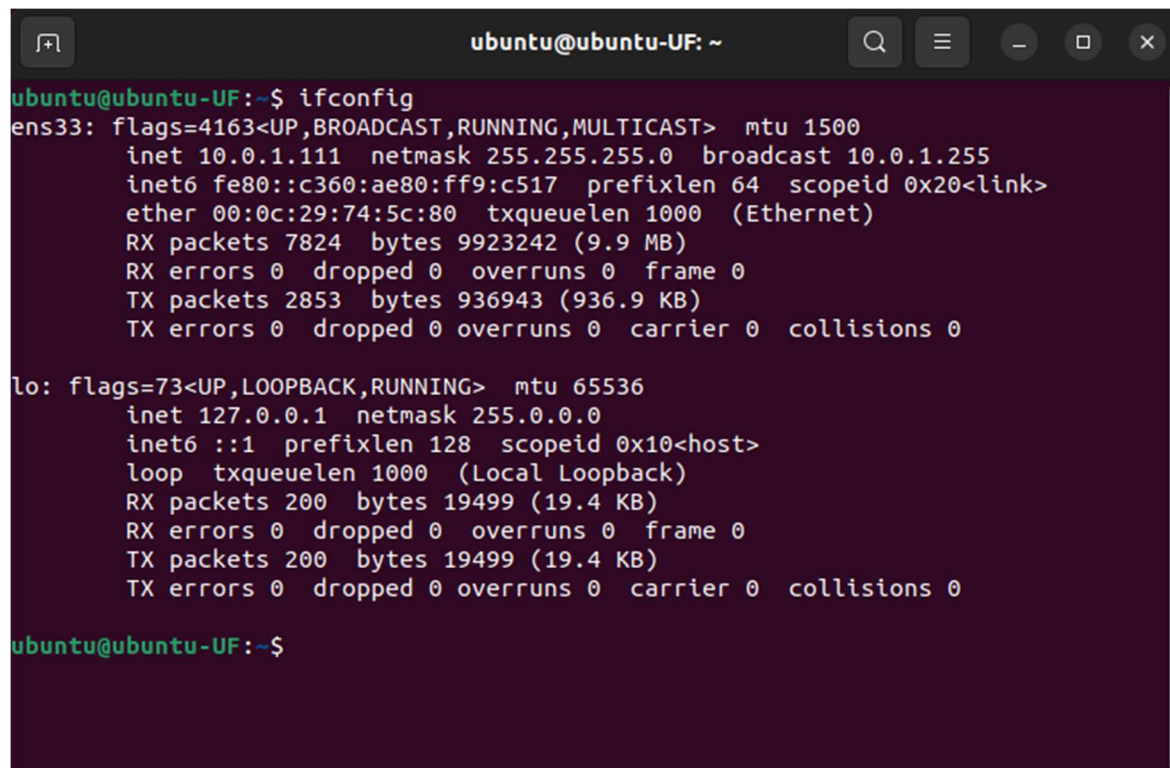


Detecting and Blocking Unauthorized Access on Linux using Fail2Ban and Splunk

In this project, I demonstrate how to configure Fail2Ban to detect and prevent unauthorized SSH login attempts. The setup includes monitoring logs, configuring Splunk for centralized logging, and simulating brute-force attacks with Hydra.

Steps & Screenshots

A terminal window titled 'ubuntu@ubuntu-UF: ~' with standard window controls. The user has entered the command 'ifconfig'. The output shows details for the 'ens33' interface (IP 10.0.1.111, MTU 1500) and the 'lo' loopback interface (IP 127.0.0.1, MTU 65536).

```
ubuntu@ubuntu-UF:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.1.111  netmask 255.255.255.0  broadcast 10.0.1.255
    inet6 fe80::c360:ae80:ff9:c517  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:74:5c:80  txqueuelen 1000  (Ethernet)
    RX packets 7824  bytes 9923242 (9.9 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2853  bytes 936943 (936.9 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 200  bytes 19499 (19.4 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 200  bytes 19499 (19.4 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

ubuntu@ubuntu-UF:~$
```

