POC TASK 1

User & Permission Misconfigurations

Setup-

Create Multiple Users :

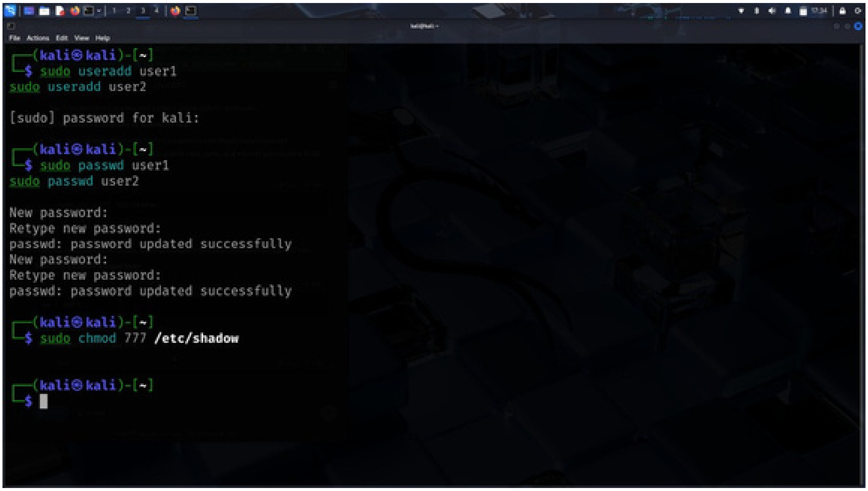
sudo useradd user1

sudo useradd user2

Set passwords for these Users :

sudo passwd user1

sudo passwd user2



```
┌──(kali㉿kali)-[~]
└─$ sudo useradd user1
sudo useradd user2

[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo passwd user1
sudo passwd user2

New password:
Retype new password:
passwd: password updated successfully
New password:
Retype new password:
passwd: password updated successfully

┌──(kali㉿kali)-[~]
└─$ sudo chmod 777 /etc/shadow


┌──(kali㉿kali)-[~]
└─$ ▌
```

Assign Incorrect Permissions to Sensitive Files:

The /etc/shadow file stores hashed passwords and should have restricted permissions. Assigning chmod 777 grants read, write, and execute permissions to all users, which is insecure: sudo chmod 777 /etc/shadow

Exploit-

With the misconfigured permissions, a low-privileged user can access sensitive system files:

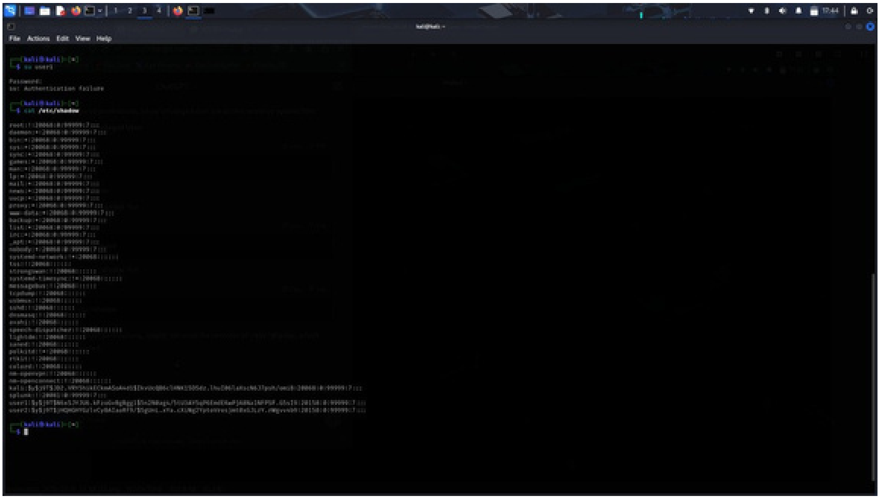
Switch to a Low-Privileged User:
su - user1

Access Sensitive Files:

View the /etc/passwd file:
cat /etc/passwd

View the /etc/shadow file:
cat /etc/shadow

Due to the improper permissions, user1 can read the contents of /etc/shadow, which should be restricted.



```
┌──(kali㉿kali)-[~]
└─$ su user1
Password:
su: Authentication failure

┌──(kali㉿kali)-[~]
└─$ cat /etc/shadow
root:!:20068:0:99999:7:::
daemon:*:20068:0:99999:7:::
bin:*:20068:0:99999:7:::
sys:*:20068:0:99999:7:::
sync:*:20068:0:99999:7:::
games:*:20068:0:99999:7:::
man:*:20068:0:99999:7:::
lp:*:20068:0:99999:7:::
mail:*:20068:0:99999:7:::
news:*:20068:0:99999:7:::
uucp:*:20068:0:99999:7:::
proxy:*:20068:0:99999:7:::
www-data:*:20068:0:99999:7:::
backup:*:20068:0:99999:7:::
list:*:20068:0:99999:7:::
irc:*:20068:0:99999:7:::
_apt:*:20068:0:99999:7:::
nobody:*:20068:0:99999:7:::
systemd-network:!*:20068::::::
tss:!:20068::::::
strongswan:!:20068::::::
systemd-timesync:!*:20068::::::
messagebus:!:20068::::::
tcpdump:!:20068::::::
usbmux:!:20068::::::
sshd:!:20068::::::
dnsmasq:!:20068::::::
avahi:!:20068::::::
speech-dispatcher:!:20068::::::
lightdm:!:20068::::::
saned:!:20068::::::
polkitd:!*:20068::::::
rtkit:!:20068::::::
colord:!:20068::::::
nm-openvpn:!:20068::::::
nm-openconnect:!:20068::::::
kali:$y$j9T$[illegible]:20068:0:99999:7:::
splunk:!:20085:0:99999:7:::
user1:$y$j9T$[illegible]:20158:0:99999:7:::
user2:$y$j9T$[illegible]:20158:0:99999:7:::

┌──(kali㉿kali)-[~]
└─$ 
```

Mitigation-

Restrict Permissions on Sensitive Files:

Set appropriate permissions for /etc/shadow:

sudo chmod 640 /etc/shadow

Verify the permissions:

ls -l /etc/shadow

The output should indicate that the file is readable and writable by the owner (root) and readable by the group (shadow), with no permissions for others.

Ensure Correct Ownership:

Set the owner and group for /etc/shadow:

sudo chown root:shadow /etc/shadow

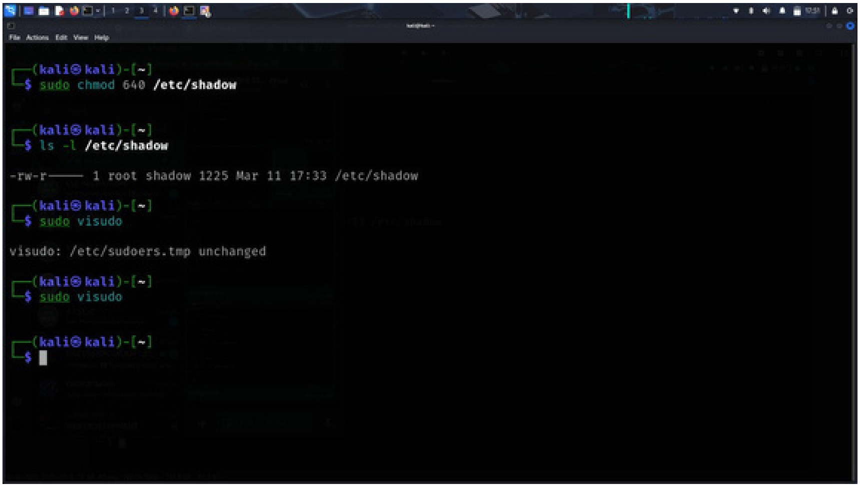
Configure Proper sudo Privileges: Edit the

sudoers file to grant specific permissions: sudo

visudo

Add or modify lines to ensure only authorized users have elevated privileges. For example, to grant user1 specific permissions:

user1 ALL=(ALL) /usr/bin/apt-get



```
┌──(kali㉿kali)-[~]
└─$ sudo chmod 640 /etc/shadow

┌──(kali㉿kali)-[~]
└─$ ls -l /etc/shadow

-rw-r----- 1 root shadow 1225 Mar 11 17:33 /etc/shadow

┌──(kali㉿kali)-[~]
└─$ sudo visudo

visudo: /etc/sudoers.tmp unchanged

┌──(kali㉿kali)-[~]
└─$ sudo visudo

┌──(kali㉿kali)-[~]
└─$ 
```

POC TASK 2

Remote Access & SSH Hardening

Setup-

Enable SSH with Root Login & Password Authentication :

Install/Open SSH Server(if not already installed)

```
sudo apt update && sudo apt install openssh-server -y
sudo systemctl enable --now ssh
```

Edit SSH Configuration:

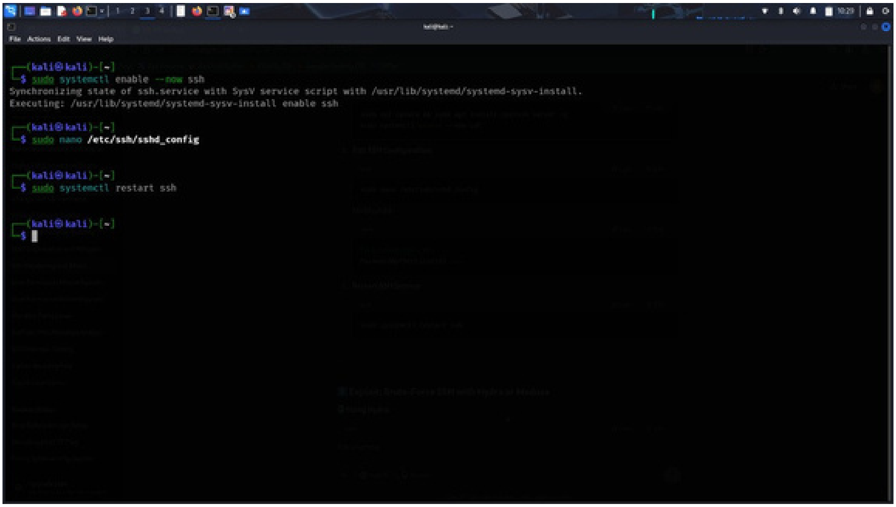
```
sudo nano /etc/ssh/sshd_config
```

Now modify these changes in nano:

Modify/Add:

PermitRootLogin yes

PasswordAuthentication yes



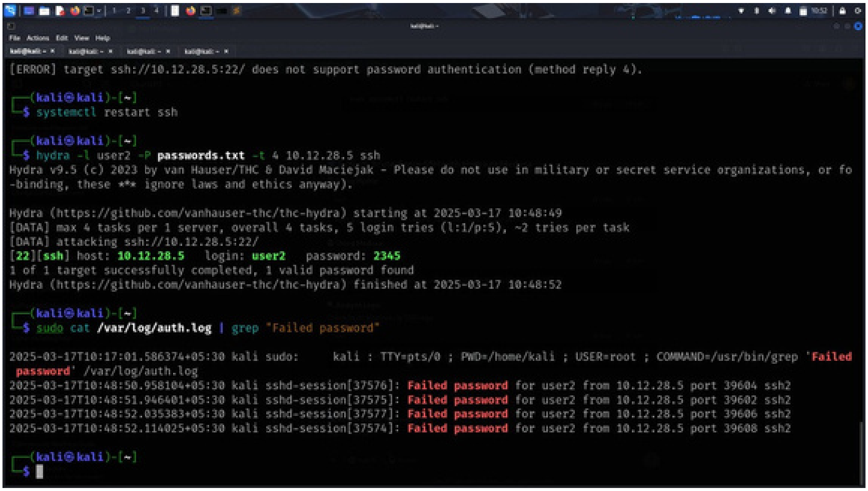
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh

┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/ssh/sshd_config

┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart ssh

┌──(kali㉿kali)-[~]
└─$ 
```

Restart SSH Service:

sudo systemctl restart ssh

Exploit-

Brute-Force SSH with Hydra or Medusa:



```
[ERROR] target ssh://10.12.28.5:22/ does not support password authentication (method reply 4).

