Proof of Concept (PoC) Report

Task 2: Remote Access & SSH Hardening

📌 Objective:

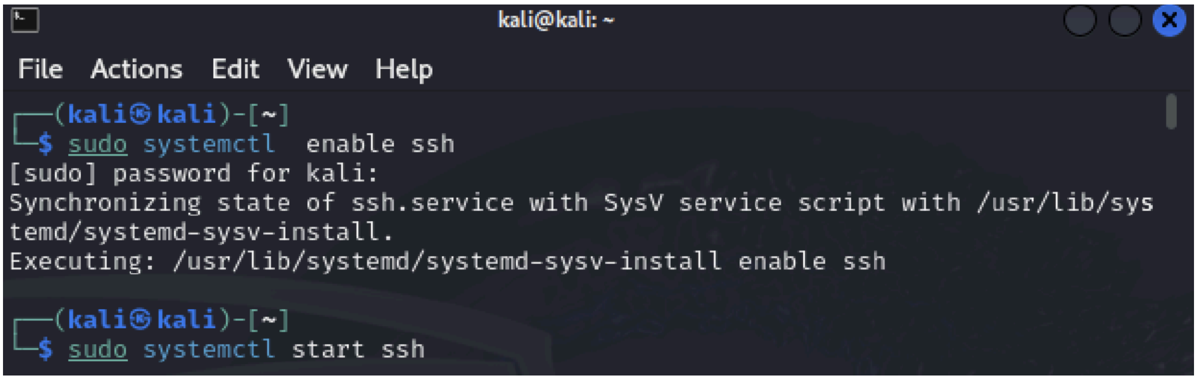
Demonstrate how insecure SSH configurations can be exploited and how to harden SSH to prevent unauthorized access.

1️⃣Setup: Enabling SSH with Weak Configuration

Step 1: Enable SSH and Allow Root Login

Command:

sudo apt update && sudo apt install openssh-server -y(If not installed)



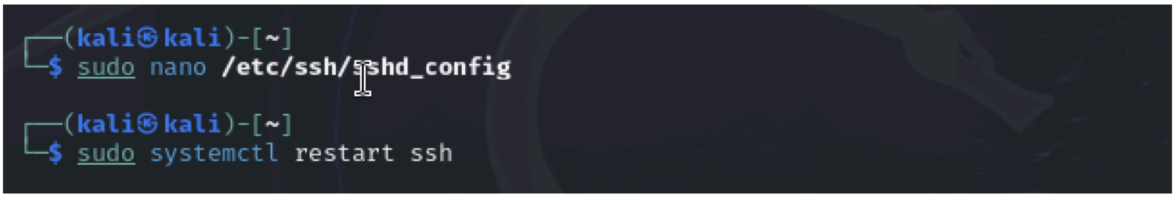
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl  enable ssh
[sudo] password for kali:
Synchronizing state of ssh.service with SysV service script with /usr/lib/sys
temd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh

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

What Does It Do?

- Installs and enables the SSH service.
- Starts the SSH service on boot.

Step 2: Modify SSH Configuration for Insecure Setup

Command:



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

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

Modify the following parameters:

- PermitRootLogin yes
- PasswordAuthentication yes
- Save and exit, then restart SSH:

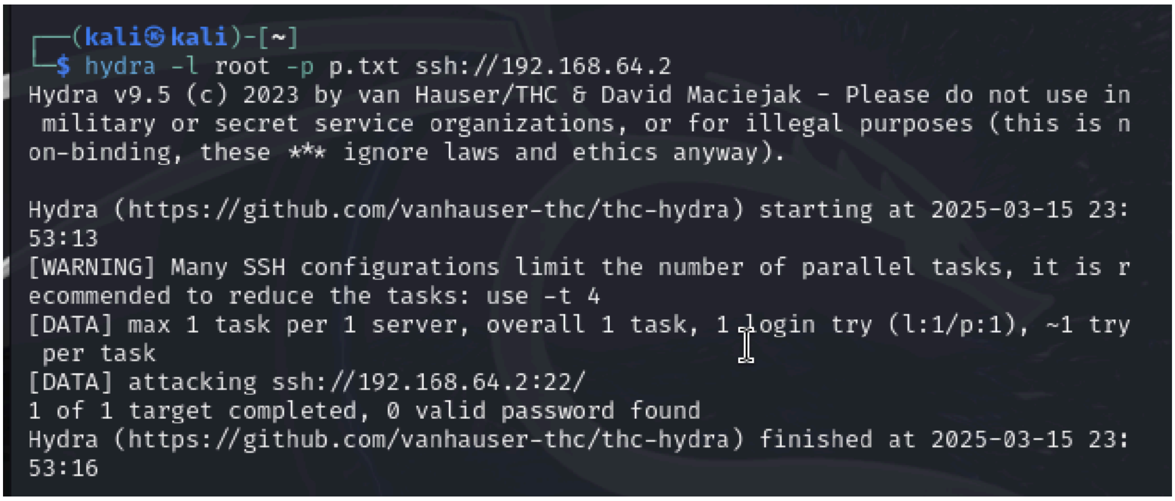
Security Risk:

