Your Browser Wears No Clothes
Why Fully Patched Browsers Remain Vulnerable

Michael Sutton
VP, Security Research
Who Am I?

Company
- Zscaler – SaaS solution for web security
- VP, Security Research

Background
- SPI Dynamics – acquired by HP
- iDefense – acquired by VeriSign

Research
- Web security
- Client-side vulnerabilities
- Fuzzing
BSoD – Beijing Olympics
Overview

Background

Attacks
- XSS
- Clickjacking

Challenges

Defense

Future
Typical Attack Cycle

- Public Vulnerability Disclosure
- Patches Deployed
- Worm/Botnet Attacks
- Public Exploit Disclosure
- Individual Attacks
Drivers of Change

**Enterprises**
- Shrinking patch windows
- Focus on DMZ protection

**Vendors**
- Security response teams
- Secure coding practices

**Technology**
- Increasingly complex web applications
- Development platforms streamline development
- Rapid pace of new web technologies
Browser Attacks vs. Naked Browser Attacks

**Browser**
- Results from flaws in browser design
- Attack triggered by anomalous traffic
- Risk is mitigated through patching

**Naked Browser**
- Results from flaws in web application design or abuse of functionality
- Attack often indistinguishable from *normal* traffic
- Patches are not available for risk mitigation
Technical Web Application
Vulnerabilities Affecting End Users

**Cause**

- Technical (e.g. XSS, CSRF, etc.) or application logic vulnerabilities permit attackers to access or control content
- Although vulnerabilities reside on the server, victims can be end users due to trust relationships
  - User data stored on the server can be accessed/Altered (web application attack – e.g. SQLi)
  - Attack can target end user data or actions via the web browser (naked browser attack)

**Risk**

- Vulnerabilities are regularly discovered on reputable sites
- End users may have no way of knowing that they have been the victim of an attack
Abuse of Functionality Affecting End Users

**Cause**
- No web application or browser vulnerability is abused
- Intended functionality is used in an unintended way
- Examples – clickjacking and URL redirection

**Risk**
- Difficult to detect as traffic is legitimate
- Who takes responsibility for protection?
Web Browser Vulnerabilities

Internet Explorer 7.x | Firefox 3.x | Safari 3.x | Opera 9.x

- Not
- Less
- Moderately
- Highly
- Extremely

Statistics courtesy of Secunia, Inc.

Copyright 2009 Zscaler, Inc.
WASC Threat Classification

1 - Authentication
- Brute Force Attack
- Insufficient Authentication
- Weak Password Recovery Validation

2 - Authorization
- Credential/Session Prediction
- Insufficient Authorization
- Insufficient Session Expiration
- Session Fixation

3 - Client-Side
- Content Spoofing
- Cross-Site Scripting
- Cross-Site Request Forgery

4 - Command Execution
- Buffer Overflow
- Format String Attack
- OS Commanding
- SQL Injection
- SSI Injection
- XPath Injection

5 - Information Disclosure
- Directory Indexing
- Information Leakage
- Path Traversal
- Predictable Resource Location

6 - Logical Attacks
- Abuse of Functionality
- Denial of Service
- Insufficient Anti-automation
- Insufficient Process Validation
WhiteHat Security Statistics

December 2008

Naked Browser Attacks

Chart Copyright © 2008 WhiteHat Security, Inc.
Cross-Site Scripting

1. Generate Traffic
2. Active Script Sent in Request
3. Active Script Embedded in Response
4. Active Script Executes (e.g. cookie theft)

Attacker

Spam

Victim

Vulnerable Web Site

Copyright 2009 Zscaler, Inc.
Google (NSDQ: GOOG) says it has repaired a security issue in its Orkut social networking site that allowed a worm to propagate among at least 400,000 Orkut users.

