SOC Lab Simulation Report

This project is for educational and demonstration purposes only.

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

1. Lab Environment Setup

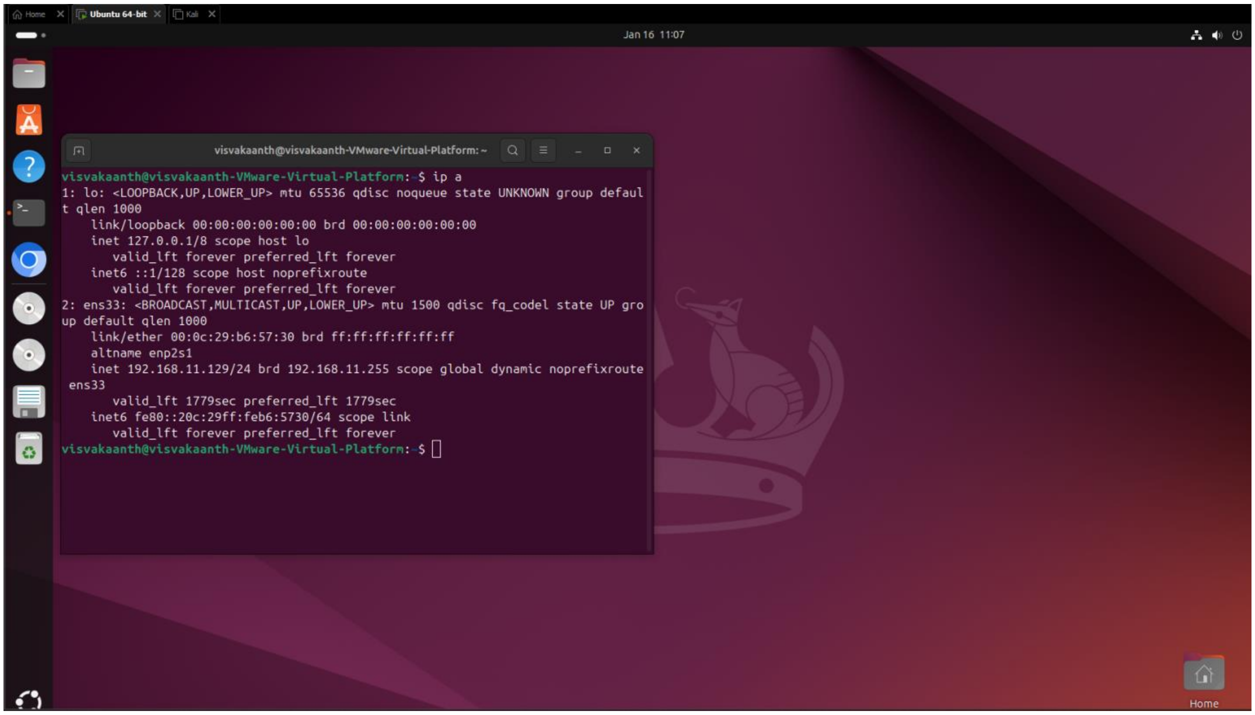
As part of this SOC lab simulation, an Ubuntu virtual machine was deployed using VMware Workstation to represent a typical employee workstation found in an enterprise environment. This machine acts as the target system for the simulated attacker activity.

