Find Blue Oceans

Through the Competitive World of Bug Bounty

Muneaki Nishimura (nishimunea)

Weekend Bug Hunter

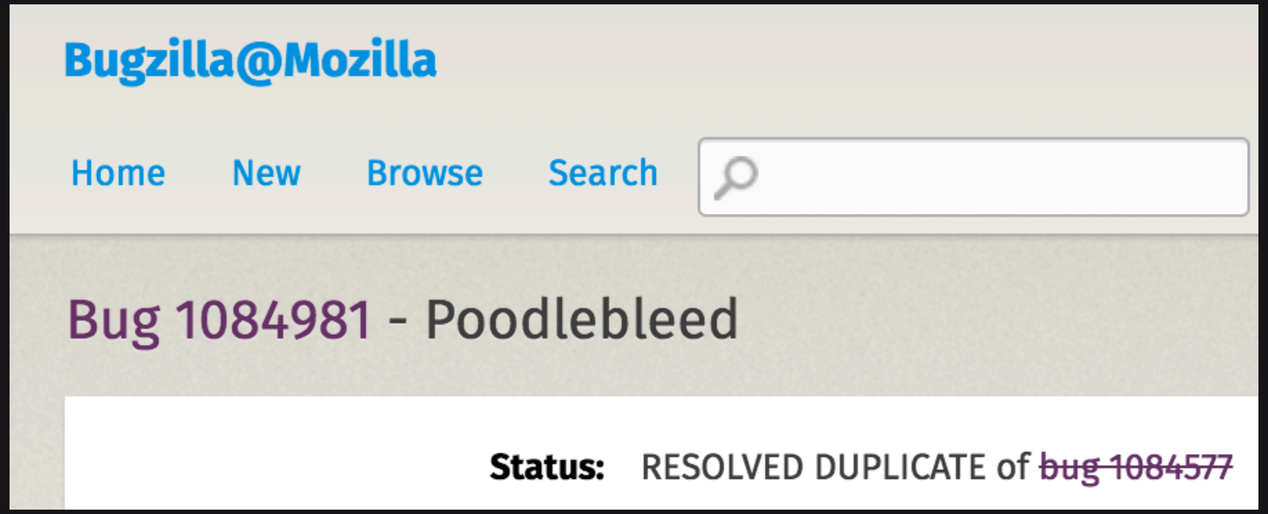
Lecturer of Web Security of Security Camp in Japan

Found **30** Bugs in Firefox
Received Reward of **$70,000+** from Mozilla

Bug 1065909	Bug 1109276	Bug 1162018	Bug 1196740	Bug 1223743
Bug 1069762	Bug 1148328	Bug 1162411	Bug 1198078	Bug 1224529
Bug 1080987	Bug 1149094	Bug 1164397	Bug 1207556	Bug 1224906
Bug 1101158	Bug 1157216	Bug 1190038	Bug 1208520	Bug 1224910
Bug 1102204	Bug 1158715	Bug 1190139	Bug 1208525	Bug 1227462
Bug 1106713	Bug 1160069	Bug 1192595	Bug 1208956	Bug 1258188

Bug Bounty Programs are **Competitive**

Required a lot of time and techniques to avoid duplicates



1084981 - Poodlebleed
https://bugzilla.mozilla.org/show_bug.cgi?id=1084981

Hunting Time is Limited (4:00-7:00 AM)

Weekdays

4:00 7:00

Hunt

Weekend

4:00 7:00

Hunt

Find and Create **Uncontested** Bounty Targets

Give you some tips from my experience of Firefox bug bounty program

Tip #1

Find Bugs in **Web Platforms**

”Fox-keh" (C) 2006 Mozilla Japan

- Browsers and networking features in OS are less competitive targets
- There are **common pitfalls** but not widely known
- Developers make similar mistakes whenever they introduce new features

Security Advisories for Firefox

Impact key

CRITICAL	Vulnerability can be used to run attacker code and install software, requiring no user interaction beyond normal browsing.
HIGH	Vulnerability can be used to gather sensitive data from sites in other windows or inject data or code into those sites, requiring no more than normal browsing actions.
MODERATE	Vulnerabilities that would otherwise be High or Critical except they only work in uncommon non-default configurations or require the user to perform complicated and/or unlikely steps