"Google takes the security of our users very seriously," a company spokesperson said in an e-mail Wednesday evening. "We worked quickly to implement a fix for the issue recently reported in Orkut. We also took steps to help prevent similar problems in the future. Service to Orkut was not disrupted during this time."
Orkut Attack

Process

• Email received from another Orkut user announcing a new scrapbook entry (message)
• Persistent XSS vulnerability allowed JavaScript to be embedded in scrapbook
• Simply viewing the entry caused addition to the "Infectados pelo Vírus do Orkut" (infected by the Orkut virus) group
• Scrapbook entry then sent to all friends and propagation continues

Risk

• Social networking sites allow and encourage user supplied content
• Weak input validation makes such attacks possible
• No user action required beyond viewing a page
• No malicious intent – attack conducted to highlight security vulnerability
Case Study: Banca Fideuram

HTTPS URL

https://www.fideuramonline.it/script/LoginServ

ATTENZIONE!

FORM INJECTED BY FRAUDSTER
Banca Fideuram Attack

Process

- Social Engineering – Spam email used to generate traffic
- IFRAME injected into login page
- Injected code obfuscated - String.fromCharCode()
- Original login form obfuscated by attacker content
- Login credentials sent to attackers in Taiwan
- Login credentials redirected to original bank site

Risk

- XSS on SSL protected page
- Traditional browser security indicators useless
  - Address bar, SSL certificate, lock and key, HTTPS, etc.
- Victim’s are unaware of attack due to successful login
'Clickjacking’ Attack Hides Behind the Mouse

Posted by Robert Vamosi
October 8, 2008 12:51 PM PDT

**On Tuesday, Adobe issued a workaround for a serious issue that could allow attackers to change the security settings within Flash.**

Termed "clickjacking," the process gives "an attacker the ability to trick a user into clicking on something only barely or momentarily noticeable," wrote WhiteHat Security CTO Jeremiah Grossman in a blog posting last month. He went on to say that while "guarding against Clickjacking was largely the browser vendors' responsibility," both he and Robert Hansen agreed to withhold further information and even canceled their talk recently at OWASP NYC AppSec 2008 Conference at the request of Adobe. In return, Adobe thanked the researchers.
Clickjacking

- Embedded Content
  - Attacker controlled site
  - 3rd party content added in IFRAME

- Layering
  - Attacker controlled content layered on top
  - Z-index property

- Obfuscation
  - Attacker content made transparent
  - Opacity property

☑️ http://FakeSite.com

Admin Interface
Reset Password OK
Adobe Flash

Website Privacy Settings
For websites you have already visited, view or change the privacy settings for access to your camera and/or microphone.

- Always ask
- Always allow
- Always deny

Visited Websites

<table>
<thead>
<tr>
<th>Privacy</th>
<th>Websites</th>
<th>Used</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>✭</td>
<td><a href="http://www.virgingalactic.com">www.virgingalactic.com</a></td>
<td>1 KB</td>
<td>100 KB</td>
</tr>
<tr>
<td>✷</td>
<td>msnbcmedia.msn.com</td>
<td>1 KB</td>
<td>100 KB</td>
</tr>
<tr>
<td>✭</td>
<td>video.google.com</td>
<td>1 KB</td>
<td>100 KB</td>
</tr>
<tr>
<td>✷</td>
<td>documents.scribd.com</td>
<td>2 KB</td>
<td>100 KB</td>
</tr>
</tbody>
</table>

1. Clear websites
2. Select targeted site
3. Always allow
Adobe Flash
IE8 Clickjacking Controls

IE8's clickjacking fix not much help, security researchers say

By Robert McMillan

January 27, 2009 (IDG News Service) New technology from Microsoft Corp. designed to protect Internet Explorer users from a powerful new Web-based attack will not fix the problem, some security researchers said Tuesday.

Microsoft released the technology yesterday as part of the Release Candidate 1 version of its upcoming Internet Explorer 8 browser, saying that the feature provides "consumer-ready" protection for an attack known as clickjacking.
BSoD – NIN Concert
Other Naked Attacks

**Cross-Site Request Forgery (CSRF)**
- Browser/server trust is abused by social engineering victim to perform an unintended action (e.g. password rest, post content, etc.)