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



```
visvakaanth@visvakaanth-VMware-Virtual-Platform:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 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

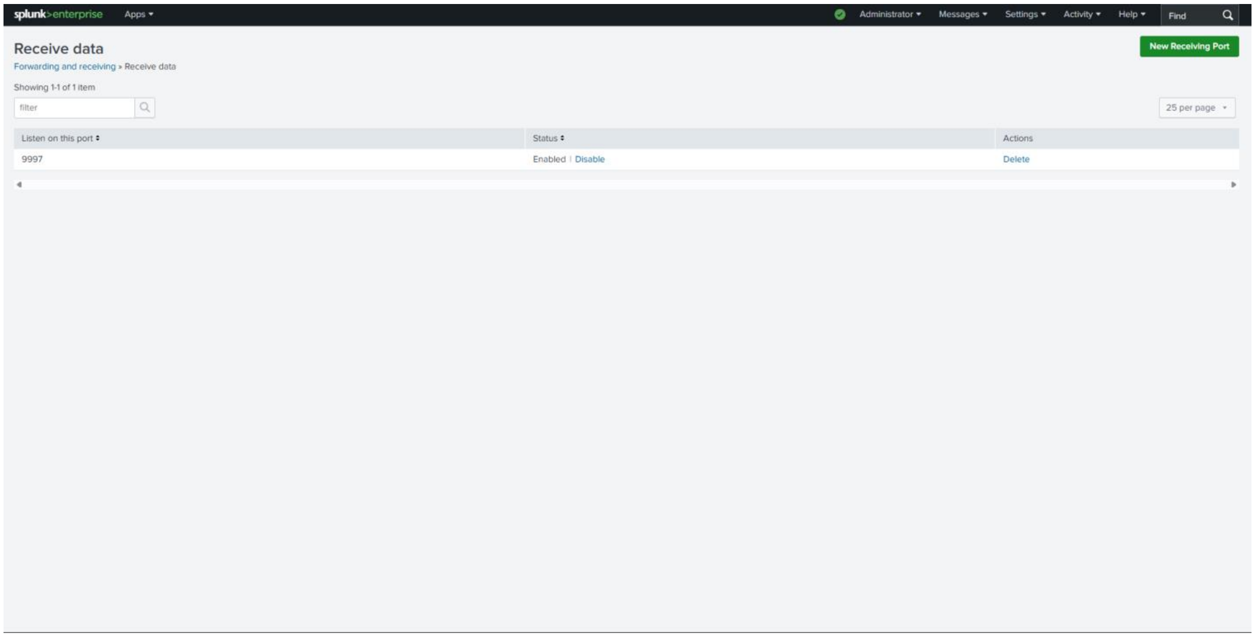
Splunk Enterprise was deployed on the Windows host system to act as the central log collection and analysis platform for this SOC lab environment.

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.



Receive data

Forwarding and receiving » Receive data

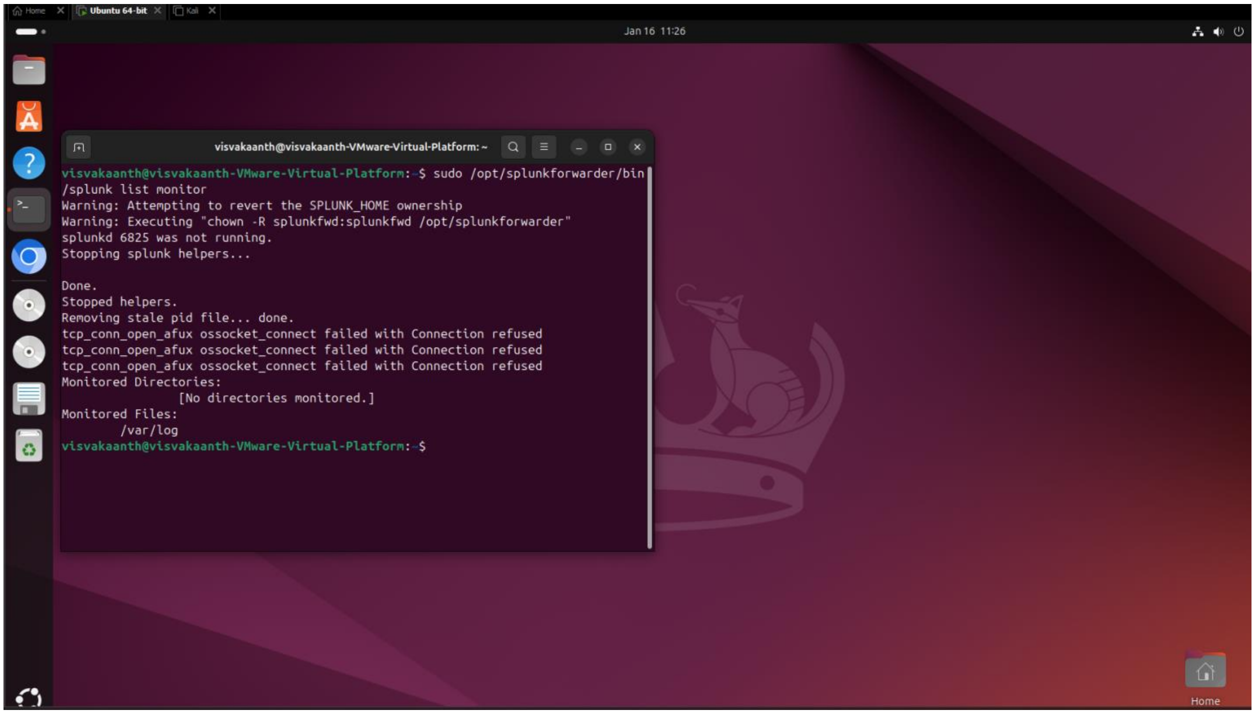
Showing 1-1 of 1 item

Listen on this port ⬍	Status ⬍	Actions
9997	Enabled | Disable	Delete

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

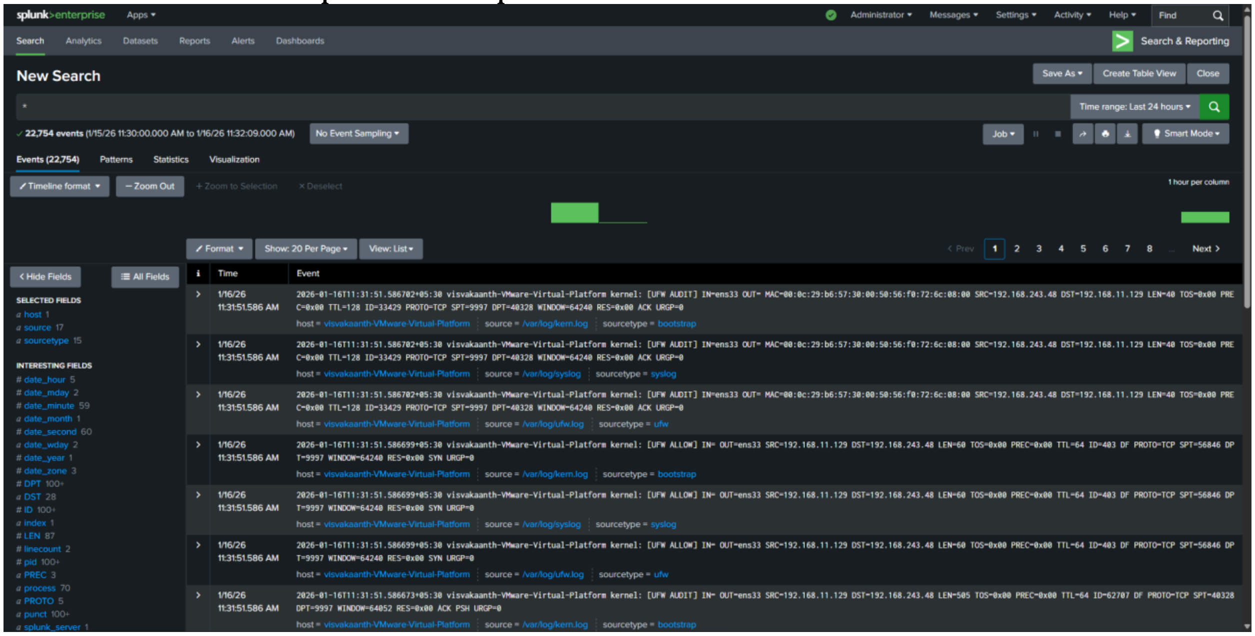
The Splunk Universal Forwarder on the Ubuntu endpoint was configured to monitor the "/var/log" directory, including authentication and system logs. These logs were forwarded in real-time to the Splunk Enterprise instance over TCP port 9997.



