

1. Lab Environment Setup

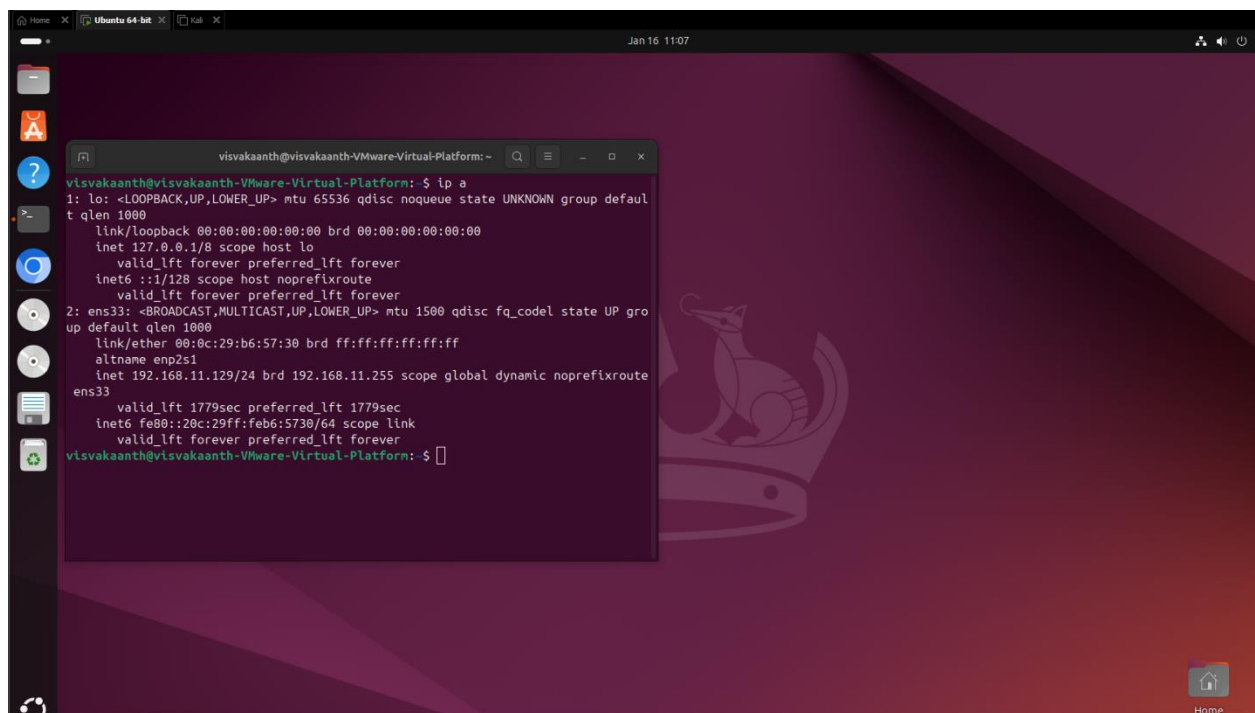
As part of this SOC lab simulation, an Ubuntu virtual machine was deployed using VMware Workstation to represent a typical employee workstation found in an enterprise environment. This machine acts as the target system for the simulated attacker activity.

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 terminal output. The terminal window is open on a desktop environment with a dark purple background and a sidebar on the left containing various application icons. The date and time 'Jan 16 11:07' are visible in the top right corner of the desktop.