┌──(kali㉿kali)-[~]
└─$ systemctl restart ssh

┌──(kali㉿kali)-[~]
└─$ hydra -l user2 -P passwords.txt -t 4 10.12.28.5 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or fo
-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-17 10:48:49
[DATA] max 4 tasks per 1 server, overall 4 tasks, 5 login tries (l:1/p:5), ~2 tries per task
[DATA] attacking ssh://10.12.28.5:22/
[22][ssh] host: 10.12.28.5   login: user2   password: 2345
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-17 10:48:52

┌──(kali㉿kali)-[~]
└─$ sudo cat /var/log/auth.log | grep "Failed password"

2025-03-17T10:17:01.586374+05:30 kali sudo:     kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep 'Failed
 password' /var/log/auth.log
2025-03-17T10:48:50.958104+05:30 kali sshd-session[37576]: Failed password for user2 from 10.12.28.5 port 39604 ssh2
2025-03-17T10:48:51.946401+05:30 kali sshd-session[37575]: Failed password for user2 from 10.12.28.5 port 39602 ssh2
2025-03-17T10:48:52.035383+05:30 kali sshd-session[37577]: Failed password for user2 from 10.12.28.5 port 39606 ssh2
2025-03-17T10:48:52.114025+05:30 kali sshd-session[37574]: Failed password for user2 from 10.12.28.5 port 39608 ssh2

┌──(kali㉿kali)-[~]
└─$ 
```

Using Hydra:

hydra -l username -P password_list.txt -t number of tries ssh

Using Medusa:

medusa -h -u root -P password_list.txt -M ssh

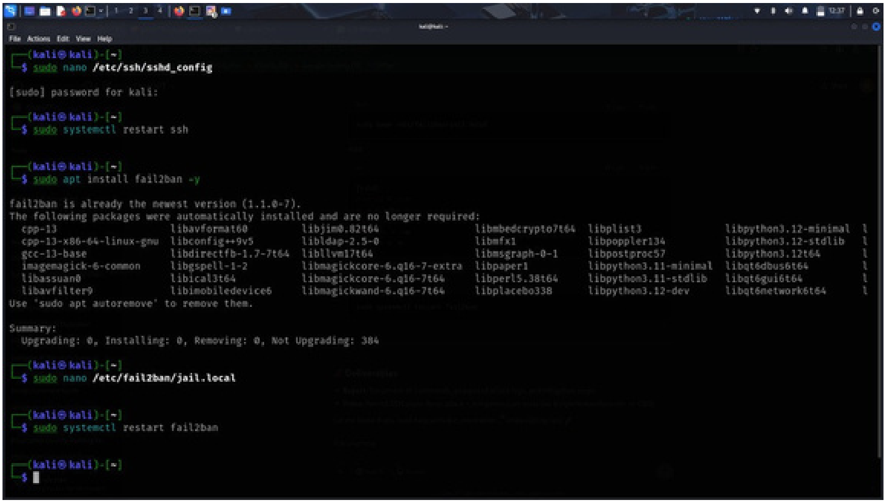
Analyze Logs and check login attempts in SSH logs:

sudo cat /var/log/auth.log | grep "Failed password"

Mitigation:

Secure SSH

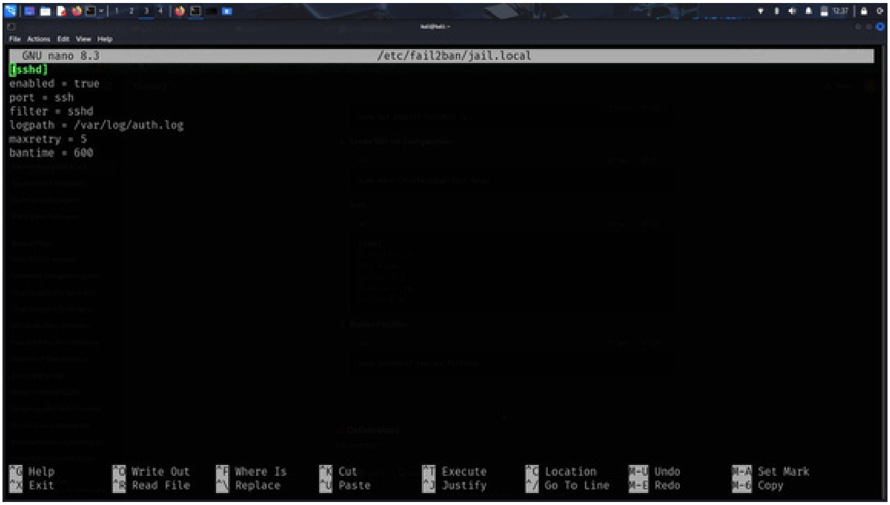
Disable Root Login & Enforce Key-Based Authentication



```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/ssh/sshd_config

[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart ssh


┌──(kali㉿kali)-[~]
└─$ sudo apt install fail2ban -y

fail2ban is already the newest version (1.1.0-7).
The following packages were automatically installed and are no longer required:
  cpp-13                  libavformat60          libjim0.82t64              libmbedcrypto7t64  libplist3              libpython3.12-minimal  l
  cpp-13-x86-64-linux-gnu libconfig++9v5         libldap-2.5-0              libmfx1            libpoppler134          libpython3.12-stdlib   l
  gcc-13-base             libdirectfb-1.7-7t64   libllvm17t64               libmsgraph-0-1     libpostproc57          libpython3.12t64       l
  imagemagick-6-common    libgspell-1-2          libmagickcore-6.q16-7-extra libpaper1         libpython3.11-minimal  libqt6dbus6t64         l
  libassuan0              libical3t64            libmagickcore-6.q16-7t64   libperl5.38t64     libpython3.11-stdlib   libqt6gui6t64          l
  libavfilter9            libimobiledevice6      libmagickwand-6.q16-7t64   libplacebo338      libpython3.12-dev      libqt6network6t64      l
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 384

┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/fail2ban/jail.local


┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart fail2ban


┌──(kali㉿kali)-[~]
└─$ █
```

Modify and edit SSH Config:

sudo nano /etc/ssh/sshd_config



```
  GNU nano 8.3                                  /etc/fail2ban/jail.local
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 600
```

PermitRootLogin no

PasswordAuthentication no

Restart SSH Service:

```
sudo systemctl restart ssh
```

Configure Fail2Ban to Block Brute-Force Attempts

Install Fail2Ban :

```
sudo apt install fail2ban -y
```

Create SSH Jail Configuration:

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

Add:

```
[sshd]

enabled = true

port = ssh

maxretry = 3

findtime = 10m

bantime = 1h
```

Restart Fail2Ban:

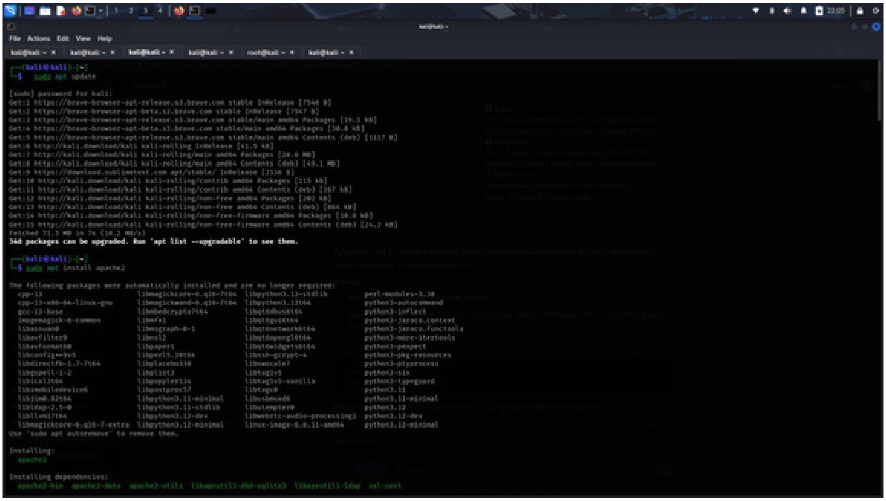
```
sudo systemctl restart fail2ban
```

POC TASK 3

Firewall & Network Security

Setup-

Install and Configure Apache Web Server:

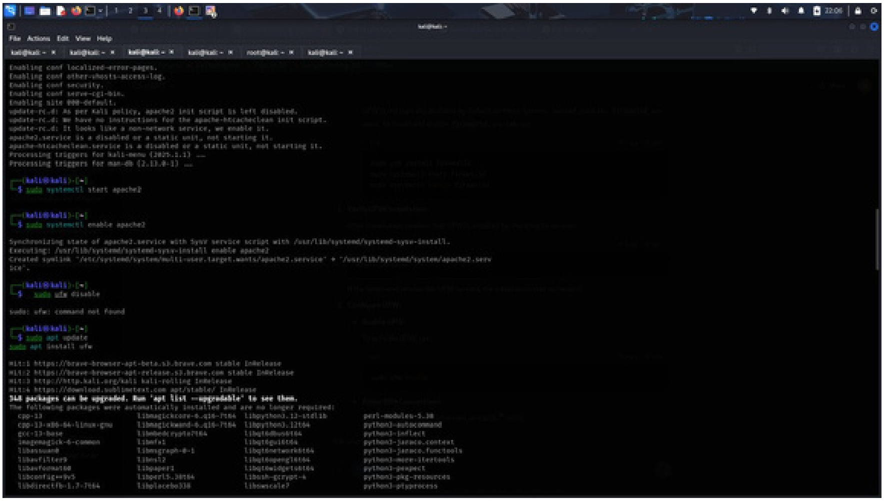


```
┌──(kali㉿kali)-[~]
└─$ sudo apt update

[sudo] password for kali:
Get:1 https://brave-browser-apt-release.s3.brave.com stable InRelease [7546 B]
Get:2 https://brave-browser-apt-beta.s3.brave.com stable InRelease [7547 B]
Get:3 https://brave-browser-apt-release.s3.brave.com stable/main amd64 Packages [19.3 kB]
Get:4 https://brave-browser-apt-beta.s3.brave.com stable/main amd64 Packages [30.0 kB]
Get:5 https://brave-browser-apt-release.s3.brave.com stable/main amd64 Contents (deb) [1117 B]
Get:6 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:7 http://kali.download/kali kali-rolling/main amd64 Packages [20.6 MB]
Get:8 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [49.1 MB]
Get:9 https://download.sublimetext.com apt/stable/ InRelease [2536 B]
Get:10 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:11 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [267 kB]
Get:12 http://kali.download/kali kali-rolling/non-free amd64 Packages [202 kB]
Get:13 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [886 kB]
Get:14 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]
Get:15 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [24.3 kB]
Fetched 71.3 MB in 7s (10.2 MB/s)
348 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(kali㉿kali)-[~]
└─$ sudo apt install apache2