```
visvakaanth@visvakaanth-VMware-Virtual-Platform:~$ sudo /opt/splunkforwarder/bin
/splunk list monitor
Warning: Attempting to revert the SPLUNK_HOME ownership
Warning: Executing "chown -R splunkfwd:splunkfwd /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.



```
*
Time range: Last 24 hours
✓ 22,754 events (1/15/26 11:30:00.000 AM to 1/16/26 11:32:09.000 AM)
```

```
SELECTED FIELDS
a host 1
a source 17
a sourcetype 15

INTERESTING FIELDS
# date_hour 5
# date_mday 2
# date_minute 59
a date_month 1
# date_second 60
a date_wday 2
# date_year 1
# date_zone 3
# DPT 100+
a DST 28
# ID 100+
a index 1
# LEN 87
# linecount 2
# pid 100+
a PREC 3
a process 70
a PROTO 5
a punct 100+
a splunk_server 1
```

```
i  Time                 Event
>  1/16/26              2026-01-16T11:31:51.586702+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN=ens33 OUT= MAC=00:0c: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 PRE
   11:31:51.586 AM      C=0x00 TTL=128 ID=33429 PROTO=TCP SPT=9997 DPT=40328 WINDOW=64240 RES=0x00 ACK URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/kern.log | sourcetype = bootstrap
>  1/16/26              2026-01-16T11:31:51.586702+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN=ens33 OUT= MAC=00:0c: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 PRE
   11:31:51.586 AM      C=0x00 TTL=128 ID=33429 PROTO=TCP SPT=9997 DPT=40328 WINDOW=64240 RES=0x00 ACK URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/syslog | sourcetype = syslog
>  1/16/26              2026-01-16T11:31:51.586702+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN=ens33 OUT= MAC=00:0c: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 PRE
   11:31:51.586 AM      C=0x00 TTL=128 ID=33429 PROTO=TCP SPT=9997 DPT=40328 WINDOW=64240 RES=0x00 ACK URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/ufw.log | sourcetype = ufw
>  1/16/26              2026-01-16T11:31:51.586699+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW ALLOW] IN= OUT=ens33 SRC=192.168.11.129 DST=192.168.243.48 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=403 DF PROTO=TCP SPT=56846 DP
   11:31:51.586 AM      T=9997 WINDOW=64240 RES=0x00 SYN URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/kern.log | sourcetype = bootstrap
>  1/16/26              2026-01-16T11:31:51.586699+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW ALLOW] IN= OUT=ens33 SRC=192.168.11.129 DST=192.168.243.48 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=403 DF PROTO=TCP SPT=56846 DP
   11:31:51.586 AM      T=9997 WINDOW=64240 RES=0x00 SYN URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/syslog | sourcetype = syslog
>  1/16/26              2026-01-16T11:31:51.586699+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW ALLOW] IN= OUT=ens33 SRC=192.168.11.129 DST=192.168.243.48 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=403 DF PROTO=TCP SPT=56846 DP
   11:31:51.586 AM      T=9997 WINDOW=64240 RES=0x00 SYN URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/ufw.log | sourcetype = ufw
>  1/16/26              2026-01-16T11:31:51.586673+05:30 visvakaanth-VMware-Virtual-Platform kernel: [UFW AUDIT] IN= OUT=ens33 SRC=192.168.11.129 DST=192.168.243.48 LEN=505 TOS=0x00 PREC=0x00 TTL=64 ID=62707 DF PROTO=TCP SPT=40328
   11:31:51.586 AM      DPT=9997 WINDOW=64052 RES=0x00 ACK PSH URGP=0
                        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/kern.log | sourcetype = bootstrap
```

