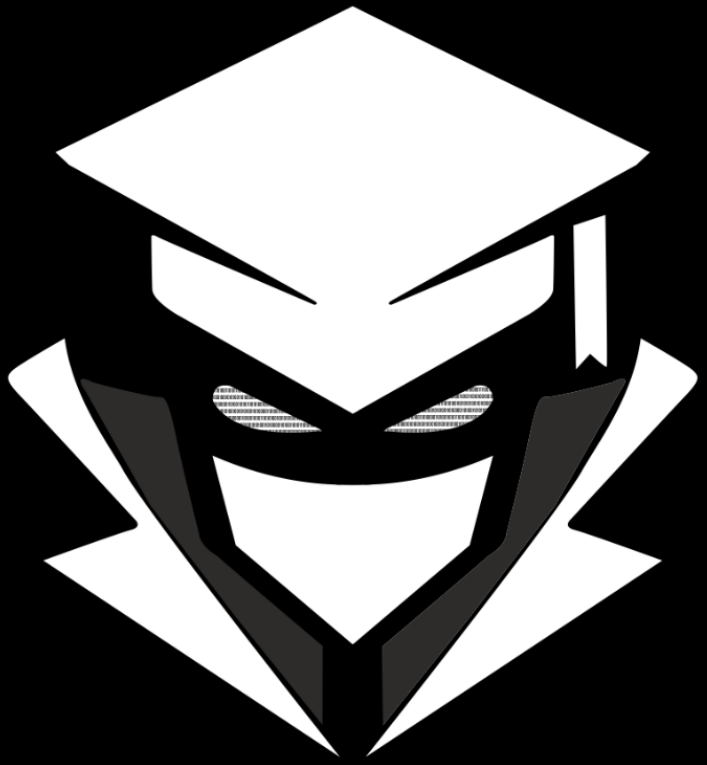

MISC

PRIVILEGE ESCALATION

What are we doing today?

Playing around with some pentesting concepts:

- Getting a reverse shell
- Upgrading to a fully interactive shell
- We'll do a practical demo at the end of the presentation

What are we doing today?

Playing around with some pentesting concepts:

- Getting a reverse shell
- Upgrading to a fully interactive shell
- We'll do a practical demo at the end of the presentation

Practising privilege escalation:

- How do we go from $ to #?
- Some guiding principles

Reverse Shells

- If we can execute arbitrary code on a remote machine, execute a command that gives us remote control

Reverse Shells

- If we can execute arbitrary code on a remote machine, execute a command that gives us remote control
- Many examples of such commands can be found out pentest-monkey's Reverse Shell Cheatsheet

Reverse Shells

- If we can execute arbitrary code on a remote machine, execute a command that gives us remote control
- Many examples of such commands can be found out pentest-monkey's Reverse Shell Cheatsheet
- Which one you use will depend on which programs/utilities are present on the machine

Reverse Shells

- If we can execute arbitrary code on a remote machine, execute a command that gives us remote control
- Many examples of such commands can be found out pentest-monkey's Reverse Shell Cheatsheet
- Which one you use will depend on which programs/utilities are present on the machine
- since `nc` is usually present by default on linux systems, this is usually a robust option

Reverse Shells

- If we can execute arbitrary code on a remote machine, execute a command that gives us remote control
- Many examples of such commands can be found out pentest-monkey's Reverse Shell Cheatsheet
- Which one you use will depend on which programs/utilities are present on the machine
- since `nc` is usually present by default on linux systems, this is usually a robust option
- With RCE, we can check for the presence of nc with `which nc`

I'm in... What now?

Enumerate

- This is why we're here! Look for more system misconfigurations/vulnerabilities
- Find out as much as we can!

I'm in... What now?

Enumerate

- This is why we're here! Look for more system misconfigurations/vulnerabilities
- Find out as much as we can!

Try to escalate privileges

- Our ultimate goal
- This facilitates further enumeration

Why privilege escalation?

In UNIX based systems, one of the most important security mitigations is restricting what users are allowed to do.

Why privilege escalation?

In UNIX based systems, one of the most important security mitigations is restricting what users are allowed to do.

For instance, most services will run in the privilege context of a service account, thus reducing the potential impact of that service becoming compromised.

Why privilege escalation?

In UNIX based systems, one of the most important security mitigations is restricting what users are allowed to do.

For instance, most services will run in the privilege context of a service account, thus reducing the potential impact of that service becoming compromised.

The unix administrative user, `root` is allowed to do anything however. If we can find a way to become this user, then we are no longer restricted in what we are able to do on a compromised system.

Enumeration: what am I looking for?

Sensitive information:

- Passwords/hashes
- ssh keys

Enumeration: what am I looking for?

Sensitive information:

- Passwords/hashes
- ssh keys

Misconfigurations:

- insecure sudo configurations
- insecure file permissions
- cron jobs running as root

Exploring misconfigurations

We are looking for misconfigurations that let us read, write or execute as root

Exploring misconfigurations

We are looking for misconfigurations that let us read, write or execute as root

Read:

- read `/etc/shadow` and crack passwords
- read any sensitive root owned files, eg ssh keys

Exploring misconfigurations

We are looking for misconfigurations that let us read, write or execute as root

Read:

- read `/etc/shadow` and crack passwords
- read any sensitive root owned files, eg ssh keys

Write:

- write a root user to `/etc/passwd`
- overwrite contents of sensitive scripts

Exploring misconfigurations

Execute

- Some processes can be escaped out of to get a shell in the current privilege context
- eg `!/bin/bash` to escape the `more` program
- python module hijacking (hint)
- See GTFObins for a nice list of binaries that might help escalate privilege

Some examples: finding files

Using the `find` command to find files with certain attributes

Some examples: finding files

Using the `find` command to find files with certain attributes

Find world writeable folders:

```
find / -perm -222 -type d 2>/dev/null
```

Some examples: finding files

Using the `find` command to find files with certain attributes

Find world writeable folders:

```
find / -perm -222 -type d 2>/dev/null
```

Find SUID binaries:

```
find / -perm -u=s -type f 2>/dev/null
```

Some examples: finding files

Using the `find` command to find files with certain attributes

Find world writeable folders:

```
find / -perm -222 -type d 2>/dev/null
```

Find SUID binaries:

```
find / -perm -u=s -type f 2>/dev/null
```

Find all files with 'password' in their name:

```
find / -iname *password* 2>/dev/null
```

Some examples: finding files

Using the `find` command to find files with certain attributes

Find world writeable folders:

```
find / -perm -222 -type d 2>/dev/null
```

Find SUID binaries:

```
find / -perm -u=s -type f 2>/dev/null
```

Find all files with 'password' in their name:

```
find / -iname *password* 2>/dev/null
```

Checkout `man find` to learn more about how to use it.

The VM:

Use virtualbox to start up the VM

- credentials: `user:password`
- `ifconfig` will tell you its IP

Navigating to this IP in your attack box will expose a trivial RCE vulnerability that you can use to practise getting a reverse shell (demo in a moment)

You can also SSH directly directly into the box using the credentials listed above.

The Challenge:

There are several privilege-escalation vulnerabilites in the machine:

- 1 takes you from `www-data` to `user`
- 1 takes you from `user` to `root`
- 2 take you from `www-data` to `root`

you have the box installed, have a play around and have fun!list of userful resources is in the github repo for this weeks workshop

PS: getting a fully interactive shell:

You might have found that your reverse shell is annoying to use

The following set of commands will upgrade your shell to be fully interactive:

In your reverse shell:

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

`ctrl-z` to background your shell

`stty raw -echo` (see `man stty` for a full explanation of this step)

then `fg` to foreground your shell