Mozilla Security

Security Advisories

Known Vulnerabilities

Bug Bounty

Firefox Hall Of Fame

Mozilla Web and Services Hall Of Fame

Security Blog

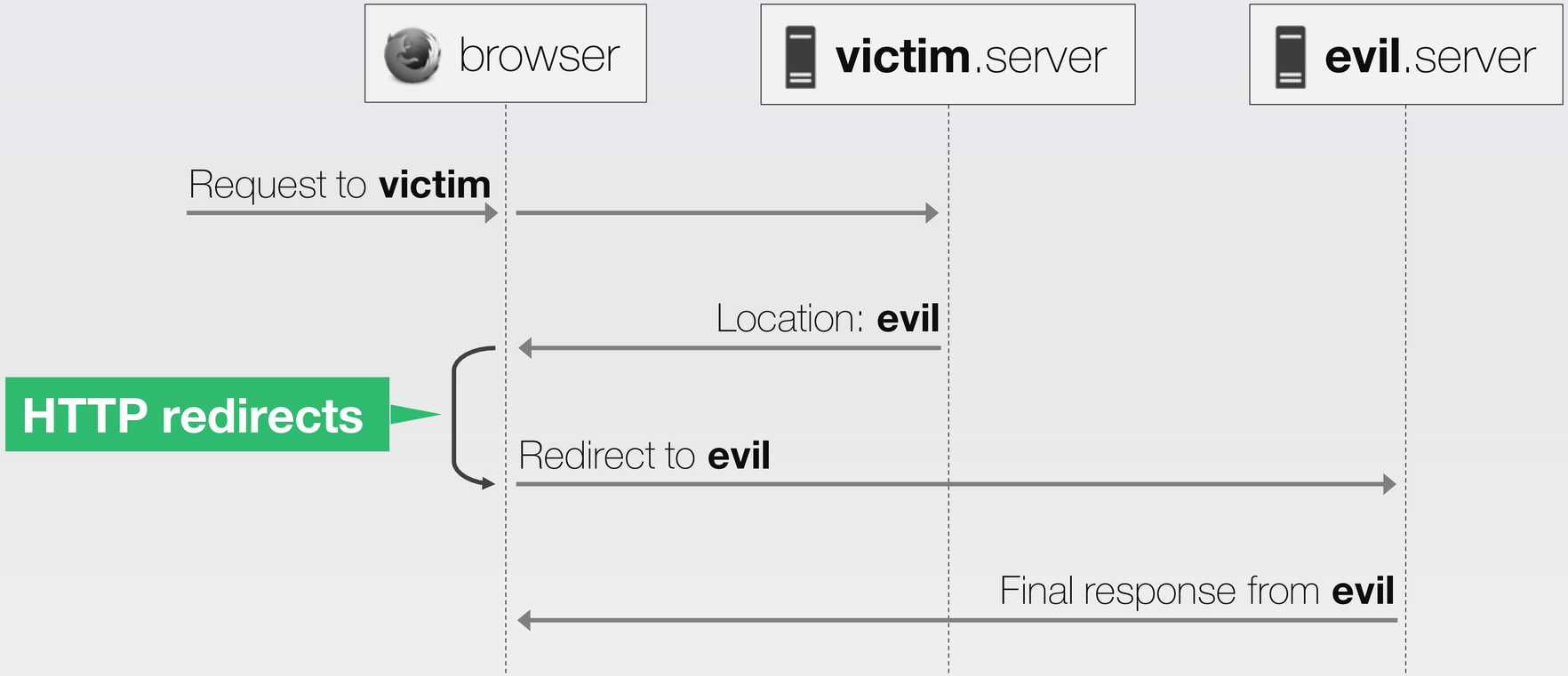
Learn **Known Bugs** from **Security Advisories**

and try the same attack scenario on similar features

Mozilla Foundation Security Advisories
https://www.mozilla.org/en-US/security/advisories/

Example

Improper Handling of HTTP Redirect


browser
victim.server
evil.server
Request to victim
Location: evil
HTTP redirects
Redirect to evil
Final response from evil