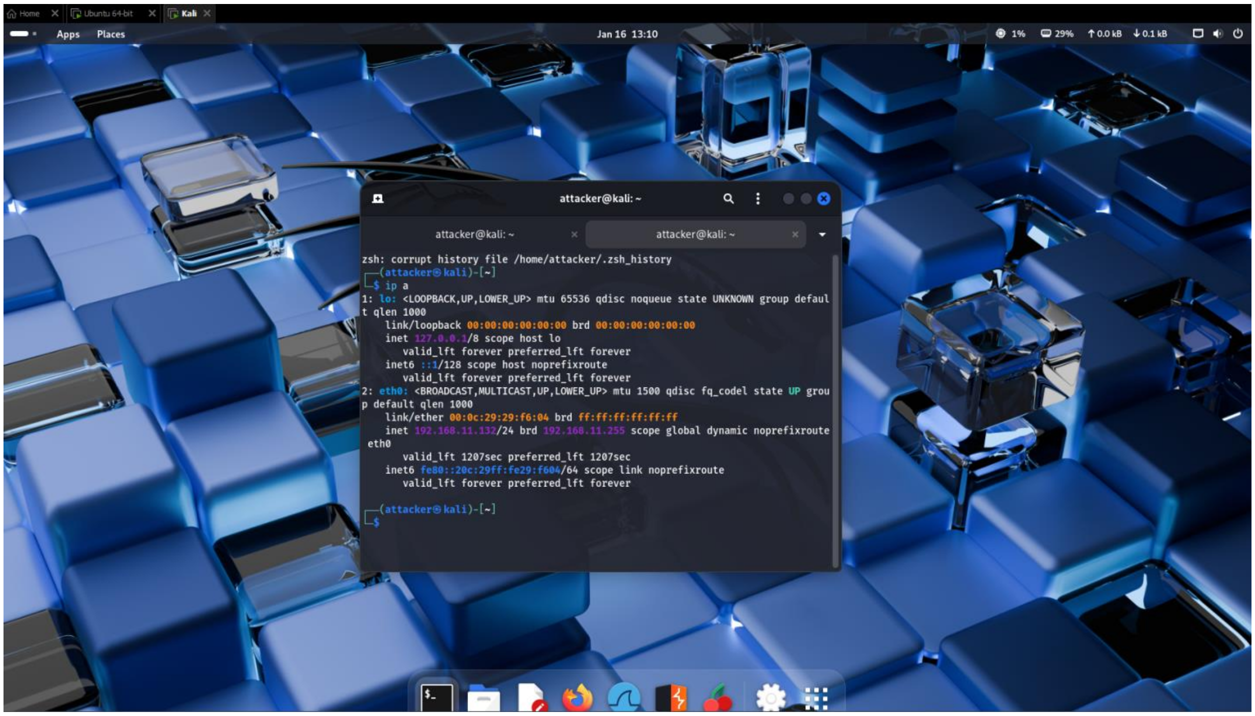
4. Attack Simulation – Reconnaissance & Port Scanning

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



```
zsh: corrupt history file /home/attacker/.zsh_history
┌──(attacker㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 00:0c:29:29:f6:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.132/24 brd 192.168.11.255 scope global dynamic noprefixroute
 eth0
       valid_lft 1207sec preferred_lft 1207sec
    inet6 fe80::20c:29ff:fe29:f604/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(attacker㉿kali)-[~]
└─$
```

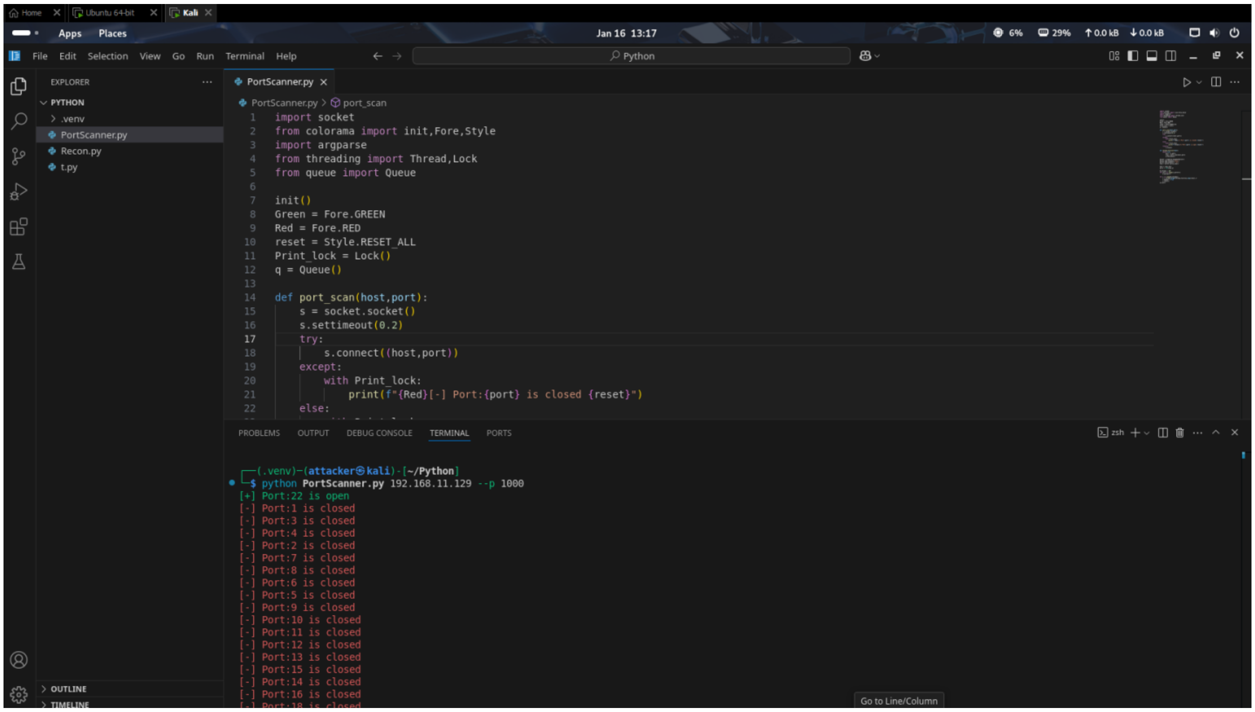
5. Custom Port Scanning Attack from Kali Linux

