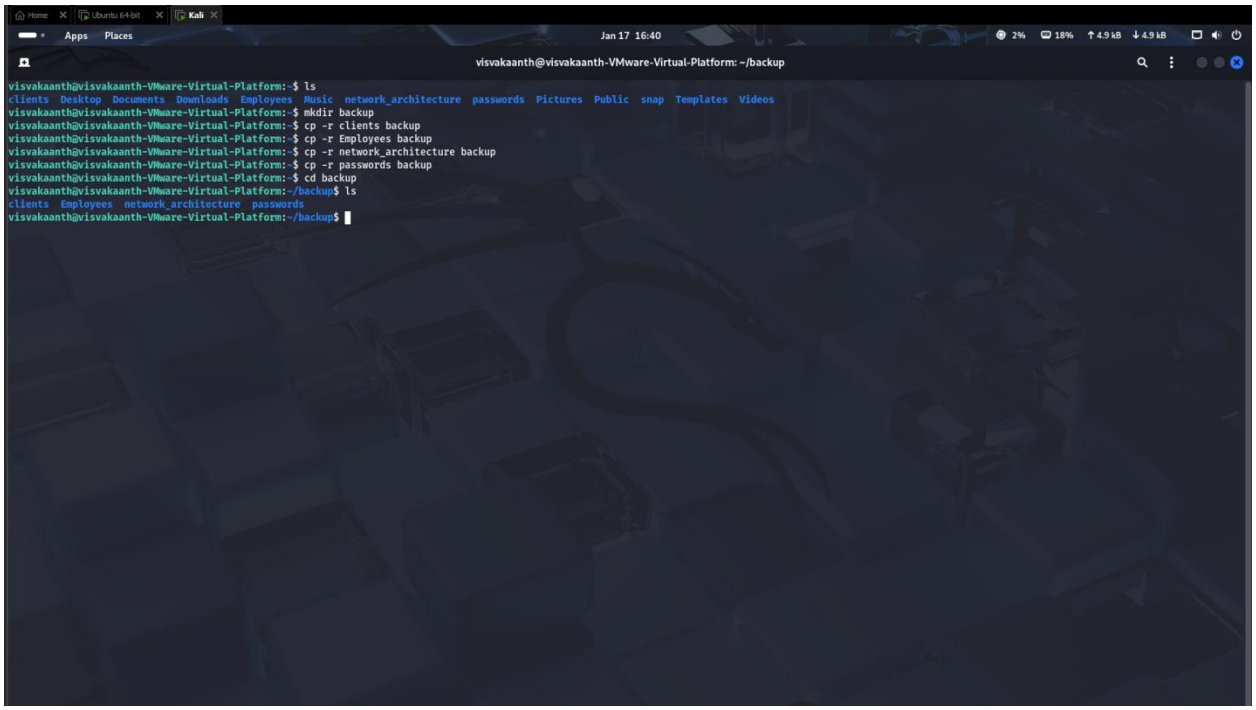
- T1074 – Data Staged

After the attacker gained initial access to the Ubuntu endpoint via an SSH brute force attack, post-exploitation activity was observed that indicated preparation for potential data exfiltration.

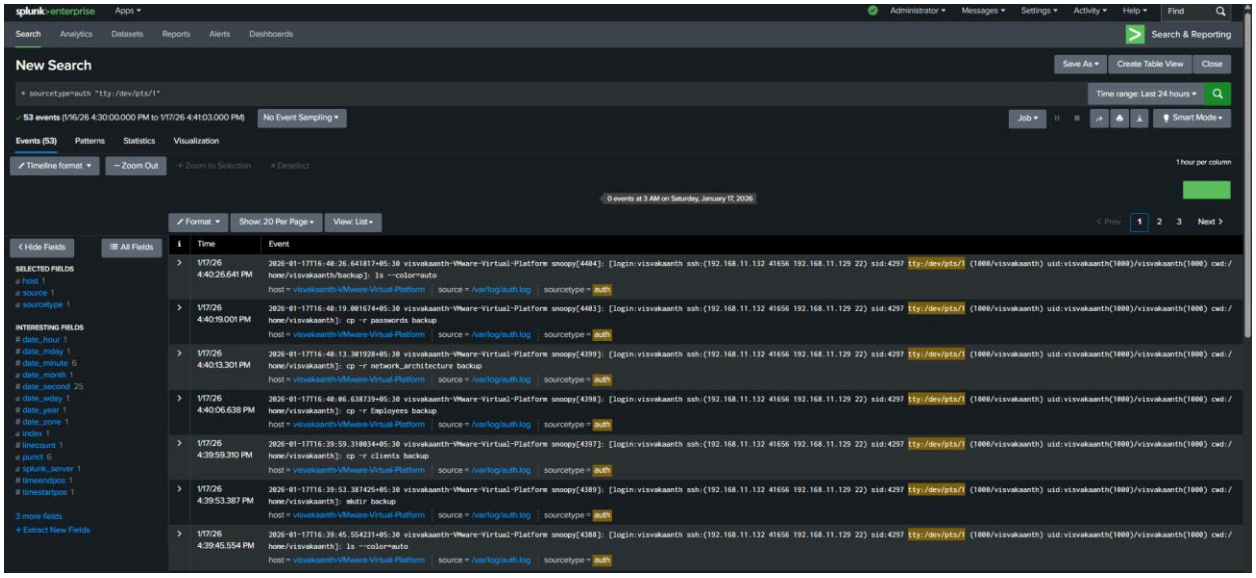
As part of this activity, the attacker aggregated files of interest from various locations on the system into a single directory named **"backup."** This technique, commonly referred to as **data staging**, is used by attackers to consolidate collected data prior to transferring it off the compromised host.