Developers **expect** following code properly gets a response **only from victim**

```
if( request.url.indexOf('http://victim.server/') === 0 )
{
    resource = http.get(request.url);
    parse(resource);
}
```

But still possible to load a resource from **evil**

```
if( request.url.indexOf('http://victim.server/') === 0 )
{
    resource = http.get(request.url);
    parse(resource);
}
```

Resource from **evil** might be used due to redirect

Similar bugs were found other than Firefox

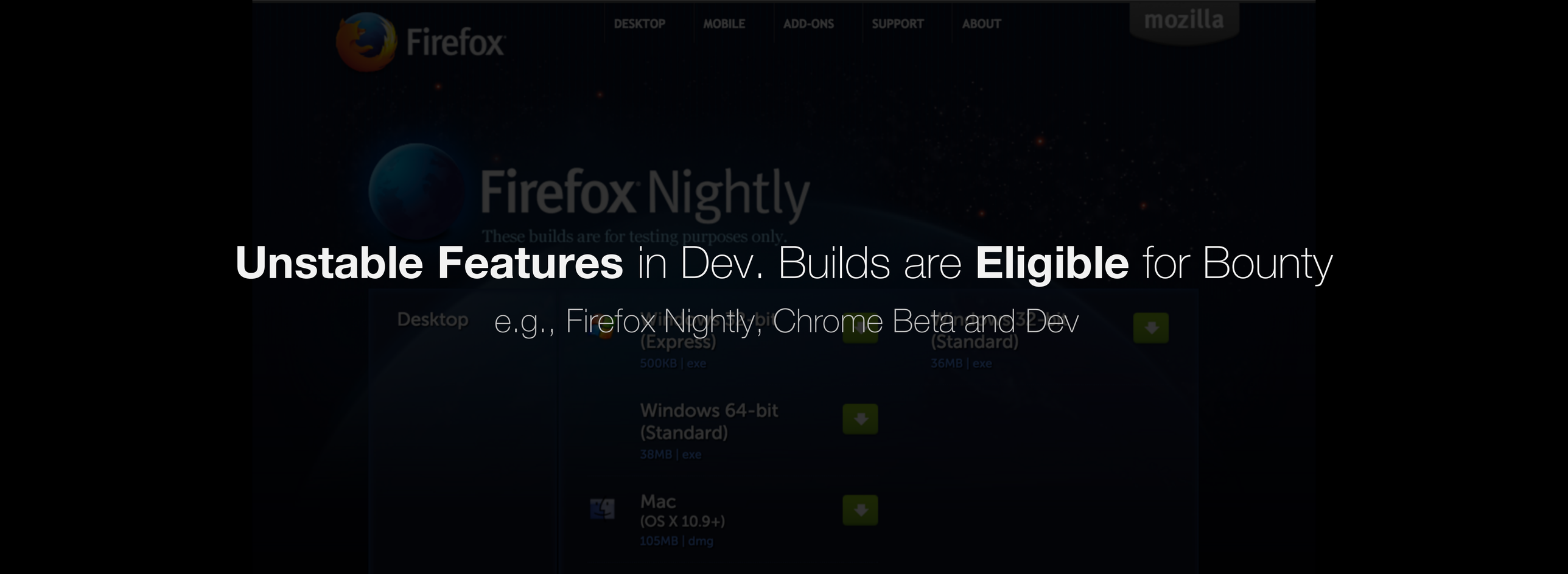
Firefox	• Bug 1111834 - Cross-origin restriction bypass in navigator.sendBeacon • Bug 1164397 - Origin confusion in cache data of Service Workers • Bug 1196740 - Cross-origin restriction bypass in Subresource Integrity (SRI)
Chrome	• CVE-2015-6762 - Cross-origin restriction bypass in CSS Font Loading API
Safari	• CVE-2016-1782 - Non-http port banning bypass in WebKit

