Password Protocols

protecting your application by enforcing good password practices on your users, and keeping them secure

Erik Goodlad
www.errorik.com
egoodlad@gmail.com

Why are we here?

- Users create weak passwords
- Phishers and hackers
- Weak security in web applications

Why are we here?

- We care about security
- We are in a position to do something

Overview

- What can we do to encourage good passwords
- What can we do to prevent attacks
- What can we do to protect user accounts
- Open Discussion

Passwords

- MySpace Passwords analyzed (~20,000)
  - password1, abc123, myspace1, password, blink182
- Web App A (~375,000)
  - password, 123456, [site1], [site2], monkey
  - qwerty, letmein, trustno1, computer, abc123
- Web App B (~2600)
  - [name of band], 12345678, password, punkrock, pennywise

How many passwords?

- Home PC, Notebook, Work PC, Work Servers x ?
- E-mail x ?, FTP, VPN, DB, Remote Desktop
- Banking, Shopping, Forums, Entertainment, etc...

What We Can Do

“Aid” The User

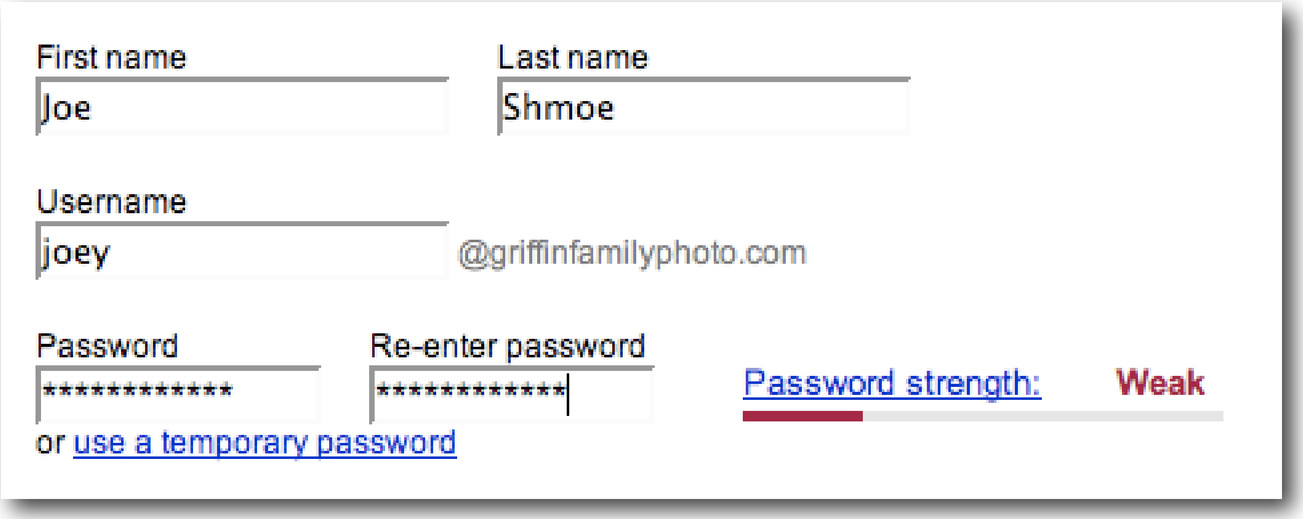
Force Users

- Enforce minimum length passwords
- mixed case / alpha + numeric + special chars
- Use blacklist of bad passwords (“password”, domain name, user’s name or other info)
- Mandatory password rotation

Downside

Encourage Good Password Practices

- Give "Tips" on what makes a good password
- Use Google-like script to alert users if they are entering a good password or not



First name: Joe

Last name: Shmoe

Username: joey @griffinfamilyphoto.com

Password: *************

Re-enter password: *************

or use a temporary password

Password strength: Weak

Advocate

- Links to password discussions
  - Security Now! Podcast #4 & #5
- Password Tools
  - SuperGenPass http://labs.zarate.org/passwd_new/