- Allowing root login makes brute-force attacks easier.
- Password-based authentication is vulnerable to brute-force attacks.

2️⃣ Exploit: Brute-Force Attack on SSH

Step 3: Perform Brute-Force Attack using Hydra

- To find the ip of the system use command (ifconfig)



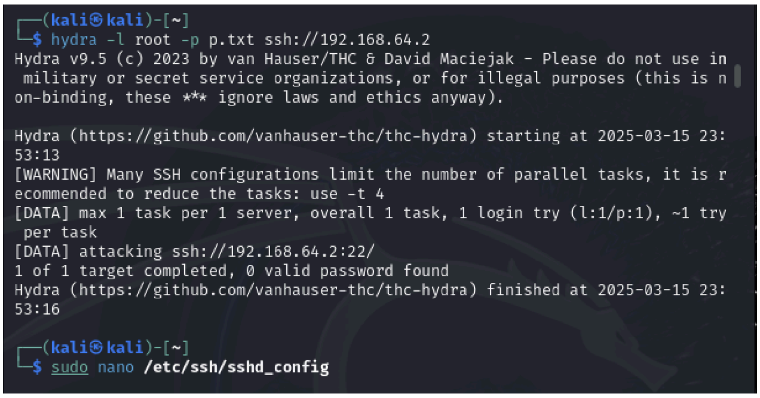
```
┌──(kali㉿kali)-[~]
└─$ hydra -l root -p p.txt ssh://192.168.64.2
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-15 23:
53:13
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try
 per task
[DATA] attacking ssh://192.168.64.2:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-15 23:
53:16
```

What Does It Do?

- Uses Hydra to brute-force SSH credentials.
- Targets the root user with a dictionary attack.