Tip #2

Find Bugs in **Unstable Features**

”Fox-keh" (C) 2006 Mozilla Japan

Firefox Nightly Builds
https://nightly.mozilla.org/



Unstable Features in Dev. Builds are **Eligible** for Bounty

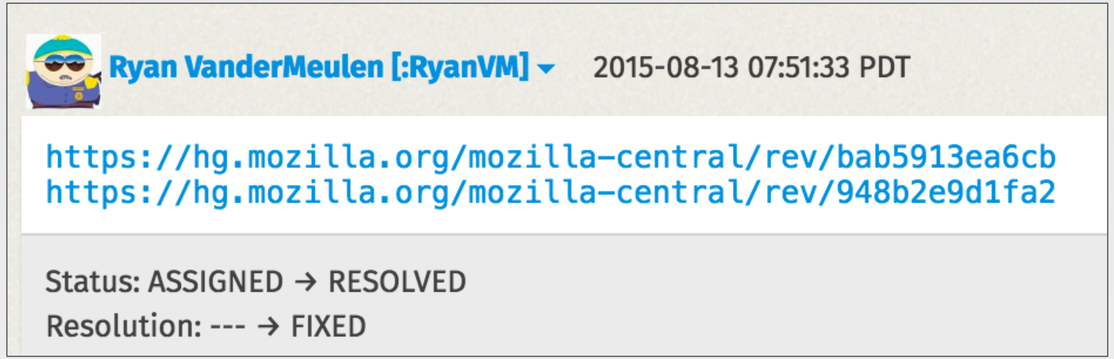
e.g., Firefox Nightly, Chrome Beta and Dev

Example

Subresource Integrity (SRI)

