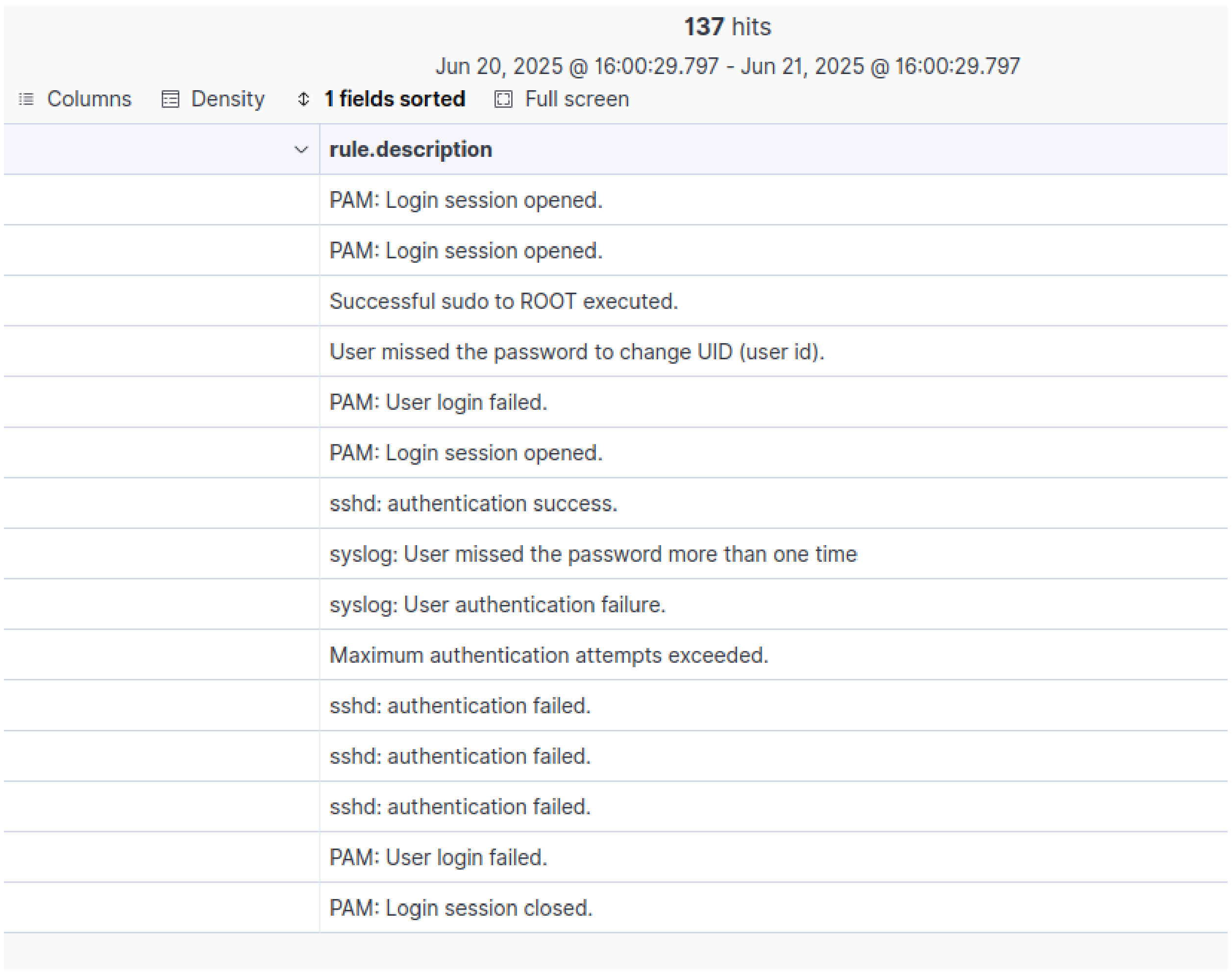
Defending attacks

Brute Force Protection on ssh.

