CS 155 April 19, 2005

Web Browser Security

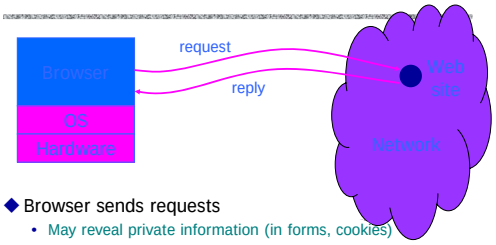
John Mitchell

Course Schedule

- Projects
  - Project 1: Assigned April 7, Due April 21
  - Project 2: Assign April 21, Due May 5
  - Project 3: Assign May 12, Due June 2 *No Late Days*
- Homework
  - HW 1: Assigned April 14, Due April 28
  - HW 2: Assign April 28, Due May 12
  - HW 3: Assign May 19, Due June 2 *No Late Days*

All assign/due dates are Thursdays (see calendar on web)
June 2 is "automatic one-week extension from May 26"

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 this year, ...

Windows Security Updates Summary for April 2005

Published: April 12, 2005

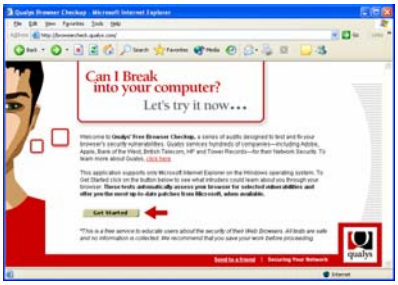
A security issue has been identified that could allow an attacker to compromise a computer running Internet Explorer and gain control over it. You can help protect your computer by installing this update from Microsoft. After you install this item, you may have to restart your computer.

Browser Security Check

http://www.verisign.com/advisor/check.html

What kind of security are they checking?

More informative test site



- Cookie Disclosure
- Clipboard Reading
- Program Execution
- File Execution
- Web Page Spoofing
- Security Zone Spoofing
- Hard Drive Access

http://browsercheck.qualys.com/

Browser security topics

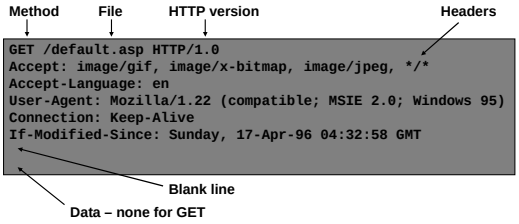
- HTTP review
- Controlling outgoing information
  - Cookies
    - Cookie mechanism, JunkBuster
  - Routing privacy
    - Anonymizer, Crowds
  - Privacy policy – P3P
- Risks from incoming executable code
  - JavaScript
  - ActiveX
  - Plug-ins
  - Java

HTTP

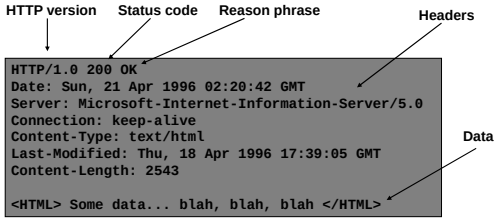
HyperText Transfer Protocol

- 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

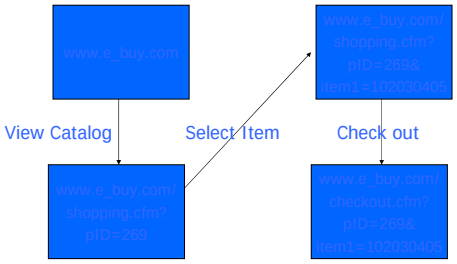


HTTP Server Status Codes

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!!!

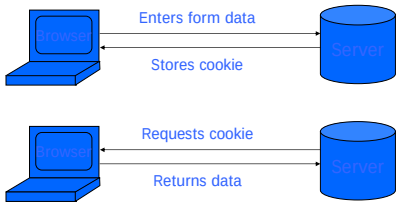
Primitive Browser Session



Store session information in URL; Easily read on network

Store info across sessions?