The following packages were automatically installed and are no longer required:
  cpp-13                      libmagickcore-6.q16-7t64  libpython3.12-stdlib         perl-modules-5.38
  cpp-13-x86-64-linux-gnu     libmagickwand-6.q16-7t64  libpython3.12t64             python3-autocommand
  gcc-13-base                 libmbedcrypto7t64         libqt6dbus6t64               python3-inflect
  imagemagick-6-common        libmfx1                   libqt6gui6t64                python3-jaraco.context
  libassuan0                  libmsgraph-0-1            libqt6network6t64            python3-jaraco.functools
  libavfilter9                libnsl2                   libqt6opengl6t64             python3-more-itertools
  libavformat60               libpaper1                 libqt6widgets6t64            python3-pexpect
  libconfig++9v5              libperl5.38t64            libssh-gcrypt-4              python3-pkg-resources
  libdirectfb-1.7-7t64        libplacebo338             libswscale7                  python3-ptyprocess
  libgspell-1-2               libplist3                 libtag1v5                    python3-six
  libical3t64                 libpoppler134             libtag1v5-vanilla            python3-typeguard
  libimobiledevice6           libpostproc57             libtagc0                     python3.11
  libjim0.82t64               libpython3.11-minimal     libusbmuxd6                  python3.11-minimal
  libldap-2.5-0               libpython3.11-stdlib      libutempter0                 python3.12
  libllvm17t64                libpython3.12-dev         libwebrtc-audio-processing1  python3.12-dev
  libmagickcore-6.q16-7-extra libpython3.12-minimal     linux-image-6.8.11-amd64     python3.12-minimal
Use 'sudo apt autoremove' to remove them.

Installing:
  apache2

Installing dependencies:
  apache2-bin  apache2-data  apache2-utils  libaprutil1-dbd-sqlite3  libaprutil1-ldap  ssl-cert
```

Begin by installing the Apache2 web server on your system. On Ubuntu, this can be achieved using the following commands:

sudo apt update
sudo apt install apache2

After installation, ensure the Apache service is running and enabled to start at boot:



```
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
update-rc.d: As per Kali policy, apache2 init script is left disabled.
update-rc.d: We have no instructions for the apache-htcacheclean init script.
update-rc.d: It looks like a non-network service, we enable it.
apache2.service is a disabled or a static unit, not starting it.
apache-htcacheclean.service is a disabled or a static unit, not starting it.
Processing triggers for kali-menu (2025.1.1) ...
Processing triggers for man-db (2.13.0-1) ...

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start apache2


┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable apache2

Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
Created symlink '/etc/systemd/system/multi-user.target.wants/apache2.service' → '/usr/lib/systemd/system/apache2.serv
ice'.

┌──(kali㉿kali)-[~]
└─$  sudo ufw disable

sudo: ufw: command not found

┌──(kali㉿kali)-[~]
└─$ sudo apt update
sudo apt install ufw

Hit:1 https://brave-browser-apt-beta.s3.brave.com stable InRelease
Hit:2 https://brave-browser-apt-release.s3.brave.com stable InRelease
Hit:3 http://http.kali.org/kali kali-rolling InRelease
Hit:4 https://download.sublimetext.com apt/stable/ InRelease
348 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  cpp-13                      libmagickcore-6.q16-7t64  libpython3.12-stdlib        perl-modules-5.38
  cpp-13-x86-64-linux-gnu     libmagickwand-6.q16-7t64  libpython3.12t64            python3-autocommand
  gcc-13-base                 libmbedcrypto7t64         libqt6dbus6t64              python3-inflect
  imagemagick-6-common        libmfx1                   libqt6gui6t64               python3-jaraco.context
  libassuan0                  libmsgraph-0-1            libqt6network6t64           python3-jaraco.functools
  libavfilter9                libmsl2                   libqt6opengl6t64            python3-more-itertools
  libavformat60               libpaper1                 libqt6widgets6t64           python3-pexpect
  libconfig++9v5              libperl5.38t64            libssh-gcrypt-4             python3-pkg-resources
  libdirectfb-1.7-7t64        libplacebo338             libswscale7                 python3-ptyprocess
```

sudo systemctl start apache2
sudo systemctl enable apache2

Disable UFW to Allow All Trace:

To permit all incoming and outgoing tra c temporarily, disable the Uncomplicated Firewall (UFW):

sudo ufw disable

Exploit-

Scan for Open Ports and Services Using Nmap and Netcat:

With the firewall disabled, an attacker can utilize tools like Nmap and Netcat to identify open ports and running services:

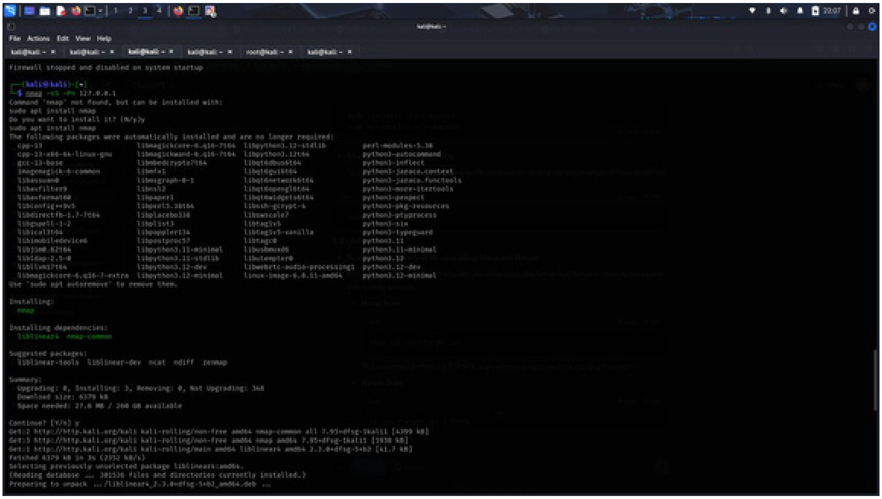
Nmap Scan:

nmap -sS -Pn <target_ip>

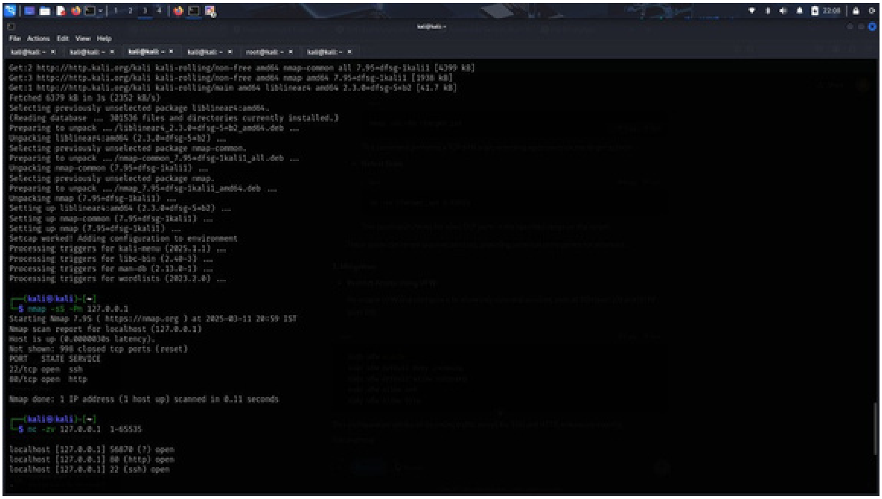
This command performs a TCP SYN scan, detecting open ports on the target system.

Netcat Scan:

nc -zv <target_ip> 1-65535



```
Firewall stopped and disabled on system startup

┌──(kali㉿kali)-[~]
└─$ nmap -sS -Pn 127.0.0.1
Command 'nmap' not found, but can be installed with:
sudo apt install nmap
Do you want to install it? (N/y)y
sudo apt install nmap
The following packages were automatically installed and are no longer required:
  cpp-13                        libmagickcore-6.q16-7t64  libpython3.12-stdlib            perl-modules-5.38
  cpp-13-x86-64-linux-gnu       libmagickwand-6.q16-7t64  libpython3.12t64                python3-autocommand
  gcc-13-base                   libmbedcrypto7t64         libqt6dbus6t64                  python3-inflect
  imagemagick-6-common          libmfx1                   libqt6gui6t64                   python3-jaraco.context
  libassuan0                    libmsgraph-0-1            libqt6network6t64               python3-jaraco.functools
  libavfilter9                  libnsl2                   libqt6opengl6t64                python3-more-itertools
  libavformat60                 libpaper1                 libqt6widgets6t64               python3-pexpect
  libconfig++9v5                libperl5.38t64            libssh-gcrypt-4                 python3-pkg-resources
  libdirectfb-1.7-7t64          libplacebo338             libswscale7                     python3-ptyprocess
  libgspell-1-2                 libplist3                 libtag1v5                       python3-six
  libical3t64                   libpoppler134             libtag1v5-vanilla               python3-typeguard
  libimobiledevice6             libpostproc57             libtagc0                        python3.11
  libjim0.82t64                 libpython3.11-minimal     libusbmuxd6                     python3.11-minimal
  libldap-2.5-0                 libpython3.11-stdlib      libutempter0                    python3.12
  libllvm17t64                  libpython3.12-dev         libwebrtc-audio-processing1     python3.12-dev
  libmagickcore-6.q16-7-extra   libpython3.12-minimal     linux-image-6.8.11-amd64        python3.12-minimal
Use 'sudo apt autoremove' to remove them.

Installing:
  nmap

Installing dependencies:
  liblinear4  nmap-common

Suggested packages:
  liblinear-tools  liblinear-dev  ncat  ndiff  zenmap

Summary:
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 348
  Download size: 6379 kB
  Space needed: 27.6 MB / 260 GB available

Continue? [Y/n] y
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.95+dfsg-1kali1 [4399 kB]
Get:3 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.95+dfsg-1kali1 [1938 kB]
Get:1 http://http.kali.org/kali kali-rolling/main amd64 liblinear4 amd64 2.3.0+dfsg-5+b2 [41.7 kB]
Fetched 6379 kB in 3s (2352 kB/s)
Selecting previously unselected package liblinear4:amd64.
(Reading database ... 301536 files and directories currently installed.)
Preparing to unpack .../liblinear4_2.3.0+dfsg-5+b2_amd64.deb ...
```



```
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.95+dfsg-1kali1 [4399 kB]
Get:3 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.95+dfsg-1kali1 [1938 kB]
Get:1 http://http.kali.org/kali kali-rolling/main amd64 liblinear4 amd64 2.3.0+dfsg-5+b2 [41.7 kB]
Fetched 6379 kB in 3s (2352 kB/s)
Selecting previously unselected package liblinear4:amd64.
(Reading database ... 301536 files and directories currently installed.)
Preparing to unpack .../liblinear4_2.3.0+dfsg-5+b2_amd64.deb ...
Unpacking liblinear4:amd64 (2.3.0+dfsg-5+b2) ...
Selecting previously unselected package nmap-common.
Preparing to unpack .../nmap-common_7.95+dfsg-1kali1_all.deb ...
Unpacking nmap-common (7.95+dfsg-1kali1) ...
Selecting previously unselected package nmap.
Preparing to unpack .../nmap_7.95+dfsg-1kali1_amd64.deb ...
Unpacking nmap (7.95+dfsg-1kali1) ...
Setting up liblinear4:amd64 (2.3.0+dfsg-5+b2) ...
Setting up nmap-common (7.95+dfsg-1kali1) ...
Setting up nmap (7.95+dfsg-1kali1) ...
Setcap worked! Adding configuration to environment
Processing triggers for kali-menu (2025.1.1) ...
Processing triggers for libc-bin (2.40-3) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for wordlists (2023.2.0) ...