3️⃣ Mitigation: Securing SSH Access

Step 4: Disable Root Login & Enforce Key-Based Authentication



```
┌──(kali㉿kali)-[~]
└─$ hydra -l root -p p.txt ssh://192.168.64.2
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 2025-03-15 23:
53:13
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try
 per task
[DATA] attacking ssh://192.168.64.2:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-15 23:
53:16

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

Modify the following parameters:

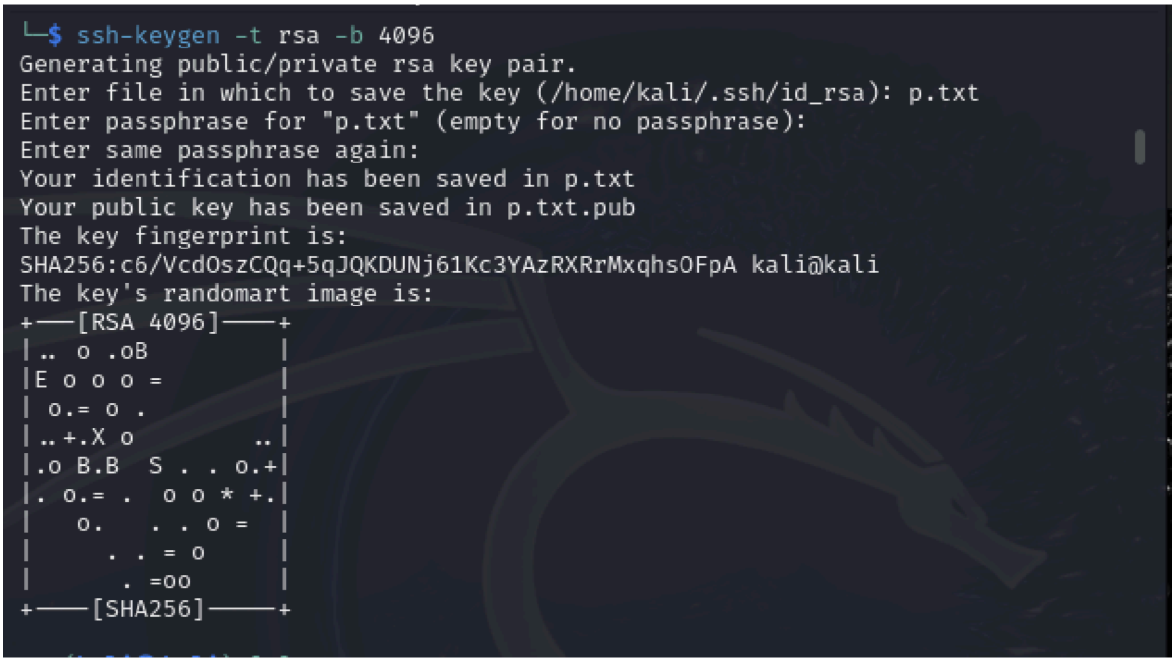
- PermitRootLogin no
- PasswordAuthentication no

- This has been done to enhance the security in the ssh config files
- Then restart the ssh

✅ Fixes: Prevents root login and enforces key-based authentication.

Step 5: Set Up Key-Based Authentication

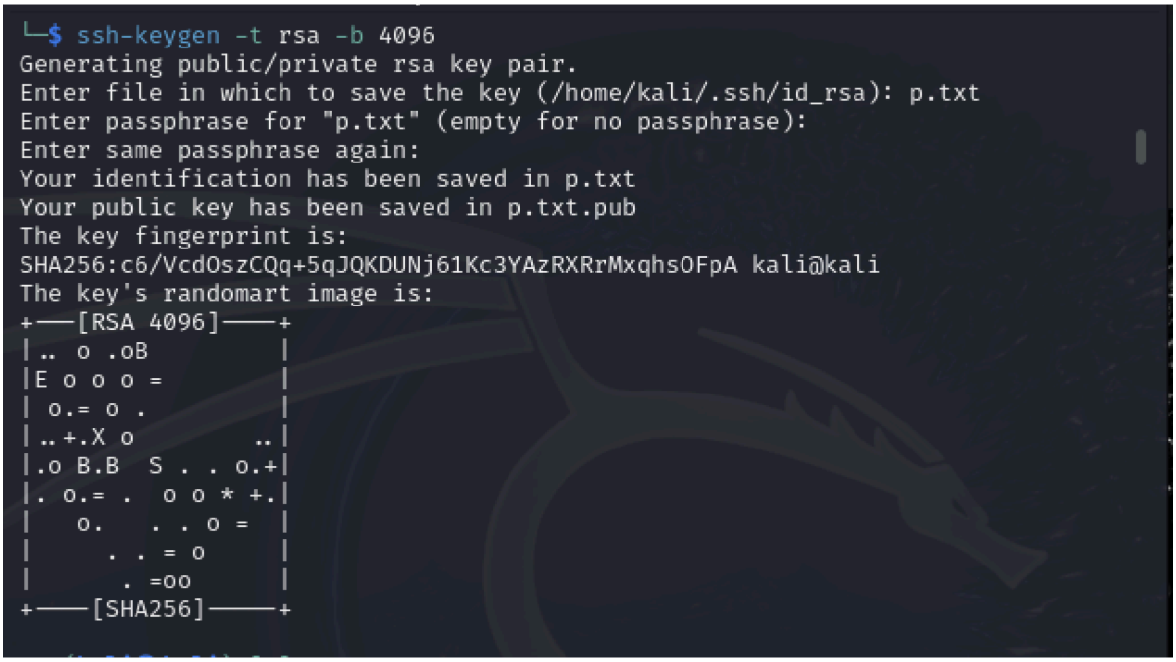
Command:



```
└─$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa): p.txt
Enter passphrase for "p.txt" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in p.txt
Your public key has been saved in p.txt.pub
The key fingerprint is:
SHA256:c6/VcdOszCQq+5qJQKDUNj61Kc3YAzRXRrMxqhsOFpA kali@kali
The key's randomart image is:
+---[RSA 4096]----+
|.. o .oB         |
|E o o o =        |
| o.= o .         |
|..+.X o        ..|
|.o B.B  S . . o.+|
|. o.= .  o o * +.|
|   o.   . . o =  |
|       . . = o   |
|        . =oo    |
+----[SHA256]-----+
```

What Does It Do?

- Generates an SSH key pair.



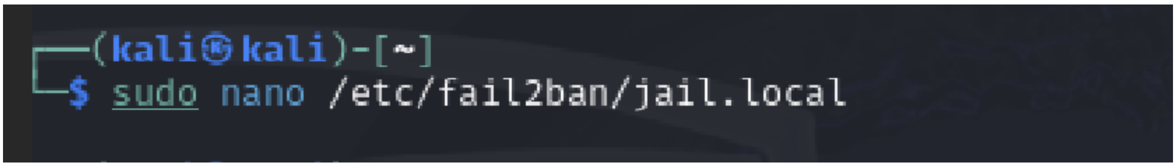
```
└─$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa): p.txt
Enter passphrase for "p.txt" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in p.txt
Your public key has been saved in p.txt.pub
The key fingerprint is:
SHA256:c6/VcdOszCQq+5qJQKDUNj61Kc3YAzRXRrMxqhsOFpA kali@kali
The key's randomart image is:
+---[RSA 4096]----+
|.. o .oB         |
|E o o o =        |
| o.= o .         |
|..+.X o        ..|
|.o B.B  S . . o.+|
|. o.= .  o o * +.|
|   o.    . . o = |
|      . . = o    |
|       . =oo     |
+----[SHA256]-----+
```

- Copies the public key to the target server for secure login.

Step 6: Configure Fail2Ban to Prevent Brute-Force Attacks

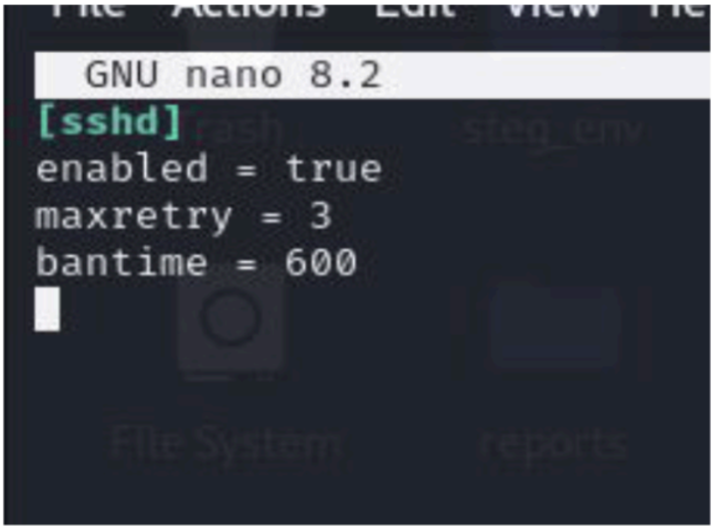
Command:

- (sudo apt install fail2ban -y) this command helps to download fail2ban which helps in protect against the brute force attack by monitoring and blocking the suspicious attempts



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

Add the following configuration:



```
GNU nano 8.2
[sshd]
enabled = true
maxretry = 3
bantime = 600
```

Save and restart Fail2Ban:

- sudo systemctl restart fail2ban

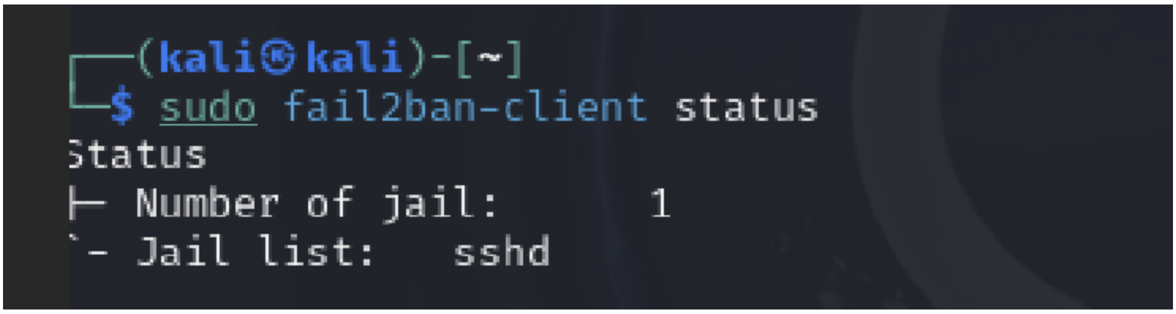
✅ Fixes:Automatically bans IPs after repeated failed login attempts.

📝 Conclusion:

Exploitation: Demonstrated how weak SSH settings allow brute-force attacks.
Mitigation: Implemented SSH hardening techniques to secure remote access.
Outcome:Attack surface significantly reduced, enhancing system security.

📍 **Status: Fixed & Hardened** ✅

- We can ensure that fail2ban status working properly by the following commands



```
┌──(kali㉿kali)-[~]
└─$ sudo fail2ban-client status
Status
|- Number of jail:      1
`- Jail list:   sshd
```