- Cookies
  - A cookie is a file created by an Internet site to store information on your computer



Http is stateless protocol; cookies add state

Browser Cookie Management

- Cookie Ownership
  - Once a cookie is saved on your computer, only the Web site that created the cookie can read it.
- Variations
  - Temporary cookies
    - Stored until you quit your browser
  - Persistent cookies
    - Remain until deleted or expire
  - Third-party cookies
    - Originates on or sent to another Web site

Third-Party Cookies

- Yahoo! Privacy Center
  - Yahoo! sends most of the advertisements you see
  - However, we also allow … third-party ad servers … to serve advertisements
  - Because your web browser must request these … from the ad network web site, these companies can send their own cookies to your cookie file …
  - **Opting Out of Third-Party Ad Servers**
    - "If you want to prevent a third-party ad server from sending and reading cookies on your computer, currently you must visit each ad network's web site individually and opt out (if they offer this capability)."

Example: Mortgage Center

<html><title>
Mortgage Center
</title><body>
… http://www.loanweb.com/ad.asp?RLID=0b70at1ep0k9

What's this?

Cookie issues

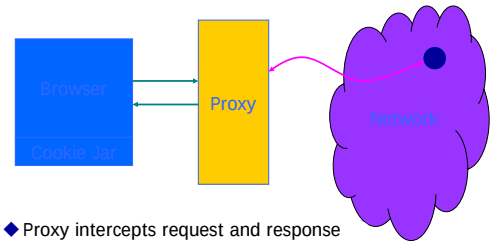
- Cookies maintain record of your browsing habits
  - Cookie stores information as set of name/value pairs
  - May include *any* information a web site knows about you
  - Sites track your activity from multiple visits to site
- Sites can share this information (e.g., DoubleClick)
  - Sites using DoubleClick place small graphic that causes user to request page from DoubleClick (see previous slide)
  - DoubleClick uses cookies to identify you on various sites

    Site can generate user ID from its own cookie and pass to DoubleClick or other site in the URL
- Browser attacks could invade your "privacy"

  08 Nov 2001

  Users of Microsoft's browser and e-mail programs could be vulnerable to having their browser cookies stolen or modified due to a new security bug in Internet Explorer (IE), the company warned today.

Managing cookie policy via proxy



- Proxy intercepts request and response
- May modify cookies before sending to Browser
- Can do other checks: filter ads, block sites, etc.

Sample Proxy: JUNKBUSTERS

- Cookie management by policy in *cookiefile*
  - Default: all cookies are silently crunched
  - Options
    - Allow cookies only to/from certain sites
    - Block cookies to browser (but allow to server)
    - Send vanilla wafers instead
- Block URLs matching any pattern in *blockfile*
  - Example: pattern /*.*/ad matches http://nomatterwhere.com/images/advert/g3487.gif

Easy to write your own http proxy; you can try *this* at home

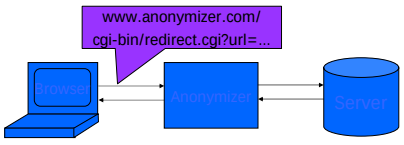
Preserving web privacy

- Your IP address may be visible to web sites
  - This may reveal your employer, ISP, etc.
  - Can link activities on different sites, different times
- Can you prevent sites from learning about you?
  - Anonymizer
    - Single site that hides origin of web request
  - Crowds
    - Distributed solution

Browsing Anonymizers

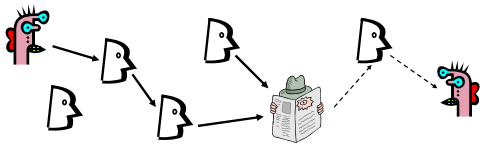
ANONYMIZER.COM
Privacy is your right.

- Web Anonymizer hides your IP address



- What does anonymizer.com know about you?

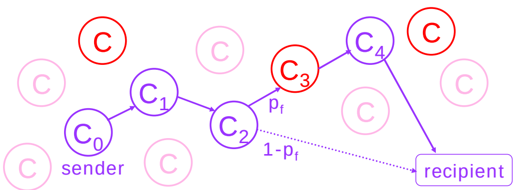
Related approach to anonymity



- Hide source of messages by routing them randomly
- Routers don't know for sure if the apparent source of the message is the actual sender or simply another router
  - Only secure against local attackers!
- Existing systems: Freenet, Crowds, etc.

Crowds

[Reiter,Rubin '98]



- Sender randomly chooses a path through the crowd
- Some routers are honest, some corrupt
- After receiving a message, honest router flips a coin
  - With probability P_f routes to the next member on the path
  - With probability $1- P_f$ sends directly to the recipient

What Does Anonymity Mean?

