An introduction to
Password Security

by **Peter Boughton**

DIGITAL CRYDN

What the talk is about

Introduction

How developers can protect users

How users can protect themselves

What is NOT covered...

A comprehensive guide to user auth.

two-factor

logging in to third-parties

password managers

single sign-on

etc

Introduction

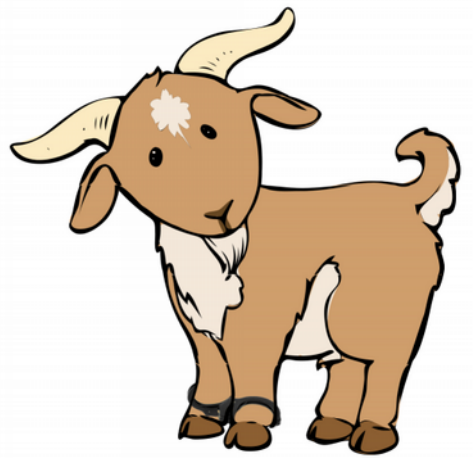
Why do we need passwords?

Identity

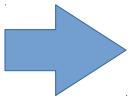
Why do we need passwords?

"Embarrased", licensed CC BY 2.0
From: https://www.flickr.com/photos/briovonsarah/5372890384

from "The Schizoid Man", episode 5 of "The Prisoner" 1960s TV series.



Wording

password  passcode

How Developers Can Protect Users

Avoid if you can

Are user accounts *really* necessary?

Cookies might be good enough.

Store passcodes correctly

Obviously, never store plaintext.

Obviously...?

Store passcodes correctly

Don’t store "encrypted".

- Encryption is a two way process.

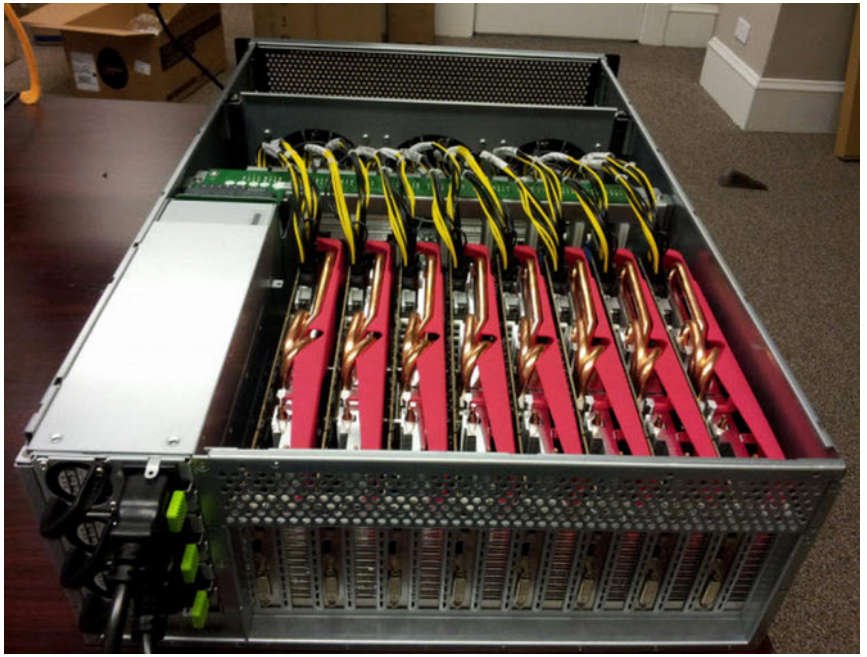
Don’t store "hashed".

- Not even "lots of times".
- MD5 is broken - never use it.
- SHA is fast - by design.

Hashing is fast

	MD5	SHA
1 GPU	8.6	3
8 GPU	135	42
25 GPU	180	63

billions per second



Store passcodes correctly

Don’t write a salted hash function.

Don’t grab magic code from the web.

Store passcodes correctly

Use a purpose-built tunable, iterative, uniquely salted one-way algorithm, that is deliberately slow, and slowable as technology improves.

(Do not attempt to write your own)

bcrypt | pbkdf2 | scrypt

Stronger passcodes

Educate your users.

Using "qwerty" is bad.

Good UI and feedback are critical.

Stronger passcodes

Most important: **Length**

Any single unit is weak - multiple units are good

- a letter is a single unit
- any character is a single unit
- a word is a single unit

Stronger passcodes

The more common a unit, the worse it is

- letters and numbers are very common
- some words are common - banana
- some punctuation is common - e.g. ! and $
- letter substitution is not a panacea
  - p4$$w0rd! is far *worse* than purplebananacube

Stronger passcodes

The hard work is done for you:

zxcvbn.js

Still needs appropriate setup

- Pass in username, dob, etc.
- Encourage strength, don't force too much

Common mistakes to avoid

Don’t write your own algorithms.

Don’t limit upper length.