```
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

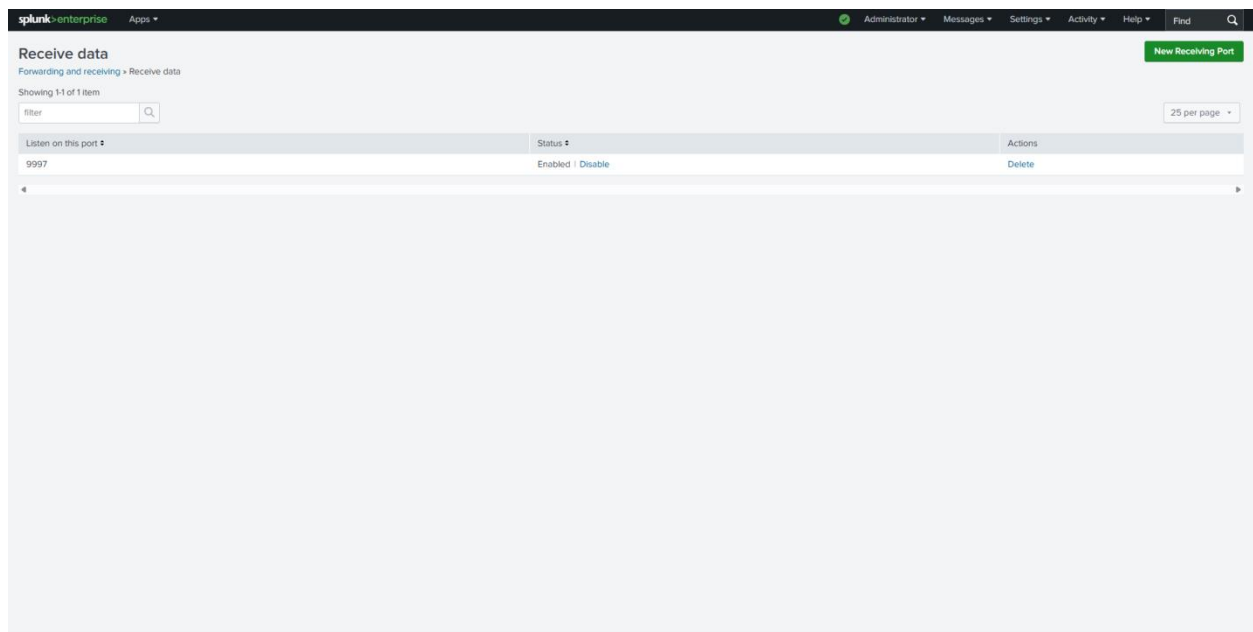
Splunk Enterprise was deployed on the Windows host system to act as the central log collection and analysis platform for this SOC lab environment.

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.

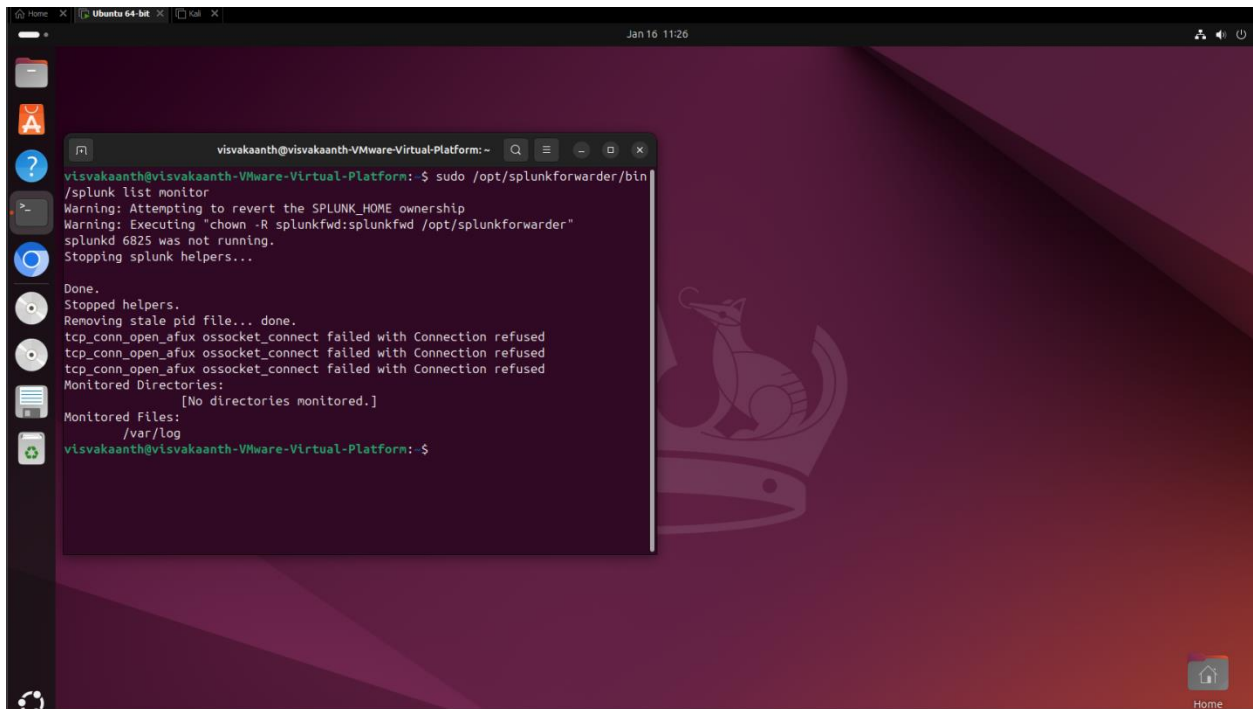
A screenshot showing Splunk listening on port 9997 is provided below for reference.



