Ethical Student Hackers

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.

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

Reminder - EGM!

We are electing **new committee members**!

Help us **write sessions**, **make challenges**, and **run the society** - great way to learn, and great on your CV!

(You don’t have to be an expert - but it’s the best way to become one)

We are looking for:

- Publicity Officer
- General Member (up to 2)

If you’re interested, please **email us** at ethicalhackers@sheffield.ac.uk

Another quick reminder - if you finished the Give it a Go, come say hi at the end of session / email us and we’ll sort your prize

SHEFFIELD ETHICAL STUDENT HACKERS

What is Linux?

- Linux is an operating system similar to Windows and Mac OS
  - Linux is everywhere, because of its ability to be very small, very customisable and the ability to run on many devices

- It is Open Source!
  - This means that anyone can read/modify the code that makes Linux
  - Anyone can contribute to the Linux project
  - Windows and Mac OS are proprietary, so you cannot see their source code.

- It's a product of UNIX - 1960s
  - Therefore it's small and modular by design
  - Is the same family as MacOS
  - 'Everything is a file'

Distributions

As Linux is open source and anyone can edit it, there are multiple distributions by different maintainers that you can download.

There are many different distributions to choose from, you just need to have a look around to see what suits your needs:

- Mint
    - Very beginner friendly and similar to Windows
- Ubuntu
    - Very beginner friendly if transitioning from mac/windows, can be used on servers
- Debian / Kali
    - The predecessor to Ubuntu and Mint, Very stable, highly worked on and maintained
- Arch Linux
    - Rolling release, intermediate/expert user base, very customizable

Standard Navigation commands

There are a lot of commands for navigating the file structure in the terminal

cd	Change directory	cd /home/user
ls	List files in the directory	ls /home/user
find	Find files on the system	find / -name user.txt -type f
grep	Search for strings within files	grep "NAME" /etc/os-release
locate	Find files on the system (find)	locate hosts
mv	Move files	mv /home/user/old /home/user/new
cp	Copy files	cp /etc/passwd /home/user/passwd
rm	Remove file	rm /home/user/old_file
nano	Edit a text file (vim and vi)	nano /etc/hosts
cat	Output contents of a file	cat /etc/hosts
touch	Create a file	touch /home/user/abc
sudo / su	Execute code as another user	sudo ls /root / su - user

Connect

In Command Prompt or Powershell type:

```
ssh user[number]@100.25.220.244 -p 2222
```

Password:

```
SESH_0310_[number]
```

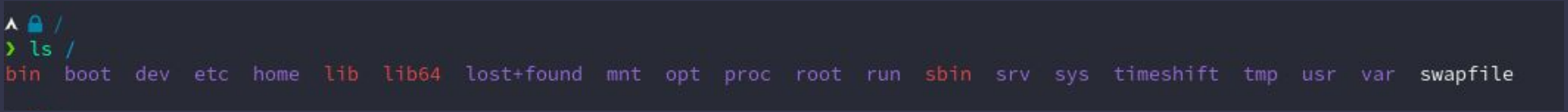
Tasks:

- What is in the file fruit.txt?
- What file is in the folder “tree”?
- What distribution and version of linux are you using?

In ConnectBot/JuiceSSH/similar:

```
user[number]@100.25.220.244:2222
```

Navigation - Folder structure



```
∧ 🔒 /
❯ ls /
bin  boot  dev  etc  home  lib  lib64  lost+found  mnt  opt  proc  root  run  sbin  srv  sys  timeshift  tmp  usr  var  swapfile
```

/bin

- Used for essential binaries (applications)

/etc

- ‘Etcetera’, contains the configuration of Linux and its applications

/home

- Contains the home directories of users

/root

- Contains the home directory of the root user

/opt

- Optional software, software that generally isn’t maintained by a package manager

/usr

- User binaries and program data

/sbin

- Binaries to be run as a root/sudo user

/var

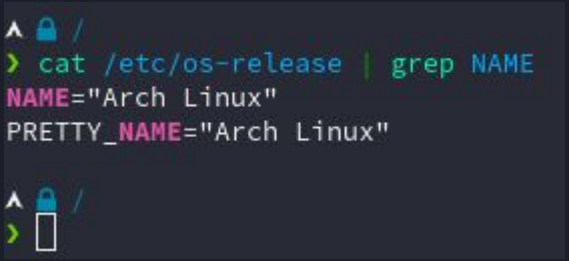
- Variable information, generally where running applications store necessary data