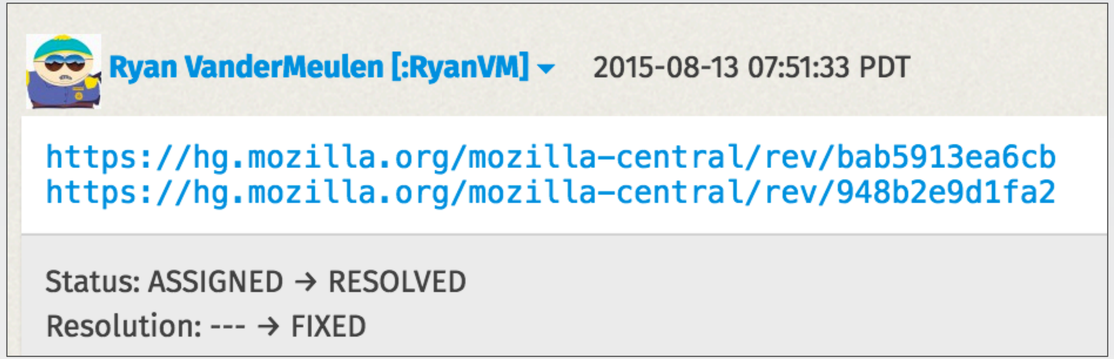
2015.08.13

SRI has been enabled in Nightly



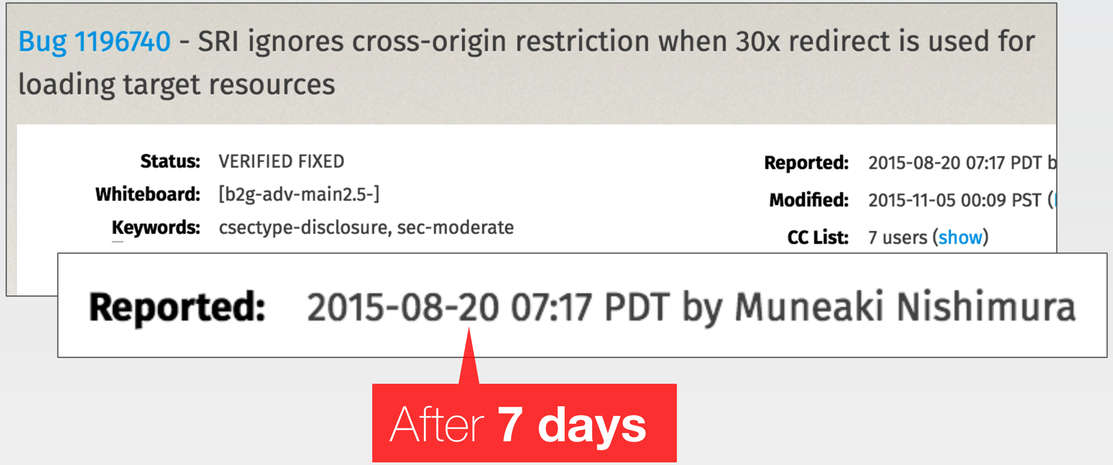
2015.08.13

SRI has been enabled in Nightly



2015.08.20

Reported the first security bug in SRI



2016.01 - Implemented Service Workers on Firefox 44

Reported an origin confusion (Bug 1162018) on Nightly 41 at **2015.05**

2016.08 - Planned to introduce Web Extensions on Firefox 48

Reported a privilege escalation (Bug 1227462) on Nightly 45 at **2015.11**

2015.12 - Determined not to support HTML Imports on Firefox

Reported a sandbox bypass (Bug 1106713) on Nightly 37 at **2014.12**

Tip #3

Find Bugs in **Sub Products**

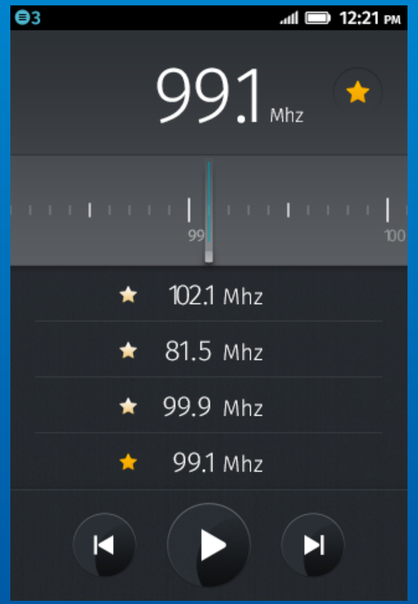
”Fox-keh" (C) 2006 Mozilla Japan

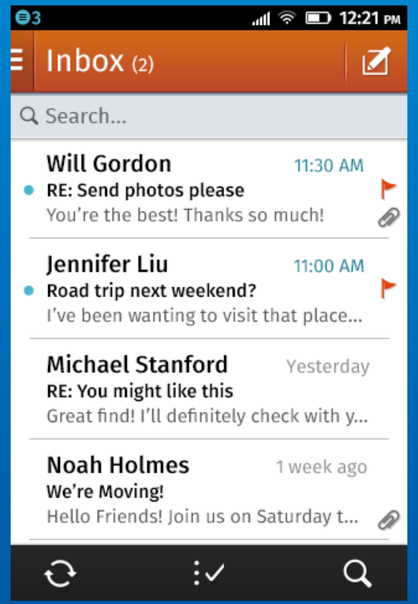
Firefox OS

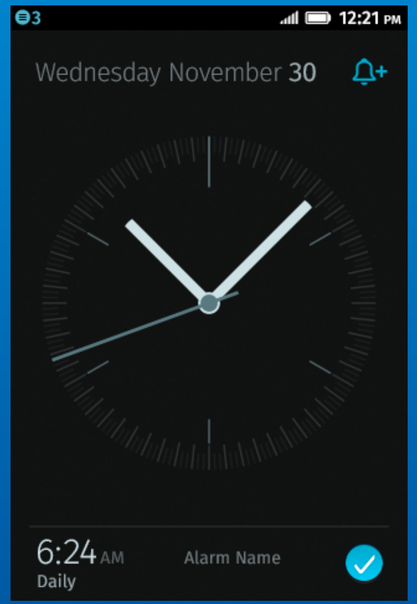
New bland name is B2G OS

- Smartphones and Smart TV OS based on Firefox browser
- All applications are made with HTML5

All applications are made with **HTML5**







All applications are made with **HTML5**

All applications are made with **HTML5**

Yes, **we know**

- Pre-installed applications run with **higher privilege**
- Protected with **Content Security Policy (CSP)**
  i.e. XSS doesn't work
- But **HTML tag injection still works** fine