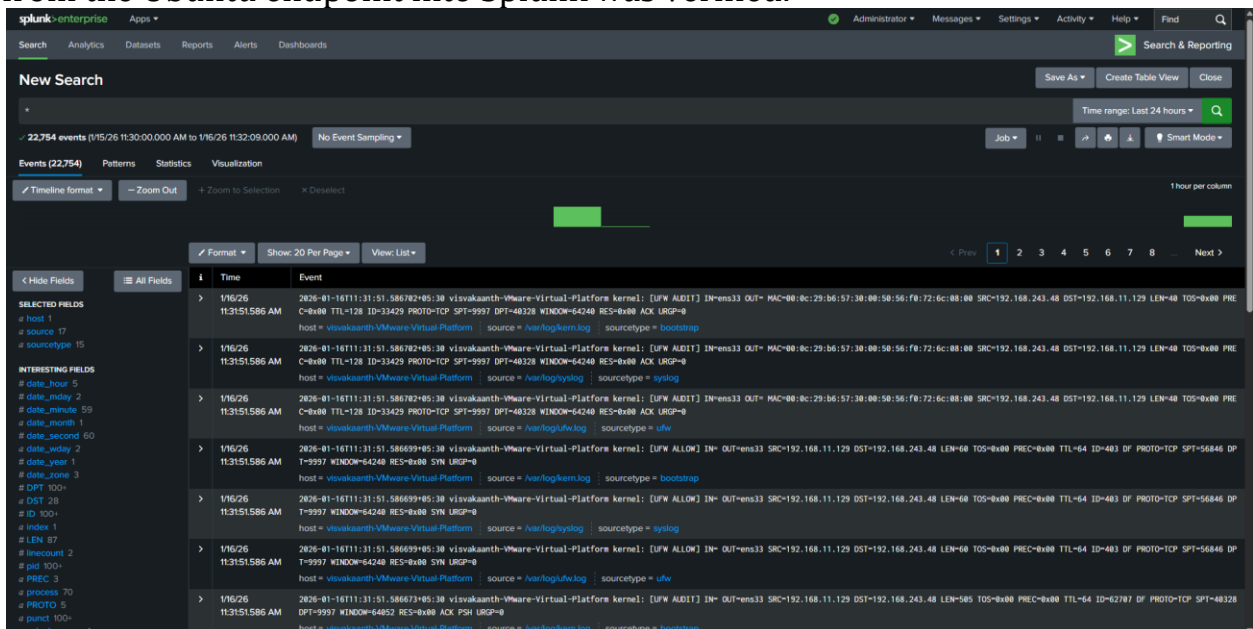
This configuration allows the Ubuntu system to forward system and authentication logs to Splunk in real time, enabling centralized monitoring and detection of suspicious activity.

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.



To validate that log forwarding was functioning correctly, real-time log ingestion from the Ubuntu endpoint into Splunk was verified.



