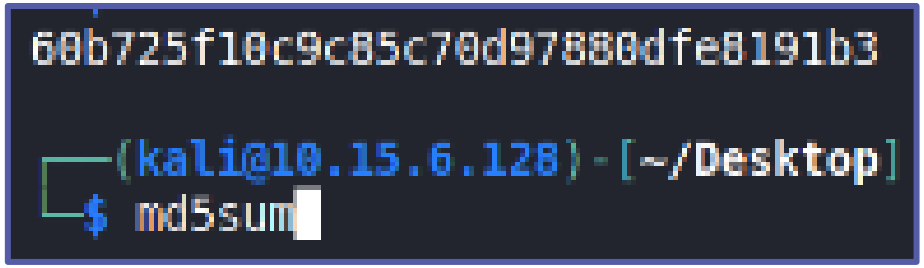
Cybersecurity

File Hashing Lab

CYBER.ORG

File Hashing Lab

- Materials needed
  - Kali Linux Virtual Machine
- Software Tools used (all from from Kali Linux OS)
  - **md5sum**
  - **sha1sum**
  - **sha256sum**
  - **sha512sum**



Objectives Covered

- Security+ Objectives (SY0-601)
  - Objective 2.1 - Explain the importance of security concepts in an enterprise environment
    - Hashing
  - Objective 2.8 - Summarize the basics of cryptographic concepts
    - Hashing
  - Objective 3.2 - Given a scenario, implement host or application security solutions
    - Database
      - Hashing
  - Objective 4.5 - Explain the key aspects of digital forensics
    - Integrity
      - Hashing

What is a file hash?

- A file hash is the hash which results from putting a file's contents through a hashing algorithm.
- A hash is a one-way function, you cannot retrieve the contents of the file from the hash but you can authenticate the contents of the file have not been altered by comparing a stored hash and running a hash on the current version of a file.
- Usage – if downloading a sensitive document or compiled code online, how can you be sure the contents have not been altered? Hashing.

The File Hashing Lab

- Setup Environment
- Locate a file
- Locate the hashes
- Generate the hash
- Compare the hashes
- Hashing “nothing”

Setup Environment

- Log into your range
- Open the Kali Linux Environment
  - You should be on your Kali Linux Desktop
  - Open the Terminal

Locate a File

- `PuTTY` is a popular SSH client for Windows users.
- SSH as you may recall is a secure way to access a remote server.
  It would be a tempting target for an attacker to compromise your SSH client! That would allow them to run all sorts of attacks. MiTM, keylogger, etc.
- To ensure integrity of the files, the creators of `PuTTY` provide the hash values of all their downloadable files.

Locate a File

- Download PuTTY
  - Navigate to https://www.putty.org
  - Click the Download PuTTY
  - Click the download link for MSI: 64-bit x86
  - This URL will download the latest 64-bit version of PuTTY for Windows.

Locate the Hashes

- Now that the file has been downloaded, how can we ensure it has not been compromised?
- Hashes are long, so be sure to copy/paste the resulting hash.
- With the cyber range open in one tab, open the PuTTY download page in another tab.
  Scroll to the very bottom to the section titled "*checksums*".
- Click on the first option: MD5
  https://the.earth.li/~sgtatham/putty/latest/md5sums

Generate the Hash

- Open a terminal and navigate to Download

  `cd Downloads`

- With putty in the current folder, generate the MD5 hash of it:

  `md5sum putty` (press the tab button to autocomplete, should see something like putty-64bit-0.76-installer.msi)

- You should get something like:

  `f838fdafd0881cf1e6040a07d78e840d`

  This is the hash for version 0.80

```
┌──(kali@10.15.7.174)-[~/Downloads]
└─$ md5sum putty-64bit-0.80-installer.msi
3b4e8f0f7cb8b32f85b60abed701ba0b  putty-64bit-0.80-installer.msi
```

Compare the Hash

- Highlight the hash output and copy it.
- Open the tab with the MD5 hashes from the PuTTY site.
- Press CTRL+F to *find* text on the page and paste in the MD5 hash.
- Did it find the hash in the long list of files on that page?
  - Yes? Great! This means nobody has tampered with the file.
  - No? Make sure you copied the entire hash and you have the correct version of hashes from the PuTTY team. Still not a match? Uh-oh! Someone may be tampering with your copy of `putty`!

