Detecting In-Flight Page Changes with Web Tripwires

Charles Reis
Steve Gribble, Yoshi Kohno, Nick Weaver
University of Washington, ICSI
ISP-Injected Ads

- Surprising reports of web page modifications
- How often does this occur?
Detecting Page Changes

- Can detect with JavaScript
- Built a Web Tripwire:
  - Runs in client’s browser
  - Finds most changes to HTML
  - Reports to user & server

http://vancouver.cs.washington.edu
How it Works

- Fetch and render original page
- Fetch JavaScript code in background
- Second, encoded copy of page
- Compare against page’s source code

http://vancouver.cs.washington.edu
Attracting Visitors

✦ Wanted view of many clients on many networks

✦ Posted to Slashdot, Digg, etc.

✦ Visits from over 50,000 unique IP addresses

http://vancouver.cs.washington.edu
Outline

- Detecting In-Flight Changes
- Measurement Results
- Dangerous Consequences
- Web Tripwires for Publishers
Many Users Affected

- 657 clients saw changes (1.3%)
  - Many made by client software
  - Some made by agents in network
- Diverse incentives
- Often concerning for publishers

http://vancouver.cs.washington.edu
Many Types of Changes

- Internet Service Providers
- Enterprise Firewalls
- Client Proxies
- Malware
Changes by ISPs

- **Injected Advertisements** (2.4%)
  - NebuAd, MetroFi, LokBox, ...

  *Revenue for ISP; annoy users*

- **Compression** (4.6%)

Growing Trend?
PerfTech, Front Porch, Adzilla, Phorm

http://vancouver.cs.washington.edu
Changes by Enterprises

- Security Checking Scripts (2.3%)
- BlueCoat Web Filter

Safer for clients; reduce risk

http://vancouver.cs.washington.edu
Changes by Client Proxies

- Popup & Ad Blockers (71%)
- Zone Alarm, Ad Muncher, ...

Less annoying; impact revenue

http://vancouver.cs.washington.edu
Changes by Malware

Adware (1 client)

http://vancouver.cs.washington.edu
Changes by Malware

- **Adware** (1 client)
- **Worms** (2 clients)

Helps malware author; risk to user

http://vancouver.cs.washington.edu
Outline

Detecting In-Flight Changes

Measurement Results

Dangerous Consequences

Web Tripwires for Publishers
Unanticipated Impact

- Some changes inadvertently broke pages
- JavaScript errors
- Interfered with MySpace / forum posts
Introduced Vulnerabilities

- **XSS** allows script injection
  - Usually fixed at server
- Some proxies made otherwise safe pages vulnerable
  - Ad Muncher, Proxomitron
- Affected most HTTP pages
  - Like a root exploit

http://vancouver.cs.washington.edu
XSS via Proxy

- Proxy injected script code
- **Page URL** was included in code
- Attacker could place script code in a valid URL
- Users who follow the URL run injected code
Example Exploit

- Redirect user to Google
- Inject script code into search form
- Append exploit code to all outgoing links

http://vancouver.cs.washington.edu
Vulnerability Aftermath

- Reported vulnerabilities; now fixed
- Web tripwires can help find vulnerabilities
  - Search for URL in page changes
Outline

- Detecting In-Flight Changes
- Measurement Results
- Dangerous Consequences
- Web Tripwires for Publishers
How to React?

- Option 1: Use HTTPS
  - Encryption prevents in-flight changes
  - But... costly and rigid
  - Can’t allow security checks, caching, etc.

http://vancouver.cs.washington.edu
Web Tripwires

- JavaScript code to detect changes
- Easy for publishers to deploy
  - Configurable toolkit
- Web tripwire service
- But... not cryptographically secure
- Can be robust in practice

http://vancouver.cs.washington.edu
# Tradeoffs

## HTTPS vs. Web Tripwires

<table>
<thead>
<tr>
<th>HTTPS</th>
<th>Web Tripwires</th>
</tr>
</thead>
<tbody>
<tr>
<td>✦ Prevents most changes, as well as some useful services</td>
<td>✦ Detects most in-flight changes</td>
</tr>
<tr>
<td>✦ Cryptographically robust</td>
<td>✦ Could face an arms race</td>
</tr>
<tr>
<td>✦ Expensive: certificates, computation, extra RTTs</td>
<td>✦ Obfuscation can challenge adversaries</td>
</tr>
<tr>
<td></td>
<td>✦ Inexpensive to deploy</td>
</tr>
</tbody>
</table>

Performance Impact

- Relative to HTTPS, web tripwires have:
  - Low latency
  - High throughput
Conclusion

- HTTP web pages are being changed in flight
  - Real negative impact for publishers & users
  - Page rewriters have dangerous power
- Web tripwires can help publishers react

This work partially supported by NSF.

http://vancouver.cs.washington.edu