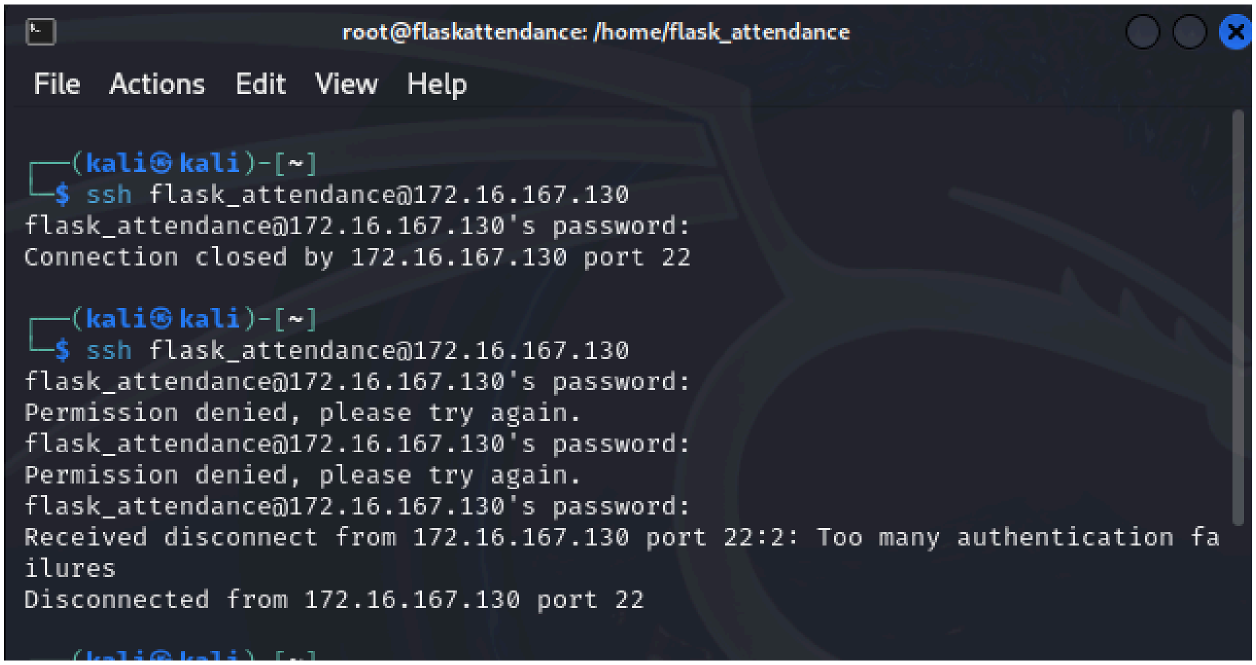


137 hits

Jun 20, 2025 @ 16:00:29.797 - Jun 21, 2025 @ 16:00:29.797

Columns Density **1 fields sorted** Full screen

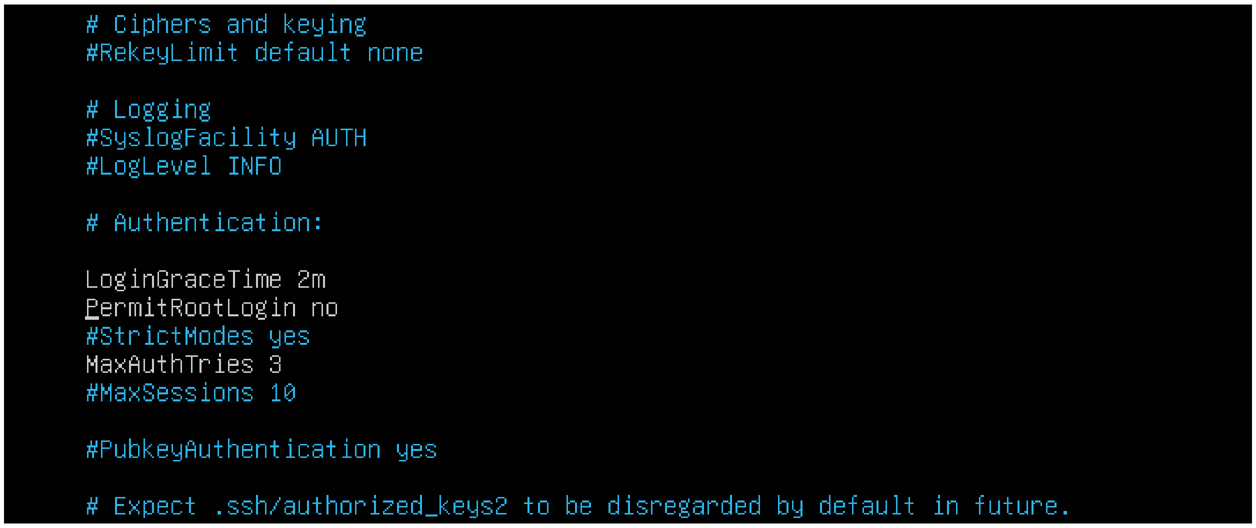
	rule.description
	PAM: Login session opened.
	PAM: Login session opened.
	Successful sudo to ROOT executed.
	User missed the password to change UID (user id).
	PAM: User login failed.
	PAM: Login session opened.
	sshd: authentication success.
	syslog: User missed the password more than one time
	syslog: User authentication failure.
	Maximum authentication attempts exceeded.
	sshd: authentication failed.
	sshd: authentication failed.
	sshd: authentication failed.
	PAM: User login failed.
	PAM: Login session closed.


root@flaskattendance: /home/flask_attendance
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ ssh flask_attendance@172.16.167.130
flask_attendance@172.16.167.130's password:
Connection closed by 172.16.167.130 port 22

┌──(kali㉿kali)-[~]
└─$ ssh flask_attendance@172.16.167.130
flask_attendance@172.16.167.130's password:
Permission denied, please try again.
flask_attendance@172.16.167.130's password:
Permission denied, please try again.
flask_attendance@172.16.167.130's password:
Received disconnect from 172.16.167.130 port 22:2: Too many authentication fa
ilures
Disconnected from 172.16.167.130 port 22
```

Here we can see that i have exceeded the limit of password attempts and it is clearly seen in the wazuh dashboard.

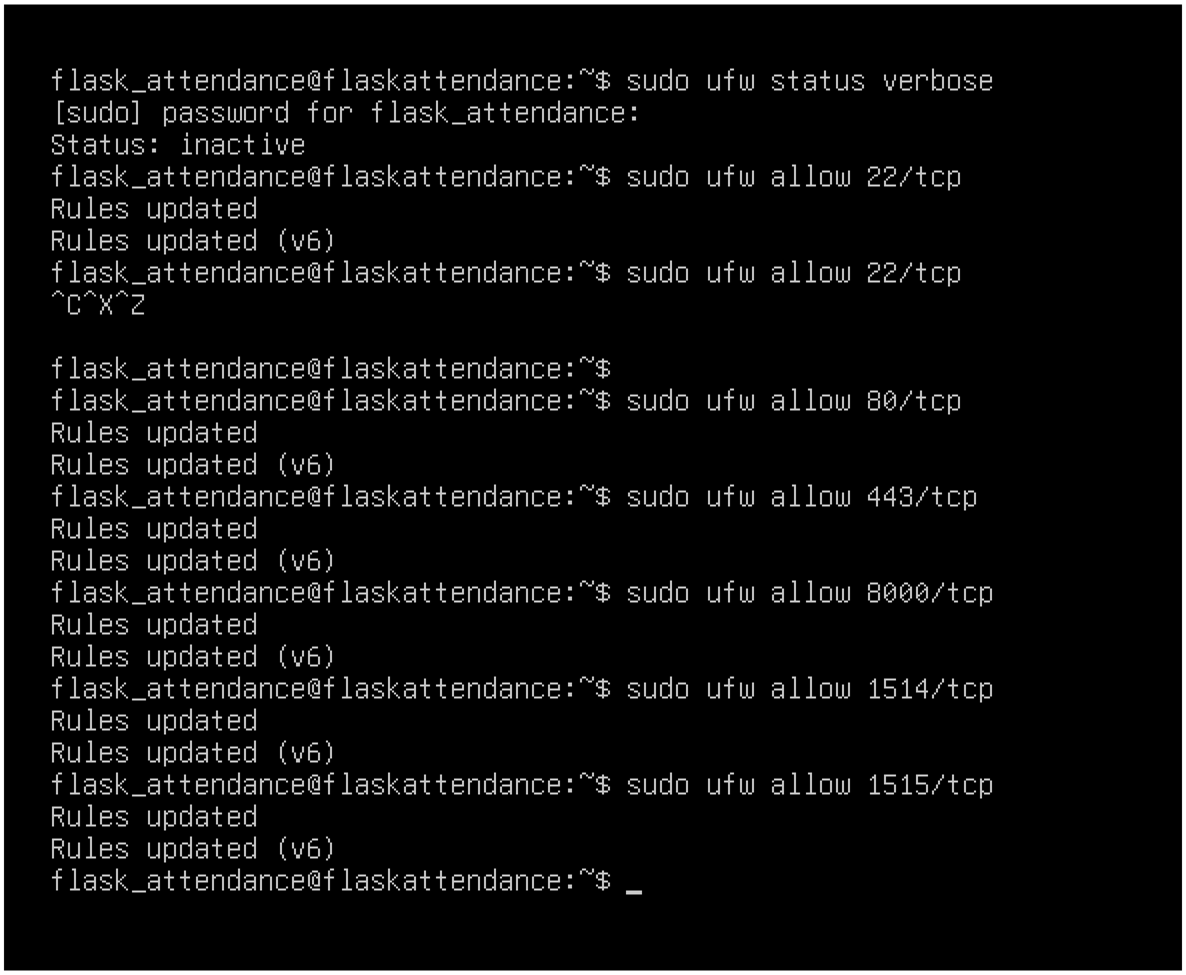
I also updated the sshd_config file for further security