Figure 1 Ubuntu IP (Victim PC)

```
kali@kali: ~/Passwords
File Actions Edit View Help

(kali@kali)-[~/Passwords]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.1.108 netmask 255.255.255.0 broadcast 10.0.1.255
    inet6 fe80::20c:29ff:fec0:644d prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:c0:64:4d txqueuelen 1000 (Ethernet)
    RX packets 312 bytes 47758 (46.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 363 bytes 40008 (39.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 19 base 0x2000

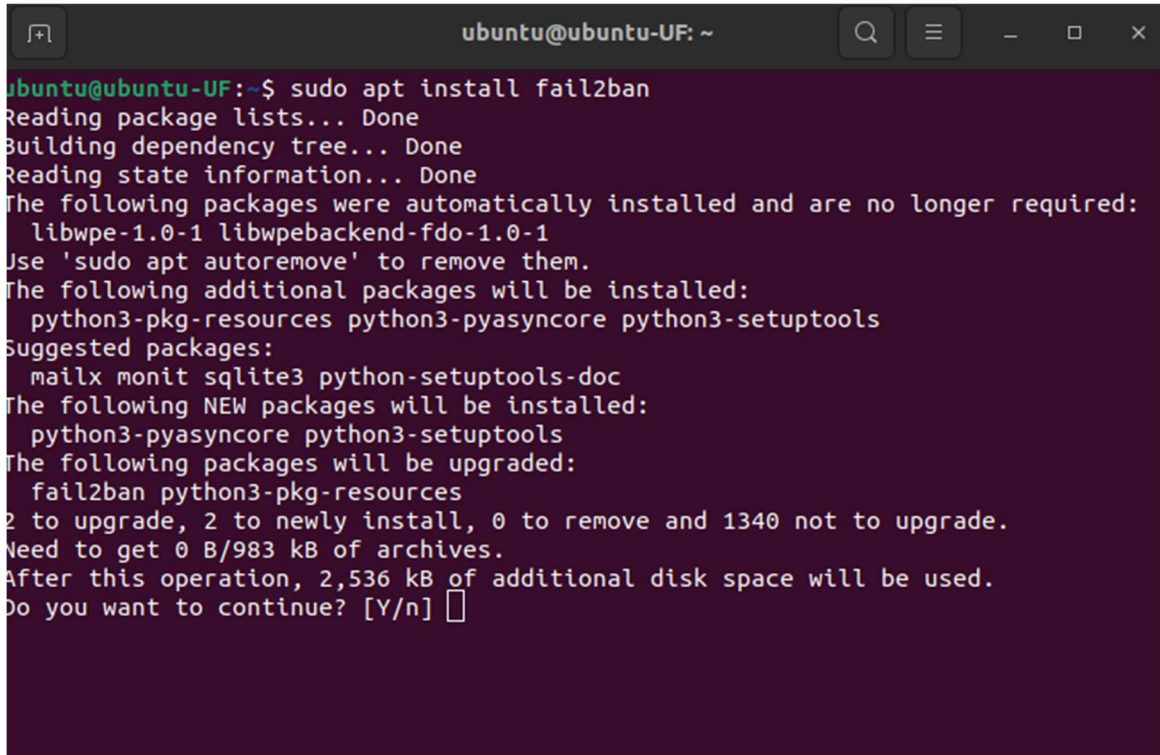
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~/Passwords]
$ █
```

Figure 2 Attacker IP

Step 1: Install Required Tools

- Installed Splunk Universal Forwarder on the victim machine.
- Installed Fail2Ban on the victim machine to monitor unauthorized access.

A terminal window titled 'ubuntu@ubuntu-UF: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'sudo apt install fail2ban' and its output. The output indicates that several packages were automatically installed and are no longer required, lists additional packages to be installed, and suggests other packages. It also shows the disk space requirements and asks for confirmation to continue.

```
ubuntu@ubuntu-UF:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-pkg-resources python3-pyasyncore python3-setuptools
Suggested packages:
  mailx monit sqlite3 python-setuptools-doc
The following NEW packages will be installed:
  python3-pyasyncore python3-setuptools
The following packages will be upgraded:
  fail2ban python3-pkg-resources
2 to upgrade, 2 to newly install, 0 to remove and 1340 not to upgrade.
Need to get 0 B/983 kB of archives.
After this operation, 2,536 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Figure 3 Fail2ban Installation

Step 2: Install Fail2Ban & Configure

Fail2Ban is an open-source intrusion prevention software that helps protect servers from brute-force attacks, unauthorized access attempts, and suspicious activities by monitoring log files and banning malicious IP addresses. It works by dynamically blocking IP addresses that exhibit malicious behavior using firewall rules.

Configure:

```
sudo nano /etc/fail2ban/jail.local
```

Added the following lines to protect the SSH service:

```
[sshd]
```

```
enabled = true
```

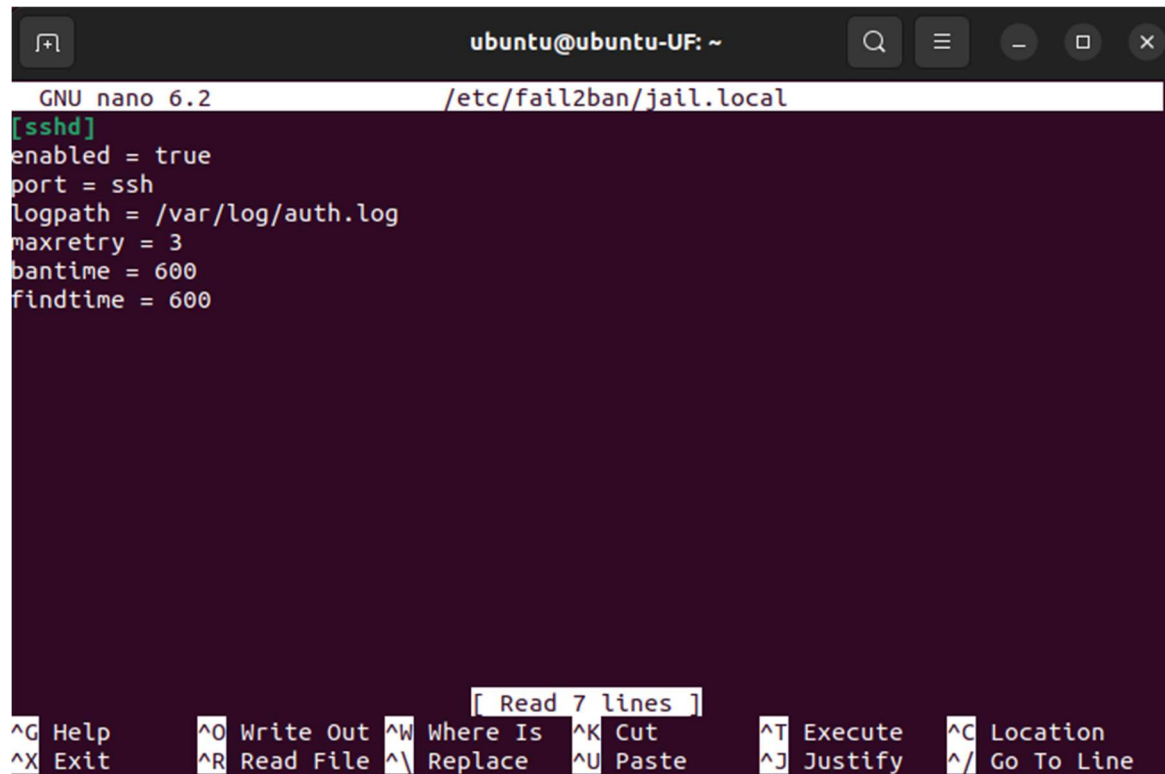
```
port = ssh
```

logpath = /var/log/auth.log

maxretry = 3

bantime = 600

findtime = 600



