Ministry of Higher Education and Scientific Research

Northern Technical University

Technical Engineering College of Mosul

Cybersecurity and Cloud Computing Techniques Engineering

Hashing Functions

Submitted by:

مصطفى عبد العزيز حمزة

Introduction

Hash functions, such as MD5, SHA-1, SHA-256, SHA-3, and BLAKE2, play a crucial role in various cryptographic applications. These functions are used for digital signatures, public-key encryption, integrity verification, message authentication, password protection, key agreement protocols, and more.

In the real world, hash functions find widespread use across different domains:

1. Cloud Storage Systems: They help identify identical files and detect modifications.
2. Git Revision Control System: Hash functions are used to uniquely identify files within a repository.
3. Host-Based Intrusion Detection Systems (HIDS): These systems use hash values to detect changes in files.
4. Network-Based Intrusion Detection Systems (NIDS): Hashes are employed to identify known malicious data passing through a network.
5. Forensic Analysis: Hash values serve as evidence that digital artifacts remain unaltered.
6. Bitcoin: The proof-of-work system relies on hash functions.

How Hash Functions Work

Hash functions take an input of variable length and produce a fixed-size output, commonly referred to as a “digest.” Their primary purpose is to protect data integrity. Unlike ciphers that ensure data confidentiality, hash functions determine whether data has been modified. They achieve this by generating unique digests for each piece of data. Ideally, two distinct pieces of data should never correspond to the same digest. If they do, the hash function is considered vulnerable to collisions.

Additionally, hash functions enhance password security. When storing passwords in a database, they create digests. Even if a data breach occurs, these digests cannot be reverse-engineered to reveal the original input.

Playing with Python and Hashlib

Python language provides a built-in module called hashlib which contains many hashing algorithms such like: sha256, sha128, and md5.

For example, to generate a sha256 hash for the word ‘Plaintext’.

```
[1]: import hashlib

[2]: hashlib.sha256('Plaintext'.encode()).hexdigest()

[2]: '0707c5d972a7029d1696f45c9268cc1dbe2215ae2d6245724f46adc7fd998c46'
```

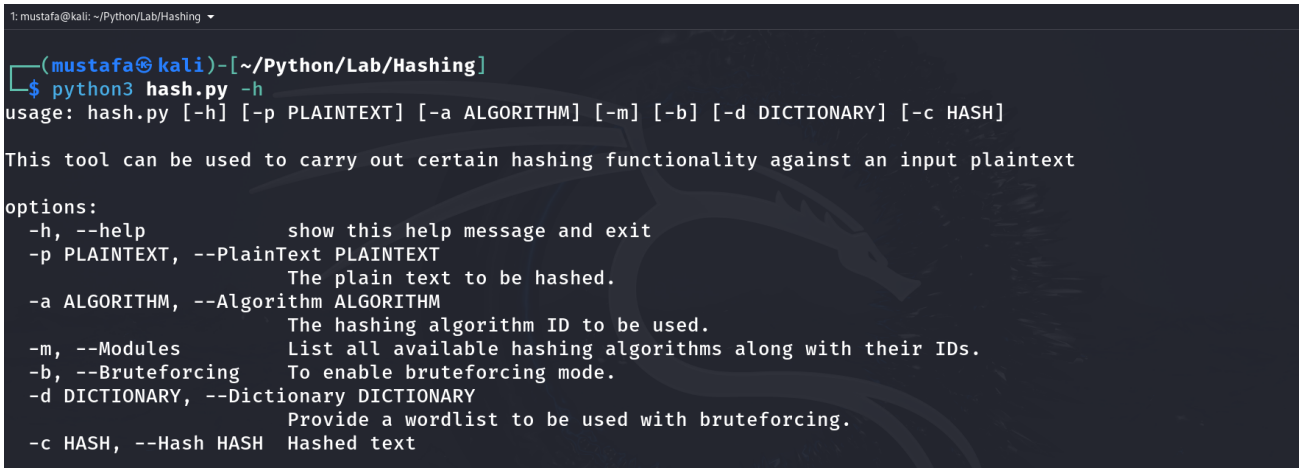
I programmed a Python tool to carry out hashing and brute-forcing processes, using the following Python modules

1- Hashlib for hashing.
2- Argparse to parse command-line arguments.
3- OS for operating system-related purposes
4- Other modules including: pandas, sys, colorama.

For the code from GitHub, scan the QR Code.

Lab

1. Show a help message



2. To list the available modules

3. To generate a sha256 hash for the word ‘Cybersecurty’

4. To generate a shake_128 hash for the words: “Never trust, always verify.”

5- To brute force the sha256 hash which was early generated. Using the rockyou wordlist.

6- To brute force the blake2s hash: **`93f9b0b205276f1c3b03464d5e890d6c99e7e22eb4d7530e53ba2e3a8066d103`**