Key Terms

Plaintext- Data before encryption or hashing. It can be a text, a photograph or other file.
Encoding- NOT a form of encryption. It is just a form of data representation like base64 or hexadecimal. It is reversible.
Hash- It is the output of the hash function.
Brute Force- Attacking cryptography by trying every different password or every different key.
Cryptanalysis- Attacking cryptography by finding weaknesses in the underlying maths.

What is Hash Function?

• It is different from encyption.
• It do not have a key, so it is considered impossible to go back to the input by using the output.
• The output of the hash function is of fixed size.
• Any small change in the input data, causes a large change in the output.

Hash Collision

• It occurs when 2 different inputs give the same output.
• Due to pegionhole effect, collisions are unavoidable.
• MD5 and SHA1 have been attacked, and made technically insecure due to engineering hash collisions. So, we shouldn't trust either algorithm for hashing passwords or data.

Uses for Hashing

Two main purposes of hashing in cybersecurity-
- To verify integrity of data.
- To verify passwords.

Hashing for password verification

We can't encrypt the passwords, as the key has to be stored somewhere. If someone gets the key, they can just decrypt the passwords.
So instead of storing the password, we just store the hash of the password. And if our database leaks then the attacker would have to crack each password to find out what the password was.
The only problem is that a user with same password will have same hash.

A “rainbow table” is a lookup table of hashes to plaintexts, so we can quickly find out what the password a user had just from the hash.

Example of rainbow table-

Hash	Password
02c75fb22c75b23dc963c7eb91a062cc	zxcvbnm
b0baee9d279d34fa1dfd71aadb908c3f	11111
c44a471bd78cc6c2fea32b9fe028d30a	asdfghjkl
d0199f51d2728db6011945145a1b607a	basketball
dcddb75469b4b4875094e14561e573d8	000000
e10adc3949ba59abbe56e057f20f883e	123456
e19d5cd5af0378da05f63f891c7467af	abcd1234
e99a18c428cb38d5f260853678922e03	abc123
fcea920f7412b5da7be0cf42b8c93759	1234567

Protecting against rainbow tables

• To protect against rainbow tables, we add a salt to the password.
• The salt is randomly generated and stored in the database and is unique to every user.
• So, that even if two users have the same password, they would have different hash due to different salt added to the password.
• Hash functions like bcyrpt and sha512crypt handle this automatically.

Recognizing Password Hashes

Automated hash recognizition tools exist such as https://pypi.org/project/hashID .
For hashes that have a prefix, the tools are reliable.
Windows passwords are hashed using NTLM (variant of md4).
On linux, password hashes are stored in /etc/shadow.
On Windows, password hashes are stored in SAM.

Quick table of the most UNIX style password prefixes that we see-

Prefix	Algorithm
1	md5crypt, used in Cisco stuff and older Linux/Unix systems
2, $2a$, $2b$, $2x$, $2y$	Bcrypt (Popular for web applications)
6	sha512crypt (Default for most Linux/Unix systems)

For more hash formats and password prefixes - https://hashcat.net/wiki/doku.php?id=example_hashes

Password Cracking

• One of the methods is rainbow tables to crack hashes that don't have a salt.
• If the hashes have salt we have to hash a large number of inputs with potentially adding the salt if there is one.
• Tools like Hashcat and John the Ripper are normally used for this.

Why Use GPUs?

• They are very good at maths involved in hash functions.
• We can use graphics card to crack most hash types much more quickly.
• Some hashing algorithms, like bcrypt, are designed so that hashing on a GPU is about the same speed as hashing on a CPU, hence resists the cracking of hash.

Note: Never use *-force* for hashcat as it can give false positives or false negatives.

Hashing For Integrity Checking

• Hashing can be used to check that files haven't been changed.
• If we put the same data in, we always get the same data out.
• If a single bit changes, the hash will change a lot.
• So, by hashing we can make sure that file haven't been modified or to make sure that they have downloaded correctly.
• We can also use hashing to find duplicate files, as they will have the same hash.

HMACs

• It is a method of using a cryptographic hashing function to verify the authenticity and integrity of data.
• They use a secret key and a hashing algorithm to produce a hash.