**Hashing**

**What is Hashing?**

**It is a one-way street** – Irreversible

Transforms plaintext into a shorter representation - Also called a hash function

Why is this needed?

hashing is used a lot of different types of systems

* Passwords
* Indexes
* File signature

A Password: will be hashed and kept in a lookup table or db locally

* The plaintext password is not kept on db
* When a password is entered, its hashed and compared to the hash stored locally

Index: Can hash a 200 digit index value into a smaller number that is still unique so it can still be used

File Signature: when a file is downloaded, it can be hashed and compared with he hash of the file from the source

* This ensures integrity, that the file is unchanged from the source to client, during the download

**MD5 Hashing** - Message Digest 5

Commonly used to verify the content of data

* Not commonly used for authentication any longer

Provides a unique output of a message digest of 128 bits ALWAYS

* No matter how many bits the data provided is – could be 1GB

Processes the data in 512-bit block, just like a block cipher

Each block is broken down into 16 words composed of 32 bits each

**SHA Hashing** – Secure Hashing Algorithms

Commonly used to hash passwords

Utilises salt in most cases

Common types

* SHA-1
* SHA-2 (SHA-224, SHA-386, SHA-512)
* SHA-3

**Salt**

Appending (or prepending) a value to the end of the ‘password’ before hashing to avoid dictionary attacks

1, 2 and 3 work similarly with some mathematical differences

* SHA-1 output a 160-bit message digest
  + No longer secure, has been brute forced in the past
  + No longer available
* SHA-2 and 3 output a 224-bit or 256-bit message