

Web Security

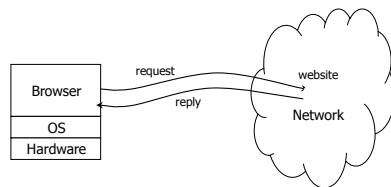
Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Goals for Today

- ◆ Web security
- ◆ Key issues
 - Browser is the new OS
 - State on client
 - Integrity (e.g., for pricing)
 - Privacy (e.g., cookies)
 - Website isolation (e.g., cross-site scripting)

Browser and Network



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"

Fixed by the February 2002 Patch

- ◆ Buffer overrun associated with an HTML directive
 - Could be used by hackers to run malicious code on a user's system
- ◆ Scripting vulnerability
 - Lets an attacker read files on a user's system
- ◆ 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
- ◆ ... and many more

On April 13, 2004, MS announced 20 new vulnerabilities

January 7, 2007

Microsoft Security Bulletin MS07-004

A remote code execution vulnerability exists in the Vector Markup Language (VML) implementation in Microsoft Windows. An attacker could exploit the vulnerability by constructing a specially crafted Web page or HTML e-mail that could potentially allow remote code execution if a user visited the Web page or viewed the message. An attacker who successfully exploited this vulnerability could take complete control of an affected system.

Maximum Severity Rating: Critical

Recommendation: Customers should apply the update immediately

Browsers are becoming "mini operating systems" - complex, running third-party code, etc.

Many Other Vulnerabilities

- ◆ Check out <http://www.microsoft.com/technet/security/>
- ◆ 44 "critical" updates related to Internet Explorer 6.0 between October 10, 2001, and January 9, 2007
- ◆ Other browsers also have problems
 - Recently Firefox in the news
 - First successful remote compromise of the iPhone exploited a bug in Safari

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

```
Method      File      HTTP version  Headers
|          |          |          |
GET /default.asp HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Connection: Keep-Alive
If-Modified-Since: Sunday, 17-Apr-96 04:32:58 GMT
```

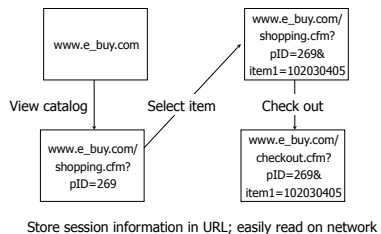
Blank line
Data - none for GET

HTTP Response

```
HTTP version  Status code  Reason phrase  Headers
|            |            |            |
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Content-Length: 2543
```

Data
<HTML> Some data... blah, blah, blah </HTML>

Primitive Browser Session



FatBrain.com circa 1999 [due to Fu et al.]

- ◆ User logs into website with his password, authenticator is generated, user is given special URL containing the authenticator

<https://www.fatbrain.com/HelpAccount.asp?l=0&p1=me@me.com&2=540555758>

- With special URL, user doesn't need to re-authenticate
 - Reasoning: user could not have not known the special URL without authenticating first. That's true, BUT...
- ◆ Authenticators are global sequence numbers
 - It's easy to guess sequence number for another user

<https://www.fatbrain.com/HelpAccount.asp?l=0&p1=SomeoneElse&2=540555752>

- Partial fix: use random authenticators
 - (Why not complete fix?)

Bad Idea: Encoding State in URL

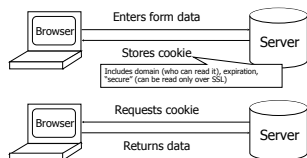
- ◆ Unstable, frequently changing URLs
- ◆ Vulnerable to eavesdropping
- ◆ There is no guarantee that URL is private
 - Early versions of Opera used to send entire browsing history, including all visited URLs, to Google

Cookies



Storing Info Across Sessions

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



HTTP is a stateless protocol; cookies add state

What Are Cookies Used For?