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



```
import socket
from colorama import init,Fore,Style
import argparse
from threading import Thread,Lock
from queue import Queue

init()
Green = Fore.GREEN
Red = Fore.RED
reset = Style.RESET_ALL
Print_lock = Lock()
q = Queue()

def port_scan(host,port):
    s = socket.socket()
    s.settimeout(0.2)
    try:
        s.connect((host,port))
    except:
        with Print_lock:
            print(f"{Red}[-] Port:{port} is closed {reset}")
    else:
```

```
┌──(.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.

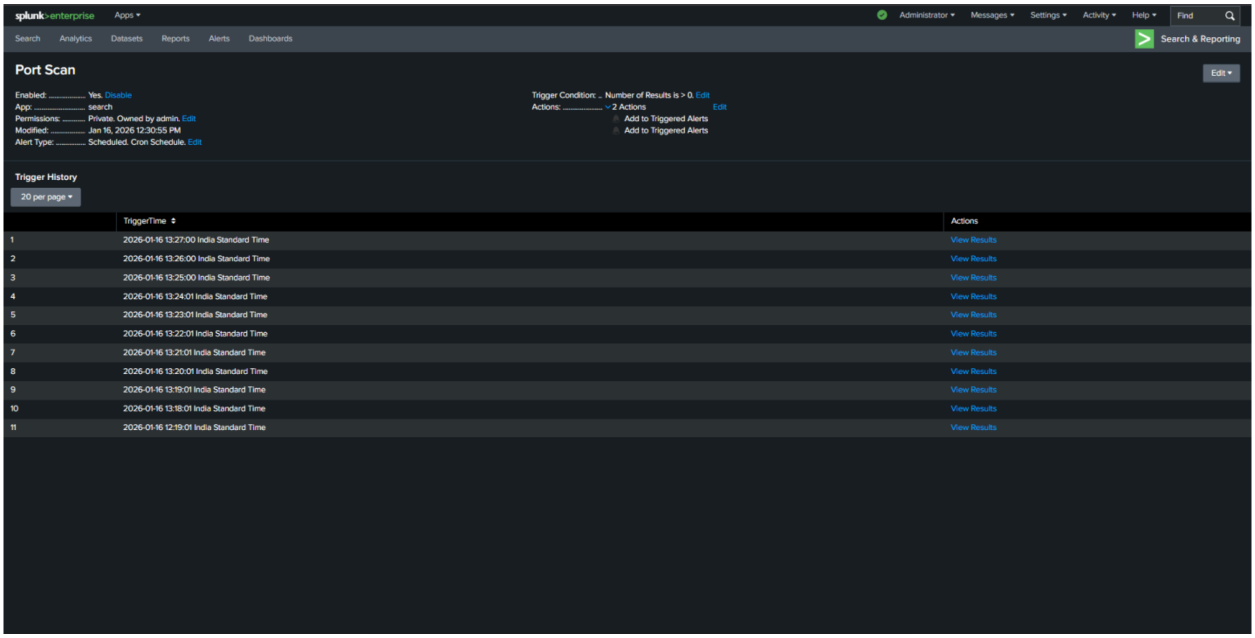
6. Detection & Alerting – Port Scanning Activity

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.



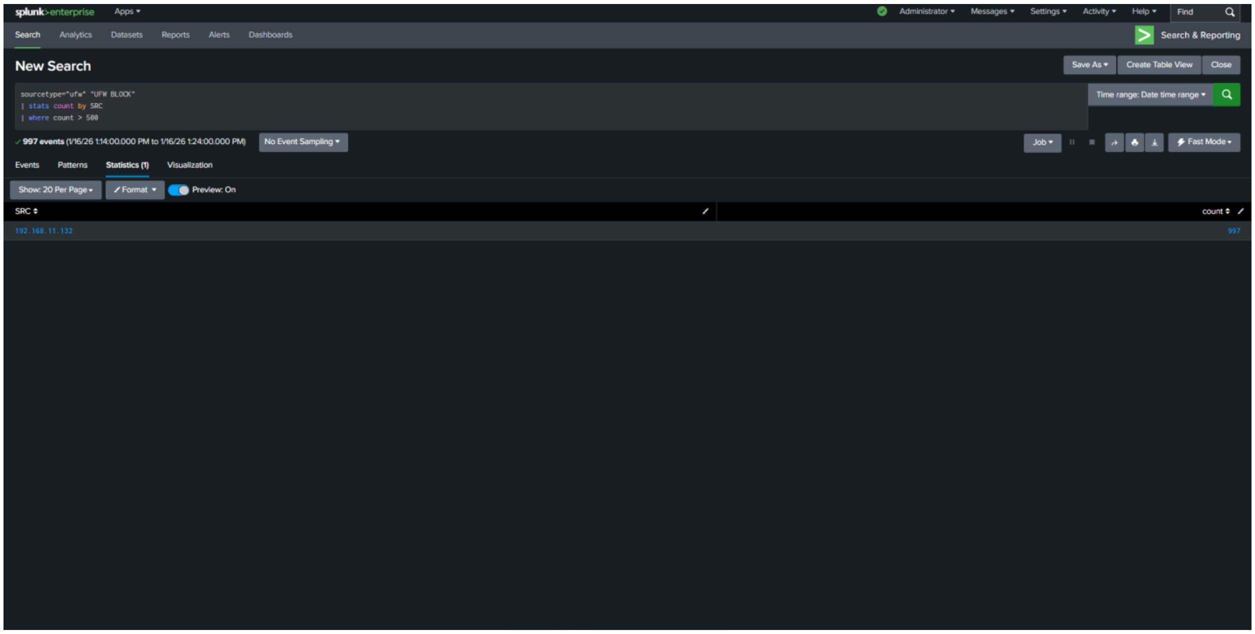
Port Scan

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 Edit
Add to Triggered Alerts
Add to Triggered Alerts

Trigger History

	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 > 500
```

✓ 997 events (1/16/26 1:14:00.000 PM to 1/16/26 1:24:00.000 PM)

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.