```
# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
```

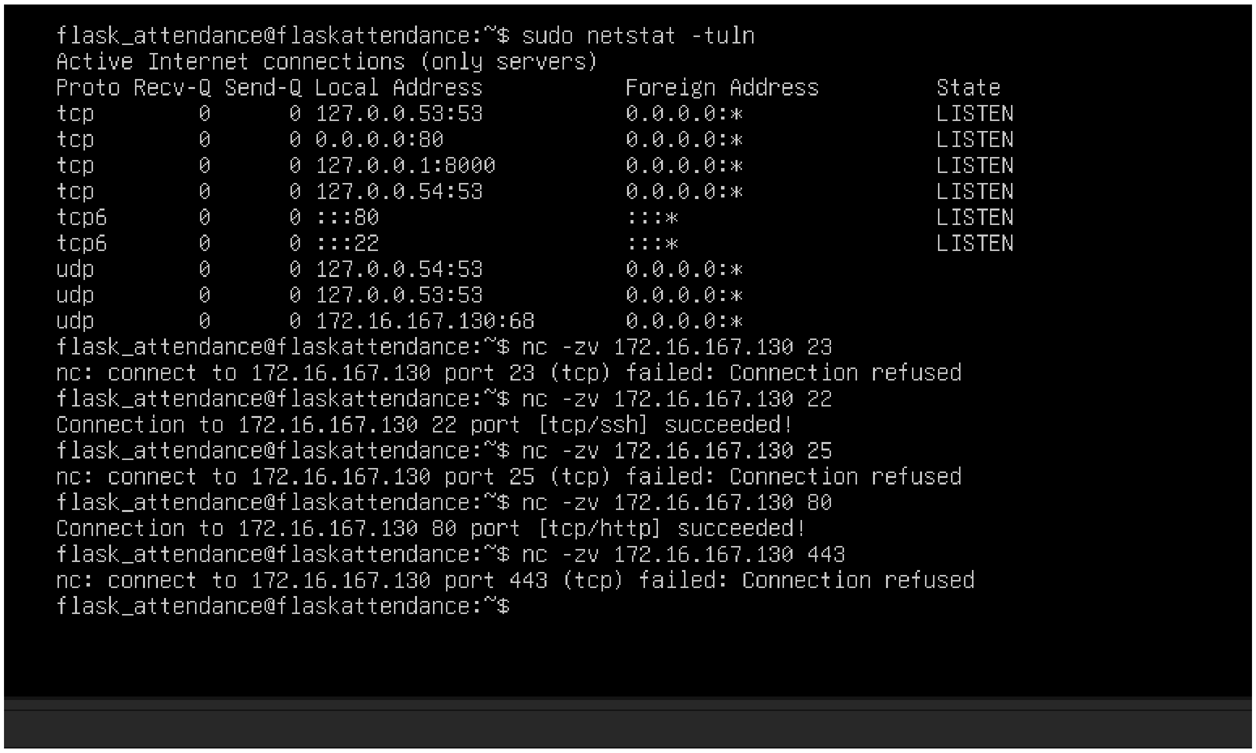
Configuring Firewall to Limit Traffic (UFW: Uncomplicated Firewall)

Blocking unwanted traffic can reduce the attack surface



```
flask_attendance@flaskattendance:~$ sudo ufw status verbose
[sudo] password for flask_attendance:
Status: inactive
flask_attendance@flaskattendance:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
flask_attendance@flaskattendance:~$ sudo ufw allow 22/tcp
^C^X^Z

flask_attendance@flaskattendance:~$
flask_attendance@flaskattendance:~$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
flask_attendance@flaskattendance:~$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
flask_attendance@flaskattendance:~$ sudo ufw allow 8000/tcp
Rules updated
Rules updated (v6)
flask_attendance@flaskattendance:~$ sudo ufw allow 1514/tcp
Rules updated
Rules updated (v6)
flask_attendance@flaskattendance:~$ sudo ufw allow 1515/tcp
Rules updated
Rules updated (v6)
flask_attendance@flaskattendance:~$ _
```

Here you can see the updated rules of firewall.

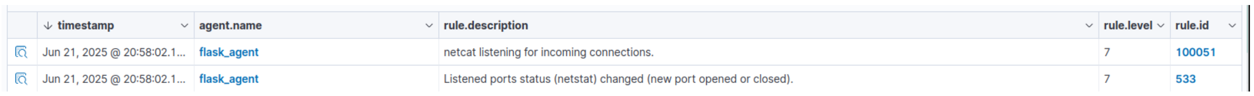


```
flask_attendance@flaskattendance:~$ sudo netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
udp        0      0 127.0.0.54:53           0.0.0.0:*
udp        0      0 127.0.0.53:53           0.0.0.0:*
udp        0      0 172.16.167.130:68       0.0.0.0:*
flask_attendance@flaskattendance:~$ nc -zv 172.16.167.130 23
nc: connect to 172.16.167.130 port 23 (tcp) failed: Connection refused
flask_attendance@flaskattendance:~$ nc -zv 172.16.167.130 22
Connection to 172.16.167.130 22 port [tcp/ssh] succeeded!
flask_attendance@flaskattendance:~$ nc -zv 172.16.167.130 25
nc: connect to 172.16.167.130 port 25 (tcp) failed: Connection refused
flask_attendance@flaskattendance:~$ nc -zv 172.16.167.130 80
Connection to 172.16.167.130 80 port [tcp/http] succeeded!
flask_attendance@flaskattendance:~$ nc -zv 172.16.167.130 443
nc: connect to 172.16.167.130 port 443 (tcp) failed: Connection refused
flask_attendance@flaskattendance:~$
```

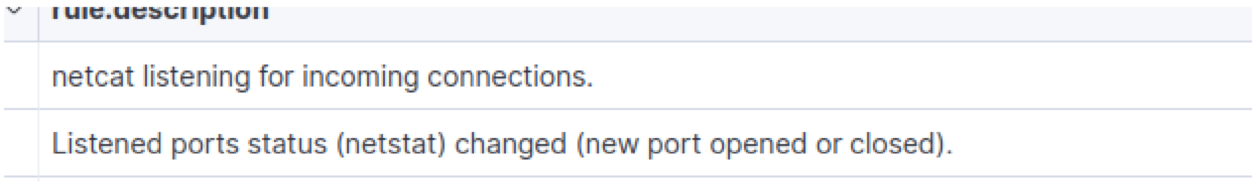
Here you can see the specific ports are only allowed and some are also refused.

Detecting unauthorized processes

You can check wazuh official documentation for this.
Here is the Link.



timestamp	agent.name	rule.description	rule.level	rule.id
Jun 21, 2025 @ 20:58:02.1...	flask_agent	netcat listening for incoming connections.	7	100051
Jun 21, 2025 @ 20:58:02.1...	flask_agent	Listened ports status (netstat) changed (new port opened or closed).	7	533



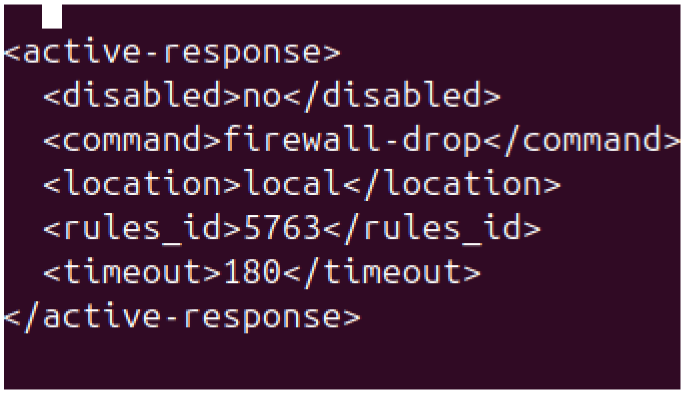
rule.description
netcat listening for incoming connections.
Listened ports status (netstat) changed (new port opened or closed).

Here you can see the netcat starts listening for incoming connections.

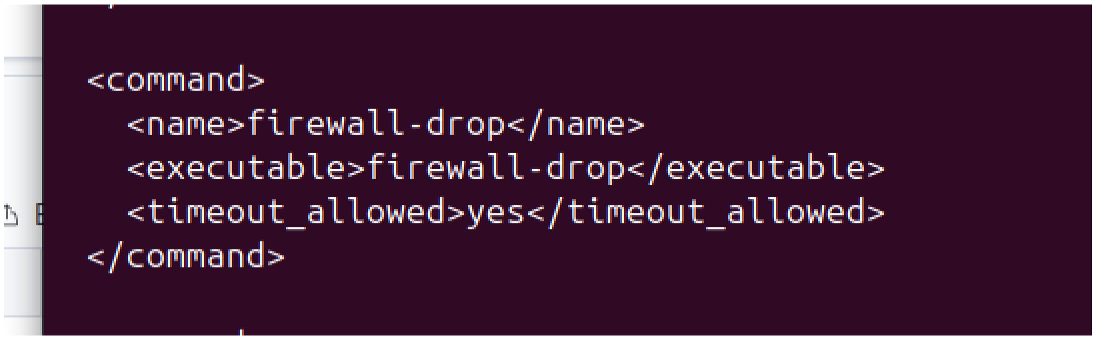
Active Response

Now what i am going to do is write a incident response script that will block malicious ip address that is trying to brute force my system. Here you can see i have written all the required code in the **server** configuration file.

`/var/ossec/etc/ossec.conf`