```
ubuntu@ubuntu-UF: ~  
GNU nano 6.2 /etc/fail2ban/jail.local  
[sshd]  
enabled = true  
port = ssh  
logpath = /var/log/auth.log  
maxretry = 3  
bantime = 600  
findtime = 600  
[ Read 7 lines ]  
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify  ^_ Go To Line
```

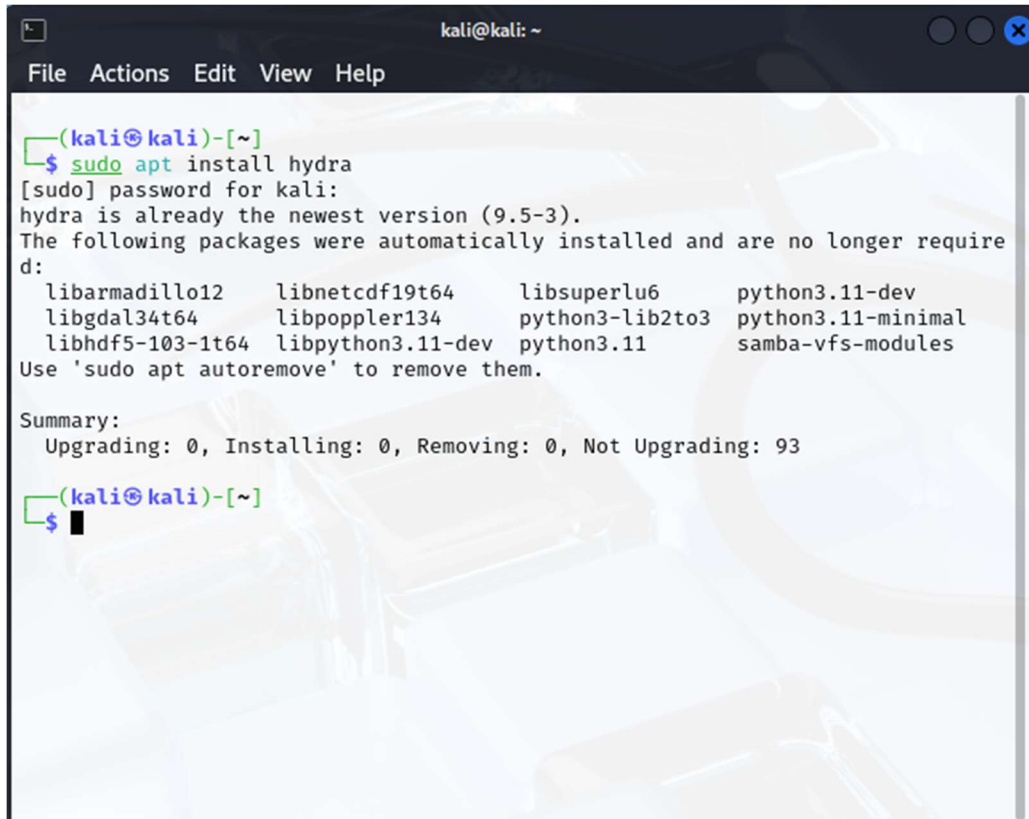
Figure 4 Fail2ban configuration

###Step 3: Created index and sourcetype in Splunk Server Dashboard

Index = fail2ban_logs

Sourcetype = fail2ban

###Step 4: Install hydra on Attack VM



