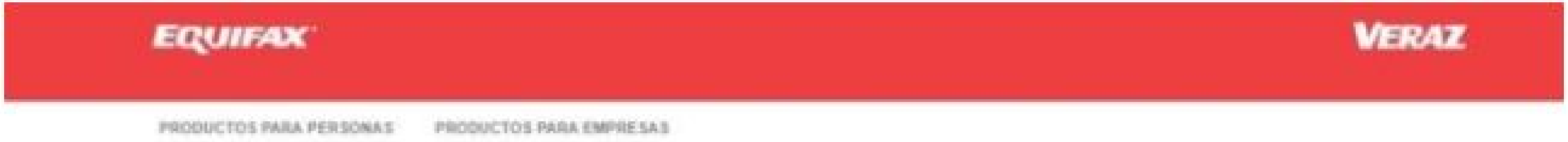
Password Cracking

Technology

Equifax had 'admin' as login and password in Argentina

13 September 2017 | Technology

Share



How are passwords stored

In practice, passwords should be stored as a hash

A **hash function** maps data of any arbitrary size to a fixed size digest (a hash)

In a **cryptographically secure** hash function, the output should appear completely random, but is deterministic

It should be infeasible to invert a cryptographically secure hash function, meaning we can’t find the input from the output

A small change in the input should make the new hash appear completely unrelated to the old hash

Hash function example

```
“hello”      -> hash(“hello”)      = b67c57dd6e2d1

“world”      -> hash(“world”)      = e7fb0394c7183

“helloworld” -> hash(“helloworld”) = 90e62b92a95c4

“password”   -> hash(“password”)   = 1293486c6015d

“password1”  -> hash(“password1”)  = ca51e14063abe
```

Hash function additional properties

Pre-image resistance

Given a hash value **h** it should be difficult to find any message **m** such that **h = hash(m).**

Second pre-image resistance

Given an input **m1** it should be difficult to find different input **m2** such that **hash(m1) = hash(m2).**

Collision resistance

It should be difficult to find two different messages **m1** and **m2** such that **hash(m1) = hash(m2).** Such a pair is called a cryptographic hash collision.

Password Cracking

Let’s say you have a hash for a password:

3608bca1e44ea6c4d268eb6db02260269892c0b42b86bbf1e77a6fa16c3c9282

How do you find the password that hashes to that value?

Brute Force Attack

Try all combinations until you find the correct one

Start with hash(“a”), then hash(“b”), … hash(“abc”) until the output matches your hash

The advantages of this method is that it will eventually find the password

The disadvantage is that it for longer passwords, it might not finish before the Sun expands to be a red giant and engulfs the Earth.

Dictionary Attack

Use a dictionary (whether it be an actual dictionary or a list of common passwords)

Hash each word in the list and see if any of the computed hashes

The advantage to this method is that it is fast and likely to find most common passwords

The disadvantage is that if the password is not in the dictionary you are using, it will not be found

Mask Attack/Hybrid attack

Let’s say you know the password is an english word followed by 2 numbers and a symbol

In this case you could run a dictionary attack, but for each word have it try all combinations of 2 numbers and a symbol at the end

This can cover more ground than a dictionary attack by itself

Rainbow Tables

Have a giant list of passwords and their corresponding hashes that you precalculated ahead of time.

Less feasible now because of salts.

Salts

In many systems, a salt is stored in addition to the password hash.

A salt is simply a unique long random string concatenated to the password before it is hashed

password+salt -> hash(password+salt) = 2dbe282e7eecb5110afac45d6

Having a unique salt for each password prevents rainbow table attacks and makes other attacks for difficult, as the work must be done for each hash salt combination as opposed to a large group of hashes all at once.