Handling and Safety

> https://github.com/HuskyHacks/PMAT-labs/blob/main/labs/0-1.HandlingAndSafety/Malware.Calc.exe.7z

Matt Kiely also known as HuskyHacks has explained pretty well, the importance on handling the malware and the safety precautions one should take on and before analyzing the malware.

Secondly, if you have created a malware or you want to share the malware with anyone else, we need to ensure that the executable/file is defanged. Now, what is defanged files.

It's simply apprending anything to the file, URL etc. to make it non-executable.

For example, you have analyzed a malware which is an executable and now you are required to be share it with your sub-ordinated. You'll do the following,

- Actual File → Malware.exe
- Defanged File → Malware.exe.KnockMe

Now, appending anythin to the file with a (.) at the end will make it non-executable because MS Windows is looking at the rightmost extension of the file. Which in this case is .KnockMe

which is not a valid extension thus, it will not run directly on the system.

Similar to the executable, we should also defang the IOCs like IP addresses and URLs. The common method which is being followed in the today's cyber market is as below.

- IP → 192.168.1.1
- Defanged IP → 192[.]168[.]1[.]1
- Domain → https://google.com/
- Defanged Domain → https[:]//google[.]com

Also, ensure to have the sample zipped which is password protected when you are keeping the sample and/or are sharing it.

Also, the most important thing while handling the malware is to calculate the hashes. This is to ensure that the file integrity remains intact and nothing in the file has changed during the download, handling and sharing etc.

There are plenty of ways to calculate hashes as there are plenty of tools available in the market.

- PowerShell → Get-FileHash -Algorithm
- Linux/Flare → md5sum, shasum etc.

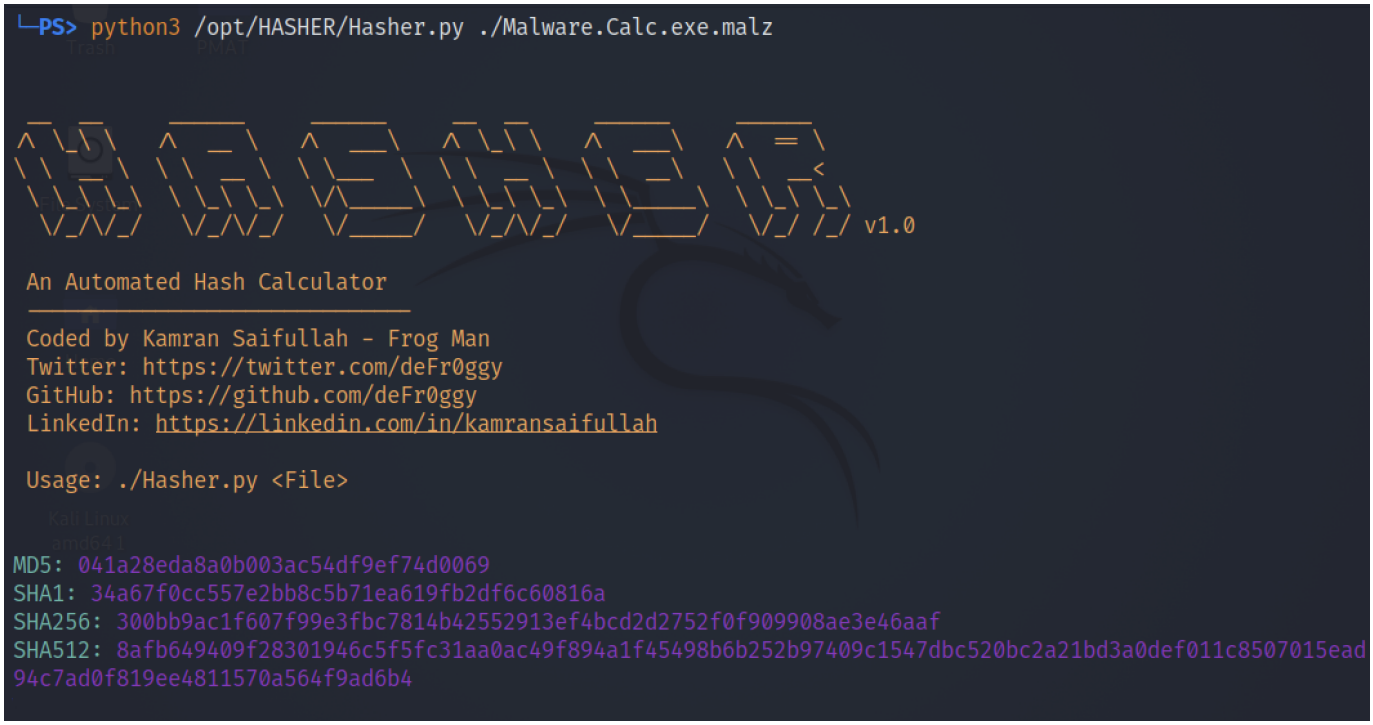
For my own self, I have created a python tool to calculate the following hashes at once for a single file.

1. MD5
2. SHA1
3. SHA256

4. SHA512

> https://github.com/deFr0ggy/HASHER

Now, we will download the executable from the PMAT-LABs GitHub Repo and will calculate it's hashes.



Thus, we have the following hashes in hand for our first sample.

1. MD5 → 041a28eda8a0b003ac54df9ef74d0069
2. SHA1 → 34a67f0cc557e2bb8c5b71ea619fb2df6c60816a
3. SHA256 → 300bb9ac1f607f99e3fbc7814b42552913ef4bcd2d2752f0f909908ae3e46aaf
4. SHA512 → 8afb649409f28301946c5f5fc31aa0ac49f894a1f45498b6b252b97409c1547dbc520bc2a21bd3a0def011c8507015ead94c7ad0f819ee4811570a564f9ad6b4

Ensure, you have s separate HOST-ONLY Adapters set for your VMs. Don't get your primary machine infected.

Follow Me

Twitter: https://twitter.com/deFr0ggy

GitHub: https://github.com/deFr0ggy

LinkedIn: https://linkedin.com/in/kamransaifullah