Linux Fundamentals

Where is Linux Used?

It's fair to say that Linux is a lot more intimidating to approach than Operating System's (OSs) such as Windows. Both variants have their own advantages and disadvantages. For example, Linux is considerably much more lightweight and you'd be surprised to know that there's a good chance you've used Linux in some form or another every day! Linux powers things such as:

- Websites that you visit
- Car entertainment/control panels
- Point of Sale (PoS) systems such as checkout tills and registers in shops
- Critical infrastructures such as traffic light controllers or industrial sensors

Flavours of Linux

The name "Linux" is actually an umbrella term for multiple OS's that are based on UNIX (another operating system). Thanks to Linux being open-source, variants of Linux come in all shapes and sizes - suited best for what the system is being used for.

For example, Ubuntu & Debian are some of the more commonplace distributions of Linux because it is so extensible. I.e. you can run Ubuntu as a server (such as websites & web applications) or as a fully-fledged desktop. For this series, we're going to be using Ubuntu.

Note: Ubuntu Server can run on systems with only 512MB of RAM!

Similar to how you have different versions Windows (7, 8 and 10), there are many different versions/distributions of Linux.

Commands

echo	Output any text that we provide
whoami	Find out what user we're currently logged in as!

Command	Full Name
ls	listing
cd	change directory
cat	concatenate
pwd	print working directory
Find	siehe unten
grep	siehe unten
wc	Count the number of entries

Outputting the Contents of a File (cat)

Whilst knowing about the existence of files is great — it's not all that useful unless we're able to view the contents of them.

We will come on to discuss some of the tools available to us that allows us to transfer

files from one machine to another in a later room. But for now, we're going to talk about simply seeing the contents of text files using a command called "**cat".**

"Cat" is short for concatenating & is a fantastic way for us to output the contents of files (not just text files!).

In the screenshot below, you can see how I have combined the use of "ls" to list the files within a directory called "Documents":

```
tryhackme@linux1:~/Documents$ ls
todo.txt
tryhackme@linux1:~/Documents$ cat todo.txt
Here's something important for me to do later!
```

Finding out the full Path to our Current Working Directory (pwd)

You'll notice as you progress through navigating your Linux machine, the name of the directory that you are currently working in will be listed in your terminal.

It's easy to lose track of where we are on the filesystem exactly, which is why I want to introduce "**pwd**". This stands for **p**rint **w**orking **d**irectory.

Using the example machine from before, we are currently in the "Documents" folder — but where is this exactly on the Linux machine's filesystem? We can find this out using this "pwd" command like within the screenshot below:

```
tryhackme@linux1:~/Documents$ pwd
/home/ubuntu/Documents
tryhackme@linux1:~/Documents$
```

Using Find

The find command is fantastic in the sense that it can be used both very simply or rather complex depending upon what it is you want to do exactly. However, let's stick to the fundamentals first.

Let's start simple and assume that we already know the name of the file we're looking for — but can't remember where it is exactly! In this case, we're looking for "passwords.txt"

If we remember the filename, we can simply use find -name passwords.txt where the command will look through every folder in our current directory for that specific file like so:

Using "find" to find a file with the name of "passwords.txt"

```
tryhackme@linux1:~$ find -name passwords.txt
./folder1/passwords.txt
tryhackme@linux1:~$
```

"Find" has managed to *find* the file — it turns out it is located in folder1/passwords.txt — sweet. But let's say that we don't know the name of the file, or want to search for every file that has an extension such as ".txt". Find let's us do that too!

We can simply use what's known as a wildcard (*) to search for anything that has .txt at the end. In our case, we want to find every .txt file that's in our current directory. We will construct a command such as find -name *.txt . Where "Find" has been able to *find* every .txt file and has then given us the location of each one:

Using "find" to find any file with the extension of ".txt"

```
tryhackme@linux1:~$ find -name *.txt
```

```
./folder1/passwords.txt
./Documents/todo.txt
tryhackme@linux1:~$
```

Using Grep

Another great utility that is a great one to learn about is the use of grep. The grep command allows us to search the contents of files for specific values that we are looking for.

Take for example, the access log of a web server. In this case, the access.log of a web server has 244 entries.

Using "wc" to count the number of entries in "access.log"

```
tryhackme@linux1:~$ wc -l access.log
244 access.log
tryhackme@linux1:~$
```

Using a command like cat isn't going to cut it too well here. Let's say for example if we wanted to search this log file to see the things that a certain user/IP address visited? Looking through 244 entries isn't all that efficient considering we want to find a specific value.

We can use grep to search the entire contents of this file for any entries of the value that we are searching for. Going with the example of a web server's access log, we want to see everything that the IP address "81.143.211.90" has visited (note that this is fictional)

Using "grep" to find any entries with the IP address of "81.143.211.90" in "access.log"

```
tryhackme@linux1:~$ grep "81.143.211.90" access.log
81.143.211.90 - - [25/Mar/2021:11:17 + 0000] "GET / HTTP/1.1" 200 417 "-" "Mozilla/5.0 (Linux; Android 7.0;
Moto G(4))"
tryhackme@linux1:~$
```

"Grep" has searched through this file and has shown us any entries of what we've provided and that is contained within this log file for the IP.

Eigene Recherche

What is an Access Log?

An access log is a record of all the requests made to your server. This includes information about the request itself, such as the request method (GET, POST, etc.), the requested URL, and the user agent. It also includes information about the server’s response, such as the status code and the size of the response.

Access logs are created by your web server, such as Apache or Nginx, and can be configured to include additional information, such as the IP address of the user making the request, the time and date of the request, and the referrer (the website that the user was on before making the request).