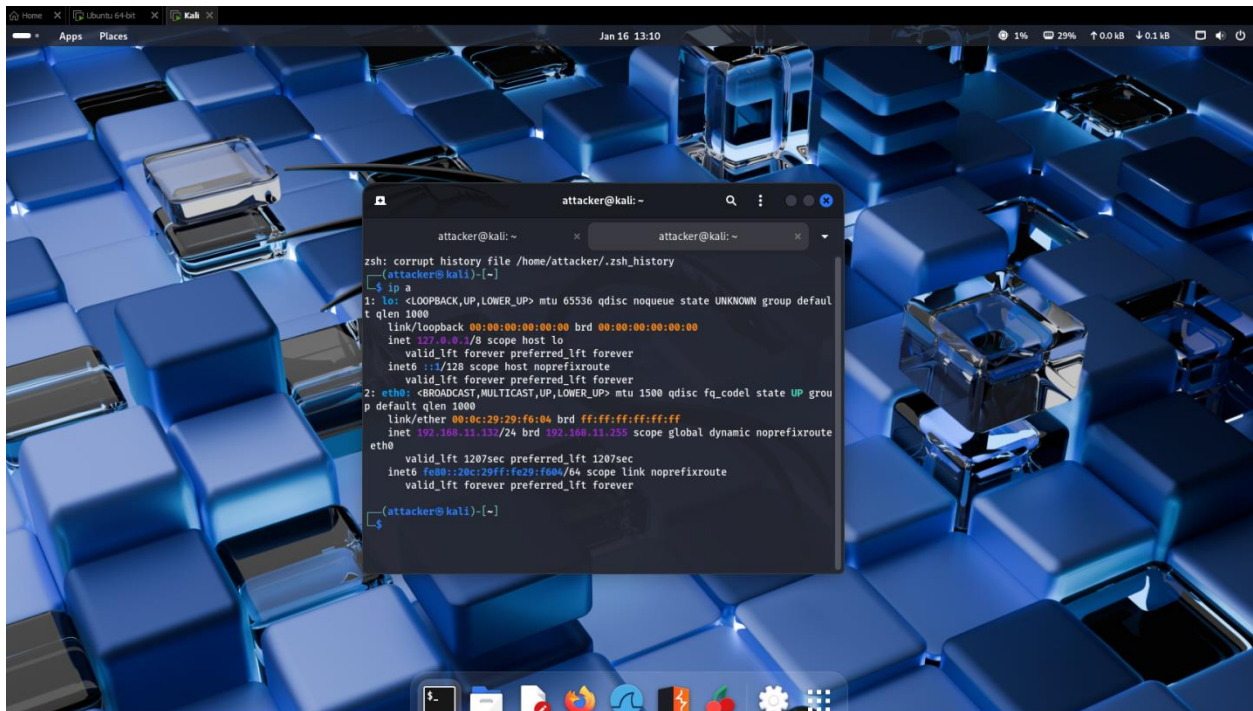
4. Attack Simulation – Reconnaissance & Port Scanning

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



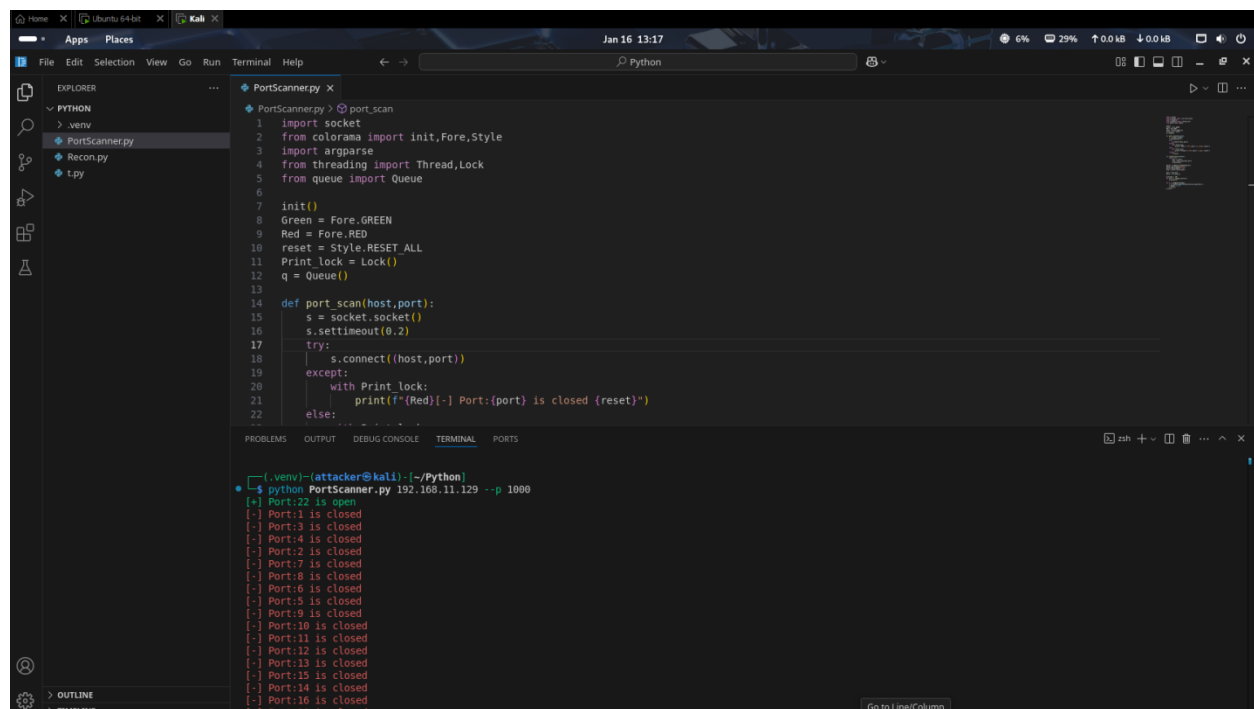
5. Custom Port Scanning Attack from Kali Linux

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         with Print_lock:
24             print(f'{Green}{+} Port:{port} is open {reset}')
25
26 if __name__ == '__main__':
27     parser = argparse.ArgumentParser()
28     parser.add_argument('ip', type=str, help='Target IP address')
29     parser.add_argument('-p', type=int, help='Port range (e.g., 1-1000)')
30     args = parser.parse_args()
31     ip = args.ip
32     port_range = args.p
33     threads = []
