Passwords in MySQL (and other misc.)

CS518

Dr. Justin F. Brunelle

Problem

- Passwords should not be stored in clear-text
- Access to the DB gives you access to all user credentials
- Single point of failure for site security
-

Solution: Hash passwords

- Use a randomized per-use secret string and hash it with the password
- “Secret String” is a *salt*
- e.g.:
  - sha256($password . $salt)
- sha256 recommended
- sha1 comes standard with php

Storage

- Each user will have a salt
  - Salt created when user created

username	Plain-text-pass	salt	Encrypted-pass
justin	mustangs	abcdefg	2704231b6ca155b6b477d3e52054c8ca98aeedd3
kent	basketball	123456	afd5616e055ed5e4e82882143a0fde8f0b574d09
taylor	football	abc123	1efa2622b5df3ee8a09e779f344aa4e7d2f9f72c

Making the passwords

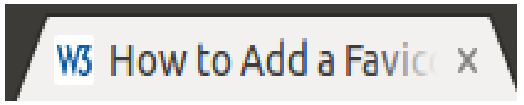
```
$> php -a
Interactive mode enabled

php > print sha1("mustangs");
5c33647c01eaf3aa534c0e83376193a32cc86a95

php > print sha1("mustangs" . "abcdefg");
2704231b6ca155b6b477d3e52054c8ca98aeedd3
```

What this gets you...

- Multiple variables in authentication:
  - No plain-text storage (can't steal passwords)
  - Salt order (prepend or append)
  - Hash algorithm (md5, sha1, sha256...)
- But does not:
  - Pass passwords security from HTML!
  - Monitor POST data, form data
  - (Use HTTPS and SSL for this)

Disclaimer

- Using your home-grown authentication/encryption is a bad idea
- See password_hash(), oauth, or Kerberos



favicon

- Shows up in your browser's tab/window/title
- Small image (16x16 or 32x32)
- Most browser look for “favicon”
- Safer to force HTML reference:

```
<!DOCTYPE html
      PUBLIC "-//W3C//DTD HTML 4.01//EN"
      "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US">
<head profile="http://www.w3.org/2005/10/profile">
<link rel="icon"
      type="image/png"
      href="http://example.com/myicon.png">
[…]
</head>
[…]
</html>
```