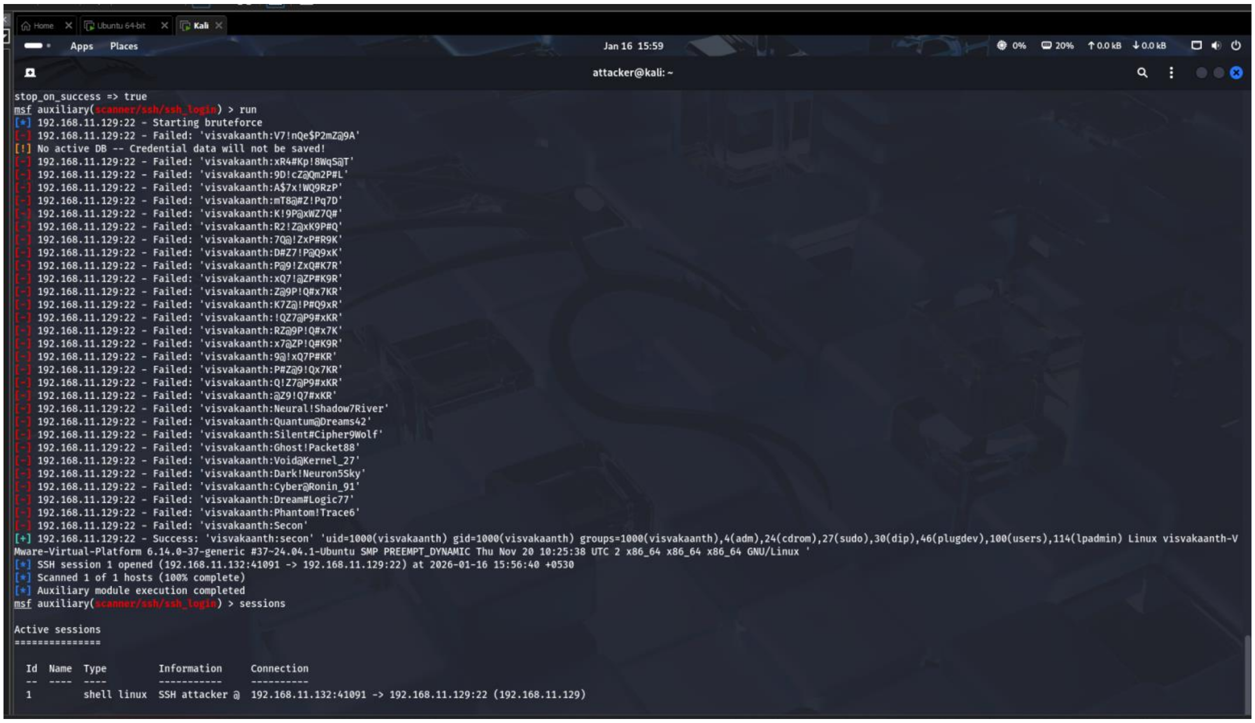
A credential-based brute force attack was executed from the Kali Linux attacker machine against the Ubuntu system using repeated authentication attempts with different username and password combinations. This technique is commonly used by attackers to gain initial access when weak or reused credentials 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:V7!nQe$P2mZ@9A'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.11.129:22 - Failed: 'visvakaanth:xR4#Kp!8WqS@T'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:9D!cZ@Qm2P#L'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:A$7x!WQ9RzP'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:mT8@#Z!Pq7D'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K!9P@xWZ7Q#'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:R2!Z@xK9P#Q'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:7Q@!ZxP#R9K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:D#Z7!P@Q9xK'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:P@9!ZxQ#K7R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:xQ7!@ZP#K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Z@9P!Q#x7KR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:K7Z@!P#Q9xR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:!QZ7@P9#xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:RZ@9P!Q#x7K'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:x7@ZP!Q#K9R'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:9@!xQ7P#KR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:P#Z@9!Qx7KR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Q!Z7@P9#xKR'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:@Z9!Q7#xKR'
[-] 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@Ronin_91'
[-] 192.168.11.129:22 - Failed: 'visvakaanth:Dream#Logic77'
[-] 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
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