- ◆ Authentication
 - Use the fact that the user authenticated correctly in the past to make future authentication quicker
- ◆ Personalization
 - Recognize the user from a previous visit
- ◆ Tracking
 - Follow the user from site to site; learn his/her browsing behavior, preferences, and so on

Cookie Management

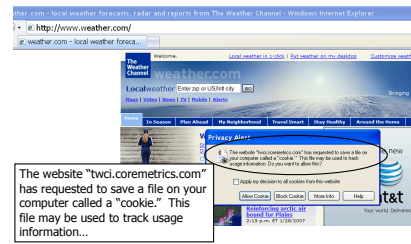
- ◆ Cookie ownership
 - Once a cookie is saved on your computer, only the website that created the cookie can read it (supposedly)
- ◆ Variations
 - Temporary cookies
 - Stored until you quit your browser
 - Persistent cookies
 - Remain until deleted or expire
 - Third-party cookies
 - Originates on or sent to another website

Privacy Issues with Cookies

- ◆ Cookie may include any information about you known by the website that created it
 - Browsing activity, account information, etc.
- ◆ Sites can share this information
 - Advertising networks
 - 2o7.net tracking cookie
- ◆ Browser attacks could invade your privacy

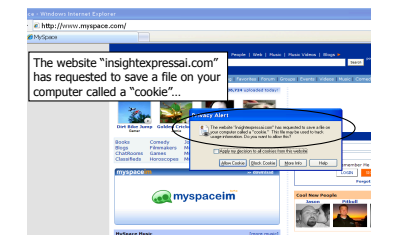
November 8, 2001:
Users of Microsoft's browser and e-mail programs could be vulnerable to having their browser cookies stolen or modified due to a new security bug in Internet Explorer (IE), the company warned today

The Weather Channel



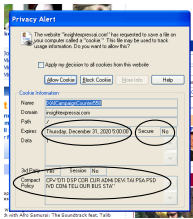
The screenshot shows the Weather Channel website with a privacy alert from twcl.coremetrics.com. The alert text reads: "The website 'twcl.coremetrics.com' has requested to save a file on your computer called a 'cookie.' This file may be used to track usage information...". The alert includes options to "Allow Cookies", "Deny Cookies", "Remember", and "Help". The website background shows a search bar for "Local weather" and navigation links like "Home", "About", "My Neighborhood", "Travel Search", "Stay Healthy", and "Access the News".

MySpace



The screenshot shows the MySpace website with a privacy alert from insightexpressal.com. The alert text reads: "The website 'insightexpressal.com' has requested to save a file on your computer called a 'cookie'...". The alert includes options to "Allow Cookies", "Deny Cookies", "Remember", and "Help". The website background shows a search bar and navigation links like "Home", "About", "My Neighborhood", "Travel Search", "Stay Healthy", and "Access the News".

Let's Take a Closer Look...



Storing State in Browser

◆ Dansie Shopping Cart (2006)

- "A premium, comprehensive, Perl shopping cart. Increase your web sales by making it easier for your web store customers to order"

```
<FORM METHOD=POST
ACTION="http://www.dansie.net/cgi-bin/scripts/cart.pl">
  Black leather purse with leather straps<BR>Fr Change this to 2.00
  <INPUT TYPE=HIDDEN NAME=name VALUE="Black leather purse">
  <INPUT TYPE=HIDDEN NAME=price VALUE="299.99">
  <INPUT TYPE=HIDDEN NAME=ab VALUE="1">
  <INPUT TYPE=HIDDEN NAME=ling VALUE="purse.jpg">
  <INPUT TYPE=HIDDEN NAME=custom VALUE="Black leather purse with
  leather straps">
  <INPUT TYPE=HIDDEN NAME="add" VALUE="Put in Shopping Cart">
</FORM>
```

Shopping Cart Form Tampering

<http://xforce.iss.net/xforce/xfdb/4621>

- ◆ Many Web-based shopping cart applications use hidden fields in HTML forms to hold parameters for items in an online store. These parameters can include the item's name, weight, quantity, product ID, and price. Any application that bases price on a hidden field in an HTML form is vulnerable to price changing by a remote user. A remote user can change the price of a particular item they intend to buy, by changing the value for the hidden HTML tag that specifies the price, to purchase products at any price they choose.