34     for port in range(1, port_range + 1):
35         thread = Thread(target=port_scan, args=(ip, port))
36         threads.append(thread)
37         thread.start()
38     for thread in threads:
39         thread.join()
40
41 (.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: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
[-] 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.

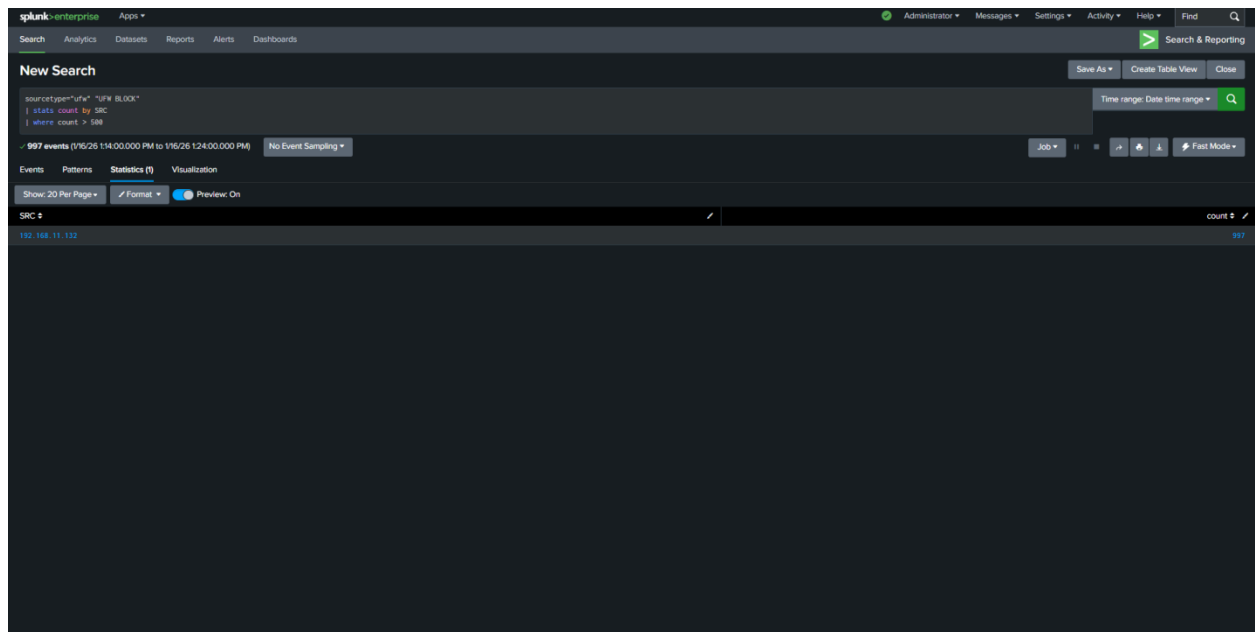
The alert is based on a search query that monitors repeated connection attempts from a single source IP address to multiple ports within a defined time window. This pattern is commonly associated with automated scanning activity performed during the reconnaissance phase of an attack.

