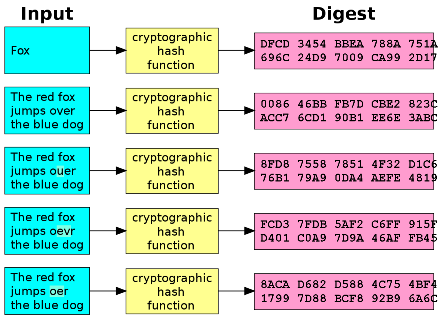
Overview: Basics of hashing and how it can help with databases

A Review on Hashing

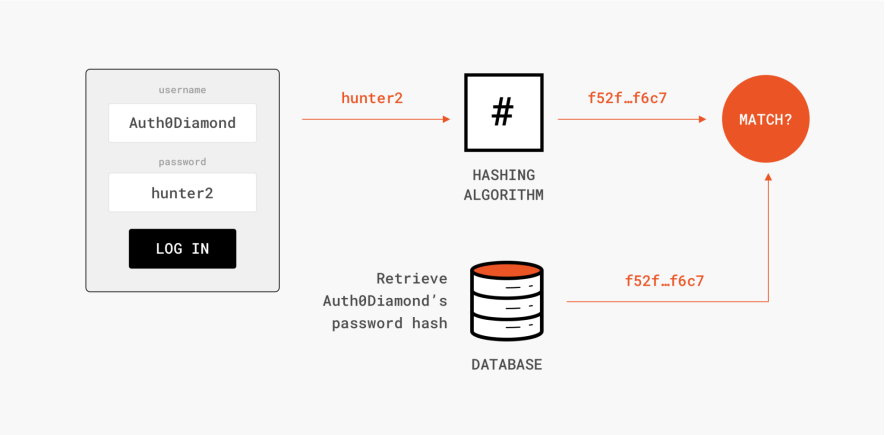

Input
Digest
Fox
The red fox jumps over the blue dog
The red fox jumps ouer the blue dog
The red fox jumps oevr the blue dog
The red fox jumps oer the blue dog
cryptographic hash function
DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

Hashing is a mathematical unidirectional function that converts a message into a unique* encoded message

PRO: If a malicious actor is accesses sensitive information **that is hashed**, they won't be able to do anything meaningful with it. Modern hashing algorithms include SHA 2, SHA 256, and MD5

*Hashing collisions can theoretically occur, but has a very low probability .

Implementing Hashing in a Database



Data Breach: Some Companies that Had Data Leaks

- Colonial Pipeline: Ransomware Attack
- SocialArks: Unsecure Database
- Accellion/Kroger: Legacy security, SQL injection
- Parler: Flawed API

Password Managers: How do they work?

Sometimes the best passwords to use are hard to remember, especially if they are all different, so that's where using a password manager can help!.

- Bitwarden
- Lastpass
- Google Password Manager

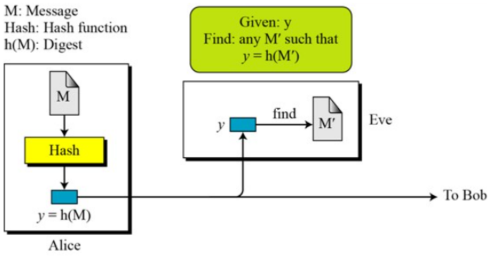
Data Attacks: different ways to obtain sensitive data

Common Mistakes

- Storing passwords as plaintext
- Encrypting passwords
- Weak passwords

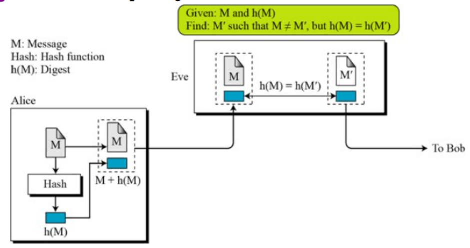
Preimage Attack

Given a hash, ***h*** , can you find the message ***m*** that generated that hash?
Extremely difficult and not practical. Would take hundreds of years to compute with a standard brute force approach.