Tips from OWASP

Open Web Application Security Project

www.owasp.org

Password Strength

- minimum size and complexity
  - mixed alphanumeric + special characters
- periodic change required
  - previous passwords restricted

Password Use

- limited failed logins per unit of time
    - log attempts (not passwords)
- error messages should not indicate whether the username or password is failing
- on successful login, notify of failed attempts

Password Change Controls

- require old password when setting new
  - password should be required for all account edits
- don’t send user name in e-mail with password

Password Storage

- Always store passwords hashed / encrypted
- Hash is preferred as it is not reversible
- Encryption might be required if your application needs the password in plaintext
  - Encryption keys must be protected
- Passwords never hard coded

Salting Your Hash Tastes Good

```
plain_hash = hash(“password”)
```

```
salted_hash = hash(“password” + salt)
```

```
dbl_salted = hash(hash(“password” + salt) + salt)
```

Salting Your Hash Tastes Good

```
plain_hash = hash(“password”)
1A833DA63A6B7E20098DAE06D06602E1
```

```
salted_hash = hash(“password” + salt)
23295088674AAA1E2CE1CC032EDC40BE
```

```
dbl_salted = hash(hash(“password” + salt) + salt)
14F4A3AC9460B39F2C8EBC48E519EE70
```

Salting Your Hash Tastes Good

```
plain_hash = hash(“password”)


salted_hash1 = hash(“password” + salt1)


salted_hash2 = hash(“password” + salt2)
```

Salting Your Hash Tastes Good

```
plain_hash = hash(“password”)
1A833DA63A6B7E20098DAE06D06602E1


salted_hash1 = hash(“password” + salt1)
23295088674AAA1E2CE1CC032EDC40BE


salted_hash2 = hash(“password” + salt2)
EDB17856B4FCE5F17D29FBD44D080052
```

Protecting Credentials In Transit

- Always use SSL
- Client-side hashing before sending to server is useless

Browser Caching

- Never use GET, always use POST
- Use all varieties of "no cache" tags
  - and HTTP no cache headers

Last Random Bits...

CaSe sEnSiTiVe SqL

MS SQL Server:

```
SELECT
  columnList
FROM
  tableName
WHERE
  CAST(userPass as VarBinary(20))
            = CAST(@inPass as VarBinary(20))
```

MySQL:

```
WHERE
  userPass COLLATE latin1_bin = inPass
```

Don't Assume E-mail Ownership

- Users will mistype their own e-mail address, don't send account info until you have verified ownership.

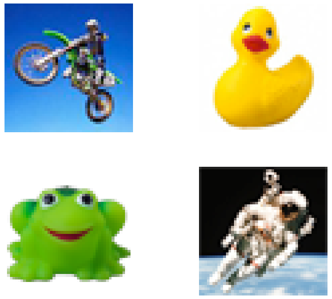
Phishing

Houston, we have a [big] problem

- Fake e-mails claiming to be from trustworthy organizations such as CitiBank, eBay, etc have a success rate of about 3%.
- A study by two students at Indiana University sending e-mails to students using publicly available information about their friends / classmates have yielded a "success rate … much higher than that of a traditional phishing attack"

Do “Tricks” Work?

- Bank of America SiteKey Study
  - Uses user selected image next to password field
    - 58 of 60 users entered their passwords even though the security image was not displayed



In 2005, Treasury department auditors posed as network technicians and attempted to get IRS employees to reveal their usernames and passwords and/or change the password to one suggested by the "technician". The result: over one-third shared their passwords. If there is any good news in the story it is that the 35% figure represents a substantial reduction from the 71% who fell for the ruse in 2001.

Who's To Blame?

- The user?
- The attacker?
- The application?
- All of the above?

Q & A

Thank You

Erik Goodlad
www.errorik.com
egoodlad@gmail.com