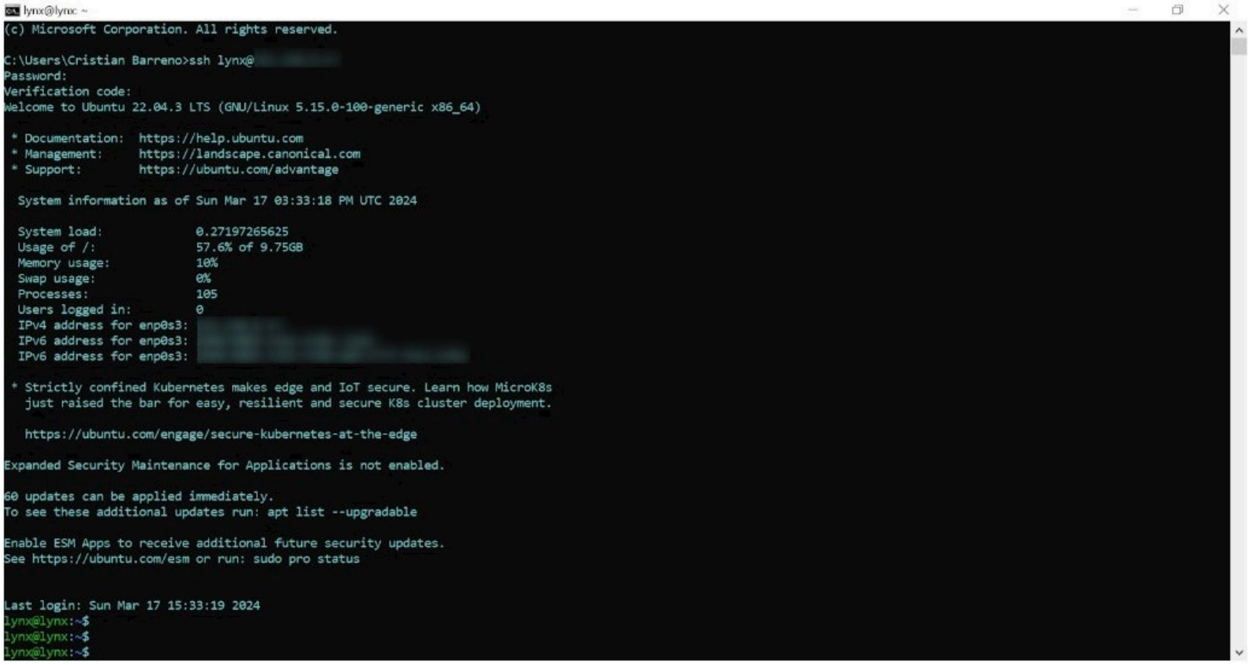
Name: Cristian Barreno

## Encoding vs Hashing vs Encryption



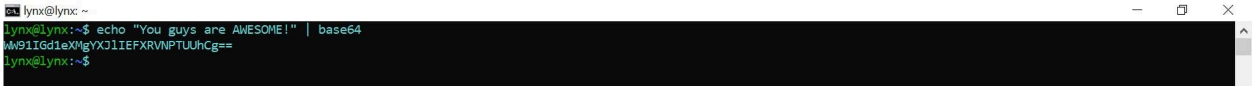
```
(c) Microsoft Corporation. All rights reserved.

C:\Users\Cristian Barreno>ssh lynx@[redacted]
Password:
Verification code:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-100-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Mar 17 03:33:18 PM UTC 2024

  System load:            0.27197265625
  Usage of /:             57.6% of 9.75GB
  Memory usage:           10%
  Swap usage:             0%
  Processes:              105
  Users logged in:        0
  IPv4 address for enp0s3: [redacted]
  IPv6 address for enp0s3:
  IPv6 address for enp0s3:

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

60 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sun Mar 17 15:33:19 2024
lynx@lynx:~$
lynx@lynx:~$
lynx@lynx:~$
```

## Task 1

### Echo

```
echo "You guys are AWESOME!" | base64
```

Collect the output.



