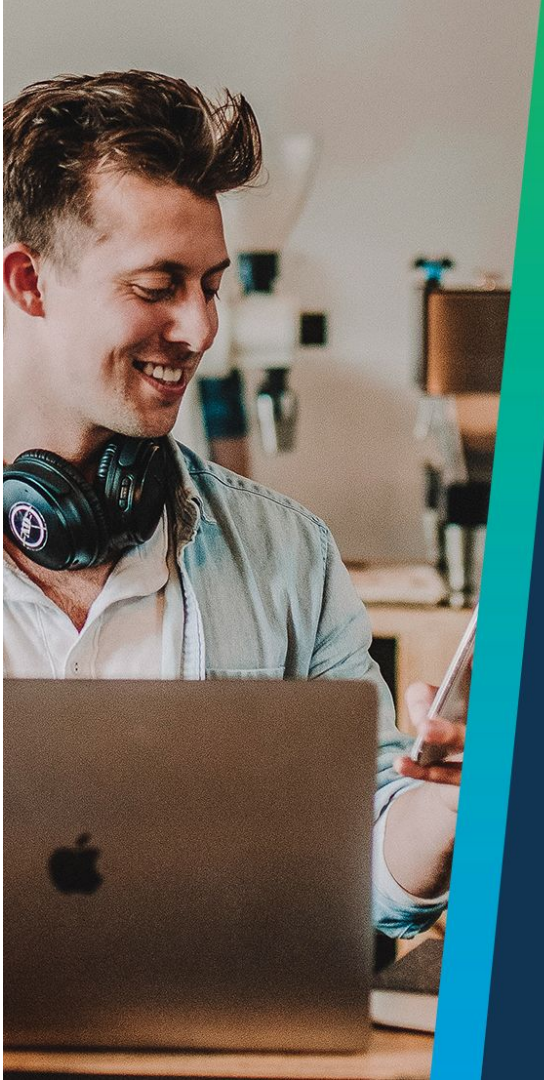


Cyber Security Bootcamp

Hyperiondev

Linux & Bash

Lecture - Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
- No question is daft or silly - ask them!
- There are Q/A sessions at the end of the session, should you wish to ask any follow-up questions.
- For all non-academic questions, please submit a query: www.hyperiondev.com/support
- Report a safeguarding incident: http://hyperiondev.com/safeguardreporting

Objectives

1. Define Linux and Bash.
2. Explore Linux commands and connect these to Bash scripting.
3. Implement Bash as a scripting language for Unix based systems.

Poll

What is linux?

Poll

What is a *shebang* (#!) in a Bash script?

Linux

Linux

- **Open Source:** Linux is free and open-source, allowing users to view, modify, and distribute the source code.
- **Security**: Linux is known for its robust security features and lower susceptibility to viruses and malware.
- **Customizability:** Users can modify every aspect of the OS, including the graphical user interface, system behavior, and installed software.
- **Stability and Performance:** Linux is stable and performs well under heavy workloads, making it ideal for servers and development environments.
- **Community Support**: A vast community of developers and users provides support, updates, and a plethora of open-source applications.
- **Lightweight:** Linux can run on older hardware with minimal resources, extending the lifespan of devices.

BASH

THE BOURNE-AGAIN SHELL

Bash

Bash is a popular Unix shell and command language in Linux and macOS. It includes scripting, file redirection, and piping, among other features.