Other hashes

- MD5 is susceptible to hash collisions.
- We can use the Secure Hashing Algorithm (SHA) to be sure.
- Which SHA?
  - `SHA1`
  - `SHA256`
  - `SHA512`

  …whichever suits your “good enough” criteria!

Hashing with SHA

- The process is the exact same as md5 only the command is different.
- `PuTTY` makes all 3 options available on the *checksums* section.
  - SHA1* = https://the.earth.li/~sgtatham/putty/latest/sha1sums
  - SHA256 = https://the.earth.li/~sgtatham/putty/latest/sha256sums
  - SHA512 = https://the.earth.li/~sgtatham/putty/latest/sha512sums
- To hash:
  - SHA1 = **`sha1sum putty.exe`**
  - SHA256 = **`sha256sum putty.exe`**
  - SHA512 = **`sha512sum putty.exe`**

*Recall that like MD5, SHA-1 is cryptographically broken and insecure.

To be very sure of your hash, use SHA-2 in the form of SHA256 or SHA512!

Hash Lengths

- SHA1 results in a 160-bit (40 character) value
- SHA256 results in a 256-bit (64 character) value
- SHA512 results in a 512-bit (128 character) value

```
┌──(kali@10.15.7.174)-[~/Downloads]
└─$ sha1sum putty-64bit-0.80-installer.msi
9da4e411e0ca62fc452cb91a3b33c7c1621f746d  putty-64bit-0.80-install
er.msi
┌──(kali@10.15.7.174)-[~/Downloads]
└─$ sha256sum putty-64bit-0.80-installer.msi
858399ee9ee49e15a78c7018dbf0dd73dba8337d6f0adb841896ba553c9a646c
putty-64bit-0.80-installer.msi
┌──(kali@10.15.7.174)-[~/Downloads]
└─$ sha512sum putty-64bit-0.80-installer.msi
599b031199b9629549ac0d172726056b6fcd8248e7ef24e36c18e06f23038ed726
b354398b73cd3fb5d8a7ce4872b3fa9fc0fa191efcd5f35a2f3d02db222313  pu
tty-64bit-0.80-installer.msi
```

How can I trust my hashing tool?

- In Cybersecurity it's good to be paranoid!
- How can you be sure your hashing tool is trustworthy?
- Confirm you're installing what you think you're installing.
  - Hash the hashing tool when you install it.
  - Hash the Linux distribution when you install it.
  - Hash each tool you install.
- Hashing an "empty value" should provide well-known, published strings…

Hashing “nothing”

- The hashing tools in Kali require some sort of file input. It needs *something*. If you do not provide a file input, it will not return a hash.
- You can create an empty file and hash that:

```
touch foo.txt
md5sum foo.txt
```

- Or you can use the famous “nothing” file in linux: /dev/null

```
md5sum /dev/null
```

- Compare the output of either method with the following hashes:

Empty Hash Values

- Hashing “nothing” should give you the following known values:

Algorithm	Hash
MD5	D41D8CD98F00B204E9800998ECF8427E
SHA1	DA39A3EE5E6B4B0D3255 BFEF95601890AFD80709
SHA256	E3B0C44298FC1C149AFBF4C8996FB924 27AE41E4649B934CA495991B7852B855
SHA512	CF83E1357EEFB8BDF1542850D66D8007 D620E4050B5715DC83F4A921D36CE9CE 47D0D13C5D85F2B0FF8318D2877EEC2F 63B931BD47417A81A538327AF927DA3E