◆ Platforms Affected:

- 3iC Commerce Ltd: ShopFactory 5.0 and earlier @Retail Any version
- Adaptic: Check It Out Any version Baron Consulting Group: Website Tool Any version
- ConCity Corporation: SalesCart Any version Crested Butte Software: EasyCart Any version
- Dansie.net: Dansie Shopping Cart Any version Intelligent Wording Systems: Intelligent Any version
- Hake & Stone: Hake & Stone Online/Any Any version McMurtry/Whitaker & Associates: Cart12 2.6
- McMurtry/Whitaker & Associates: Cart12 3.0 sknutzer@netethu.no: CartMan 1.04
- Rich Media Technologies: JustASACommerce 5.0 SmartCart: SmartCart Any version
- Web Express: Shoppen 1.2

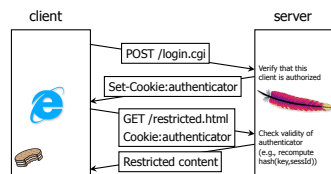
Storing State in Browser Cookies

- ◆ Set-cookie: price=299.99
- ◆ User edits the cookie... cookie: price=29.99
- ◆ What's the solution?
- ◆ Add a MAC to every cookie, computed with the server's secret key
 - Price=299.99; HMAC(ServerKey, 299.99)

Web Authentication via Cookies

- ◆ Need authentication system that works over HTTP and does not require servers to store session data
 - Why is it a bad idea to store session state on server?
- ◆ Servers can use cookies to store state on client
 - When session starts, server computes an authenticator and gives it back to browser in the form of a cookie
 - Authenticator is a value that client cannot forge on his own
 - Example: $MAC(\text{server's secret key, session id})$
 - With each request, browser presents the cookie
 - Server recomputes and verifies the authenticator
 - Server does not need to remember the authenticator

Typical Session with Cookies

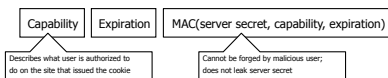


Authenticators must be unforgeable and tamper-proof
(malicious client shouldn't be able to compute his own or modify an existing authenticator)

WSJ.com circa 1999 [due to Fu et al.]

- ◆ Idea: use $\text{user, hash}(\text{user, key})$ as authenticator
 - Key is secret and known only to the server. Without the key, clients can't forge authenticators.
- ◆ Implementation: $\text{user, crypt}(\text{user, key})$
 - `crypt()` is UNIX hash function for passwords
 - `crypt()` truncates its input at 8 characters
 - Usernames matching first 8 characters end up with the same authenticator
 - No expiration or revocation
- ◆ It gets worse... This scheme can be exploited to extract the server's secret key

Better Cookie Authenticator



- ◆ Main lesson: don't roll your own!
 - Homebrewed authentication schemes are often flawed
- ◆ There are standard cookie-based schemes

Web Applications

- ◆ Online banking, shopping, government, etc. etc.
- ◆ Website takes input from user, interacts with back-end databases and third parties, outputs results by generating an HTML page
- ◆ Often written from scratch in a mixture of PHP, Java, Perl, Python, C, ASP
- ◆ Security is rarely the main concern, though that is hopefully changing.
 - Poorly written scripts with inadequate input validation
 - Sensitive data stored in world-readable files
 - Recent push from Visa and Mastercard to improve security of data management (PCI standard)

JavaScript

- ◆ Language executed by browser
 - Can run before HTML is loaded, before page is viewed, while it is being viewed or when leaving the page
- ◆ Often used to exploit other vulnerabilities
 - Attacker gets to execute some code on user's machine
 - Cross-scripting: attacker inserts malicious JavaScript into a Web page or HTML email; when script is executed, it steals user's cookies and hands them over to attacker's site

Scripting

```
<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>
```

Script defines a page-specific function

Function gets executed when some event happens (onLoad, onKeyPress, onMouseMove...)

JavaScript Security Model

- ◆ Script runs in a "sandbox"
 - Not allowed to access files or talk to the network
- ◆ Same-origin policy
 - Can only read properties of documents and windows from the same server, protocol, and port
 - If the same server hosts unrelated sites, scripts from one site can access document properties on the other
- ◆ User can grant privileges to signed scripts
 - UniversalBrowserRead/Write, UniversalFileRead, UniversalSendMail

Risks of Poorly Written Scripts

◆ For example, echo user's input

`http://naive.com/search.php?term="Britney Spears"`
search.php responds with
`<html> <title>Search results</title>
<body>You have searched for <?php echo $_GET[term]?>... </body>`

Or

`GET/ hello.cgi?name=Bob`
hello.cgi responds with
`<html>Welcome, dear Bob</html>`

Stealing Cookies by Cross Scripting