```
<active-response>
  <disabled>no</disabled>
  <command>firewall-drop</command>
  <location>local</location>
  <rules_id>5763</rules_id>
  <timeout>180</timeout>
</active-response>
```

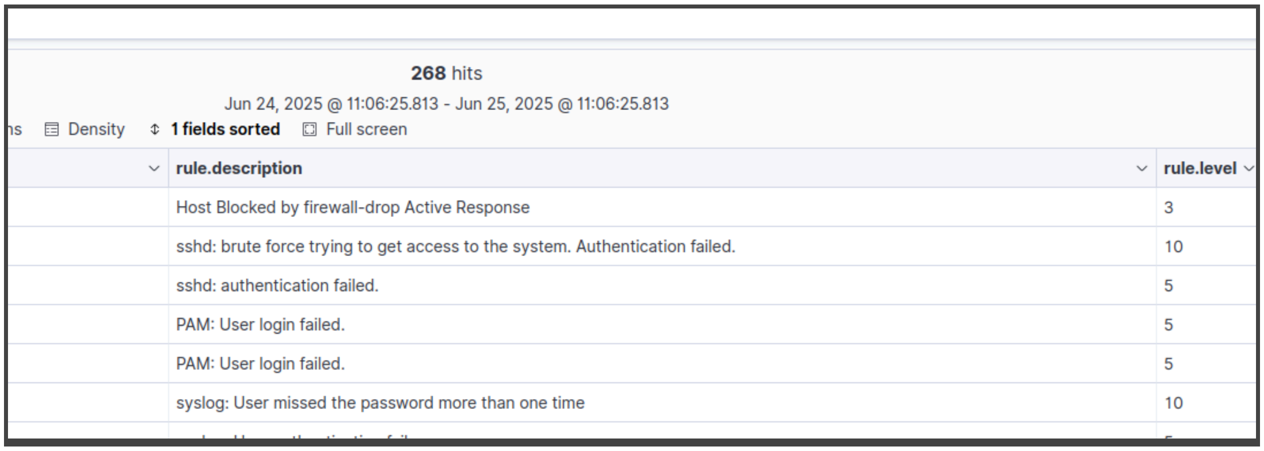