Example

Special Iframe Tag Injection

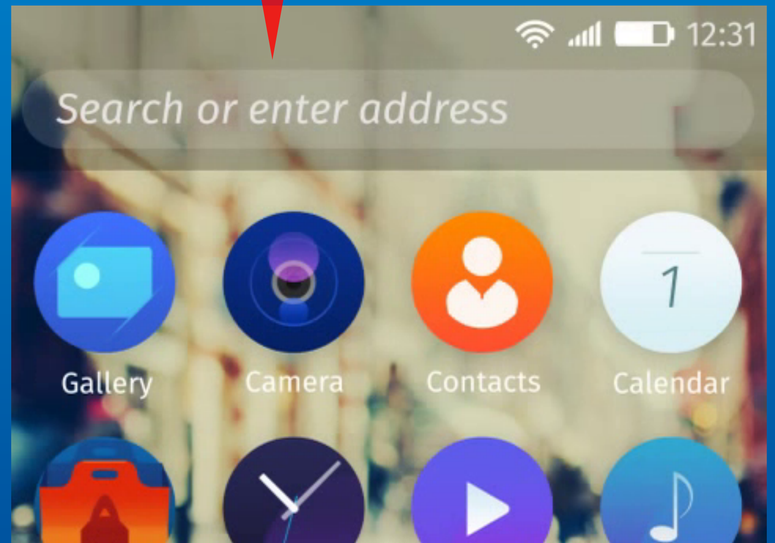
Firefox OS supports special iframe that can **embed another app** in the frame

```
<iframe mozbrowser remote
  mozapp='app://fm.gaiamobile.org/manifest.webapp'
  src='app://fm.gaiamobile.org/index.html' />
```

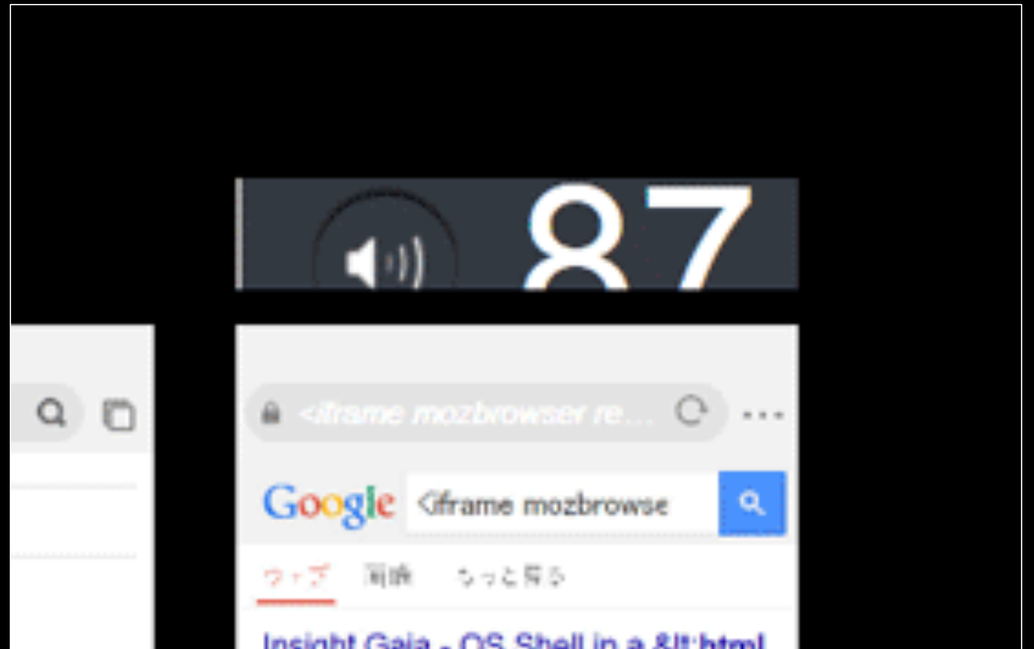
Embed **FM Radio** app.