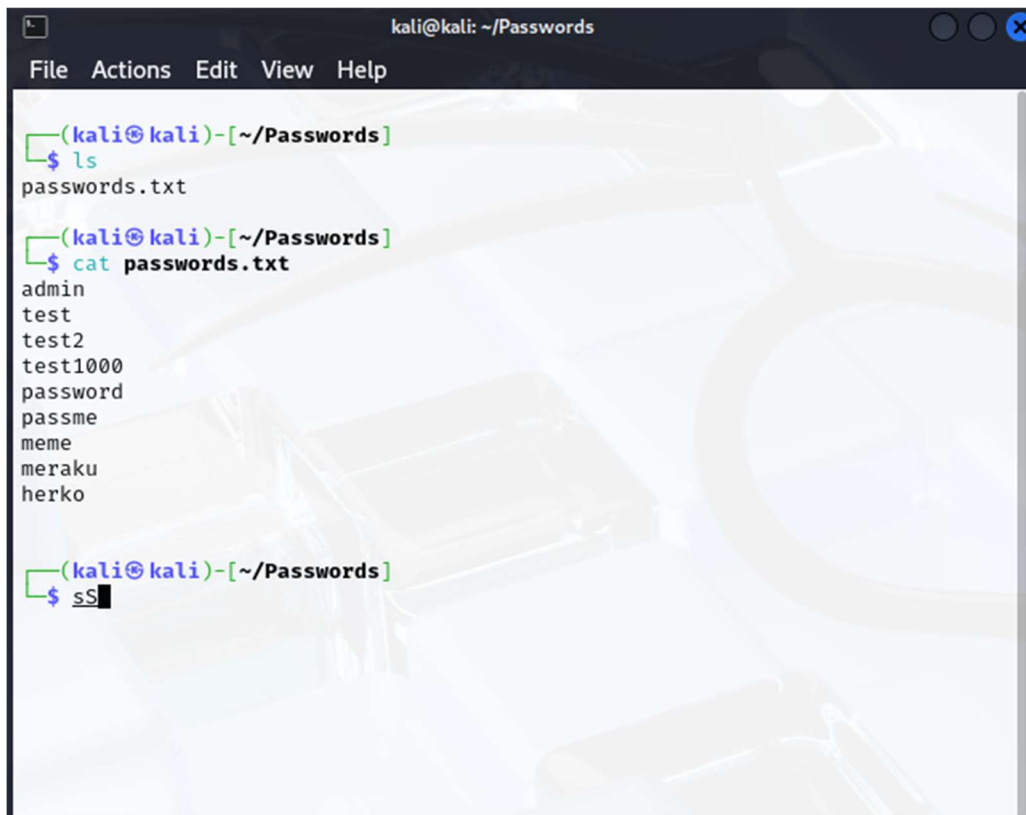
```
(kali㉿kali)-[~]
└─$ sudo apt install hydra
[sudo] password for kali:
hydra is already the newest version (9.5-3).
The following packages were automatically installed and are no longer required:
  libarmadillo12  libnetcdf19t64  libsuperlu6  python3.11-dev
  libgdal34t64   libpoppler134   python3-lib2to3  python3.11-minimal
  libhdf5-103-1t64  libpython3.11-dev  python3.11  samba-vfs-modules
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 93

