Web Browser Security

John Mitchell

Course Schedule

- Projects
  - Proj 1: Assigned April 11, Due April 27
  - Proj 2: Assigned May 2, Due May 18
  - Proj 3: Assigned May 18, Due June 8 *No Late Days*
- Homework
  - HW 1: Assigned April 20, Due May 4
  - HW 2: Assigned May 11, Due May 25
  - HW 3: no HW3 this year

Project 2: Web Application Security

CS155, Spring 2006

Part 1 due: Thursday, May 11 th

Part 2 due: Thursday, May 18 th [Note: CGI workspace may take up to 24 hours]

Further clarifications and fixes will be posted in the FAQ.

Part 1: Attacks

The fictional ZobCoin Foundation is hosting a simple web application at zobcoin.org, allowing registered users to post profiles and transfer "zobcoin" credits between each other. Each registered user starts with 10 zobcoin.

You will draft a list of attacks on zobcoin.org that exploit vulnerabilities in the website's design. Each attack presents a distinct scenario and unique goals and constraints, although in some cases you may be able to re-use parts of your code.

Although many real-world attackers do not have the source code for the web sites they are attacking, you are one of the lucky ones: source code is available here. You don't actually need to look at the site's source code until Part 2.

Your attacks will run in a restricted sandbox environment that can only connect to zobcoin.org and attack.zobcoin.org. We will run your attacks after logging clearing the database of registered users, so any data you submitted to zobcoin.org while working on this assignment will not be present during grading.

Part 2: Defenses

In the second part of the project, you will modify the website to fix the vulnerabilities.

Setup

Web server. You will need a web server that can run PHP scripts while you are working on the second part of the assignment. We'll be testing your attacks on the student CGI-enabled Stanford personal web spaces, so we recommend that you use the same configuration to ensure that your web site will work. You can enable CGI on your Rainbow account by following these instructions. Activations may take up to 24 hours, so be sure to get a head start.

Project files. Once you have CGI-enabled your personal web space, run the following command from a Leland machine to install the project website on your personal web space:

```
cp -r /usr/class/cs155/projects/p2/website ~/public_html
```

SQL database. The website uses a flat file SQL database called sql.db to manage persistent user data. The database engine needs to be able to write to your website, so run this command to give it access:

```
fs setacl ~/public_html/website/db rapidjoe:cgi-user-data rlidw
```

Your site can now be accessed at http://cgi.stanford.edu/~username/website/

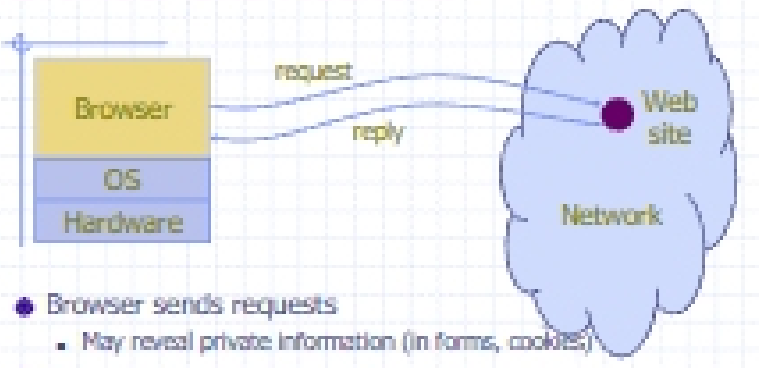
Goals

- Use input validation or encoding to prevent Attacks A, C, and D.
- Implement a method for authenticating POST requests by logged-in users to prevent Attack B and similar attacks.
- Ensure the site is free of any other cross-site scripting vulnerabilities you find.
- Do not change the site's appearance as performed via normal use.

Outline

- Browser review
  - Bugs happen
  - HTTP, scripts, events, DOM
  - Session state and cookies
- Protecting the browser environment
  - Execution sandbox
  - Access policies, signed scripts
- Privacy and confidentiality for sensitive information
  - Protecting the file system, OS, platform
  - Protecting information associated with other browser processes (e.g., other windows)
  - Protecting the user against deception
  - Protecting against traffic analysis

Browser and Network



- Browser sends requests
  - May reveal private information (in forms, cookies)
- Browser receives information, code
  - May corrupt state by running unsafe code
- Interaction susceptible to network attacks
  - Consider network security later in the course

INTERNETWEEK.com Tuesday, February 12, 2002

- Microsoft Issues New IE Browser Security Patch

  By Richard Karpinski

  - Microsoft has released a security patch that closes some major holes in its Internet Explorer browser
  - The so-called "cumulative patch" fixes six different IE problems ...
  - Affected browsers include Internet Explorer 5.01, 5.5 and 6.0.
  - Microsoft rated the potential security breaches as "critical."

Feb 2002 patch addresses:

- A buffer overrun associated with an HTML directive ... Hackers could use this breach to run malicious code on a user's system.
- A scripting vulnerability that would let an attacker read files on a user's systems.
- A vulnerability related to the display of file names ... Hackers could ... misrepresent the name of a file ... and trick a user into downloading an unsafe file.
- A vulnerability that would allow a Web page to improperly invoke an application installed on a user's system to open a file on a Web site.
- ... more ...

MS announced 20 vulnerabilities on April 13, 2004 !!!