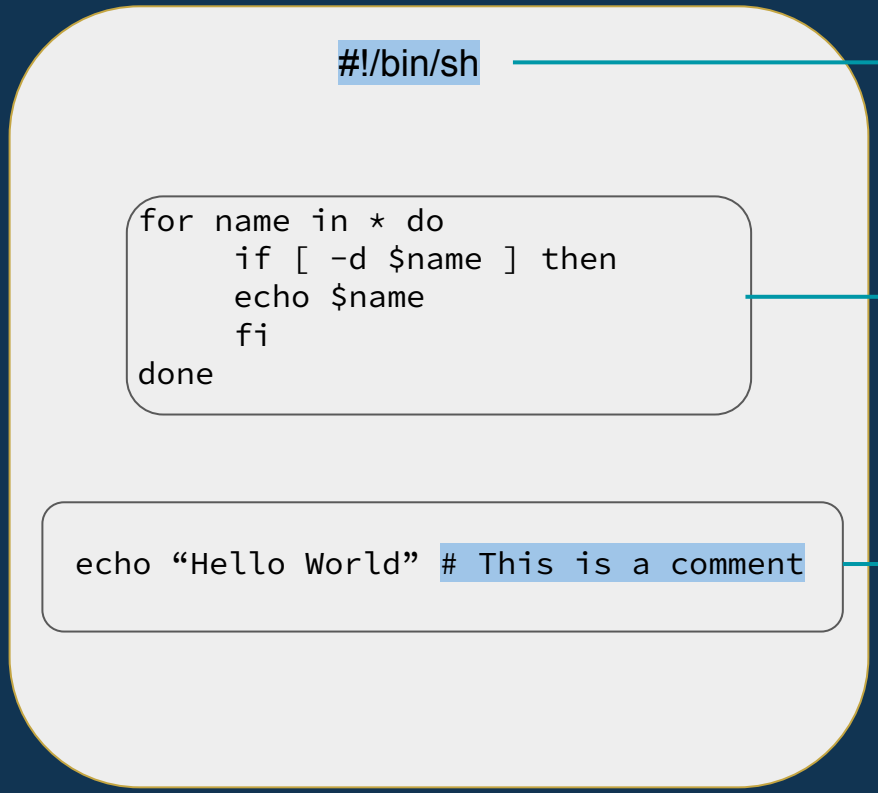
Shell scripting is a critical skill for cyber security professionals because it allows them to automate routine tasks, perform system monitoring, and detect and respond to security threats quickly.

Commands

- **ls** – Displays information about files in the current directory.
- **pwd** – Displays the current working directory.
- **mkdir** – Creates a directory.
- **cd** – To navigate between different folders.
- **rmdir** – Removes empty directories from the directory lists.
- **cp** – Moves files from one directory to another.
- **mv** – Rename and Replace the files
- **rm** – Delete files
- **uname** – Command to get basic information about the OS
- **locate**– Find a file in the database.
- **df** – Check the details of the file system
- **wc** – Check the lines, word count, and characters in a file using different options

- **locate**– Find a file in the database.
- **touch** – Create empty files
- **ln** – Create shortcuts to other files
- **cat** – Display file contents on terminal
- **clear** – Clear terminal
- **ps-** Display the processes in terminal
- **man** – Access manual for all Linux commands
- **grep-** Search for a specific string in an output
- **echo-** Display active processes on the terminal
- **wget** – download files from the internet
- **whoami-** Create or update passwords for existing users
- **sort-** sort the file content
- **cal-** View Calendar in terminal
- **whereis** – View the exact location of any command types after this command

Anatomy of a Shell Script



shebang!

UNIX! Commands, Syntax (looping, selecting/referring to variables)

Comments

File Ownership(UGO)

User

The file's owner is a user. A file's creator automatically becomes its owner. Consequently, a user may also be referred to as an owner.

Group

Multiple users may be included in a user group. All members of a group will have the same access privileges to the file under Linux. Consider a project where several people need access to a file. You could add all users to a group and give the group read/write access to all files instead of giving each user their own individual permissions. This would prevent anyone outside of the group from reading or changing the files.

Other

Simply anyone else who is authorized to access a file. This User has not created the file nor does it belong to a user group that could be the owner of the file. In actuality, it refers to everyone else. As a result, setting permissions for the world also refers to setting permissions for others.

File Permissions

Octal Representation	String Representation	Permission Type
0	---	No permission
1	--x	Execute only
2	-w-	Write only
3	-wx	Write & Execute
4	r--	Read only
5	r-x	Read & Execute
6	rw-	Read & Write
7	rwx	All (Read, Write & Execute)

File Permissions

```
drwxr-xr-x. 1 supervillain supervillain   22 Jun  5 18:53 folder
-rw-r--r--. 1 supervillain supervillain 8669 Jun  5 20:48 forecast.txt
```

Lesson Conclusion and Recap

- Linux Basics: You learned the importance of Linux. Understanding these basics empowers you to perform tasks more efficiently and customize your environment to suit your needs.

- We covered the key components of Bash scripting, including shebangs, text editors, file permissions, variables, conditional statements, loops, and arrays. These skills allow you to automate repetitive tasks, create powerful scripts, and manage system operations effectively.

- Through examples like changing file permissions, managing software updates, and creating scripts for specific tasks, you gained hands-on experience in using Linux commands and Bash scripts to solve real-world problems.

Hyperiondev

Q & A Section

Please use this time to ask any questions relating to the topic explained, should you have any

Hyperiondev

Thank you for joining us

Take regular breaks.
Stay hydrated.
Avoid prolonged screen time.
Remember to have fun :)