```
  <command>
    <name>firewall-drop</name>
    <executable>firewall-drop</executable>
    <timeout_allowed>yes</timeout_allowed>
  </command>
```

Now after that restart the wazuh manager

`sudo systemctl restart wazuh-manager`

Here you can see i after configuring the conf file, it has successfully blocked the host using the firewall-drop Active response.



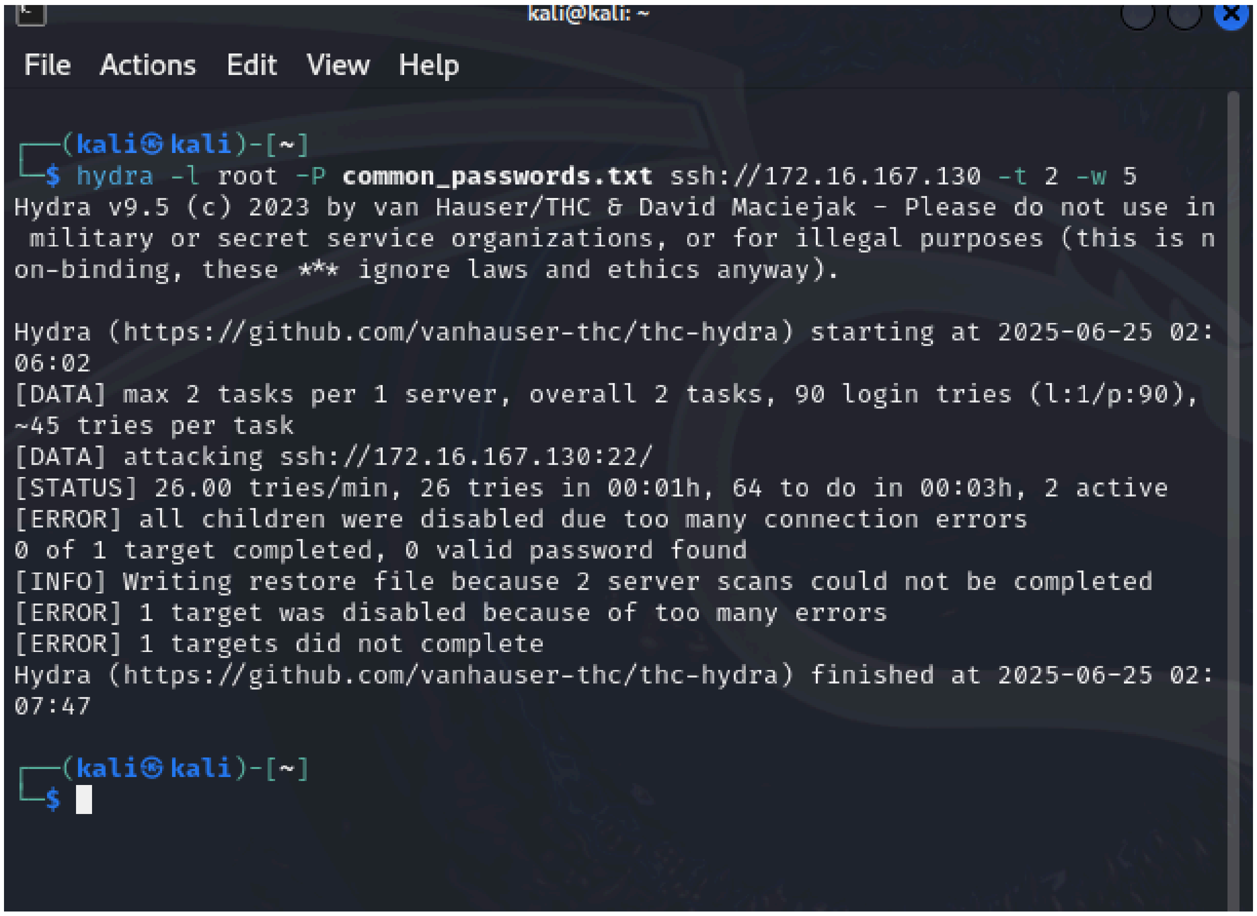
268 hits

Jun 24, 2025 @ 11:06:25.813 - Jun 25, 2025 @ 11:06:25.813

Density   1 fields sorted   Full screen

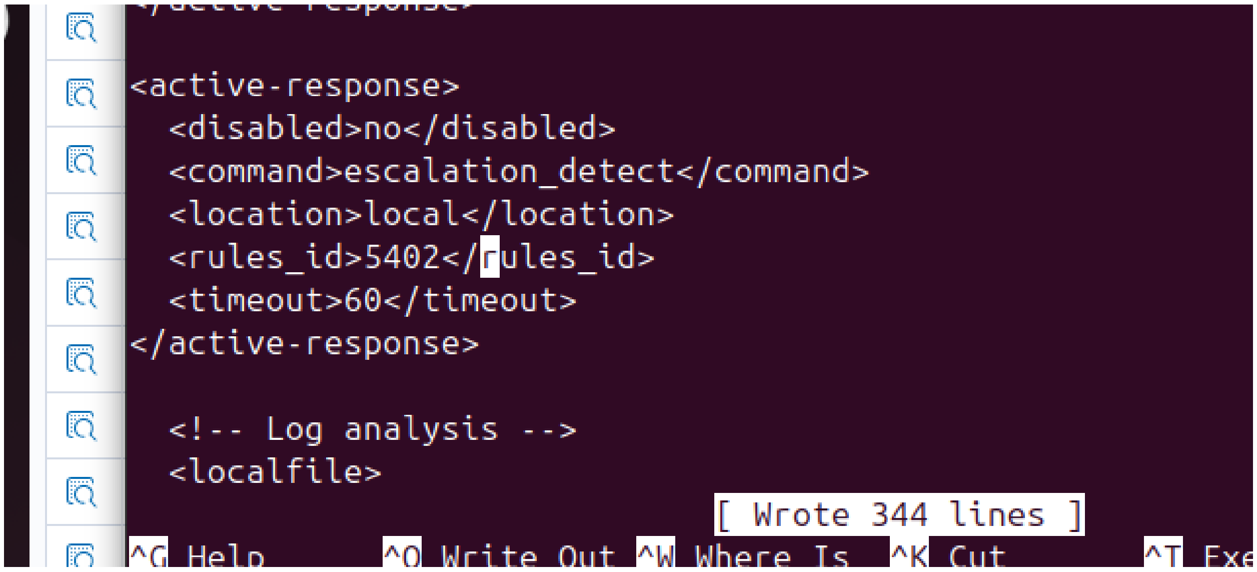
rule.description	rule.level
Host Blocked by firewall-drop Active Response	3
sshd: brute force trying to get access to the system. Authentication failed.	10
sshd: authentication failed.	5
PAM: User login failed.	5
PAM: User login failed.	5
syslog: User missed the password more than one time	10

And you can see in the hydra output, that target didn't completed because of host blockage by firewall-drop Active Response.

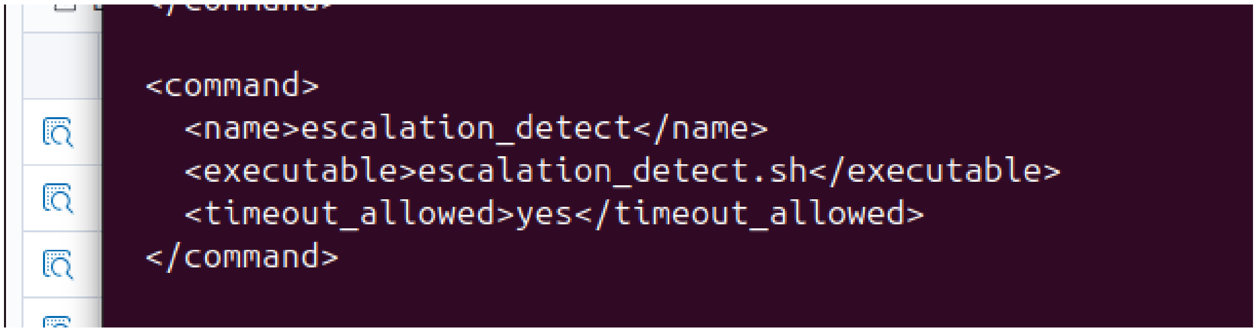


```
┌──(kali㉿kali)-[~]
└─$ hydra -l root -P common_passwords.txt ssh://172.16.167.130 -t 2 -w 5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-25 02:
06:02
[DATA] max 2 tasks per 1 server, overall 2 tasks, 90 login tries (l:1/p:90),
~45 tries per task
[DATA] attacking ssh://172.16.167.130:22/
[STATUS] 26.00 tries/min, 26 tries in 00:01h, 64 to do in 00:03h, 2 active
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
[INFO] Writing restore file because 2 server scans could not be completed
[ERROR] 1 target was disabled because of too many errors
[ERROR] 1 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-25 02:
07:47

┌──(kali㉿kali)-[~]
└─$ 
```

Adding a custom active response script for privilege escalation.

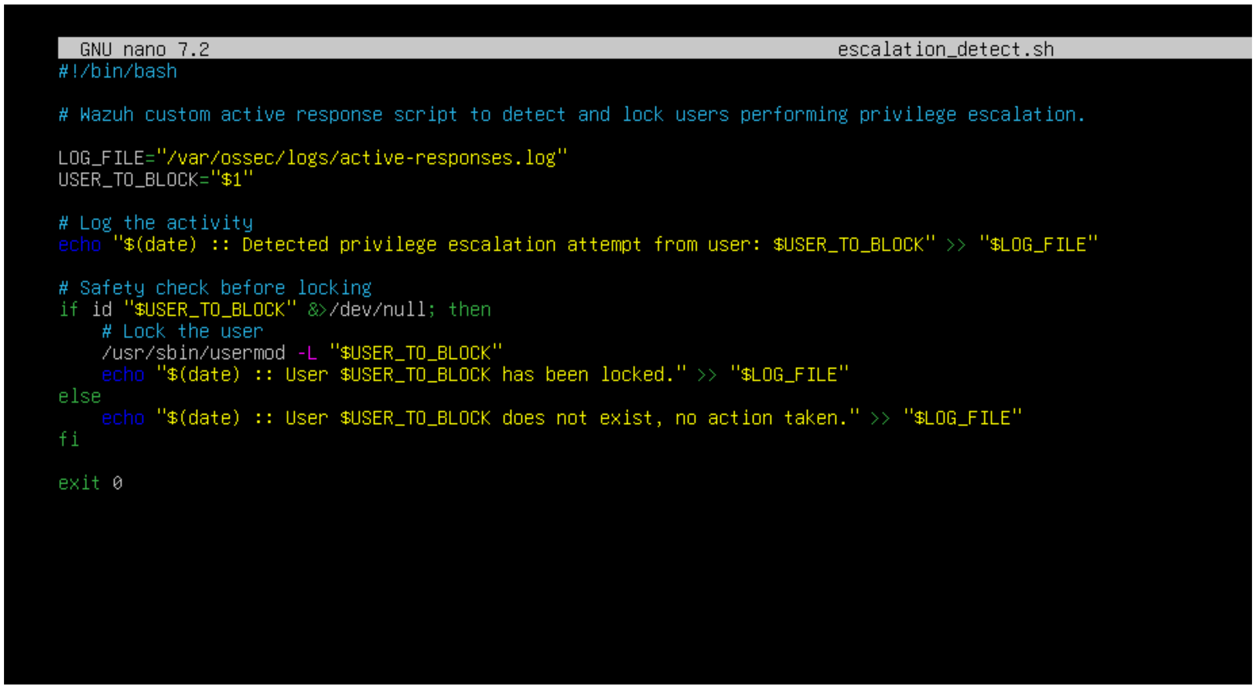