The staging activity occurred over the established SSH session originating from the Kali Linux attacker machine.

A screenshot demonstrating the creation of the staging directory and subsequent file aggregation is provided below.



Host-based audit logs captured the creation of the staging directory as well as multiple file copy operations initiated by the compromised user account. When correlated with earlier SSH authentication logs, this activity strongly indicates attacker-driven behavior rather than legitimate administrative actions.



10. Data Exfiltration Activity

MITRE ATT&CK:

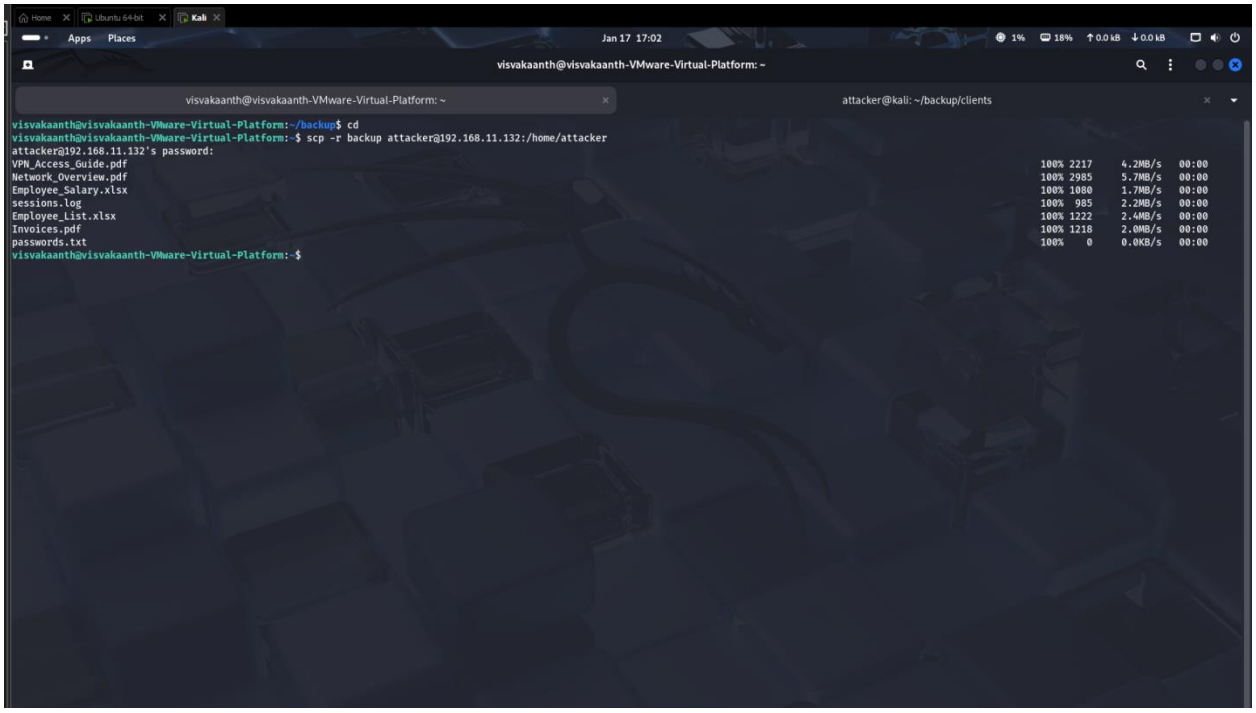
- **T1048 – Exfiltration Over Alternative Protocol**

Following the completion of data staging on the compromised Ubuntu endpoint, the attacker initiated data exfiltration by transferring the aggregated files to the external attacker-controlled system.

The exfiltration was performed using the Secure Copy Protocol (SCP) over the existing SSH connection from the Ubuntu host to the Kali Linux attacker machine. This method allows attackers to exfiltrate data while leveraging encrypted channels, making the activity more difficult to distinguish from legitimate administrative traffic.