┌──(kali㉿kali)-[~]
└─$ nmap -sS -Pn 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-11 20:59 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds

┌──(kali㉿kali)-[~]
└─$ nc -zv 127.0.0.1  1-65535

localhost [127.0.0.1] 56870 (?) open
localhost [127.0.0.1] 80 (http) open
localhost [127.0.0.1] 22 (ssh) open
```

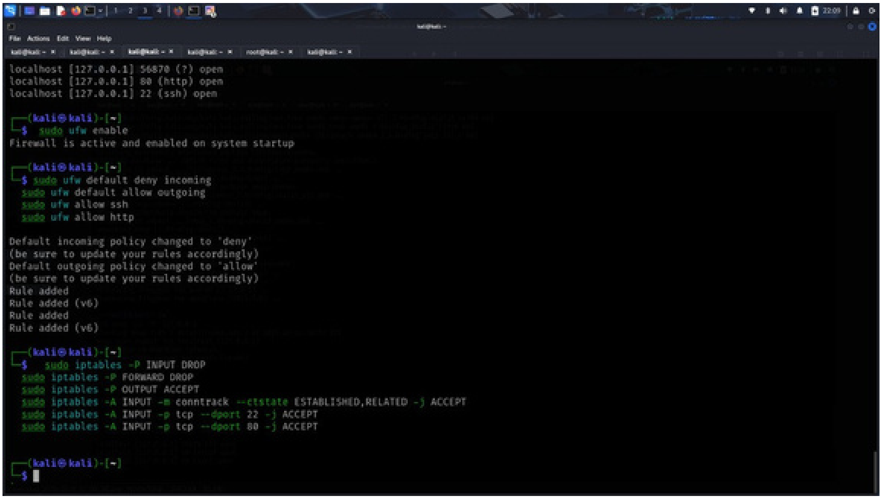
This command checks for open TCP ports in the specified range on the target.

These scans can reveal exposed services, providing potential entry points for attackers.

Mitigation-

Restrict Access Using UFW:
Re-enable UFW and configure it to allow only essential services, such as SSH (port 22) and HTTP (port 80).

This configuration denies all incoming tra c except for SSH and HTTP, enhancing security.



```
localhost [127.0.0.1] 56870 (?) open
localhost [127.0.0.1] 80 (http) open
localhost [127.0.0.1] 22 (ssh) open

┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup

┌──(kali㉿kali)-[~]
└─$ sudo ufw default deny incoming
  sudo ufw default allow outgoing
  sudo ufw allow ssh
  sudo ufw allow http

Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
Rule added
Rule added (v6)
Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$   sudo iptables -P INPUT DROP
  sudo iptables -P FORWARD DROP
  sudo iptables -P OUTPUT ACCEPT
  sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
  sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
  sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT


┌──(kali㉿kali)-[~]
└─$ ▌
```

Implement iptables Rules to Block Unnecessary Tra c:
For more granular control, iptables can be used to define specific rules:

```
 sudo iptables -P INPUT DROP

sudo iptables -P FORWARD DROP

sudo iptables -P OUTPUT ACCEPT

sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

These commands set default policies to drop incoming and forwarding tra c, accept outgoing tra c, and allow established connections along with SSH and HTTP tra c.

POC TASK 4

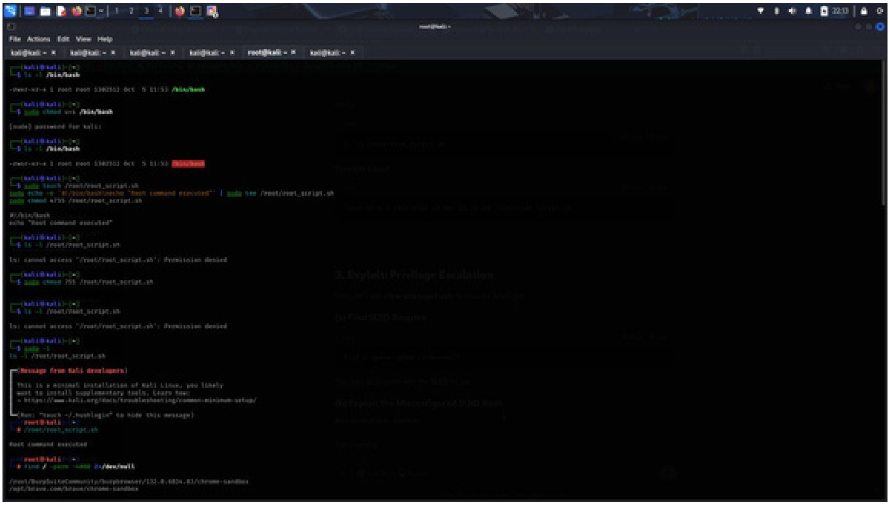
SUID & Privilege Escalation

Setup-

We will intentionally set up a misconfigured SUID binary and a root-owned script to demonstrate privilege escalation.

Enable SUID on /bin/bash :

```
sudo chmod u+s /bin/bash
```



```
┌──(kali㉿kali)-[~]
└─$ ls -l /bin/bash
-rwxr-xr-x 1 root root [illegible] Oct  5 11:53 /bin/bash

┌──(kali㉿kali)-[~]
└─$ sudo chmod u+s /bin/bash
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ ls -l /bin/bash
-rwsr-xr-x 1 root root [illegible] Oct  5 11:53 /bin/bash

┌──(kali㉿kali)-[~]
└─$ sudo touch /root/root_script.sh
sudo echo -e '#!/bin/bash\necho "Root command executed"' | sudo tee /root/root_script.sh
sudo chmod 4755 /root/root_script.sh

#!/bin/bash
echo "Root command executed"

┌──(kali㉿kali)-[~]
└─$ ls -l /root/root_script.sh
ls: cannot access '/root/root_script.sh': Permission denied

┌──(kali㉿kali)-[~]
└─$ sudo chmod 755 /root/root_script.sh

┌──(kali㉿kali)-[~]
└─$ ls -l /root/root_script.sh
ls: cannot access '/root/root_script.sh': Permission denied

┌──(kali㉿kali)-[~]
└─$ sudo -i
ls -l /root/root_script.sh
┌──(Message from Kali developers)
│
│ This is a minimal installation of Kali Linux, you likely
│ want to install supplementary tools. Learn how:
│ ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
└──(Run: "touch ~/.hushlogin" to hide this message)
┌──(root㉿kali)-[~]
└─# /root/root_script.sh
Root command executed

┌──(root㉿kali)-[~]
└─# find / -perm -4000 2>/dev/null
/root/BurpSuiteCommunity/burpbrowser/[illegible]/chrome-sandbox
/opt/brave.com/brave/chrome-sandbox
```

Verify it:

```
ls -l /bin/bash
```

Expected output:

```
-rwsr-xr-x 1 root root 1183448 Feb 11 10:32 /bin/bash
```

Now, any user who executes /bin/bash -p will inherit **root** privileges.

Create a Root-Owned SUID Script (Insecure!)

```
sudo touch /root/root_script.sh
```

```
sudo echo -e '#!/bin/bash\necho "Root command executed"' | sudo tee /root/root_script.sh
```

```
sudo chmod 4755 /root/root_script.sh
```

Verify:

```
ls -l /root/root_script.sh
```

Expected output:

```
-rwsr-xr-x 1 root root 44 Mar 11 12:00 /root/root_script.sh
```

Exploit:
Privilege Escalation

Now, let's use a **low-privileged user** to escalate privileges.

Find SUID Binaries :

`find / -perm -4000 2>/dev/null` This lists all binaries with the **SUID** bit set.

Exploit the Misconfigured SUID Bash :

As a normal user, execute:

```
/bin/bash -p
```

Since `/bin/bash` has the SUID bit set, it runs with **root privileges**.

Exploit: Privilege Escalation

Now, let's use a **low-privileged user** to escalate privileges.

Find SUID Binaries

`find / -perm -4000 2>/dev/null` This lists all

binaries with the **SUID** bit set.

Exploit the Misconfigured SUID Bash

As a normal user, execute:

```
/bin/bash -p
```

Since /bin/bash has the SUID bit set, it runs with **root privileges**.

Verify root access:

```
whoami
```

Expected output:

```
root
```

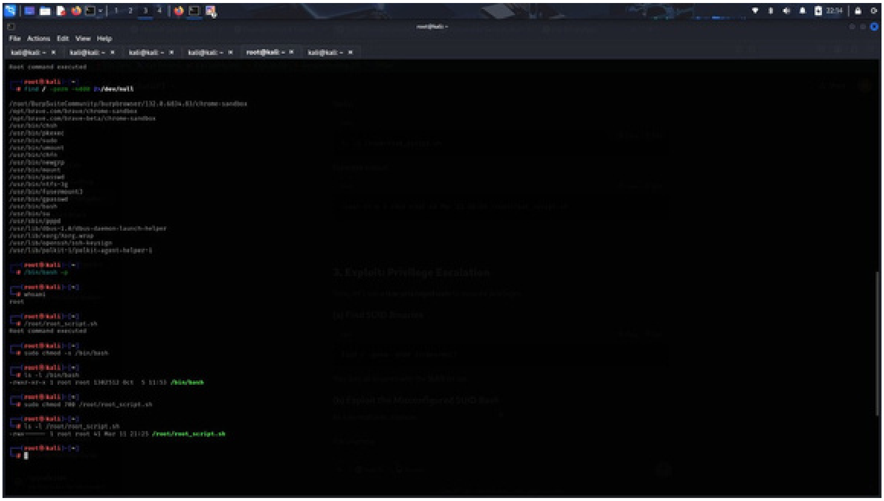
Exploit the SUID Script

Another way to exploit SUID misconfigurations is via a root-owned script.