Piping & Redirection

So we know we can run commands in the terminal, but how do we run multiple commands and get the output we want from them? In the terminal there are 3 different ‘outputs’

stdin: 0 - Standard input
stdout: 1 - Standard output
stderr: 2 - Standard error

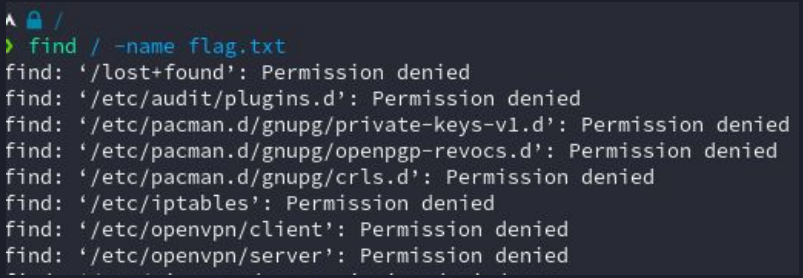
Piping



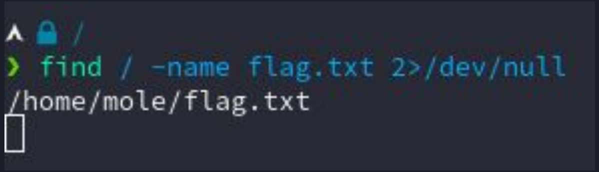
```
^ 🔒 /
❯ cat /etc/os-release | grep NAME
NAME="Arch Linux"
PRETTY_NAME="Arch Linux"

^ 🔒 /
❯ 
```

Redirecting errors



```
^ 🔒 /
❯ find / -name flag.txt
find: ‘/lost+found’: Permission denied
find: ‘/etc/audit/plugins.d’: Permission denied
find: ‘/etc/pacman.d/gnupg/private-keys-v1.d’: Permission denied
find: ‘/etc/pacman.d/gnupg/openpgp-revocs.d’: Permission denied
find: ‘/etc/pacman.d/gnupg/crls.d’: Permission denied
find: ‘/etc/iptables’: Permission denied
find: ‘/etc/openvpn/client’: Permission denied
find: ‘/etc/openvpn/server’: Permission denied
```



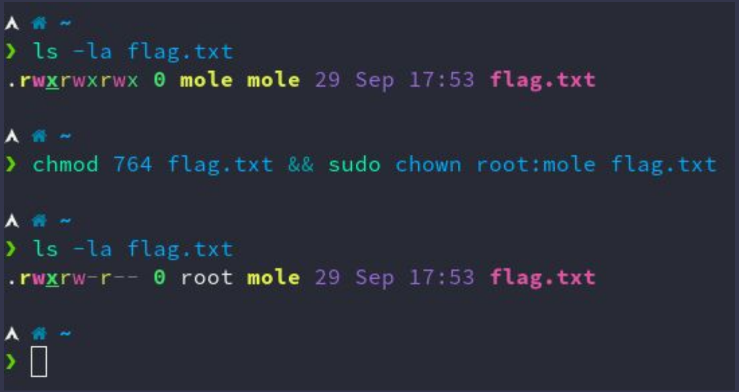
```
^ 🔒 /
❯ find / -name flag.txt 2>/dev/null
/home/mole/flag.txt
```

Permissions

Like every other operating system, Linux has file permissions that can be set. This stops data from being misused by members of the OS (if they're set correctly...)

Each file has 3 different type of permissions, Read, Write and Execute. These permissions can be individually assigned to the user who owns the file, the group who owns the file, and then everyone else.

You can also set the permissions and user/groups of files with the chmod and chown commands respectively. Read = 4, write = 2, execute = 1

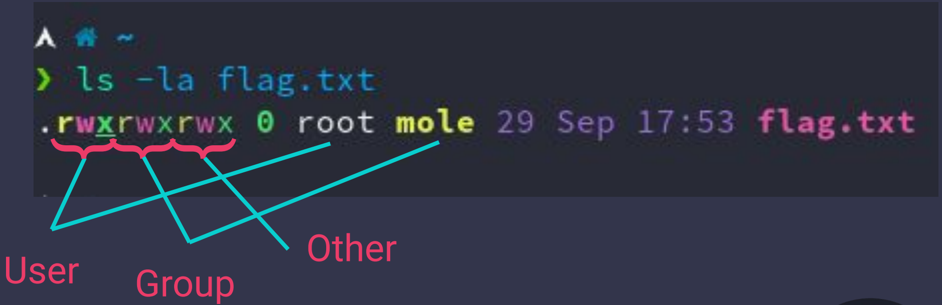


```
⋀ 🏠 ~
❯ ls -la flag.txt
.rwxrwxrwx 0 mole mole 29 Sep 17:53 flag.txt

⋀ 🏠 ~
❯ chmod 764 flag.txt && sudo chown root:mole flag.txt

⋀ 🏠 ~
❯ ls -la flag.txt
.rwxrw-r-- 0 root mole 29 Sep 17:53 flag.txt

⋀ 🏠 ~
❯ ▯
```



```
⋀ 🏠 ~
❯ ls -la flag.txt
.rwxrwxrwx 0 root mole 29 Sep 17:53 flag.txt
```