```
<active-response>
  <disabled>no</disabled>
  <command>escalation_detect</command>
  <location>local</location>
  <rules_id>5402</rules_id>
  <timeout>60</timeout>
</active-response>

  <!-- Log analysis -->
  <localfile>
                                        [ Wrote 344 lines ]
^G Help         ^O Write Out ^W Where Is  ^K Cut         ^T Exe
```



```
 <command>
   <name>escalation_detect</name>
   <executable>escalation_detect.sh</executable>
   <timeout_allowed>yes</timeout_allowed>
 </command>
```

After that I added the custom script in that location `/var/ossec/active-response/bin`



```
  GNU nano 7.2                                                                  escalation_detect.sh
#!/bin/bash

# Wazuh custom active response script to detect and lock users performing privilege escalation.

LOG_FILE="/var/ossec/logs/active-responses.log"
USER_TO_BLOCK="$1"

# Log the activity
echo "$(date) :: Detected privilege escalation attempt from user: $USER_TO_BLOCK" >> "$LOG_FILE"

# Safety check before locking
if id "$USER_TO_BLOCK" &>/dev/null; then
    # Lock the user
    /usr/sbin/usermod -L "$USER_TO_BLOCK"
    echo "$(date) :: User $USER_TO_BLOCK has been locked." >> "$LOG_FILE"
else
    echo "$(date) :: User $USER_TO_BLOCK does not exist, no action taken." >> "$LOG_FILE"
fi

exit 0
```

After that change the files permission as explained in the wazuh documentation.

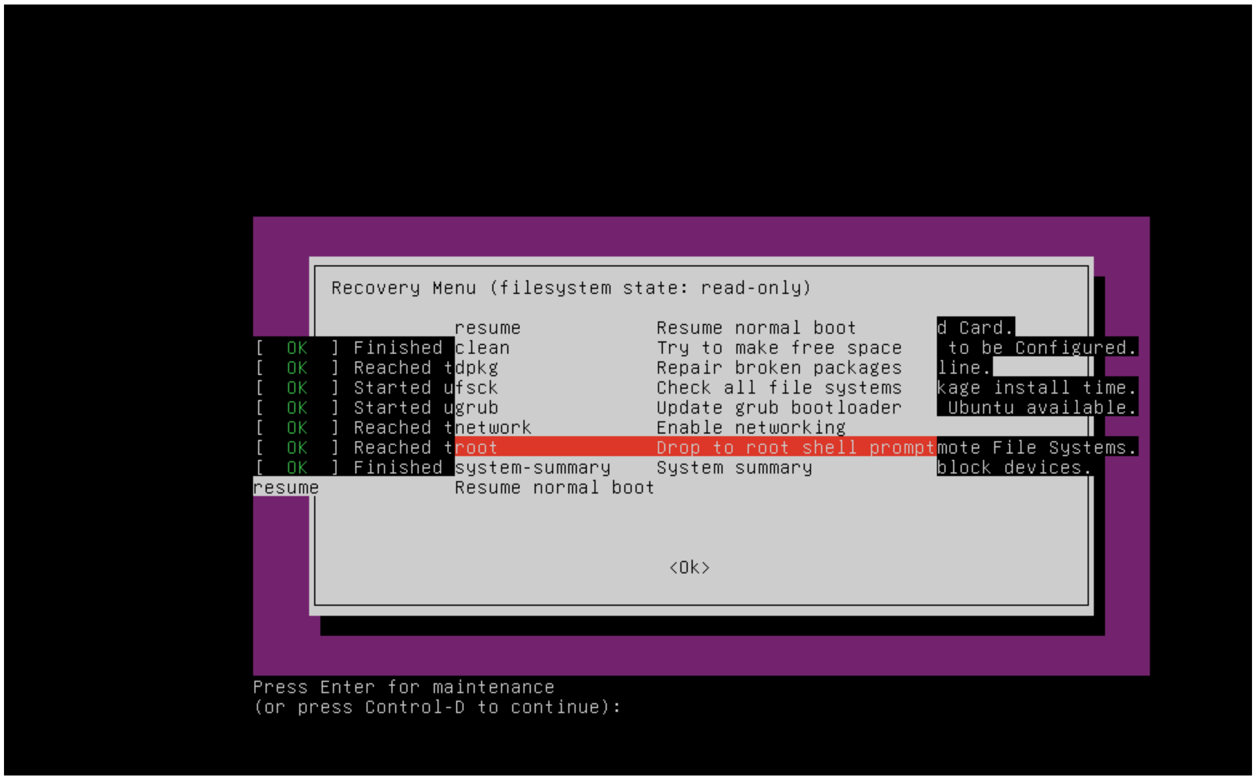
```
sudo chmod 750 /var/ossec/active-response/bin/escalation-detect.sh
sudo chown root:wazuh /var/ossec/active-response/bin/escalation-detect.sh
```

Now we are good to go,
Lets run it.

```
#!/bin/bash

LOG_FILE="/var/ossec/logs/active-responses.log"

echo "$(date) :: Emergency triggered - Rebooting system" >> "$LOG_FILE"

/sbin/shutdown -r now "Triggered by Wazuh active-response"