Try running:

```
/root/root_script.sh
```

If accessible, it runs with **root privileges** due to the SUID bit.



```
Root command executed

┌──(root㉿kali)-[~]
└─# find / -perm -4000 2>/dev/null

/root/BurpSuiteCommunity/burpbrowser/132.0.6834.83/chrome-sandbox
/opt/brave.com/brave/chrome-sandbox
/opt/brave.com/brave-beta/chrome-sandbox
/usr/bin/chsh
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/umount
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/passwd
/usr/bin/ntfs-3g
/usr/bin/fusermount3
/usr/bin/gpasswd
/usr/bin/bash
/usr/bin/su
/usr/sbin/pppd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/lib/openssh/ssh-keysign
/usr/lib/polkit-1/polkit-agent-helper-1

┌──(root㉿kali)-[~]
└─# /bin/bash -p

┌──(root㉿kali)-[~]
└─# whoami
root

┌──(root㉿kali)-[~]
└─# /root/root_script.sh
Root command executed

┌──(root㉿kali)-[~]
└─# sudo chmod -s /bin/bash

┌──(root㉿kali)-[~]
└─# ls -l /bin/bash
-rwxr-xr-x 1 root root 1302152 Oct  5 11:53 /bin/bash

┌──(root㉿kali)-[~]
└─# sudo chmod 700 /root/root_script.sh

┌──(root㉿kali)-[~]
└─# ls -l /root/root_script.sh
-rwx------ 1 root root 41 Mar 11 21:25 /root/root_script.sh

┌──(root㉿kali)-[~]
└─# 
```

Exploit:

Privilege Escalation

Now, let's use a **low-privileged user** to escalate privileges.

Find SUID Binaries

`find / -perm -4000 2>/dev/null` This lists all binaries with the **SUID** bit set.

Exploit the Misconfigured SUID Bash :

As a normal user, execute: /bin/bash -p Since **/bin/bash** has the SUID bit set, it runs with **root privileges**.

Verify root access:

```
whoami
```

Expected output:

```
root
```

Exploit the SUID Script

Another way to exploit SUID misconfigurations is via a root-owned script.

Try running:

`/root/root_script.sh` If accessible, it runs with **root privileges** due to the SUID bit.

Mitigation:

Securing the System

Remove SUID from /bin/bash

sudo chmod -s /bin/bash **Verify:** `ls -l /bin/bash`

Expected output:

```
-rwxr-xr-x 1 root root 1183448 Feb 11 10:32 /bin/bash
```

The **SUID bit is removed**.

Secure the Root-Owned Script

```
sudo chmod 700 /root/root_script.sh
```

This ensures **only root** can execute it.

Verify:

```
ls -l /root/root_script.sh
```

Expected output:

```
-rwx------ 1 root root 44 Mar 11 12:00 /root/root_script.sh
```

Use Sudo Instead

Instead of setting SUID, use sudo with **restricted permissions**:

```
sudo visudo
```

Add:

```
user ALL=(ALL:ALL) /path/to/safe/script.sh
```

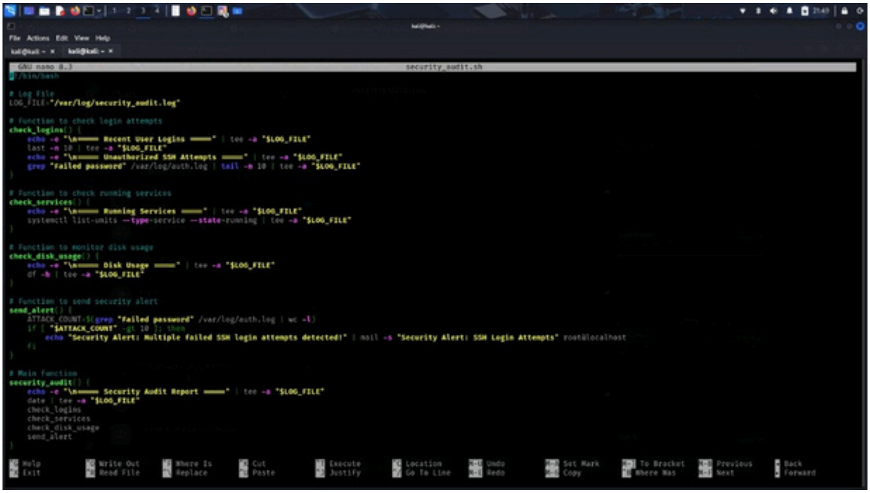
This allows the user to execute only **specific commands** with sudo.

POC TASK 5

Automated Security Auditing & Scripting

Bash Script:

Write a bash script using nano called **security_audit.sh**



```
GNU nano 8.3                          security_audit.sh
#!/bin/bash

# Log File
LOG_FILE="/var/log/security_audit.log"

# Function to check login attempts
check_logins() {
    echo -e "\n===== Recent User Logins =====" | tee -a "$LOG_FILE"
    last -n 10 | tee -a "$LOG_FILE"
    echo -e "\n===== Unauthorized SSH Attempts =====" | tee -a "$LOG_FILE"
    grep "Failed password" /var/log/auth.log | tail -n 10 | tee -a "$LOG_FILE"
}

# Function to check running services
check_services() {
    echo -e "\n===== Running Services =====" | tee -a "$LOG_FILE"
    systemctl list-units --type=service --state=running | tee -a "$LOG_FILE"
}

# Function to monitor disk usage
check_disk_usage() {
    echo -e "\n===== Disk Usage =====" | tee -a "$LOG_FILE"
    df -h | tee -a "$LOG_FILE"
}

# Function to send security alert
send_alert() {
    ATTACK_COUNT=$(grep "Failed password" /var/log/auth.log | wc -l)
    if [ "$ATTACK_COUNT" -gt 10 ]; then
        echo "Security Alert: Multiple failed SSH login attempts detected!" | mail -s "Security Alert: SSH Login Attempts" root@localhost
    fi
}

# Main function
security_audit() {
    echo -e "\n===== Security Audit Report =====" | tee -a "$LOG_FILE"
    date | tee -a "$LOG_FILE"
    check_logins
    check_services
    check_disk_usage
    send_alert
}
```

#!/bin/bash

Log File

LOG_FILE="/var/log/security_audit.log"

Function to check login attempts

check_logins() {

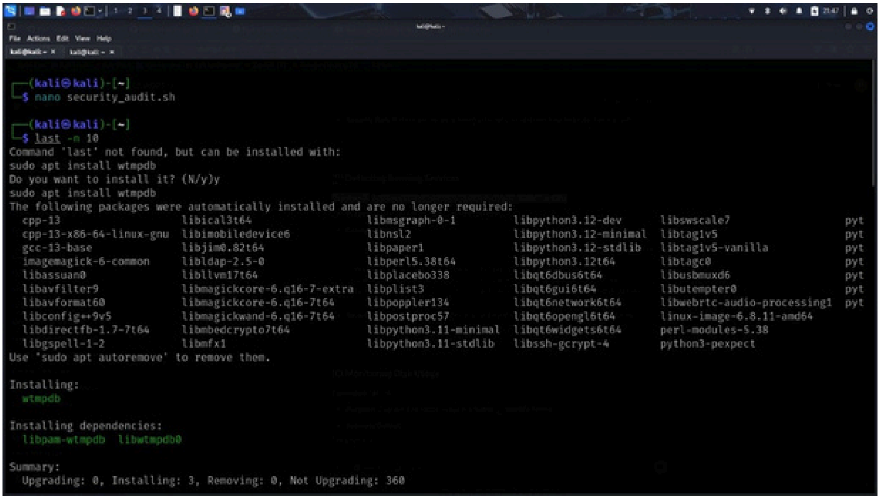
```
 echo -e "\n===== Recent User Logins =====" | tee -a "$LOG_FILE"

last -n 10 | tee -a "$LOG_FILE"

echo -e "\n===== Unauthorized SSH Attempts =====" | tee -a "$LOG_FILE"

grep "Failed password" /var/log/auth.log | tail -n 10 | tee -a "$LOG_FILE"

}


# Function to check running services

check_services() {

echo -e "\n===== Running Services =====" | tee -a "$LOG_FILE"

systemctl list-units --type=service --state=running | tee -a "$LOG_FILE"

}


# Function to monitor disk usage

check_disk_usage() {

echo -e "\n===== Disk Usage =====" | tee -a "$LOG_FILE"

df -h | tee -a "$LOG_FILE"

}


# Function to send security alert

send_alert() {

ATTACK_COUNT=$(grep "Failed password" /var/log/auth.log | wc -l)

if [ "$ATTACK_COUNT" -gt 10 ]; then
```

```
        echo "Security Alert: Multiple failed SSH login attempts detected!" | mail -s "Security
Alert: SSH Login Attempts" root@localhost

    fi

}

 # Main function

 security_audit() {

 echo -e "\n===== Security Audit Report =====" | tee -a "$LOG_FILE"

 date | tee -a "$LOG_FILE"

     check_logins

     check_services

     check_disk_usage

     send_alert

 }


 # Execute the script

 security_audit
```

Checking User Login Attempts

Command:

```
last -n 10
```



```
┌──(kali㉿kali)-[~]
└─$ nano security_audit.sh

┌──(kali㉿kali)-[~]
└─$ last -n 10
Command 'last' not found, but can be installed with:
sudo apt install wtmpdb
Do you want to install it? (N/y)y
sudo apt install wtmpdb
The following packages were automatically installed and are no longer required:
  cpp-13                  libical3t64                libmsgraph-0-1         libpython3.12-dev       libswscale7                pyt
  cpp-13-x86-64-linux-gnu libimobiledevice6          libnsl2                libpython3.12-minimal   libtag1v5                  pyt
  gcc-13-base             libjim0.82t64              libpaper1              libpython3.12-stdlib    libtag1v5-vanilla          pyt
  imagemagick-6-common    libldap-2.5-0              libperl5.38t64         libpython3.12t64        libtagc0                   pyt
  libassuan0              libllvm17t64               libplacebo338          libqt6dbus6t64          libusbmuxd6                pyt
  libavfilter9            libmagickcore-6.q16-7-extra libplist3             libqt6gui6t64           libutempter0               pyt
  libavformat60           libmagickcore-6.q16-7t64   libpoppler134          libqt6network6t64       libwebrtc-audio-processing1 pyt
  libconfig++9v5          libmagickwand-6.q16-7t64   libpostproc57          libqt6opengl6t64        linux-image-6.8.11-amd64
  libdirectfb-1.7-7t64    libmbedcrypto7t64          libpython3.11-minimal  libqt6widgets6t64       perl-modules-5.38
  libgspell-1-2           libmfx1                    libpython3.11-stdlib   libssh-gcrypt-4         python3-pexpect
Use 'sudo apt autoremove' to remove them.

Installing:
  wtmpdb

Installing dependencies:
  libpam-wtmpdb  libwtmpdb0

Summary:
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 360
```

Purpose: Lists the last 10 user login attempts.

Example Output: pts/0

```
root            192.168.1.100 Mon Mar 11 12:00 still logged in
user1 pts/1     192.168.1.101 Mon Mar 11 11:45 - 11:55 (00:10)
```

Security Risk: Identifies old, inactive accounts or unauthorized logins.

Command: grep "Failed password" /var/log/auth.log | tail -n 10

Purpose: Finds failed SSH login attempts from /var/log/auth.log.

Example Output:

```
r 11 12:30:01 server sshd[12345]: Failed password for invalid user admin from 192.168.1.200
```

Security Risk:

If there are multiple failed attempts, an attacker may be brute-forcing SSH

Detecting Running Services

Command:

```
systemctl list-units --type=service --state=running
```

Purpose: Lists currently running system services.

Example Output:

```
UNIT                LOAD ACTIVE SUB      DESCRIPTION
apache2.service          loaded active running The Apache HTTP Server
ssh.service              loaded active running OpenBSD Secure Shell server
```

Security Risk: Unnecessary services (e.g., old database servers) can expose vulnerabilities.