(kali㉿kali)-[~]
└─$
```

Figure 5 Hydra installation on Attacker PC

###Step 5: Created a dictionary folder as Passwords > passwords.txt



```
kali@kali: ~/Passwords
File Actions Edit View Help

(kali@kali)-[~/Passwords]
$ ls
passwords.txt

(kali@kali)-[~/Passwords]
$ cat passwords.txt
admin
test
test2
test1000
password
passme
meme
meraku
herko

(kali@kali)-[~/Passwords]
$ sS
```

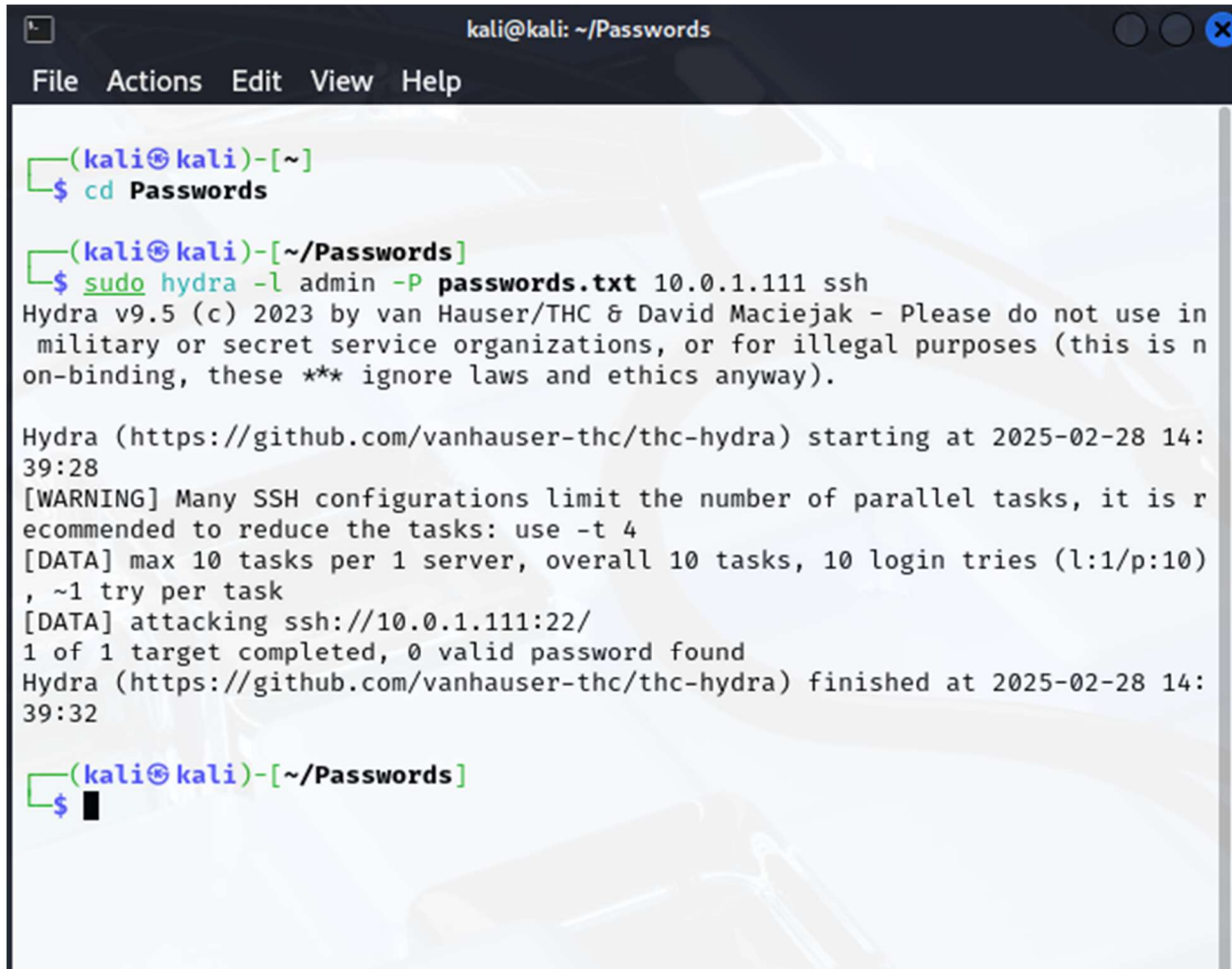
The image shows a terminal window titled 'kali@kali: ~/Passwords'. The window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal output shows the user running 'ls' which lists 'passwords.txt'. Then, the user runs 'cat passwords.txt' which displays a list of passwords: 'admin', 'test', 'test2', 'test1000', 'password', 'passme', 'meme', 'meraku', and 'herko'. Finally, the user runs 'sS' and the cursor is positioned after the command.

Figure 6 Dictionary List Folder

###Step 6: Perform the brute-force attack

"The following command attempts SSH login to the target machine using a dictionary attack:"

hydra -l admin(username) -P passwords.txt 10.0.1.111(target IP) ssh

A screenshot of a Kali Linux terminal window. The window title is 'kali@kali: ~/Passwords'. The menu bar shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows the following commands and output:

```
(kali@kali)-[~]  
$ cd Passwords  
  
(kali@kali)-[~/Passwords]  
$ sudo hydra -l admin -P passwords.txt 10.0.1.111 ssh  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in  
military or secret service organizations, or for illegal purposes (this is n  
on-binding, these *** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-28 14:  
39:28  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r  
ecommended to reduce the tasks: use -t 4  
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:1/p:10)  
, ~1 try per task  
[DATA] attacking ssh://10.0.1.111:22/  
1 of 1 target completed, 0 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-28 14:  
39:32  
  