- Beyond suspicion
  - The observed source of the message is no more likely to be the actual sender than anybody else
- Probable innocence
  - Probability <50% that the observed source of the message is the actual sender

    Guaranteed by Crowds if there are sufficiently few corrupt routers
- Possible innocence
  - Non-trivial probability that the observed source of the message is not the actual sender

Something you can try at home

- Find out what sites know about you
  - Anonymizer.com, other sites will tell you want they can find about your IP address
  - Many other sites offer this too …

www.anonymizer.com

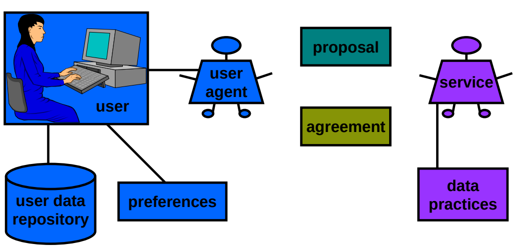
Try Private Surfing FREE!
Make your online activities invisible and untrackable to online snoops.
Just type a URL & click "GO."

Also: free privacy toolbar (?)

How web sites use your information

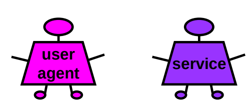
- You may enter information to buy product
  - Name, address, credit card number, …
- How will web site use this information
  - Charge your card and mail your purchase
  - Give sales information to other businesses?
- Platform for privacy preferences (P3P)
  - Framework for reaching agreement on use of personal information
  - Enforcement at server side is another matter…

Basic P3P Concepts



Credit: Lorrie Cranor

A Simple P3P Conversation



User agent: Get index.html

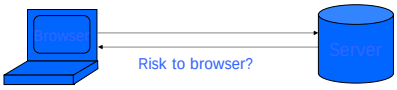
Service: Here is my P3P proposal - I collect click-stream data and computer information for web site and system administration and customization of site

User agent: OK, I accept your proposal

Service: Here is index.html

Controlling information *from* web

- Data is harmless (?)
- Risks come from code received from web
  - Scripts in web pages
  - ActiveX controls and Browser extensions
  - Applets



JavaScript

- Language executed by browser
- Used in many attacks (to exploit other vulnerabilities)
  - Cookie attack from earlier slide (08 Nov 2001):

    With the assistance of some JavaScript code, an attacker could construct a Web page or HTML-based e-mail that could access any cookie in the browser's memory or those stored on disk …
- JavaScript runs
  - Before the HTML is loaded, before the document is viewed
  - While the document is viewed, or as the browser is leaving

PwdHash browser extension

- Extra code that reads user pwd, applies hash
  - Provide unique, high-quality password
  - Stronger authentication with existing servers
- Problem
  - Javascript on malicious web page can try to intercept user password from PwdHash
  - Javascript properties
    - Script activated with user changes focus
    - Script can read input, may run before PwdHash
      - Keyboard monitoring and logging
      - Spoof parts of web browser UI
      - Communicate across network

ActiveX

- ActiveX controls reside on client's machine, activated by HTML object tag on the page
  - ActiveX controls are not interpreted by browser
  - Compiled binaries executed by client OS
  - Controls can be downloaded and installed
- Security model relies on three components
  - Digital signatures to verify source of binary
  - IE policy can reject controls from network zones
  - Controls marked by author as *safe for initialization*, *safe for scripting* which affects the way control used

Once accepted, installed and started, no control over execution

Installing Controls

If you install and run, no further control over the code.

In principle, browser/OS could apply sandboxing, other techniques for containing risks in native code.

Risks associated with controls

- MSDN Warning
  - An ActiveX control can be an extremely insecure way to provide a feature
- Why?
  - A COM object, control can do any user action
    - read and write Windows registry
    - access the local file system
  - Other web pages can attack a control
    - Once installed, control can be accessed by any page
    - Page only needs to know class identifier (CLSID)
- Recommendation: use other means if possible

http://msdn.microsoft.com/library/default.asp?url=/code/list/ie.asp

IE Browser Helper Objects (Extensions)

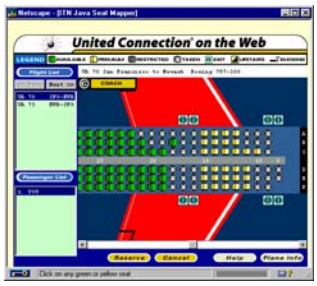
- COM components loaded when IE starts up
- Run in same memory context as the browser
- Perform any action on IE windows and modules
  - Detect browser events
    - GoBack, GoForward, and DocumentComplete
  - Access browser menu, toolbar and make changes
  - Create windows to display additional information
  - Install hooks to monitor messages and actions
- Summary: No protection from extensions