Second Preimage Attack

Given a message, ***m***, and the hash for that message ***H(m)***, find a hash ***H(m')*** where *m' ≠ m.*
Same problem as before…

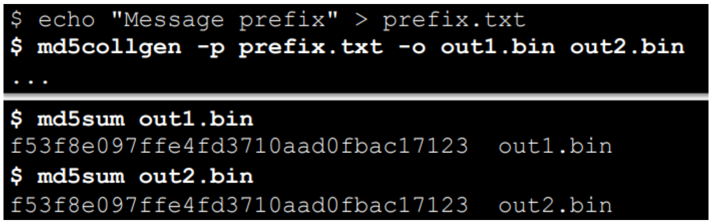


Hash Collisions

Goal: Generate two messages/files that have the same hash.
Techniques:

- Prefix attack → For MD5, approximately 2^{39} MD5 function calls need to be done (Stevens, et.al 2012)
- Birthday attack

The most common way of exploiting MD5 and SHA1
There are existing tools that can generate a collision for two files (relatively) quickly (!)



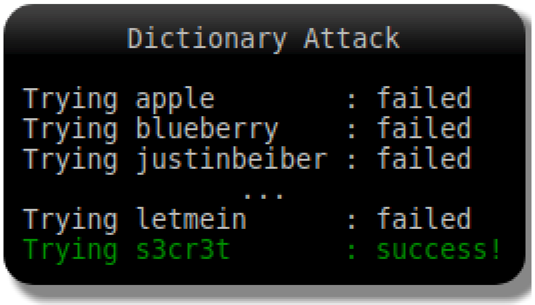
```
$ echo "Message prefix" > prefix.txt
$ md5collgen -p prefix.txt -o out1.bin out2.bin
...
$ md5sum out1.bin
f53f8e097ffe4fd3710aad0fbac17123  out1.bin
$ md5sum out2.bin
f53f8e097ffe4fd3710aad0fbac17123  out2.bin
```

SQL/Data Injection

A common attack vector that utilizes malicious SQL input with the goal of accessing information not intended.
We expect the user to input a valid username and password.
What if they try ***pass;select * from PASSWORDS*** for the password?
This could be interpreted as a separate SQL statement.

Dictionary Attack

Iterate through a list of words / common passwords and try the hash of the word as the password.
Brute Force Attack
Only works for weaker passwords.



```
       Dictionary Attack

Trying apple         : failed
Trying blueberry     : failed
Trying justinbeiber  : failed
             ...
Trying letmein       : failed
Trying s3cr3t        : success!
```

Rainbow Tables: What makes a rainbow table special

What is a Rainbow Table?

Like a dictionary of Passwords

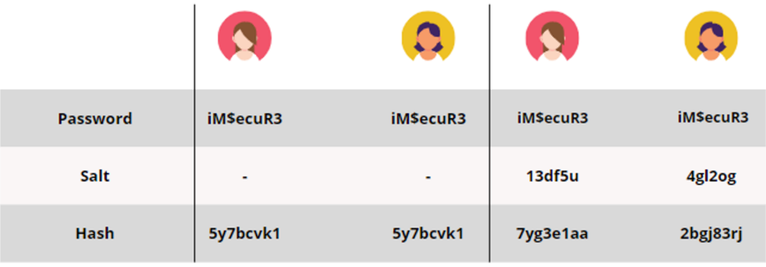
How does a Rainbow Table Work?

Chain of one way hash and reduction functions, start at plaintext, end at some hash.
A password had been found if a collision occurs

Salt Hashing

Technique to defend against password cracking attacks

Add random* bits ("salt") into the strings before hashing a newly created password



Password	iM$ecuR3	iM$ecuR3	iM$ecuR3	iM$ecuR3
Salt	-	-	13df5u	4gl2og
Hash	5y7bcvk1	5y7bcvk1	7yg3e1aa	2bgj83rj

- Guarantee that two users will not have the same hashed password
- Strengthens password and makes the reducing part difficult

Rainbow vs Salt

Rainbow and Salt
Salt prevents collisions from occurring.
Increased space requirements means less likely rainbow table contains hashed password