(kali@kali)-[~/Passwords]  
$
```

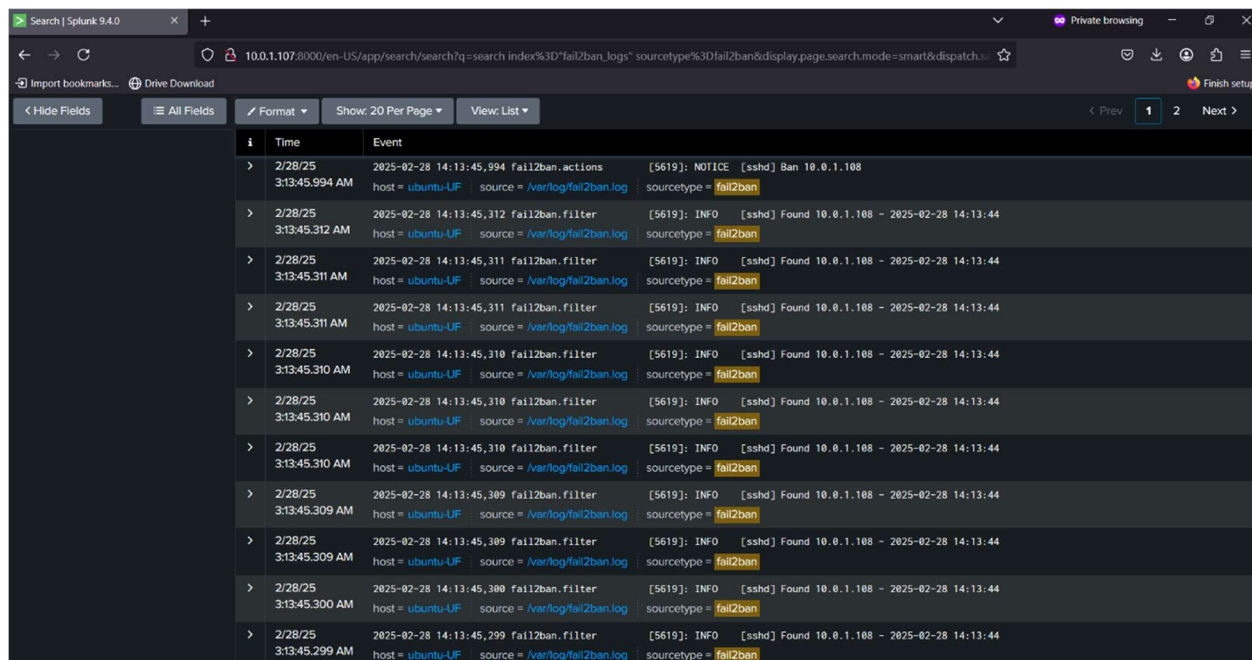
Figure 7 Brute-force simulate

###Step 7: Check the logs on Ubuntu VM (Victim PC)

2025-02-28 14:13:45,994 fail2ban.actions	[5619]: NOTICE	[sshd] Ban 10.0.1.108
2025-02-28 14:13:47,518 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,518 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,519 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,519 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,519 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,520 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,520 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,520 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47
2025-02-28 14:13:47,520 fail2ban.filter	[5619]: INFO	[sshd] Found 10.0.1.108 - 2025-02-28 14:13:47

Figure 8 Ban notice logs on ubuntu

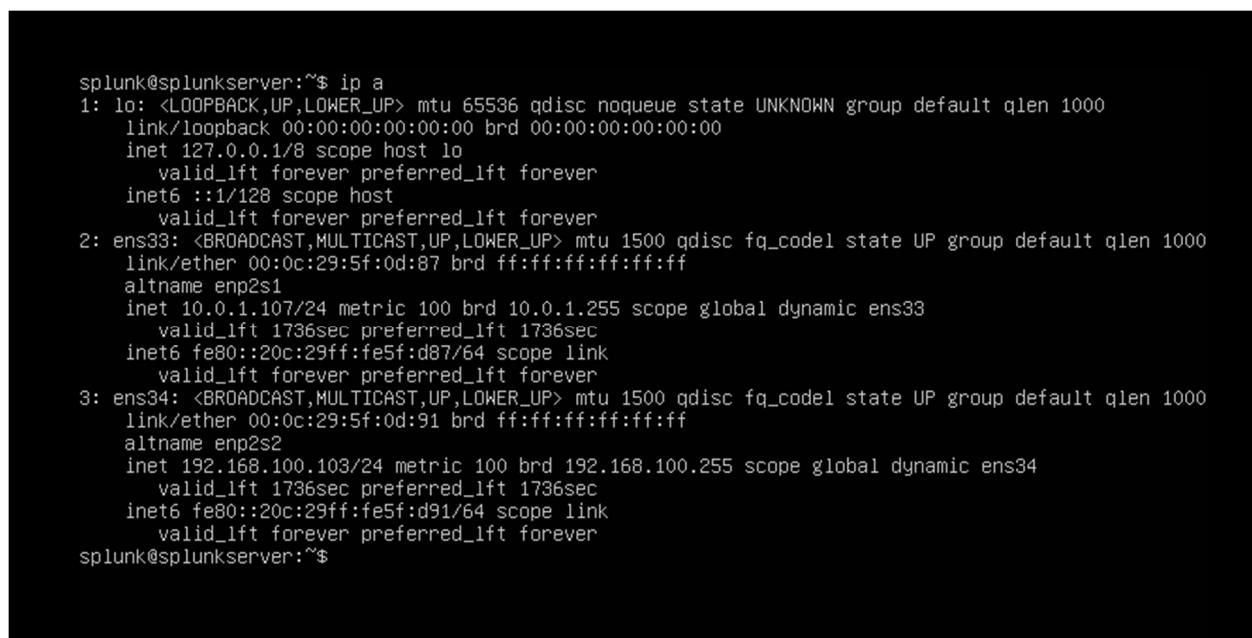
###Step 8: Check the logs on Splunk Dashboard



The screenshot shows the Splunk Dashboard interface with a search bar at the top containing the query: `search index%3Dfail2ban_logs sourcetype%3Dfail2ban&display.page.search.mode=smart&dispatch...`. Below the search bar, there are tabs for 'Hide Fields', 'All Fields', 'Format', 'Show: 20 Per Page', and 'View: List'. The main table displays a list of events with columns for Time, Event, and sourcetype. The events are filtered to show only those with sourcetype 'fail2ban'.

Time	Event	sourcetype
2/28/25 3:13:45.994 AM	2025-02-28 14:13:45,994 fail2ban.actions [5619]: NOTICE [sshd] Ban 10.0.1.108 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.312 AM	2025-02-28 14:13:45,312 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.311 AM	2025-02-28 14:13:45,311 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.311 AM	2025-02-28 14:13:45,311 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.310 AM	2025-02-28 14:13:45,310 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.310 