LAB: Basic Hashing

NOTE: If you are using updated software the screenshots may be different

UNIX or Mac OS X

Login to your server:

```
[groupN.apnictraining.net] [192.168.30.N]
```

- Replace N with your group number
- Username `apnic` and password `training`

1. Create a document for Hashing. Open a terminal window and type the following:

   ```
   echo "This is the first document we will HASH." > hash_file
   ```

2. View the file content

   ```
   cat hash_file
   ```

3. Generate a md5 hash of the document

   ```
   md5sum hash_file
   ```

   The output will be:

   ```
   3ef84e12ff8099f4a91c9251850e7cc3 hash_file
   ```

4. Now edit the `hash_file` and calculate the md5 hash again.

   ```
   echo "This is the first document we will HASH" > hash_file
   ```

   - This will delete the full stop(.)

5. Calculate the md5 hash again;

   ```
   md5sum hash_file
   ```

   The output will be:

```
afc1bc9dc0afa58cb38f73523ae2e1ec hash_file
```

6. Notice how changing just one character results in an entirely different hash value!
7. Repeat the same using the SHA algorithm (default is SHA-1), and observe the hash values.

```
shasum hash_file
```

Windows - File Checksum Integrity Verifier

NOTE: If you are using updated software the screenshots may be different

1. On your Windows machine. Download Microsoft's File Checksum Integrity Verifier from the following link:

   ```
   https://www.microsoft.com/en-us/download/details.aspx?id=11533
   ```

   Extract it to `c:\FileChecksum`. It will extract two (2) files, `fciv.exe` is a command line utility that computes verifies hashes of files and `ReadMe.txt` is the instructions.

2. Create a document for Hashing. Open a command prompt and type the following:

   ```
   cd c:\FileChecksum
   echo "This is the first document we will HASH." > hash_file
   ```

3. To check the hash of `hash_file` file, type the following:

   ```
   fciv.exe hash_file
   ```

   The output will be similar to the following: `db90c62d9463bd1fbdd821bded209204 hash_file`

4. The default hash is `MD5`. To check the `SHA-1` use the flag `-sha1`

   ```
   fciv.exe -sha1 hash_file
   ```

   The output will be similar to the following:

   ```
   2c42fbd816057a54a06c9b54e145dd9b35ff4634 hash_file
   ```

   To check for `MD5` & `SHA-1` use the flag `-both`

   ```
   fciv.exe -both hash_file
   ```

   The output will be similar to the following:

```
MD5                              SHA-1
 ----------------------------------------------------------------------------
db90c62d9463bd1fbdd821bded209204 2c42fbd816057a54a06c9b54e145dd9b35ff4634 hash
_file
```

5. Now edit the `hash_file` and calculate the md5 hash again.

```
echo "This is the first document we will HASH" > hash_file
```

   - This will delete the full stop(.)

6. Calculate the md5 hash again;

```
fciv.exe hash_file
```

   The output will be similar to the following:

```
cebf34f8183f6167e93e62c436af074d hash_file
```

7. Notice how changing just one character results in an entirely different hash value!

8. Repeat the same using the SHA algorithm (default is SHA-1), and observe the hash values.

```
fciv.exe -sha1 hash_file
```

LAB 2: Hashing important files

NOTE: If you are using updated software the screenshots may be different

UNIX or Mac OS X

Login to your server:

```
[groupN.apnictraining.net] [192.168.30.N]
```

- Replace N with your group number
- Username `apnic` and password `training`

1. Create a document storing the hash values of important files. Open a terminal window and type the following:

```
find /etc -name passwd -exec md5sum {} \; > passwd_hash_v1.txt
find /etc -name *.conf -exec md5sum {} \; > conf_hash_v1.txt
find /etc -name *.yaml -exec md5sum {} \; > yaml_hash_v1.txt
```

2. View the hash values that have been saved to the file

```
cat passwd_hash_v1.txt
```

3. To change the contents of the passwd file, create a new user

```
sudo useradd -s /bin/bash -d /home/user01/ -m -G sudo user01
```

4. Create another copy of the hash value for the passwd file

```
find /etc -name passwd -exec md5sum {} \; > passwd_hash_v2.txt
```

5. Check if the hashes are different.

```
diff -y passwd_hash_v1.txt passwd_hash_v2.txt
```

   The ouptut shows there is a different hash value. From this you could then run a schedule task, to create a file for comparison of hash values for important files that should not change. When a change occurs this should be investigated or logged.

If time permits, create a list of important files, that should be monitored for changes. ```