Linux for New Hackers - Intro

Sheffield Ethical Student Hackers

SHEFFIELD Ethical Student Hackers

Breaking into 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.

SHEFFIELD Ethical Student Hackers
Breaking into security.

Code of Conduct

- Before proceeding past this point you must read and agree 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://wiki.shefesh.com/doku.php?id=code_conduct

SHEFFIELD Ethical Student Hackers
Breaking into security.

Today's session

- Assumption made that you're a total beginner
- What Linux is and what it's used for
- The command line vs GUI
- Basic navigation commands
- Accounts and privileges
- Understanding files, directories and permissions
- Package management
- Getting help from within the OS

SHEFFIELD Ethical Student Hackers

Breaking into security.

Linux

- Started with the Linux kernel upon which all Linux-based distributions are built upon
- Kernel – underlying software (monolithic for Linux) which interacts with the hardware of the machine. OS interacts with the kernel through APIs, kernel interacts with the hardware
- Pronounced Lie-nux (named after Linus Trovaldis) so apologies that I keep pronouncing it incorrectly
- Ubuntu, CentOS, Debian and Linux Mint are all examples of Linux-based distros
    - Distro – an OS built upon the Linux kernel, distros are differentiated by the packages bundled with them and how processes are managed

Linux – where it's used

- You're almost guaranteed to interact with a Linux-based device on a daily basis
- Smartphones (Android), embedded systems, web servers, desktops
- Free as in beer and free as in speech
- Massive, actively maintained open source project backed by volunteers and large organizations
- Customise to own requirements
- Extremely lightweight but also feature-rich

Coming from other Operating Systems

- Don’t worry – plenty of GUIs (desktop environments) exist for Linux, GNOME is the most common
- We’re focusing on getting comfortable with the CLI, it’s one of the features that makes Linux so powerful
- For hacking, it’s an essential skill, equally for anyone interested in Linux Systems Administration.
- Sometimes you just won’t have a choice (servers, embedded systems)

The Shell

- A text-based command interpreter.
- Takes input from **stdin** (keyboard) and outputs to **stdout** (display). Errors go to **stderr** which is also typically the display
- Bash is the most common shell used, but all shells behave largely the same
- If you're using a desktop environment, Ctrl+Alt+T is usually the shortcut to open a terminal emulator where you can interact with the shell

For today's session

- If you want to also give some commands a go, the easiest option would be to use JSLinux
- https://bellard.org/jslinux/
- In future sessions where we're interacting with the system more, it'd be advisable to configure a Virtual Machine (instructions will go on the Wiki before the next session)
- If you can't get a VM working on your laptop (missing VT-x, insufficient specs, VirtualBox weirdness) – I can configure a terminal for you in the cloud before the next session. Message the page/jkr16 on Discord

Remotely accessing a Linux system

- SSH (Secure Shell) is a network protocol used for remote access with Linux systems
- Runs on TCP port 22
- Can authenticate with username and password, or preferably with keys + passphrase

Who am I?

- Now that we're on the system, it'd be good to know who we're logged in as
- w -> who -> whoami
- The symbol after the ~ is also a useful indicator
  - # denotes you are root
- Root is the most privileged account in a Linux system, able to modify and execute everything
  - With great power comes great responsibility
  - Do you need to run everything as root

Hacking note

- Becoming ‘root’ of a system is usually the ultimate goal for any hacker in a system
- With root permissions you essentially have full control of all resources on a system
- Explains why Jack regularly says *‘Popping a root shell’*

Sudo

- Sometimes processes will require superuser privileges. This can be achieved using the sudo command
  - Su (superuser) do (do something)
- The process' owner must have the correct permissions
  - Sudoers group
  - sudo –l
- Also possible to allow sudoers to become root without needing to share the root password
  - sudo su (superuser do switch user)

Adding/deleting users

- Some distros feature aliases for other commands. A notable one is the adduser command
- Traditionally, users are added with the useradd command. Arguments need to be passed and flags set depending on how you'd like to configure the user.
- Running adduser on some distros will instead run through a script which makes it significantly more friendly to add a user to the system

- Commands in Linux tend to be self-explanatory by name, deleting a user can be done with deluser or userdel

Where am I?

