BTA 2023 ©

Encoding vs Hashing vs Encryption

EXERCISE 1

Task 1

Echo

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

Collect the output. (No, I’m not telling you, you have to prove it to yourself that you got it right)

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

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 lets 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.

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
```

The output should be as follows:

```
de54f727fe1eb6d1d0ca9411db64024a  malware.txt
```

Now let’s hash the file, which STILL contains the malware in a encoded format.

```
md5sum notmalwarenoreally.txt
```

The output should be as follows:

```
6606d6223afdfe6c969d31197e9400e1  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.