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebgen/html/bho.asp

Java

- Java is general programming language
- Web pages may contain Java code
- Java executed by Java Virtual Machine
  - Special security measures associated with Java code from remote URLs

Java Applet



- Local window
- Download
  - Seat map
  - Airline data
- Local data
  - User profile
  - Credit card
- Transmission
  - Select seat
  - Encrypted msg

Security Risks

- Annoyance or inconvenience
  - Display large window that ignores mouse input
  - Play irritating sound and do not stop
  - Consume CPU cycles, memory, network bandwidth …
- Export confidential information
  - Communication is generally possible
  - Prevent access to password file, credit card number, …
  - Subtle attack: trick dialog boxes …
- Modify or compromise system
  - Delete files, call system functions

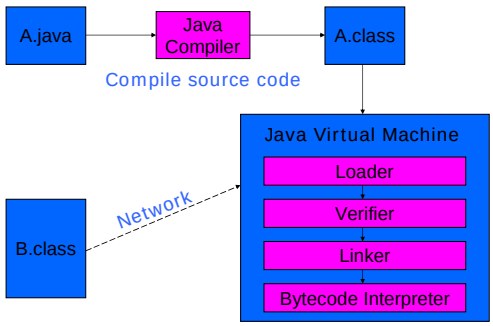
Mobile code security mechanisms

- Examine code before executing
  - Java bytecode verifier performs critical tests
  - Beyond the Browser: code modification
    - Replace standard calls by calls to “safe” versions
    - Check parameters to standard methods to make sure they are in appropriate ranges
- Interpret code and trap risky operations
  - Java bytecode interpreter does run-time tests
  - Security manager applies local access policy

Java Background

- Compiler and Virtual Machine
  - Compiler produces bytecode
  - Virtual machine loads classes on demand, verifies bytecode properties, interprets bytecode
- Why this design?
  - Portability
    - Transmit bytecode across network
  - Minimize machine-dependent part of implementation
    - Do optimization on bytecode when possible
    - Keep bytecode interpreter simple

Java Virtual Machine Architecture



Class loader

- Runtime system loads classes as needed
  - When class is referenced, loader searches for file of compiled bytecode instructions
- Default loading mechanism can be replaced
  - Define alternate ClassLoader object
    - Extend the abstract ClassLoader class and implementation
  - Can obtain bytecodes from network
    - VM restricts applet communication to site that supplied applet

Verifier

- Bytecode may not come from standard compiler
  - Evil hacker may write dangerous bytecode
- Verifier checks correctness of bytecode
  - Every instruction must have a valid operation code
  - Every branch instruction must branch to the start of some other instruction, not middle of instruction
  - Every method must have a structurally correct signature
  - Every instruction obeys the Java type discipline

Last condition is fairly complicated

Why is typing a security feature?

- Java security mechanisms rely on type safety
- Examples
  - General: Unchecked cast lets prog make any call

```
int (*fp)()     /* variable "fp" is a function pointer          */
...
fp = addr;      /* assign address stored in an integer var      */
(*fp)(n);       /* call the function at this address            */
```

  - Security manager has private fields that store permission information
    - Access to these fields would defeat the security mechanism

Type Safety of JVM

- Load-time type checking
- Run-time type checking
  - All casts are checked to make sure type safe
  - All array references are checked to be within bounds
  - References are tested to be not null before dereference
- Additional features
  - Automatic garbage collection
  - NO pointer arithmetic

If program accesses memory, the memory is allocated to the program and declared with correct type

How do we know verifier is correct?

- Many early attacks based on verifier errors
- Formal studies prove correctness
  - Abadi and Stata
  - Freund and Mitchell
    - Found error in initialize-before-use analysis

JVM uses stack machine

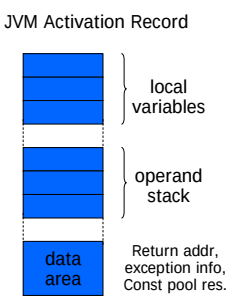
◆Java

```
Class A extends Object {
    int i
    void f(int val) { i = val + 1;}
}
```

◆Bytecode

```
Method void f(int)
    aload 0   ; object ref this
    iload 1   ; int val
    iconst 1
    iadd      ; add val +1
    putfield #4 <Field int i>
    return
```

refers to const pool



Java Object Initialization

```
Point p = new Point(3);
p.print();

1:  new Point
2:  dup
3:  iconst 3
4:  invokespecial <method Point(int)>
5:  invokevirtual <method print()>
```

◆No easy pattern to match.

