SY110
Hashing & Passwords

Major Brian Hawkins, USMC

U.S. Naval Academy

Fall AY 2018

Outline

Review

- What's the difference between steganography and encryption?
- Symmetric encription
  - Ceasar Shift Cipher - how to break?
  - Vigenere Cipher - better, but still breakable

Hash Functions

A hash function takes an input (string, file, etc.) and outputs a number. Hash functions:

- Help guarantee message **integrity**
  - See if a message has been tampered with
- Are used for password validation
  - Can compare a stored hash of the correct password...
  - to the hash of the submitted password.
  - If the same, then submitted password is correct
  - Thus, hashes also support **authentication** and **non-repudiation**

Hashes

Cryptographic hashes should have several properties:

- It should be easy to compute the hash of an input
- It should be very difficult to determine what input produced a particular hash (*one-way* function)
- It should be very difficult to find two values that hash to the same value

`https://simple.wikipedia.org/wiki/Cryptographic_hash_function`

Passwords

Often, we don't want to store every user's password in a file on a server to authenticate. Why?

A hash from the past...

We could simply store the hash of the password on the server.

- Then, when a user wants to log on, we can compute the hash and compare it to that user's hash we have stored.

```
username --- hash of password
user1234 --- N91asbADB9AaC
CompSciGuy --- XQFk9asdldA72
```

Salt

Couldn't I just precompute the hashes of all possible passwords?
Yes! I could just then look up the hash in my *dictionary*, and find what created that hash.

Not just for rubbing in wounds...

- We just throw a little *salt* in the mix
- By concatenating the password with a unique salt value (random value)
- We effectively make it much more difficult to precompute all possible hashes

Key stretching & Throttling

Server administrators often employ two other techniques to make life more difficult for would-be password theives:

- Key Stetching
  - Instead of storing a hash, store the $10,000^{th}$ hash of the hash of the . . . of the hash of the password
  - Makes cracking the password take 10,000 times as long
  - Effective against *offline* attacks
- Throttling
  - Wait a small, fixed amount of time before informing a user of an incorrect password
  - If accidental, only a mild annoyance. If trying to brute-force attack a password, can make it impossible

md5 and Password Activity

Questions

Questions?