Correlation with earlier audit and authentication logs shows that the exfiltration activity occurred shortly after data staging and successful SSH authentication, indicating a clear and structured attack progression

A screenshot demonstrating the SCP-based file transfer from the Ubuntu endpoint to the Kali system is provided below.



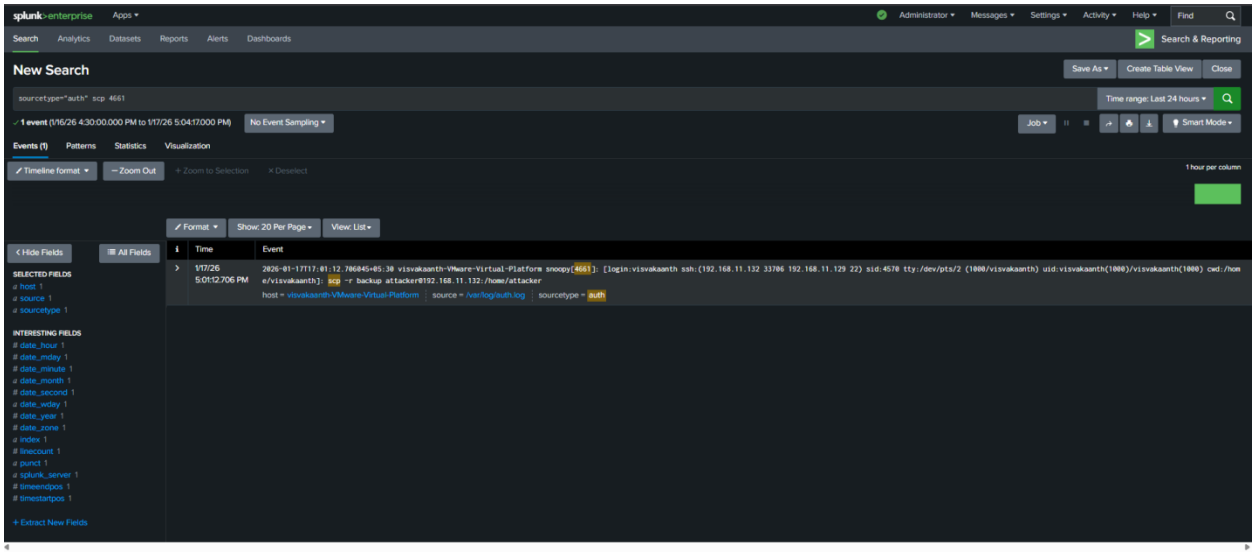
11. Detection of Data Exfiltration

Following the data staging activity, Splunk Enterprise successfully captured log events indicating outbound file transfer activity from the compromised Ubuntu endpoint to an external system.

The logs show the use of the Secure Copy Protocol (SCP) over SSH, suggesting that the attacker leveraged an encrypted channel to exfiltrate the staged data. While SCP traffic may resemble legitimate administrative activity, its occurrence immediately after unauthorized access and data staging strongly indicates malicious behavior.

This activity aligns with **MITRE ATT&CK technique T1048 – Exfiltration Over Alternative Protocol**, under the **Exfiltration** tactic.

A screenshot of the relevant Splunk log events highlighting the SCP-based data transfer is provided below.



12. Conclusion

This project successfully demonstrated a full end-to-end attack lifecycle within a controlled SOC lab environment, from initial reconnaissance to data exfiltration, while highlighting the importance of effective log collection and correlation for detection.

A hardened Ubuntu endpoint was deployed alongside a Kali Linux attacker system, with centralized log ingestion configured through Splunk Forwarder and Splunk Enterprise. Baseline security controls were applied using the Uncomplicated Firewall (UFW), restricting inbound access to SSH only, simulating a typical enterprise workstation configuration.

Reconnaissance activity originating from the attacker system was detected through port scanning, which revealed SSH as the only exposed service. Splunk alerts configured to identify scanning behavior successfully triggered, providing early visibility into suspicious activity. Subsequent brute force attempts against the SSH service resulted in multiple failed authentication events followed by a successful login, all of which were detected through SSH brute force alerts.

Following initial access, post-exploitation activity was observed in the form of data staging. Host-based audit logs captured the creation of a staging directory and file aggregation behavior, indicating preparation for data exfiltration. These events were correlated with prior authentication activity, confirming attacker-driven behavior.

Finally, data exfiltration was performed using the Secure Copy Protocol (SCP) over the established SSH session. Splunk logs captured outbound file transfer activity consistent with encrypted exfiltration techniques commonly used by attackers to evade network-based detection.

Overall, this lab demonstrates how layered detection—combining network events, authentication logs, and host-based auditing—enables security teams to identify, correlate, and understand attacker behavior across the full intrusion lifecycle. The project reinforces the importance of centralized logging, alert tuning, and post-exploitation visibility in real-world SOC operations.