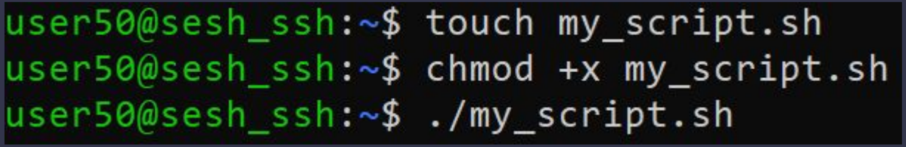
Sheffield Ethical Student Hackers

Scripts

You can put bash commands in a file and have these run in order.

- They usually have the .sh extension
- They will start with #!/bin/bash
- You need to have execute permissions for a script before you can run it

```
#!/bin/bash
echo Text > file.txt
rm file.txt
```



```
user50@sesh_ssh:~$ touch my_script.sh
user50@sesh_ssh:~$ chmod +x my_script.sh
user50@sesh_ssh:~$ ./my_script.sh
```

Installing software

Different distros have different ways of installing software, for example Ubuntu uses the apt package manager.

Package managers

- Make it incredibly easy to install and maintain software packages - A lot easier than Windows
- They use repositories to store the packages, you can add your own repositories to query from too
- Generally updating all packages can be done in one or two commands
- You don’t need to manually update software packages

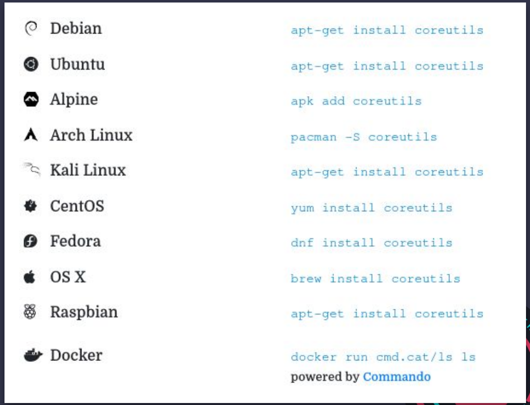
`sudo apt update` - Ubuntu update package repositories

`sudo apt upgrade` - Ubuntu upgrade packages

`apt list --installed` - Ubuntu list installed packages

`sudo apt install coreutils` - Install a package

https://command-not-found.com/



Debian	`apt-get install coreutils`
Ubuntu	`apt-get install coreutils`
Alpine	`apk add coreutils`
Arch Linux	`pacman -S coreutils`
Kali Linux	`apt-get install coreutils`
CentOS	`yum install coreutils`
Fedora	`dnf install coreutils`
OS X	`brew install coreutils`
Raspbian	`apt-get install coreutils`
Docker	`docker run cmd.cat/ls ls`

powered by Commando

Cancelling Commands

Ctrl + C - will stop a command that is currently running

Ctrl + D - will exit SSH and return to Command Prompt or PowerShell

Exiting Editors:

Nano - Ctrl + X -> y -> [Enter]

Vi - [Esc] -> : -> wq -> [Enter]

Challenges

1. Remove the file called “remove_this”
2. What version of gzip is installed? Can you output only this line?
3. Where is the file “magic.mgc” located? Can you hide the error messages?
4. Output the date command into date.txt
5. Make and run a script that prints your name.
6. What is the name of the user that is a fruit? What happens when you try to delete the account?
7. What command is your user running 5 instances of? End only one of of these processes.

Some of these might require you to look up how to do these.

Need More?

https://overthewire.org/wargames/

Start with Bandit: https://overthewire.org/wargames/bandit/

Connect: `ssh bandit0@bandit.labs.overthewire.org -p 2220`

Creds: bandit0:bandit0

OvertheWire is a great way to learn weird tricks that are unique to Unix (and might be helpful when hacking)

Linux Security - A Teaser Trailer

What happens when file permissions are incorrect?

What happens when scheduled tasks are run as privileged users?

What happens when you can modify files used by privileged programs?

How do you harden SSH? What if someone has a terrible password?

How can you limit access to network services?

How can you use what we've learned today to obscure payloads?

For fun… what does this do? :(){ :|:& };:

Have a play with Linux - it's highly customisable, and the control you have can help with Cyber

Knowledge of paths, privileges, and more will help you debug, especially shells!

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

10th October - Guest Talk from YHROCU

17th October - EGM + Introduction to Automation

24th October - BadUSB (Hardware)

Any Questions?

www.shefesh.com

Thanks for coming!