◆Multiple refs to same uninitialized object.

Bug in Sun's JDK 1.1.4

◆Example:

variables 1 and 2 contain references to two different objects,
verifier thinks they are aliases

```
1:  jsr 10
2:  store 1
3:  jsr 10
4:  store 2
5:  load 2
6:  init P
7:  load 1
8:  use P
9:  halt
10: store 0
11: new P
12: ret 0
```

Java Security Mechanisms

◆Sandboxing
- Run program in restricted environment
  - Analogy: child's sandbox with only safe toys
- This term refers to
  - Features of loader, verifier, interpreter that restrict program
  - Java Security Manager, a special object that acts as access control "gatekeeper"

◆Code signing
- Use cryptography to determine who wrote class file
  - Info used by security manager

Java Sandbox

◆Four complementary mechanisms
- Class loader
  - Separate namespaces for separate class loaders
  - Associates *protection domain* with each class
- Verifier and JVM run-time tests
  - NO unchecked casts or other type errors, NO array overflow
  - Preserves private, protected visibility levels
- Security Manager
  - Called by library functions to decide if request is allowed
  - Uses protection domain associated with code, user policy
  - Enforcement uses stack inspection

Security Manager

◆Java library functions call security manager

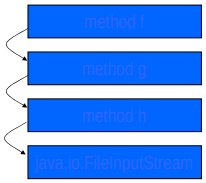
◆Security manager object answers at run time
- Decide if calling code is allowed to do operation
- Examine protection domain of calling class
  - Signer: organization that signed code before loading
  - Location: URL where the Java classes came from
- Uses the system policy to decide access permission

Sample SecurityManager methods

checkExec	Checks if the system commands can be executed.
checkRead	Checks if a file can be read from.
checkWrite	Checks if a file can be written to.
checkListen	Checks if a certain network port can be listened to for connections.
checkConnect	Checks if a network connection can be created.
checkCreate ClassLoader	Check to prevent the installation of additional ClassLoaders.

Stack Inspection

- Permission depends on
  - Permission of calling method
  - Permission of all methods above it on stack
    - Up to method that is trusted and asserts this trust



Many details omitted

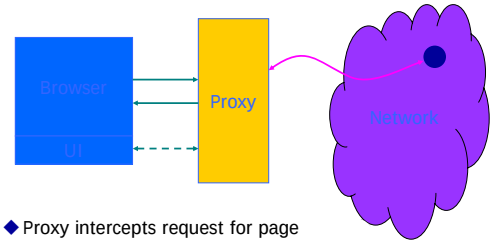
Stories: Netscape font / passwd bug; Shockwave plug-in

Beyond JVM security

- JVM does not prevent
  - Denial of service attacks
    - Applet creates large windows and ignores mouse
  - Certain network behavior
    - Applet can connect to port 25 on client machine, forge email (on some implementations)
  - URL spoofing
    - Applet can write false URL on browser status line
  - Annoying behavior
    - Applet can play loud sound
    - Applet can reload pages in new windows

Additional Security

Modify code in proxy [Shin, M...]



- Proxy intercepts request for page
- May modify before sending to browser
- Can do other checks: filter ads, block sites, etc.

Bytecode Modification Techniques

- Class-level replacement
  - Define subclass of library (or other) class
  - Replace references to class with subclass (const pool)
  - Works because of subtyping
  - Not possible if class is final
- Method-level replacement
  - Change function calls to new function
  - Generally, check or modify arguments and call original function

Sample bytecode modification

- SafeWindow class
  - Subclass of standard Window class
    - Do not allow windows larger than maximum
    - Do not allow more than max number of windows
- Restrict network activity
  - Replace call to Socket object constructor
    - Do not allow socket connection to port 25
- Maintain appearance of browser window
  - Replace calls to AppletContext methods
    - Displayed URL must match actual hyperlink

Summary: Browser security

- Browser uses network and local disk
  - Potential for outside access to local data
- Brower interprets code from network
  - HTML, JavaScript, ActiveX, Java
- Browser installs, executes plug-ins
  - Acrobat, Shockwave, ...
- Malicious code can pose risks
  - Consume resources, Steal information, Compromise system

We'll see many of these issues in other forms when we discuss OS security, network security