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.

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 a 'Search & Reporting' button are visible. The main content area is titled 'Port Scan' and shows alert configuration details. The 'Trigger Condition' is set to 'Number of Results is > 0', and the 'Actions' section shows '2 Actions' including 'Add to Triggered Alerts'. A 'Trigger History' table is displayed below, showing 11 entries of triggered alerts with timestamps and 'View Results' links.

	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



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:V7Inq6$P2mZg9A'
[-] No active DB - Credential data will not be saved!
[-] 192.168.11.129:22 - Failed: 'visvakaanth:xR4Kp!8MqSgT'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:9D1cZQm2P8L'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:A57x1WQ9RzP'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:mT8gRZ!Pq7D'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K1PpQmZ7Qc'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:R2!ZgXSP8Q'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:7Q3!ZxP8R9K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:DWZ7!PqQ9xK'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Pq9!ZxQ8K7R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:xQ7!qZP8K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Zg9P!Q8x7K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:k7Zg!P8Q9xR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:lQZ7gP98xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:RZg9P!Q8x7K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:x7gZP!Q8K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:9g!xQ7P8K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:PRZg!Qx7KR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Q1Z7gP98xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:gZ9!Q7xKR'
[-] 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:Void!Kernel_27'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Dark!Neuron5Sky'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:cyber@omin_91'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Dream!csg77'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Phantom!Traces'
[-] 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
Mware-Virtual-Platform 6.14.0-37-generic #37-24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov 20 10:25:38 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.

new search

+ "smb" "failed" | rex "from (\"[a-z0-9-\\._]+\\._\\._\\._\")" | search src="192.168.11.132"

Time range: Date time range

30 events (1/6/26 3:53:00.000 PM to 1/6/26 4:03:00.000 PM)

No Event Sampling

Job

Timeline format

Zoom Out

Zoom to Selection

Deselected

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

filecount 1

spark_server 1

src 1

Extract New Fields

	Time	Event
>	1/6/26 3:56:38.220 PM	2026-01-16T15:56:38.226232+05:30 visvakaanth-Vmware-Virtual-Platform [9816] failed password for visvakaanth from 192.168.11.132 port 34035 ssh2 host = visvakaanth-Vmware-Virtual-Platform source = /var/log/auth.log sourcetype = auth
>	1/6/26 3:56:39.201 PM	2026-01-16T15:56:39.281547+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:33.192158+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:29.543358+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:25.893467+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:22.878378+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:19.435983+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:16.752669+05:30 visvakaanth-Vmware-Virtual-Platform [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	2026-01-16T15:56:14.748448+05:30 visvakaanth-Vmware-Virtual-Platform [9999] 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	2026-01-16T15:56:18.765179+05:30 visvakaanth-Vmware-Virtual-Platform [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
>	1/6/26 3:56:07.120 PM	2026-01-16T15:56:07.128483+05:30 visvakaanth-Vmware-Virtual-Platform [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
>	1/6/26 3:56:02.803 PM	2026-01-16T15:56:02.883759+05:30 visvakaanth-Vmware-Virtual-Platform [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
>	1/6/26 3:55:58.501 PM	2026-01-16T15:55:58.581996+05:30 visvakaanth-Vmware-Virtual-Platform [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