Monitoring Disk Usage

```
Command: df -h
```

Purpose: Displays disk space usage in a **human-readable** format.

Example Output:

```
Filesystem     Size Used Avail Use% Mounted on
/dev/sda1      50G 45G 5G 90% /
```

Security Risk: If disk space is **over 90%**, attackers might try a **Denial-of-Service (DoS) attack** by filling up logs or storage.

Sending Security Alerts

Command: grep "Failed password" /var/log/auth.log | wc -l

Purpose: Counts the number of failed SSH login attempts.

Example Output:

```
15
```

Action: If this count is greater than 10, an alert is sent.

```
Command: mail -s "Security Alert: SSH Login Attempts" root@localhost
```

Purpose: Sends an email alert.

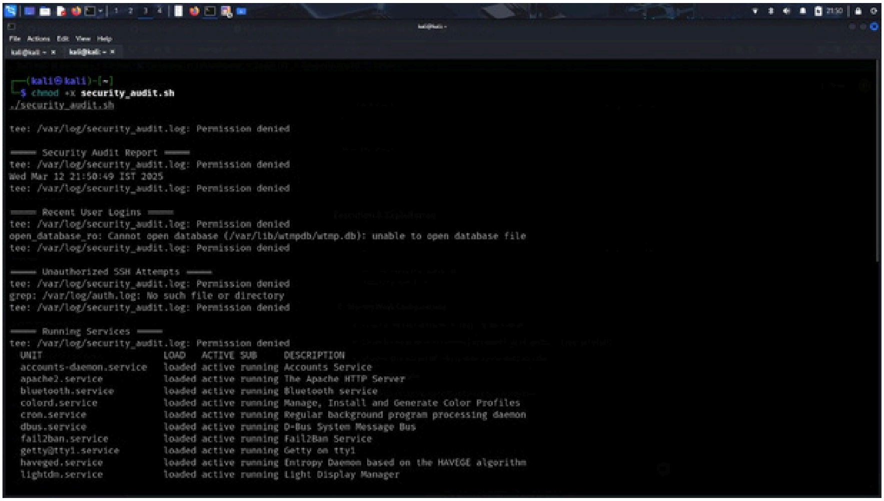
Alternative: Install and configure mailutilsfor external emails:

```
sudo apt install mailutils
```

Running the Script

Make the script executable:

```
chmod +x security_audit.sh
```



```
┌──(kali㉿kali)-[~]
└─$ chmod +x security_audit.sh
./security_audit.sh

tee: /var/log/security_audit.log: Permission denied

===== Security Audit Report =====
tee: /var/log/security_audit.log: Permission denied
Wed Mar 12 21:50:49 IST 2025
tee: /var/log/security_audit.log: Permission denied

===== Recent User Logins =====
tee: /var/log/security_audit.log: Permission denied
open_database_ro: Cannot open database (/var/lib/wtmpdb/wtmp.db): unable to open database file
tee: /var/log/security_audit.log: Permission denied

===== Unauthorized SSH Attempts =====
tee: /var/log/security_audit.log: Permission denied
grep: /var/log/auth.log: No such file or directory
tee: /var/log/security_audit.log: Permission denied

===== Running Services =====
tee: /var/log/security_audit.log: Permission denied
  UNIT                          LOAD   ACTIVE SUB     DESCRIPTION
  accounts-daemon.service       loaded active running Accounts Service
  apache2.service               loaded active running The Apache HTTP Server
  bluetooth.service             loaded active running Bluetooth service
  colord.service                loaded active running Manage, Install and Generate Color Profiles
  cron.service                  loaded active running Regular background program processing daemon
  dbus.service                  loaded active running D-Bus System Message Bus
  fail2ban.service              loaded active running Fail2Ban Service
  getty@tty1.service            loaded active running Getty on tty1
  haveged.service               loaded active running Entropy Daemon based on the HAVEGE algorithm
  lightdm.service               loaded active running Light Display Manager
```

Run the script:

```
./security_audit.sh
```

Expected output:

```
pgsql

===== Security Audit Report =====

Wed Mar 11 12:30:00 UTC 2025



===== Recent User Logins =====

(root) pts/0 192.168.1.100 Mon Mar 11 12:00 still logged in



===== Unauthorized SSH Attempts =====

Mar 11 12:30:01 server sshd[12345]: Failed password for invalid user admin from
192.168.1.200


===== Running Services =====

apache2.service        loaded active running The Apache HTTP Server



===== Disk Usage =====

Filesystem     Size Used Avail Use% Mounted on

/dev/sda1      50G 45G 5G 90% /
```

Automating with Cron

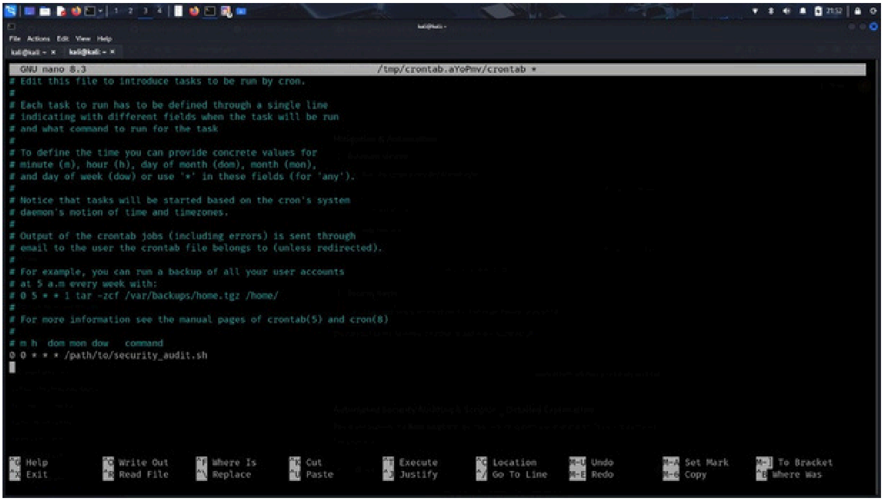
To run the script automatically every day at midnight, use:

```
crontab -e
```

Add this line:

```
0 0 * * * /path/to/security_audit.sh
```

This ensures the script runs **daily at midnight**.



```
  GNU nano 8.3                                        /tmp/crontab.aYoPmv/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 0 * * * /path/to/security_audit.sh

^G Help        ^O Write Out   ^F Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^B Where Was
```

POC TASK 6

Log Analysis & Intrusion Detection

Setup-

Enable System Logging

Ensure system logging is active and capturing SSH events.

Verify logging is enabled:

sudo systemctl status rsyslog

sudo journalctl -xe | grep ssh

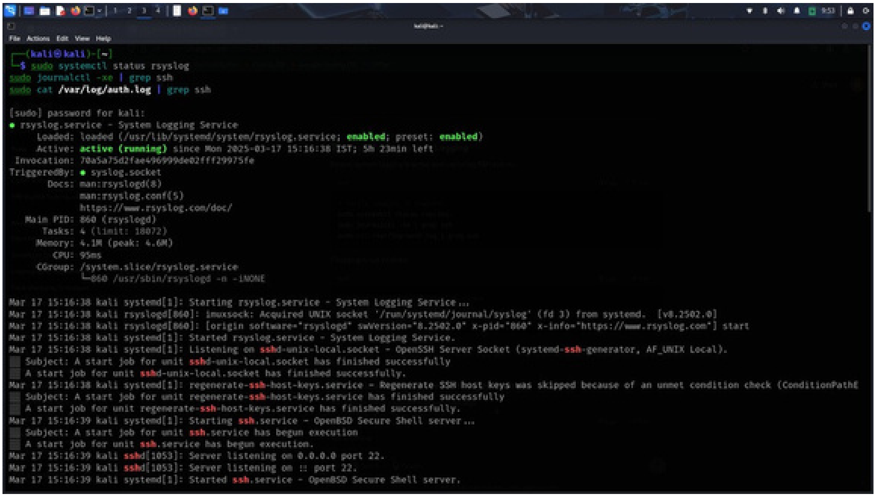
sudo cat /var/log/auth.log | grep ssh

If logging is not enabled:

Enable and restart rsyslog if needed;

sudo systemctl enable rsyslog

sudo systemctl restart rsyslog



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status rsyslog
sudo journalctl -xe | grep ssh
sudo cat /var/log/auth.log | grep ssh

[sudo] password for kali:
● rsyslog.service - System Logging Service
     Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-03-17 15:16:38 IST; 5h 23min left
 Invocation: 70a5a75d2fae496999de02fff29975fe
TriggeredBy: ● syslog.socket
       Docs: man:rsyslogd(8)
             man:rsyslog.conf(5)
             https://www.rsyslog.com/doc/
   Main PID: 860 (rsyslogd)
      Tasks: 4 (limit: 18072)
     Memory: 4.1M (peak: 4.6M)
        CPU: 95ms
     CGroup: /system.slice/rsyslog.service
             └─860 /usr/sbin/rsyslogd -n -iNONE

Mar 17 15:16:38 kali systemd[1]: Starting rsyslog.service - System Logging Service...
Mar 17 15:16:38 kali rsyslogd[860]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3) from systemd.  [v8.2502.0]
Mar 17 15:16:38 kali rsyslogd[860]: [origin software="rsyslogd" swVersion="8.2502.0" x-pid="860" x-info="https://www.rsyslog.com"] start
Mar 17 15:16:38 kali systemd[1]: Started rsyslog.service - System Logging Service.
Mar 17 15:16:38 kali systemd[1]: Listening on sshd-unix-local.socket - OpenSSH Server Socket (systemd-ssh-generator, AF_UNIX Local).
░░ Subject: A start job for unit sshd-unix-local.socket has finished successfully
░░ A start job for unit sshd-unix-local.socket has finished successfully.
Mar 17 15:16:38 kali systemd[1]: regenerate-ssh-host-keys.service - Regenerate SSH host keys was skipped because of an unmet condition check (ConditionPathE
░░ Subject: A start job for unit regenerate-ssh-host-keys.service has finished successfully
░░ A start job for unit regenerate-ssh-host-keys.service has finished successfully.
Mar 17 15:16:39 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
░░ Subject: A start job for unit ssh.service has begun execution
░░ A start job for unit ssh.service has begun execution.
Mar 17 15:16:39 kali sshd[1053]: Server listening on 0.0.0.0 port 22.
Mar 17 15:16:39 kali sshd[1053]: Server listening on :: port 22.
Mar 17 15:16:39 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

Simulate Multiple Failed SSH Login Attempts :

Run a brute-force simulation to generate logs and Attempt SSH login with incorrect credentials

```
ssh user@localhost
```

Enter incorrect passwords multiple times

Alternatively, simulate automated attacks with Hydra:

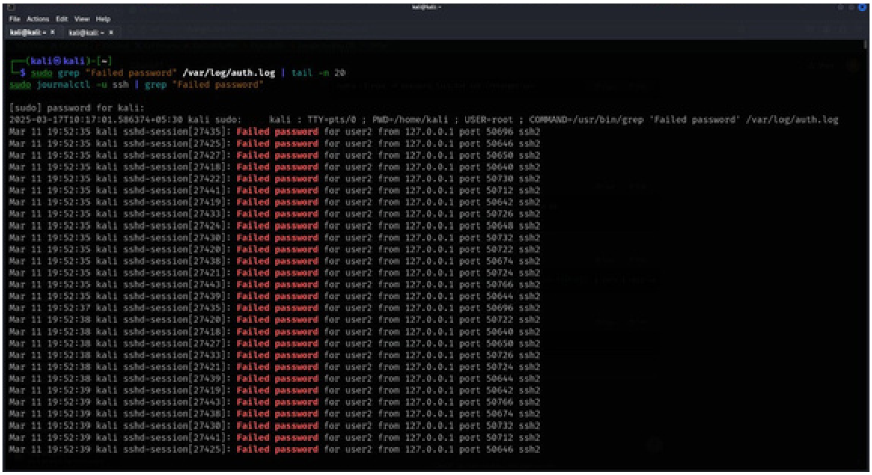
```
hydra -l root -P password_list.txt ssh://<target-ip>
```

Exploit: Log Analysis

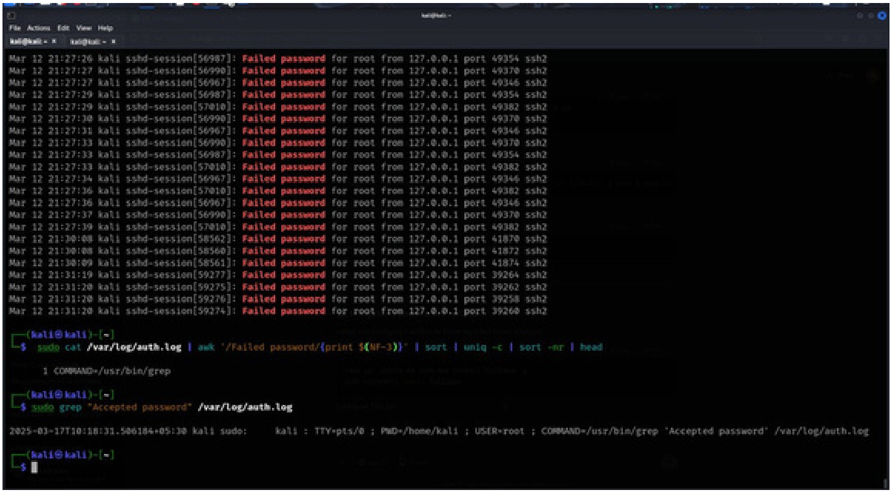
Extract failed login attempts using grep.

```
sudo grep "Failed password" /var/log/auth.log | tail -n 20
```

```
sudo journalctl -u ssh | grep "Failed password"
```



```
┌──(kali㉿kali)-[~]
└─$ sudo grep "Failed password" /var/log/auth.log | tail -n 20
sudo journalctl -u ssh | grep "Failed password"

[sudo] password for kali:
2025-03-17T10:17:01.586374+05:30 kali sudo:     kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep 'Failed password' /var/log/auth.log
Mar 11 19:52:35 kali sshd-session[27435]: Failed password for user2 from 127.0.0.1 port 50696 ssh2
Mar 11 19:52:35 kali sshd-session[27425]: Failed password for user2 from 127.0.0.1 port 50646 ssh2
Mar 11 19:52:35 kali sshd-session[27427]: Failed password for user2 from 127.0.0.1 port 50650 ssh2
Mar 11 19:52:35 kali sshd-session[27418]: Failed password for user2 from 127.0.0.1 port 50640 ssh2
Mar 11 19:52:35 kali sshd-session[27422]: Failed password for user2 from 127.0.0.1 port 50730 ssh2
Mar 11 19:52:35 kali sshd-session[27441]: Failed password for user2 from 127.0.0.1 port 50712 ssh2
Mar 11 19:52:35 kali sshd-session[27419]: Failed password for user2 from 127.0.0.1 port 50642 ssh2
Mar 11 19:52:35 kali sshd-session[27433]: Failed password for user2 from 127.0.0.1 port 50726 ssh2
Mar 11 19:52:35 kali sshd-session[27424]: Failed password for user2 from 127.0.0.1 port 50648 ssh2
Mar 11 19:52:35 kali sshd-session[27430]: Failed password for user2 from 127.0.0.1 port 50732 ssh2
Mar 11 19:52:35 kali sshd-session[27420]: Failed password for user2 from 127.0.0.1 port 50722 ssh2
Mar 11 19:52:35 kali sshd-session[27438]: Failed password for user2 from 127.0.0.1 port 50674 ssh2
Mar 11 19:52:35 kali sshd-session[27421]: Failed password for user2 from 127.0.0.1 port 50724 ssh2
Mar 11 19:52:35 kali sshd-session[27443]: Failed password for user2 from 127.0.0.1 port 50766 ssh2
Mar 11 19:52:35 kali sshd-session[27439]: Failed password for user2 from 127.0.0.1 port 50644 ssh2
Mar 11 19:52:37 kali sshd-session[27435]: Failed password for user2 from 127.0.0.1 port 50696 ssh2
Mar 11 19:52:38 kali sshd-session[27420]: Failed password for user2 from 127.0.0.1 port 50722 ssh2
Mar 11 19:52:38 kali sshd-session[27418]: Failed password for user2 from 127.0.0.1 port 50640 ssh2
Mar 11 19:52:38 kali sshd-session[27427]: Failed password for user2 from 127.0.0.1 port 50650 ssh2
Mar 11 19:52:38 kali sshd-session[27433]: Failed password for user2 from 127.0.0.1 port 50726 ssh2
Mar 11 19:52:38 kali sshd-session[27421]: Failed password for user2 from 127.0.0.1 port 50724 ssh2
Mar 11 19:52:38 kali sshd-session[27439]: Failed password for user2 from 127.0.0.1 port 50644 ssh2
Mar 11 19:52:39 kali sshd-session[27419]: Failed password for user2 from 127.0.0.1 port 50642 ssh2
Mar 11 19:52:39 kali sshd-session[27443]: Failed password for user2 from 127.0.0.1 port 50766 ssh2
Mar 11 19:52:39 kali sshd-session[27438]: Failed password for user2 from 127.0.0.1 port 50674 ssh2
Mar 11 19:52:39 kali sshd-session[27430]: Failed password for user2 from 127.0.0.1 port 50732 ssh2
Mar 11 19:52:39 kali sshd-session[27441]: Failed password for user2 from 127.0.0.1 port 50712 ssh2
Mar 11 19:52:39 kali sshd-session[27425]: Failed password for user2 from 127.0.0.1 port 50646 ssh2
```



```
Mar 12 21:27:26 kali sshd-session[56987]: Failed password for root from 127.0.0.1 port 49354 ssh2
Mar 12 21:27:27 kali sshd-session[56990]: Failed password for root from 127.0.0.1 port 49370 ssh2
Mar 12 21:27:27 kali sshd-session[56967]: Failed password for root from 127.0.0.1 port 49346 ssh2
Mar 12 21:27:29 kali sshd-session[56987]: Failed password for root from 127.0.0.1 port 49354 ssh2
Mar 12 21:27:29 kali sshd-session[57010]: Failed password for root from 127.0.0.1 port 49382 ssh2
Mar 12 21:27:30 kali sshd-session[56990]: Failed password for root from 127.0.0.1 port 49370 ssh2
Mar 12 21:27:31 kali sshd-session[56967]: Failed password for root from 127.0.0.1 port 49346 ssh2
Mar 12 21:27:33 kali sshd-session[56990]: Failed password for root from 127.0.0.1 port 49370 ssh2
Mar 12 21:27:33 kali sshd-session[56987]: Failed password for root from 127.0.0.1 port 49354 ssh2
Mar 12 21:27:33 kali sshd-session[57010]: Failed password for root from 127.0.0.1 port 49382 ssh2
Mar 12 21:27:34 kali sshd-session[56967]: Failed password for root from 127.0.0.1 port 49346 ssh2
Mar 12 21:27:36 kali sshd-session[57010]: Failed password for root from 127.0.0.1 port 49382 ssh2
Mar 12 21:27:36 kali sshd-session[56967]: Failed password for root from 127.0.0.1 port 49346 ssh2
Mar 12 21:27:37 kali sshd-session[56990]: Failed password for root from 127.0.0.1 port 49370 ssh2
Mar 12 21:27:39 kali sshd-session[57010]: Failed password for root from 127.0.0.1 port 49382 ssh2
Mar 12 21:30:08 kali sshd-session[58562]: Failed password for root from 127.0.0.1 port 41870 ssh2
Mar 12 21:30:08 kali sshd-session[58560]: Failed password for root from 127.0.0.1 port 41872 ssh2
Mar 12 21:30:09 kali sshd-session[58561]: Failed password for root from 127.0.0.1 port 41874 ssh2
Mar 12 21:31:19 kali sshd-session[59277]: Failed password for root from 127.0.0.1 port 39264 ssh2
Mar 12 21:31:20 kali sshd-session[59275]: Failed password for root from 127.0.0.1 port 39262 ssh2
Mar 12 21:31:20 kali sshd-session[59276]: Failed password for root from 127.0.0.1 port 39258 ssh2
Mar 12 21:31:20 kali sshd-session[59274]: Failed password for root from 127.0.0.1 port 39260 ssh2

┌──(kali㉿kali)-[~]
└─$ sudo cat /var/log/auth.log | awk '/Failed password/{print $(NF-3)}' | sort | uniq -c | sort -nr | head

      1 COMMAND=/usr/bin/grep

┌──(kali㉿kali)-[~]
└─$ sudo grep "Accepted password" /var/log/auth.log

2025-03-17T10:18:31.506184+05:30 kali sudo:     kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep 'Accepted password' /var/log/auth.log

┌──(kali㉿kali)-[~]
└─$ 
```

Find brute-force attempts (multiple failures from the same IP):

sudo cat /var/log/auth.log | awk '/Failed password/{print $(NF-3)}' | sort | uniq -c | sort -nr | head

Identify successful logins:

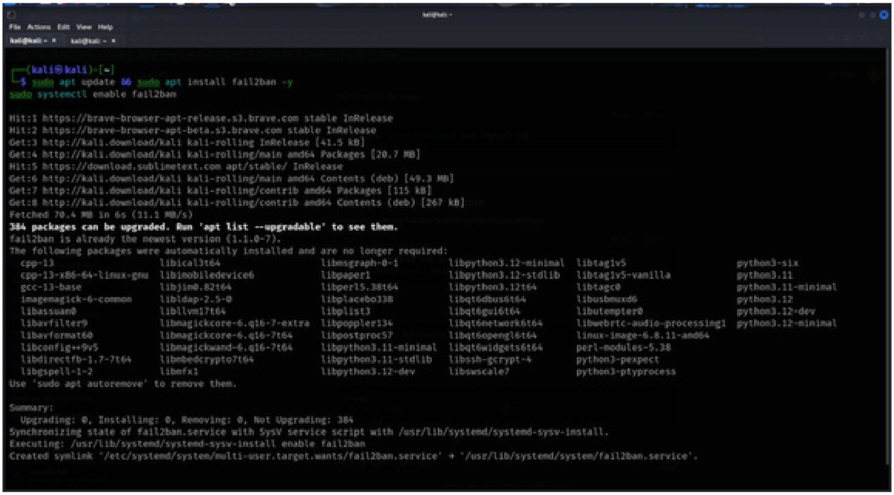
sudo grep "Accepted password" /var/log/auth.log