exit 0
```

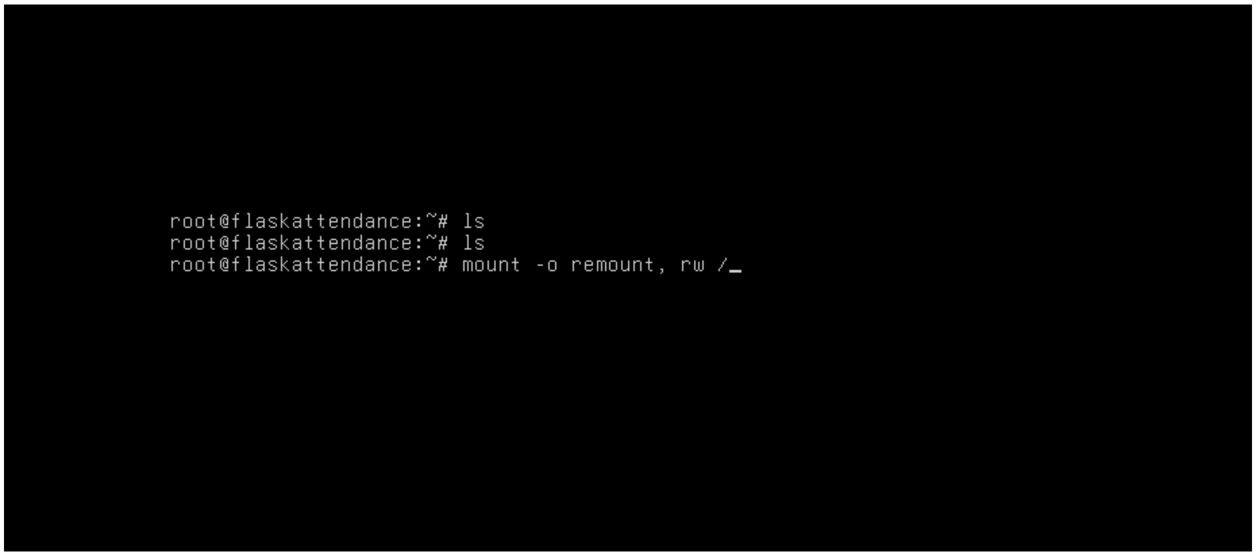
When i just logged in as root, the system shutdown instantly. Lets check if the logs are updated there

A funny thing happened ,
I also have written a system locked script previously, which did its work but i didnt knew that, and after rebooting the system, i am locked out.

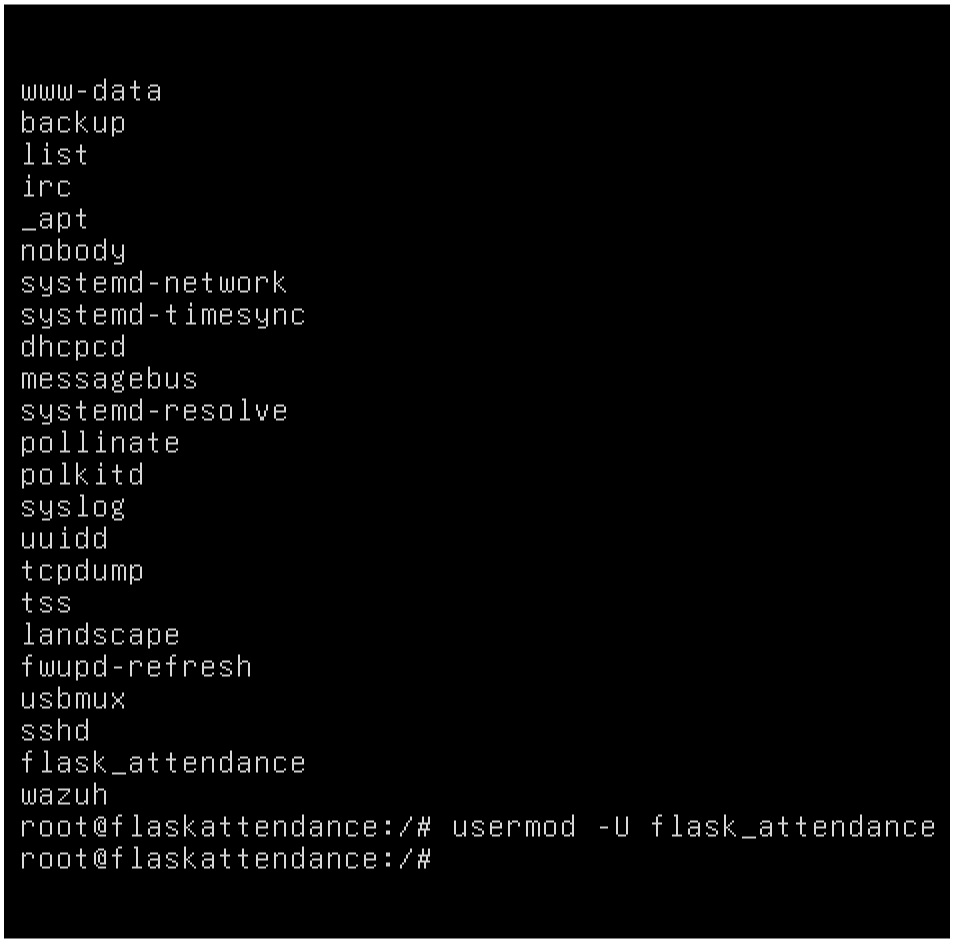


```
Recovery Menu (filesystem state: read-only)

                resume               Resume normal boot        d Card.
[  OK  ] Finished clean              Try to make free space     to be Configured.
[  OK  ] Reached tdpkg               Repair broken packages    line.
[  OK  ] Started ufsck               Check all file systems    kage install time.
[  OK  ] Started ugrub               Update grub bootloader     Ubuntu available.
[  OK  ] Reached tnetwork            Enable networking
[  OK  ] Reached troot               Drop to root shell promptmote File Systems.
[  OK  ] Finished system-summary     System summary            block devices.
resume          Resume normal boot