- pwd
- The ‘root’ of the filesystem is at ‘/’ – note that this is not the same as the root user
- Regular users typically have a home directory where all their personal files are stored
  - /home/username
- The root **user** has a home directory at /root
- The ~ in the command prompt denotes you are in your home directory
- This can be used for relative file paths from your home
- Note that levels in the file path are separated by / as opposed to \ in Windows systems

Moving around in Linux

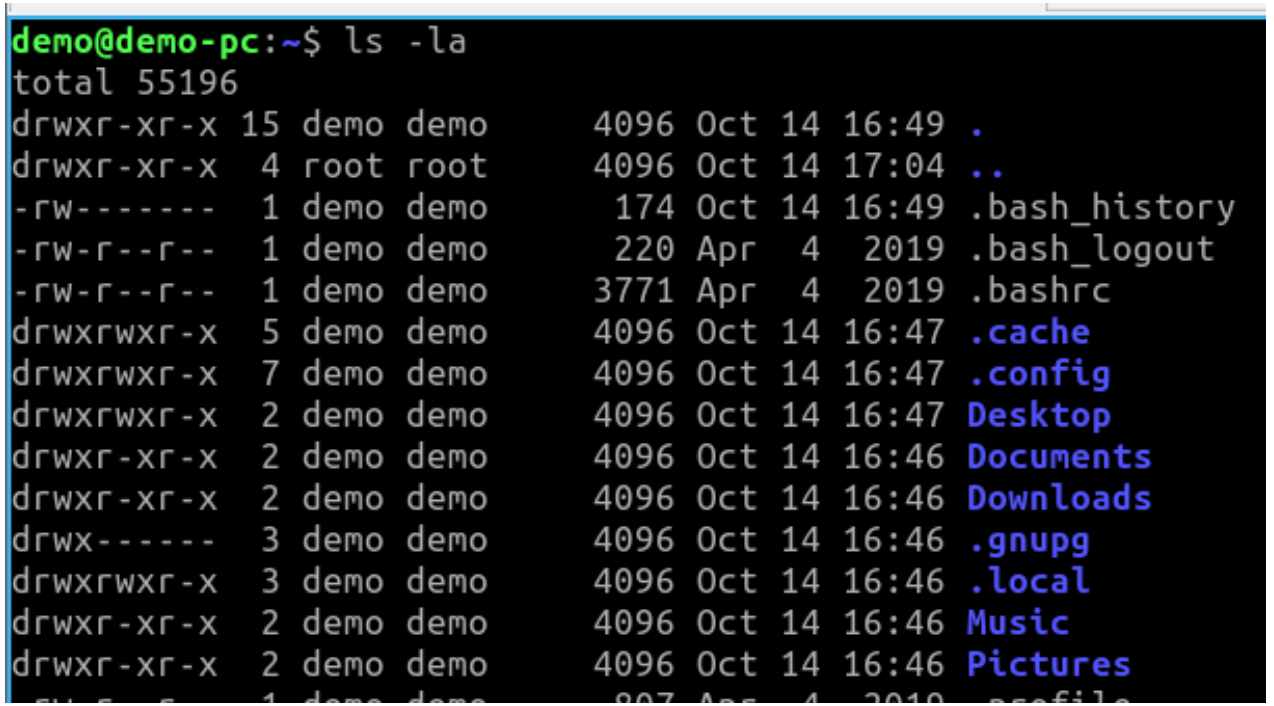
- Changing directories (folders) is done with the cd command
- Navigation is all relative to the current directory position
  - E.g. if I am in /home/john and do ‘cd SuperSecretHacks’ I’ll move to the directory /home/john/SuperSecretHacks
- Prepend ‘/’ to the argument for absolute file paths
- The argument ‘.’ denotes the current present working directory
- You can cd back to your home directory with ‘cd ~’ or simply cd
- You can skip back to the last directory you were in with ‘cd –’

Listing files

- Listing the contents of a directory is done with the ‘ls’ command
- Various flags will modify the verbosity/format of the output
  - ls –l will format the output as a vertical list
  - ls –a will show ‘hidden files’
  - Combine together to ls –la or ‘ll’ for a vertical comprehensive list of files in the directory
  - ls –R will recursively list (the ls command will travel down all possible paths and list contents of each subdirectory)
- Hidden files in Linux have names beginning with ‘.’
- ls –l also shows us some more interesting information

File permissions in Linux

- String of 10 characters
  - Can also be digits
- First character denotes type
  - - is a regular file
  - d is a directory
- Permissions are grouped
  - User – owner of the file
  - Group – members of the same group as owner
  - Other – everyone else