AM	2025-02-28 14:13:45,310 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.310 AM	2025-02-28 14:13:45,310 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.309 AM	2025-02-28 14:13:45,309 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.309 AM	2025-02-28 14:13:45,309 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.300 AM	2025-02-28 14:13:45,300 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban
2/28/25 3:13:45.299 AM	2025-02-28 14:13:45,299 fail2ban.filter [5619]: INFO [sshd] Found 10.0.1.108 - 2025-02-28 14:13:44 host = ubuntu-UF source = /var/log/fail2ban.log	fail2ban

Figure 9 Ban logs on Splunk Server



The screenshot shows a terminal window with the command `splunk@splunkserver:~$ ip a` and its output. The output displays the network configuration for the Splunk Server IP, including the loopback interface `lo` and the ethernet interface `ens33`.

```
splunk@splunkserver:~$ 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
        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:29:5f:0d:87 brd ff:ff:ff:ff:ff:ff
    altnam enp2s1
    inet 10.0.1.107/24 metric 100 brd 10.0.1.255 scope global dynamic ens33
        valid_lft 1736sec preferred_lft 1736sec
    inet6 fe80::20c:29ff:fe5f:d87/64 scope link
        valid_lft forever preferred_lft forever
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:5f:0d:91 brd ff:ff:ff:ff:ff:ff
    altnam enp2s2
    inet 192.168.100.103/24 metric 100 brd 192.168.100.255 scope global dynamic ens34
        valid_lft 1736sec preferred_lft 1736sec
    inet6 fe80::20c:29ff:fe5f:d91/64 scope link
        valid_lft forever preferred_lft forever
splunk@splunkserver:~$
```

Figure 10 Splunk Server IP

Conclusion:

This project successfully implemented Fail2Ban to detect and prevent brute-force attacks on an Ubuntu system. The logs were analyzed in Splunk, confirming that the attacker IP was automatically banned after multiple failed login attempts.