URL Rewriting for Good, not Evil
Using Alternative Resource Locators

Bryan Sullivan
Senior Security Program Manager, SDL
Microsoft
Top Web Vulns Have a Common Factor

- Cross-Site Scripting
  - OWASP #1
- Cross-Site Request Forgery
  - Growing fast
- Open Redirect Phishing
  - Lots of MSRC cases
Propagation Via Poisoned Hyperlinks

- **XSS**
  - `foo.aspx?bar=<script>alert('xss')</script>

- **XSRF**
  - `foo.aspx?action=buy&symbol=GM`

- **Redirect Phishing**

- Redirectors (TinyURL, bit.ly) make things worse
Browser History Theft

- Use any of the following:
  - Script
  - CSS
  - iframe timing attacks

- Can’t list all, but can check specific sites or searches
  - www.verylargebank.com
  - www.bing.com/search?q=scarlett+johannson
Solution: Personalize Hyperlinks

- Not URLs but PRLs (Personalized Resource Locators)
- Malicious link created by an attacker could only be used by him/her

- We already have an implementation mechanism:

  URL Rewriting
URL Rewriting in Brief

http://www.site.com/.foo.html

http://www.site.com/{sessionID}/foo.html

• This usually causes more problems than it solves
  ▫ Session hijacking
  ▫ Session fixation
Example

http://www.xbox.com/{abc123...}/rockband.aspx
Rewrite with Canary, not Session ID

• Outbound:
  1. Server creates shared secret token (canary)
  2. Store canary value in session state
  3. Rewrite canary into URL
  4. Pass SID in cookie as usual

• Inbound:
  1. Server compares incoming canary against stored
  2. If missing or mismatched, reject request
Poisoned Links are Now Useless

www.site.com/{a1b2...}/foo.aspx?action=buy&symbol=GM

• Send it around in an email
• Post it on a page
• Hide the payload with a redirector

• None of these matter, because victim can’t use it
History Theft Becomes Infeasible

- Assume GUIDs are used for canaries
- Attacker must check all of these:

  
  www.site.com/{00000000-0000-0000-0000-000000000000}/
  www.site.com/{00000000-0000-0000-0000-000000000001}/
  www.site.com/{00000000-0000-0000-0000-000000000002}/
  ...

- $3.4 \times 10^{38}$ possibilities
  - This would take a really, really long time to check
Stateless Alternative: Timed URLs

- **Outbound:**
  1. Get the current date/time
  2. Create a keyed hash of the timestamp
  3. Write the timestamp and hash into the URL

- **Inbound:**
  1. If timestamp or hash missing, reject request
  2. If timestamp and hash mismatch, reject request
  3. If timestamp older than specified expiration age (i.e., 5 minutes), reject request
Poisoned Links are Almost Useless

http://www.site.com/{07.30.2009...}/?action=buy&symbol=GM

• Links work for everyone, but only for a short lifespan
  ◦ 5 minutes or whatever the server has configured

• Seriously limits potential damage
History Theft Still Infeasible

- Attacker must make requests, store keyed hashes
- Assume millisecond granularity for timestamp
- Attacker must check all of these:

  www.site.com/{2009-07-30-T1330000000-HASH}/
  www.site.com/{2009-07-30-T1330000001-HASH}/
  www.site.com/{2009-07-30-T1330000002-HASH}/
  ...

Appropriate Cryptography

• You must include a hash of the timestamp
  ▫ Otherwise attacker could create poisoned URLs with arbitrary expiration dates (+10 years)

• You must key the hash
  ▫ Otherwise attacker could precompute a valid hash

• Use SHA-2
  ▫ If you’re going to go to this much trouble, use a secure algorithm
Landing Pages

• You must designate one or more pages as “landing pages”
  ▫ These do not require canaries or keyed timestamps
  ▫ Otherwise no one will be able to use the site

[poandpo.com]
Bypassing Defenses

• External XSS will completely defeat these defenses
  ▫ Landing page
  ▫ Different application, same domain
• Use XSS to inject XHR
  ▫ Read token + redirect
  ▫ Read token + modify DOM

• POST redirection will defeat timed URLs
Temporary URL Bypass Technique

1. Attacker sets up malicious page [www.evil.com]
2. When called, malicious page sends request to protected page to determine valid token
3. Malicious page then redirects user to valid page

• Attacker now only needs to lure user to his malicious page as usual
  ◦ Phishing, etc
Other Unfortunate Side Effects

- Can’t email links
- Can’t bookmark links
- Search engines can’t index the site
Best Usage Scenario

• Don’t apply to entire site
• Apply to secure subdomain

• www.verylargebank.com (regular URLs)
  ▫ Locations, hours
  ▫ Current interest rates
• secure.verylargebank.com (alternative URLs)
  ▫ Account balances
  ▫ Transfers
Conclusions

• Alternative URLs can be useful as defense-in-depth
• Don’t just apply them globally
• Continue to find & fix vulnerabilities

• More resources
  ▫ MSDN Magazine, March 2009, Security Briefs
  ▫ blogs.msdn.com/sdl
  ▫ My alias: bryansul