SECURE SERVER-SIDE PASSWORD STORAGE

Welcome!
I'm Jan, I'm a programmer.
I hope you don't learn anything.
This is where most developers roll there eyes and stop listening.
This talk is for server-side developers.

THE THEORY IS HARD, LET'S GO SHOPPING

I am not a security expert.

EMBARRASSMENT OF IGNORANCE

It is embarrassing to admit you don't know something.
Might hinder you from learning.
Maybe you think you are not smart enough.

Bollocks, I'm not very smart and I got through it.
I was embarrassed to now know about this enough.
I will never understand the math behind this.

PROBLEM?

What are we solving here?
Somebody steals the user database.
How can we make it so that the thieves can't pretend to be you?

I DON’T CARE

Don't be LinkedIn (6M), last.fm (?), eHarmony (?).
Trust.
Money.

Users re-use passwords, don't be the one to leak it.

PLAINTEXT

No.

NO

No.

HASH

MD4, MD5, SHA1, SHA256, SHA512, SHA3.

Designed to check integrity of large sets of data.

FAST.

Rainbow tables. (not really worth the effort)

330MB worth of MD5/s
40s to check *any* 6-char password.

HASH + SALT

Clever trick, avoids rainbow tables.
Still not good enough, cracking is still too fast.

bcrypt

Ah!
Bcrypt introduced a "work factor":
1. Uses more CPU.
2. Adjustable to keep up with faster hardware.

2010 hardware: 40 seconds (md5) -> 12 years

Yay!

scrypt

Ah2!

Scrypt introduces a "memory factor".
Allows you to configure how much RAM is used.

Yay2!

PBKDF2

Password-Based Key Derivation Function

Think bcrypt + salts.

Plus MASSIVE security research compared to bcrypt/scrypt.
WPA uses this.
FileVault uses this.
Use this.

Peanut Butter Koala Dream Factory
Please Be Kind Digital Fighter

MIGRATE

1. Set all users to "not upgraded".
2. On the next user log-in:
 - verify password
 - if user is "not upgraded"
  - run plaintext password through pbkdf2
  - store new "hash"
  - set user to "is upgraded"
3. For stale users, send them an email and ask to log in.

DON'T STORE

PASSWORDS

Alternatives:

- OAuth
- Facebook Connect
- Twitter Login

RESOURCES

http://en.wikipedia.org/wiki/PBKDF2

http://blog.zoller.lu/2012/06/storing-password-securely-hashses-salts.html

http://codahale.com/how-to-safely-store-a-password/

http://www.unlimitednovelty.com/2012/03/dont-use-bcrypt.html

http://codahale.com/a-lesson-in-timing-attacks/

http://queue.acm.org/detail.cfm?id=2254400

http://www.openwall.com/presentations/PHDays2012-Password-Security/PHDays2012-Password-Security.pdf

THANKS!

BONUS

TIMING

ATTACKS

ugh