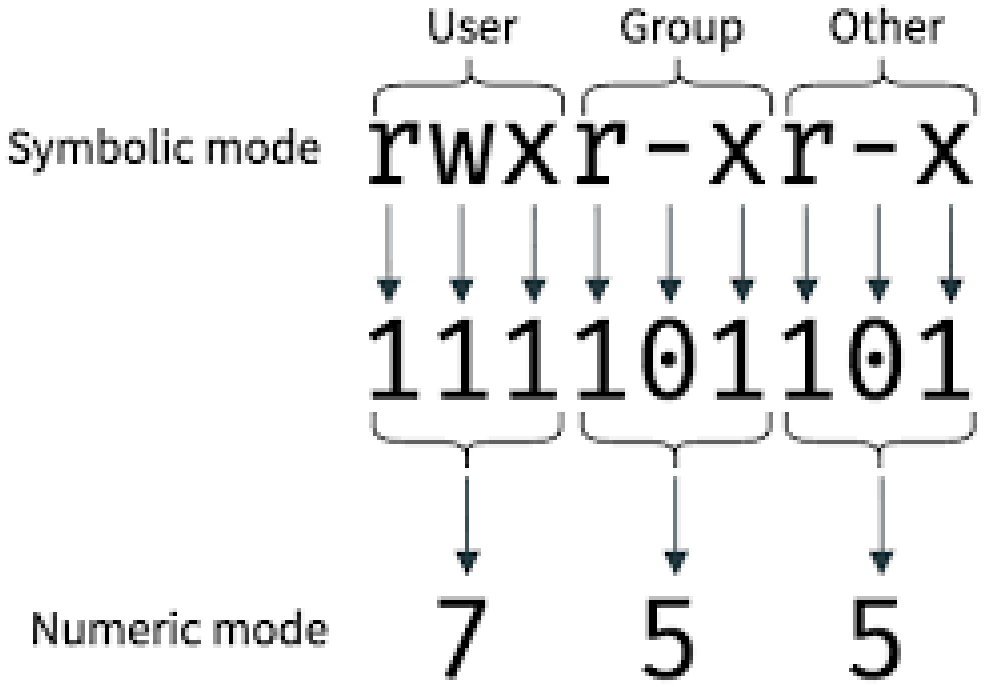
Groups in Linux

- Users can belong to groups in Linux, this can be useful for delegating different permissions dependent upon role
- Group-wide permissions can be useful for shared environments and resources
- Users can belong to multiple groups and are also a member of their own group
- Sudoers belong to the sudo/wheel group

Explaining file permissions in Linux

- UGO as before
- Example: - rwx r-x r--
- Breaking down:
  - This is a file
  - The owner of the file can read, write and execute the file
  - Members of the same group as the owner can read and execute the file
  - Everyone else can read the file
- An ‘s’ instead of an ‘x’ denotes a SUID bit, file will be executed with permissions of the owner
- Can also be represented in a numeric form (most likely encountered with FTP/webservers)

File permissions in Linux



SHEFFIELD Ethical Student Hackers
Breaking into security.

Changing permissions of files

- chmod (change mode)
- Can use numeric form or with characters
- Need to quickly make a file executable?
  - chmod +x – keep existing permissions but add ability to execute

Hacking note

- An adversary can leverage poorly configured file permissions to elevate their privileges (aiming to become root or a sudo user)
- For example, if I have a script which is owned by root and can be edited and executed by another user
  - They could replace the contents of the file to run a script to start a shell process as the root user
- Equally, some applications run as root have ways of breaking out into command execution

Deleting in Linux

- Use this carefully!
- rm command - remove
- Will only remove files by default, to use on directories add the –r flag
  - rm –r
- This will ask you for confirmation of deletion of each file
  - Can be annoying for large directories, add the –f force flag to bypass
- Extreme caution with the –rf flag, you could accidentally break your system
- Never do ‘rm –rf /’

Reading text files

- The quickest way to read the contents of a file is to use the ‘cat’ command
- ‘cat /home/john/dontopen.txt’
- cat will output the contents to **stdout**
- Other related commands include:
  - tail – read x number of lines from the bottom up
  - head – read x number of lines from the top down
- For larger documents – less or more.
  - Less is functionally better than more…
  - Allows for scrolling

Basic file creation

- You can quickly make an empty file with the ‘touch’ command
- More usefully, you can redirect an ‘echo’ command into a file with the > operator
  - echo ‘Can you hear me?’ > cave.txt
- To append to files use the >> (shovel) operator
  - echo ‘Yes I can!’ >> cave.txt

More advanced text editing

- We will cover this in the next session!
- People get weirdly passionate about their preferred text editor
- nano – beginner friendly and becoming more prominent
- vi – the hardcore enthusiasts choice, has been around forever
- vim – viMproved – adds more functionality to vi, consequently very powerful
- emacs – if that's your kinda thing

Package management

- This is implemented differently in different distros
  - Debian based distros use .deb with dpkg/apt
  - RHEL based distros use .rpm with rpm/yum
- Repositories are where packages are stored for download and install
- Package management implemented properly as opposed to Windows – no stray .exes
- Centralised updates for packages and a standardised method
- More practical examples next week

Getting help inside of Linux

- If you aren't sure of the correct usage of a command, you can utilise the manpages
  - 'man nano' would bring up the man(ual) pages for the nano application
- Some applications have a significantly shorter guide with the --help flag
  - More for understanding syntax
- If you know what sort of functionality you want, but can't remember the command name
  - apropos will search all the man pages for keywords