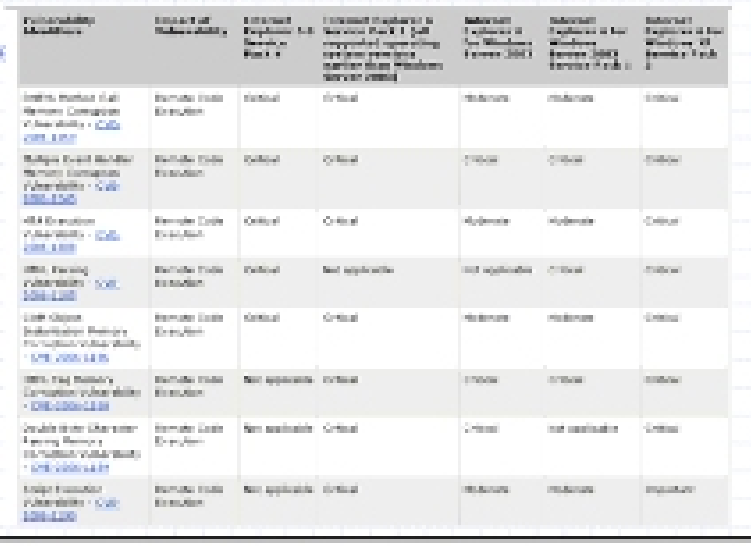
And then again last year, ...

Windows Security Updates Summary for April 2005

Published: April 12, 2005

A security issue has been identified that could allow an attacker to compromise a computer running Internet Explorer and gain control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.

Microsoft Security Bulletin MS06-013, April 2006



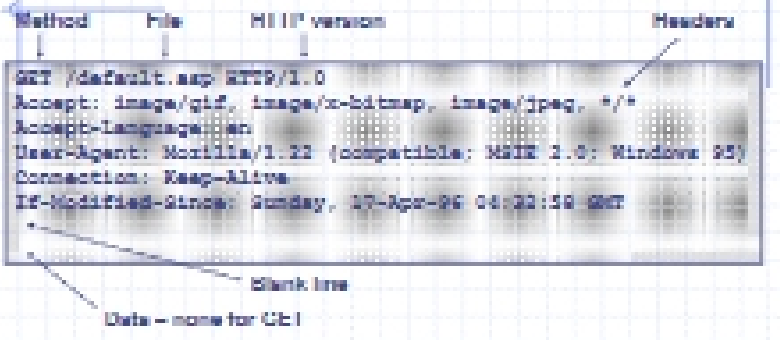
Browser security topics

- Review HTTP, scripting
- Controlling outgoing information
  - Cookies
    - Cookie mechanism, JunkBuster
  - Routing privacy
    - Anonymizer, Crowds
  - Privacy policy – P3P
- Risks from incoming executable code
  - JavaScript
  - ActiveX
  - Plug-ins
  - Java

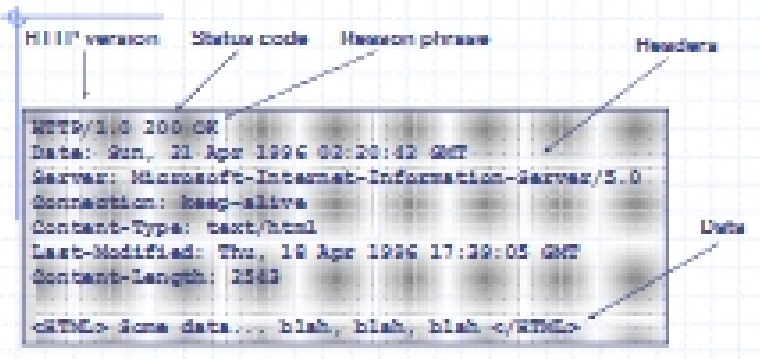
HyperText Transfer Protocol

- Used to request and return data
  - Methods: `GET`, `POST`, `HEAD`, ...
- Stateless request/response protocol
  - Each request is independent of previous requests
  - Statelessness has a significant impact on design and implementation of applications
- Evolution
  - HTTP 1.0: simple
  - HTTP 1.1: more complex

HTTP Request



HTTP Response



HTTP Server Status Codes

Code	Description
200	OK
201	Created
301	Moved Permanently
302	Moved Temporarily
400	Bad Request – not understood
401	Unauthorized
403	Forbidden – not authorized
404	Not Found
500	Internal Server Error

- Return code 401
  - Used to indicate HTTP authorization
  - HTTP authorization has serious problems!!!

HTML and Scripting

```
<html>
    ...
   <P>
<script>
    var num1, num2, sum
    num1 = prompt("Enter first number")
    num2 = prompt("Enter second number")
    sum = parseInt(num1) + parseInt(num2)
    alert("Sum = " + sum)
</script>
    ...
</html>
```

Browser receives content, displays HTML and executes scripts

Events

Mouse event causes page-defined function to be called

```
<script type="text/javascript">
    function whichButton(event) {
    if (event.button==1) {
            alert("You clicked the left mouse button!") }
    else {
            alert("You clicked the right mouse button!")
    }}
</script>
...
<body onmousedown="whichButton(event)">
...
</body>
```

Other events: onLoad, onMouseMove, onKeyPress, onUnLoad

Document object model (DOM)

- Object-oriented interface used to read and write documents
  - web page in HTML is structured data
  - DOM provides representation of this hierarchy
- Examples
  - Properties: document.alinkColor, document.URL, document.forms[ ], document.links[ ], document.anchors[ ]
  - Methods: document.write(document.referrer)
- Also Browser Object Model (BOM)
  - Window, Document, Frames[], History, Location, Navigator (type and version of browser)