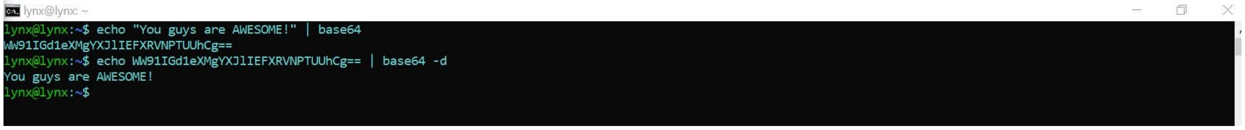
```
lynx@lynx:~$ echo "You guys are AWESOME!" | base64
WW91IGd1eXMgYXJlIEFXRVNPTUUhCg==
lynx@lynx:~$
```

## Task 2

```
echo “<output from previous command>” | base64 -d
```

The -d in base64 decodes instead of encodes

The output should be exactly what you put in the encode in Task 1



```
lynx@lynx:~$ echo "You guys are AWESOME!" | base64
WW91IGd1eXMgYXJlIEFXRVNPTUUhCg==
lynx@lynx:~$ echo WW91IGd1eXMgYXJlIEFXRVNPTUUhCg== | base64 -d
You guys are AWESOME!
lynx@lynx:~$
```

## Task 3

`echo This is evil naughty naughty malware > malware.txt`

This redirects the output of the echo to a file. No error or feedback means that it completed successfully.
Validate that it worked by:

`cat malware.txt`

The output should be exactly what you put in the echo but its saved to the file.
Now let's encode the output of the file and then save the encoded output to a different file.

`cat malware.txt | base64 > notmalwarenoreally.txt`

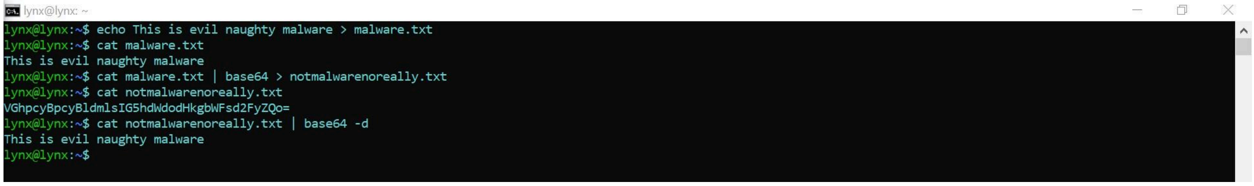
The encoded output should be exactly what you put in the echo but its saved to the file.
Validate that it worked by:

`cat notmalwarenoreally.txt`

It should be encoded and NOT human readable.
Now let's validate, by reversing the output of the encoded message.

`cat notmalwarenoreally.txt | base64 -d`

You should get This is evil naughty naughty malware
This proves that the encoding and decoding was successful.



```
lynx@lynx:~$ echo This is evil naughty malware > malware.txt
lynx@lynx:~$ cat malware.txt
This is evil naughty malware
lynx@lynx:~$ cat malware.txt | base64 > notmalwarenoreally.txt
lynx@lynx:~$ cat notmalwarenoreally.txt
VGhpcyBpcyBldmlsIG5hdWdodHkgbWFsd2FyZQo=
lynx@lynx:~$ cat notmalwarenoreally.txt | base64 -d
This is evil naughty malware
lynx@lynx:~$
```

## Task 4 – HASHING and validating:

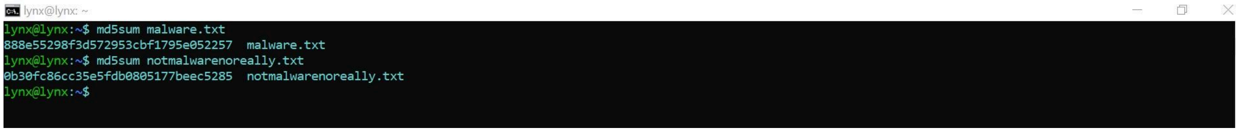
In linux one can hash a file by using the `md5sum` command.
Let's hash the original file that we are pretending is malware.

`md5sum malware.txt`

Now let's hash the file, which STILL contains the malware in an encoded format.

```
md5sum notmalwarenoreally.txt
```

Please note the above 2 hashes are different, this means that if I was using signature based antivirus, it WOULD NOT CATCH the encoded and obfuscated malware



```
lynx@lynx:~$ md5sum malware.txt
888e55298f3d572953cbf1795e052257  malware.txt
lynx@lynx:~$ md5sum notmalwarenoreally.txt
0b30fc86cc35e5fdb0805177beec5285  notmalwarenoreally.txt
lynx@lynx:~$
```