Mitigation: Implement Fail2Ban

Install and configure Fail2Ban to block repeated failed attempts.

sudo apt update && sudo apt install fail2ban -y

sudo systemctl enable fail2ban

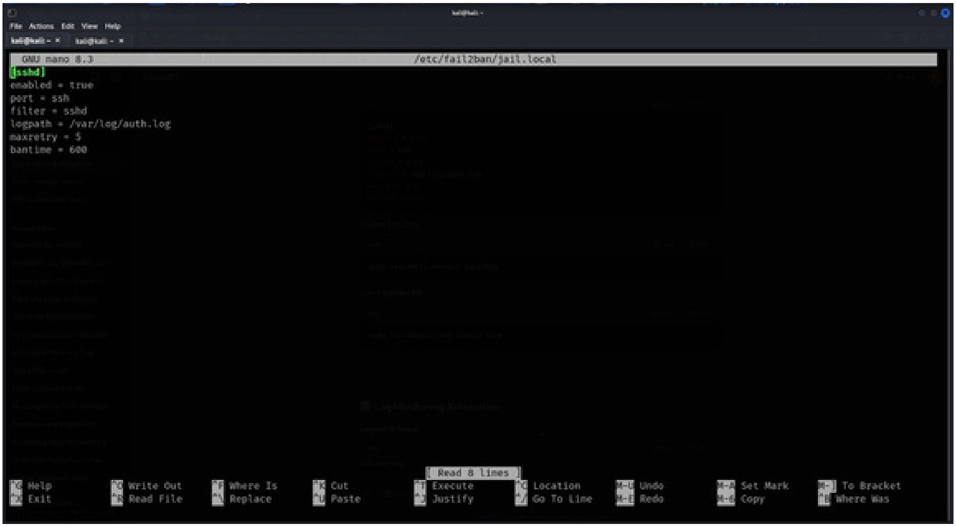


```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt install fail2ban -y
sudo systemctl enable fail2ban

Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Hit:2 https://brave-browser-apt-beta.s3.brave.com stable InRelease
Get:3 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Packages [20.7 MB]
Hit:5 https://download.sublimetext.com apt/stable/ InRelease
Get:6 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [49.3 MB]
Get:7 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:8 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [267 kB]
Fetched 70.4 MB in 6s (11.1 MB/s)
384 packages can be upgraded. Run 'apt list --upgradable' to see them.
fail2ban is already the newest version (1.1.0-7).
The following packages were automatically installed and are no longer required:
  cpp-13                  libical3t64                libmsgraph-0-1          libpython3.12-minimal  libtag1v5                     python3-six
  cpp-13-x86-64-linux-gnu libimobiledevice6          libpaper1               libpython3.12-stdlib   libtag1v5-vanilla             python3.11
  gcc-13-base             libjim0.82t64              libperl5.38t64          libpython3.12t64       libtagc0                      python3.11-minimal
  imagemagick-6-common    libldap-2.5-0              libplacebo338           libqt6dbus6t64         libusbmuxd6                   python3.12
  libassuan0              libllvm17t64               libplist3               libqt6gui6t64          libutempter0                  python3.12-dev
  libavfilter9            libmagickcore-6.q16-7-extra libpoppler134          libqt6network6t64      libwebrtc-audio-processing1   python3.12-minimal
  libavformat60           libmagickcore-6.q16-7t64   libpostproc57           libqt6opengl6t64       linux-image-6.8.11-amd64
  libconfig++9v5          libmagickwand-6.q16-7t64   libpython3.11-minimal   libqt6widgets6t64      perl-modules-5.38
  libdirectfb-1.7-7t64    libmbedcrypto7t64          libpython3.11-stdlib    libssh-gcrypt-4        python3-pexpect
  libgspell-1-2           libmfx1                    libpython3.12-dev       libswscale7            python3-ptyprocess
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 384
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban
Created symlink '/etc/systemd/system/multi-user.target.wants/fail2ban.service' → '/usr/lib/systemd/system/fail2ban.service'.
```

Configure SSH Jail:

sudo nano /etc/fail2ban/jail.local

Add these lines:



```
  GNU nano 8.3                         /etc/fail2ban/jail.local
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 600

                                    [ Read 8 lines ]
^G Help       ^O Write Out   ^F Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit       ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^B Where Was
```

[sshd]

enabled = true

port = ssh

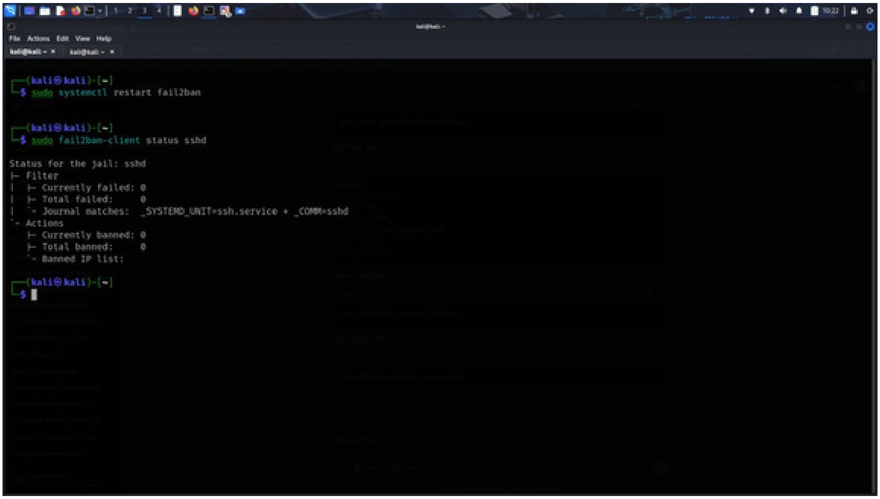
filter = sshd

logpath = /var/log/auth.log

maxretry = 5 bantime = 600

Restart Fail2Ban:

sudo systemctl restart fail2ban



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart fail2ban

┌──(kali㉿kali)-[~]
└─$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- Journal matches:  _SYSTEMD_UNIT=ssh.service + _COMM=sshd
`- Actions
   |- Currently banned: 0
   |- Total banned:     0
   `- Banned IP list:

┌──(kali㉿kali)-[~]
└─$ 
```

Check banned IPs:

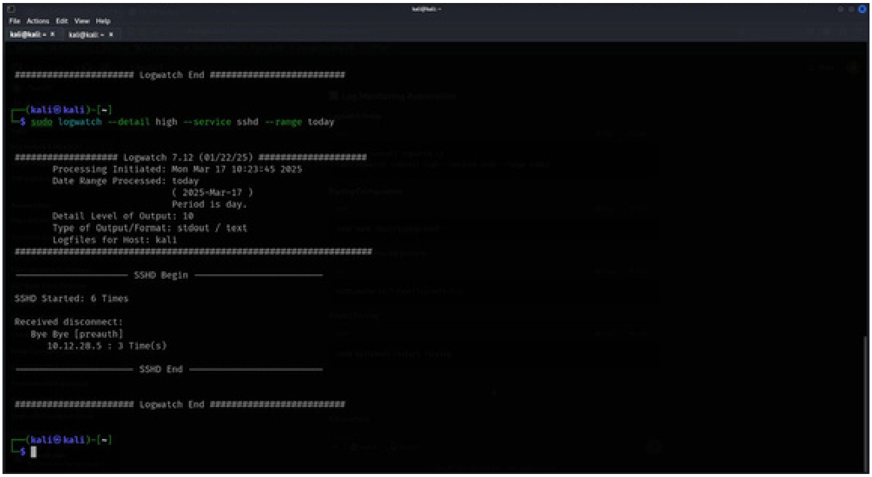
sudo fail2ban-client status sshd

Log Monitoring Automation

Logwatch Setup :

sudo apt install logwatch -y sudo logwatch --detail high -

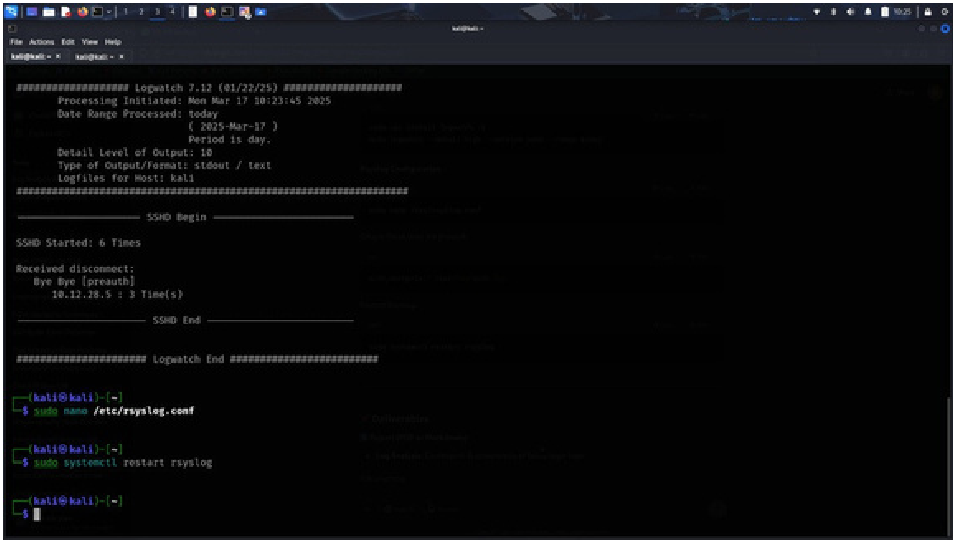
-service sshd --range today



```
###################### Logwatch End #########################

┌──(kali㉿kali)-[~]
└─$ sudo logwatch --detail high --service sshd --range today

################### Logwatch 7.12 (01/22/25) ####################
        Processing Initiated: Mon Mar 17 10:23:45 2025
        Date Range Processed: today
                              ( 2025-Mar-17 )
                              Period is day.
        Detail Level of Output: 10
        Type of Output/Format: stdout / text
        Logfiles for Host: kali
##################################################################

 --------------------- SSHD Begin ------------------------

 SSHD Started: 6 Times

 Received disconnect:
    Bye Bye [preauth]
       10.12.28.5 : 3 Time(s)

 ---------------------- SSHD End -------------------------

###################### Logwatch End #########################

┌──(kali㉿kali)-[~]
└─$ 
```

Rsyslog Configuration :

sudo nano /etc/rsyslog.conf



```
#################### Logwatch 7.12 (01/22/25) ####################
        Processing Initiated: Mon Mar 17 10:23:45 2025
        Date Range Processed: today
                              ( 2025-Mar-17 )
                              Period is day.
        Detail Level of Output: 10
        Type of Output/Format: stdout / text
        Logfiles for Host: kali
##################################################################

 --------------------- SSHD Begin ------------------------

 SSHD Started: 6 Times

 Received disconnect:
    Bye Bye [preauth]
       10.12.28.5 : 3 Time(s)

 ---------------------- SSHD End -------------------------

###################### Logwatch End #########################

┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/rsyslog.conf

┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart rsyslog

┌──(kali㉿kali)-[~]
└─$ 
```

Ensure these lines are present:

```
auth,authpriv.* /var/log/auth.log
```

Restart Rsyslog:

```
sudo systemctl restart rsyslog
```