Do disable focusing after keydown.

Don’t expire passwords every X days/weeks.

- it doesn’t add security.
- it makes passwords weaker.

Common mistakes to avoid

Do tell users about changes.

Do prevent too many attempts.

- for same account
- from same source (e.g. IP or subnet)

Don’t unlock accounts when finally correct.

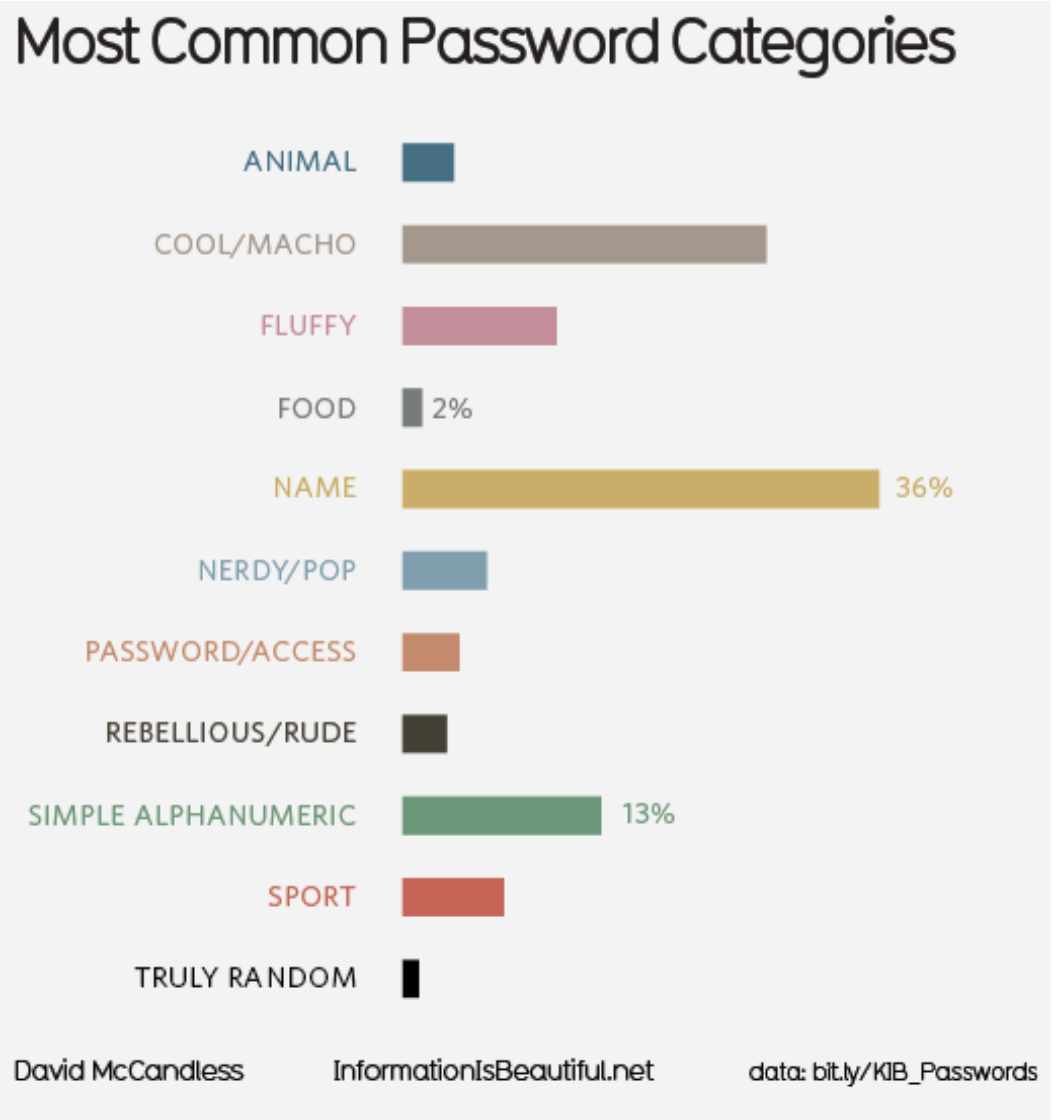
Don’t submit over HTTP - plaintext.

How Users Can Protect Themselves

Tips

Do NOT use...

- personal details
- favourite things
- keyboard patterns



Tips

"Nobody would think I'm stupid enough to use X"
(yes, they will)

You don't always need to remember.

Tips

Don’t use the same password.

Consider device input options.

Block third party JS and adverts.

Creating a Strong Passcode

- Write down six letters in a notebook/diary.
- Imagine a nonsense phrase using them.
- Use at least one uncommon mispelling.
- Any punctuation helps, unpredictable is better.

Complain About Bad Practices

If length is limited.

If pass is ever sent to you.

If no punctuation or symbols allowed.

If mothers maiden name allows resetting.

Thanks for listening

Questions?