**HTTP Response Splitting**
- Ability to inject CRLF characters into the headers of a response, thereby generating two responses to a single request – one fully attacker controlled
- Can be used to poison web caches with attacker controlled content

**Content Spoofing**
- Ability to override the content of a web page
- Valuable for phishing attacks
- Can leverage browser vulnerabilities or weaknesses in web application logic

**DNS Cache Poisoning**
- LAN or Internet based attacks (aka Dan Kaminsky attack)
- Allows for traffic redirection to attacker controlled sites

**URL Redirection**
- Sites use redirection techniques to track users leaving the site
- Can be abused by phishers attempting to hide destination site
Challenges

- Legitimate Traffic
  - Identifying attacks can be like looking for hay in a haystack

- Unique Attacks
  - Small changes in content/encoding render signatures useless

- Targeted attacks
  - Difficult to anticipate/identify
Defending Against Attack

Server vs. Client

- Virtually all solutions/papers focus on securing web applications, not browsers
- This protects the DMZ, but not the desktop

Protecting Servers is Easy

- Hundreds of desktops for every server
- Server content has change control
- Administrators have security knowledge
Existing Solutions

Host Based

- NoScript
  - Firefox extension
  - XSS and clickjacking detection
- Internet Explorer 8
  - XSS detection
  - Clickjacking protection – requires web app. component

Network Based

- Some IDS/IPS signatures for specific attacks (e.g. XSS vuln. on XYZ blogging application)
Boarding Dr. Watson
Defense In Depth

Monitor
- Identify anomalous traffic patterns

Manage
- Control what users can do on the web, not just where they can go

Merge
- Incorporate third party data feeds

Educate
- Empower users to proactively identify risks
Monitor

Logging
- Consolidate logs from separate Internet gateways
- Web proxy and/or DNS logs
- Consider SaaS solutions for logging

Analysis
- Establish baseline patterns for normal traffic
- Look at moving averages as opposed to fixed time periods
- Identify sudden spikes in traffic, especially to previously non-existent destinations

Reporting
- Reports must be reviewed to be meaningful – assign ownership
- Continually adjust thresholds to limit false positives
Manage

- Not everyone requires equivalent Web access
- Identify meaningful roles
- Manage centrally via LDAP/AD

Roles

- Allow/deny functionality, not just access
- E.g. Marketing can post to content to Facebook while others can only view profiles

Functionality
Merge

**Sources**
- Commercial data feeds
  - SiteScout, CommTouch, Sunbelt Software
  - Free
    - Browser based blacklists
    - PhishTank. Google SafeBrowsing, OpenDNS

**Integration**
- Custom
- Secure web gateways
- SaaS web security solutions
- DNS blacklists

**Metrics**
- Regularly check reports – what is being blocked and for whom?
- Evaluate value provided by various data sources

---

Copyright 2009 Zcaler, Inc.
Educate

Lather
- Empower users through education
- Not just to avoid risks but to recognize the need for escalation

Rinse
- Provide regular content – slow but steady wins the race
- Use multiple formats – we all learn differently

Repeat
- Keep it coming – we forget and the world changes
- Test users
- Don’t rely on education alone!
Future

Vendors

- Need to take responsibility for naked attacks
- Applications need to be proactively secure
  - Not just blacklists (e.g. phishing/malicious URLs)
- Application developers (e.g. IE) need to look to development platforms (e.g. .Net) for inspiration

Attackers

- Increased use of targeted attacks
- Malicious web based worms
- Abuse of web APIs
Restaurant Virus?
Questions?

I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. www.mrburns.nl before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions. I will use Google before asking dumb questions.

Michael Sutton - VP, Security Research
http://research.zscaler.com
Michael.Sutton@zscaler.com