                              <Ok>

Press Enter for maintenance
(or press Control-D to continue):
```

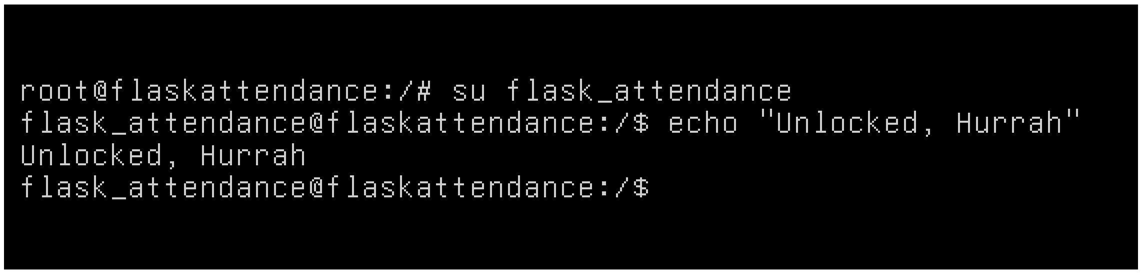
Now configuring it again using recovery mode.



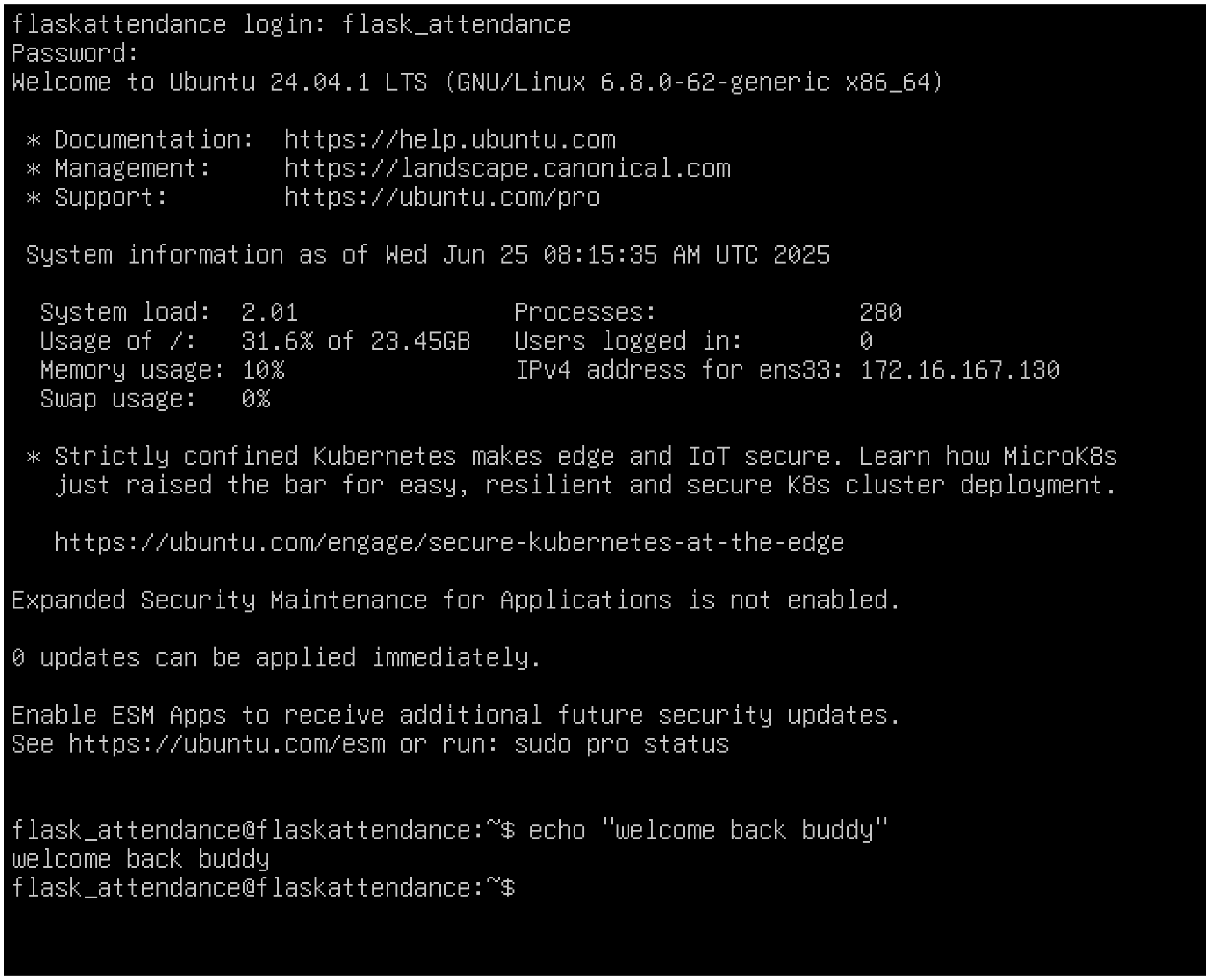
```
root@flaskattendance:~# ls
root@flaskattendance:~# ls
root@flaskattendance:~# mount -o remount, rw /_
```



```
www-data
backup
list
irc
_apt
nobody
systemd-network
systemd-timesync
dhcpcd
messagebus
systemd-resolve
pollinate
polkitd
syslog
uuidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
flask_attendance
wazuh
root@flaskattendance:/# usermod -U flask_attendance
root@flaskattendance:/#
```

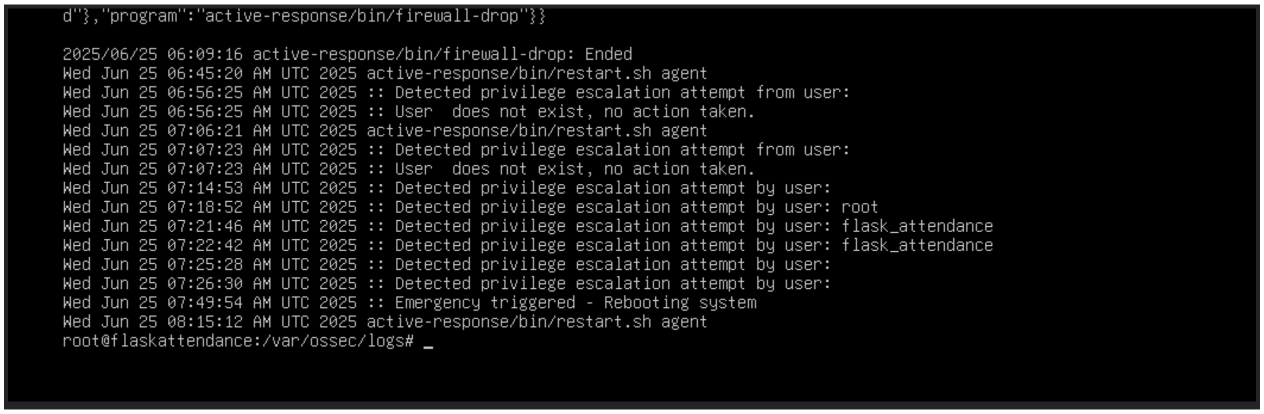


```
root@flaskattendance:/# su flask_attendance
flask_attendance@flaskattendance:/$ echo "Unlocked, Hurrah"
Unlocked, Hurrah
flask_attendance@flaskattendance:/$
```



```
flaskattendance login: flask_attendance
Password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Wed Jun 25 08:15:35 AM UTC 2025

  System load:  2.01               Processes:             280
  Usage of /:   31.6% of 23.45GB   Users logged in:       0
  Memory usage: 10%                IPv4 address for ens33: 172.16.167.130
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


flask_attendance@flaskattendance:~$ echo "welcome back buddy"
welcome back buddy
flask_attendance@flaskattendance:~$
```

Lets check the logs file
`/var/ossec/logs/active-responses.log`



```
d"},"program":"active-response/bin/firewall-drop"}}

2025/06/25 06:09:16 active-response/bin/firewall-drop: Ended
Wed Jun 25 06:45:20 AM UTC 2025 active-response/bin/restart.sh agent
Wed Jun 25 06:56:25 AM UTC 2025 :: Detected privilege escalation attempt from user:
Wed Jun 25 06:56:25 AM UTC 2025 :: User  does not exist, no action taken.
Wed Jun 25 07:06:21 AM UTC 2025 active-response/bin/restart.sh agent
Wed Jun 25 07:07:23 AM UTC 2025 :: Detected privilege escalation attempt from user:
Wed Jun 25 07:07:23 AM UTC 2025 :: User  does not exist, no action taken.
Wed Jun 25 07:14:53 AM UTC 2025 :: Detected privilege escalation attempt by user:
Wed Jun 25 07:18:52 AM UTC 2025 :: Detected privilege escalation attempt by user: root
Wed Jun 25 07:21:46 AM UTC 2025 :: Detected privilege escalation attempt by user: flask_attendance
Wed Jun 25 07:22:42 AM UTC 2025 :: Detected privilege escalation attempt by user: flask_attendance
Wed Jun 25 07:25:28 AM UTC 2025 :: Detected privilege escalation attempt by user:
Wed Jun 25 07:26:30 AM UTC 2025 :: Detected privilege escalation attempt by user:
Wed Jun 25 07:49:54 AM UTC 2025 :: Emergency triggered - Rebooting system
Wed Jun 25 08:15:12 AM UTC 2025 active-response/bin/restart.sh agent
root@flaskattendance:/var/ossec/logs# _
```

Now i can successfully see the logs, of what i just did with the system.