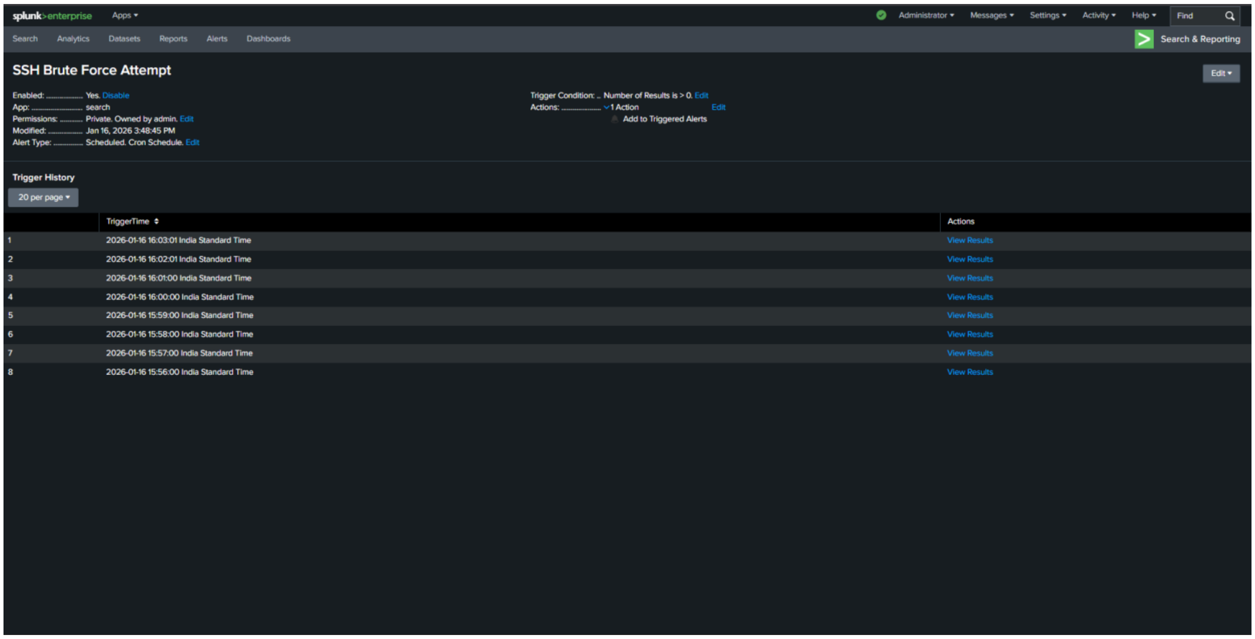
To detect credential-based attacks targeting the Ubuntu endpoint, a custom alert was configured in Splunk to identify excessive failed SSH authentication attempts from a single source IP address within a defined time window.

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.



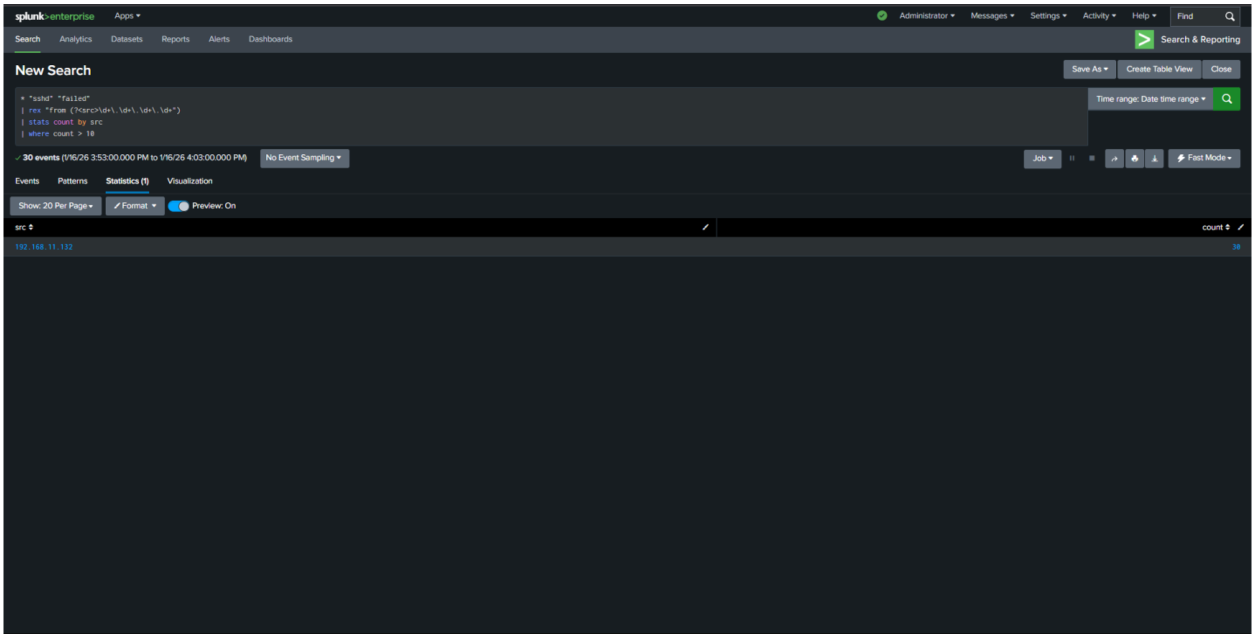
SSH Brute Force Attempt

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

Trigger Condition: .. Number of Results is > 0. Edit
Actions: .... 1 Action Edit
Add to Triggered Alerts

Trigger History

	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

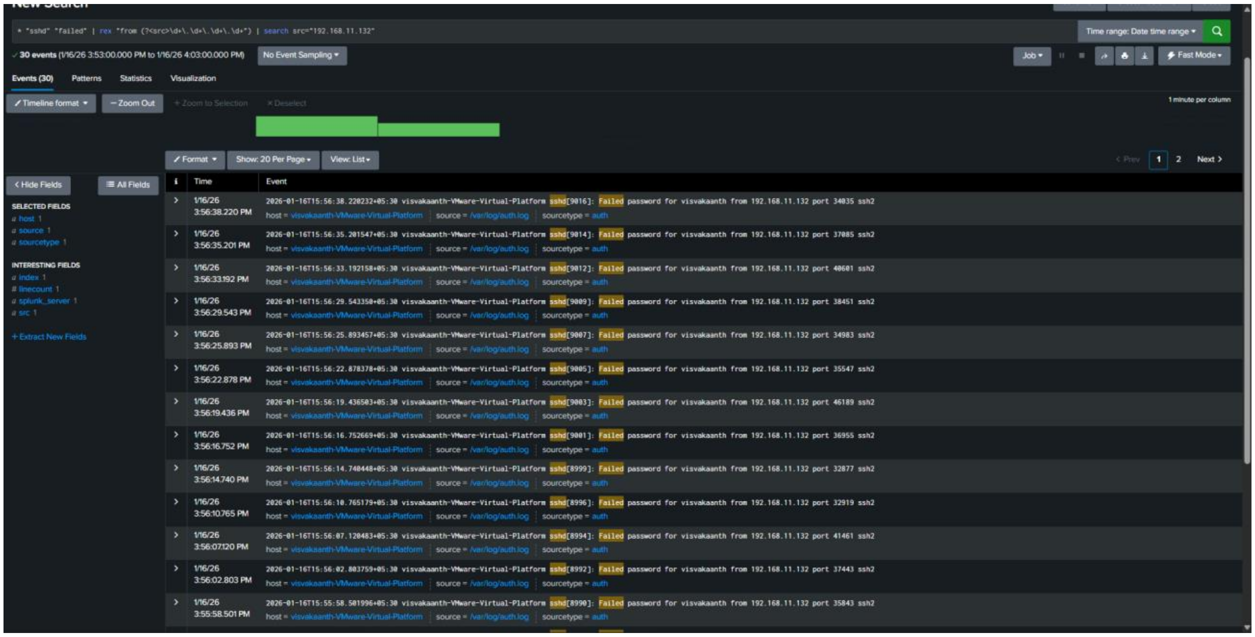


New Search

```
* "sshd" "failed"
| rex "from (?<src>\d+\.\d+\.\d+\.\d+")
| stats count by src
| where count > 10
```

✓ 30 events (1/16/26 3:53:00.000 PM to 1/16/26 4:03:00.000 PM) No Event Sampling

src	count
192.168.11.132	30



```
* "sshd" "failed" | rex "from (?<src>\d+\.\d+\.\d+\.\d+") | search src="192.168.11.132"
```

✓ 30 events (1/16/26 3:53:00.000 PM to 1/16/26 4:03:00.000 PM) No Event Sampling

```
i  Time                  Event
>  1/16/26               2026-01-16T15:56:38.220232+05:30 visvakaanth-VMware-Virtual-Platform sshd[9016]: Failed password for visvakaanth from 192.168.11.132 port 34035 ssh2
   3:56:38.220 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:35.201547+05:30 visvakaanth-VMware-Virtual-Platform sshd[9014]: Failed password for visvakaanth from 192.168.11.132 port 37085 ssh2
   3:56:35.201 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:33.192158+05:30 visvakaanth-VMware-Virtual-Platform sshd[9012]: Failed password for visvakaanth from 192.168.11.132 port 40601 ssh2
   3:56:33.192 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:29.543350+05:30 visvakaanth-VMware-Virtual-Platform sshd[9009]: Failed password for visvakaanth from 192.168.11.132 port 38451 ssh2
   3:56:29.543 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:25.893457+05:30 visvakaanth-VMware-Virtual-Platform sshd[9007]: Failed password for visvakaanth from 192.168.11.132 port 34983 ssh2
   3:56:25.893 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:22.878378+05:30 visvakaanth-VMware-Virtual-Platform sshd[9005]: Failed password for visvakaanth from 192.168.11.132 port 35547 ssh2
   3:56:22.878 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:19.436503+05:30 visvakaanth-VMware-Virtual-Platform sshd[9003]: Failed password for visvakaanth from 192.168.11.132 port 46189 ssh2
   3:56:19.436 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:16.752669+05:30 visvakaanth-VMware-Virtual-Platform sshd[9001]: Failed password for visvakaanth from 192.168.11.132 port 36955 ssh2
   3:56:16.752 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:14.740448+05:30 visvakaanth-VMware-Virtual-Platform sshd[8999]: Failed password for visvakaanth from 192.168.11.132 port 32877 ssh2
   3:56:14.740 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:10.765179+05:30 visvakaanth-VMware-Virtual-Platform sshd[8996]: Failed password for visvakaanth from 192.168.11.132 port 32919 ssh2
   3:56:10.765 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:07.120483+05:30 visvakaanth-VMware-Virtual-Platform sshd[8994]: Failed password for visvakaanth from 192.168.11.132 port 41461 ssh2
   3:56:07.120 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:56:02.803759+05:30 visvakaanth-VMware-Virtual-Platform sshd[8992]: Failed password for visvakaanth from 192.168.11.132 port 37443 ssh2
   3:56:02.803 PM        host = visvakaanth-VMware-Virtual-Platform | source = /var/log/auth.log | sourcetype = auth
>  1/16/26               2026-01-16T15:55:58.501996+05:30 visvakaanth-VMware-Virtual-Platform sshd[8990]: Failed password for visvakaanth from 192.168.11.132 port 35843 ssh2
   3:55:58.501 PM        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.