Inject special iframe

Type **<iframe mozbrowser mozapp…>**



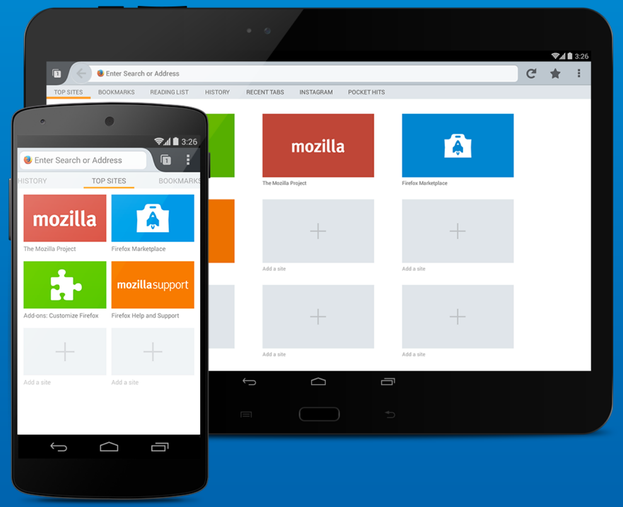
FM Radio works



Finally reported **7** similar bugs and

Received reward of **$20,000+** from Mozilla

Bug 1065909	Bug 1109276	Bug 1162018	Bug 1196740	Bug 1223743
Bug 1069762	Bug 1148328	Bug 1162411	Bug 1198078	Bug 1224529
Bug 1080987	Bug 1149094	Bug 1164397	Bug 1207556	Bug 1224906
Bug 1101158	**Bug 1157216**	**Bug 1190038**	Bug 1208520	Bug 1224910
Bug 1102204	Bug 1158715	**Bug 1190139**	Bug 1208525	Bug 1227462
Bug 1106713	**Bug 1160069**	**Bug 1192595**	Bug 1208956	Bug 1258188



Firefox for **Android**

- Firefox for Android is also **in scope** of their bounty program

- There are many Android specific features and pitfalls
  e.g. improper intent handling

Example

UXSS in Intent URL Scheme

Intent scheme URL links let you **launch another app** from a web page

```
<a href='intent://maps.google.com/maps#Intent;scheme=http;
package=com.google.android.apps.map;
S.browser_fallback_url=https%3A%2F%2Fmaps.google.com;end'>
```

Intent scheme URL link let you **launch another app** from a web page

Application name you want to launch

```
<a href='intent://maps.google.com/maps#Intent;scheme=http;
package=com.google.android.apps.map;
S.browser_fallback_url=https%3A%2F%2Fmaps.google.com;end'>
```

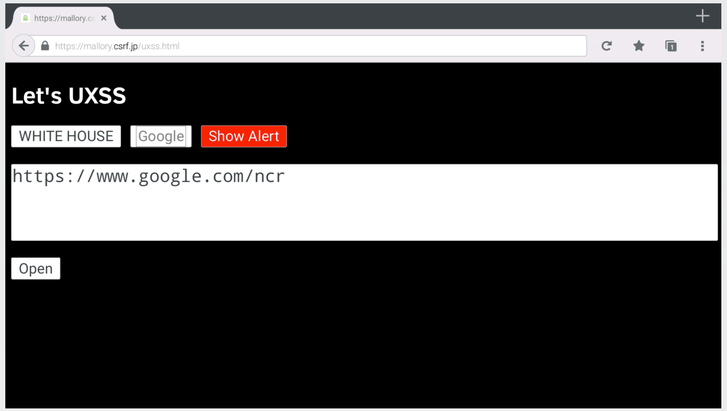
Web site URL opened if application doesn't exist

Firefox unintentionally allowed to use **any kinds of URL** as a fallback

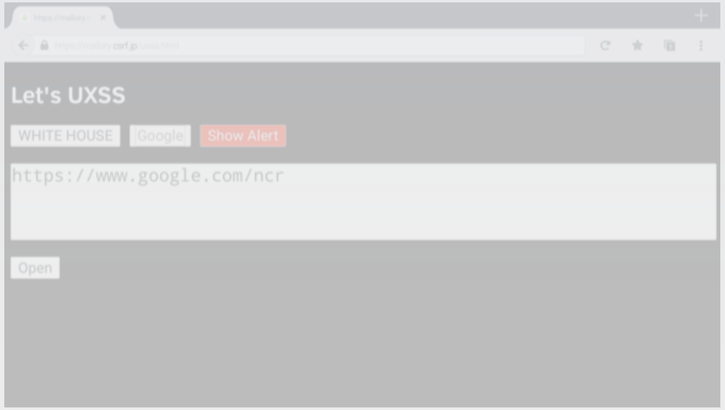
```
<a href='intent://maps.google.com/maps#Intent;scheme=http;
package=com.google.android.apps.map;
S.browser_fallback_url=javascript%3Aalert(1);end'>
```

JavaScript URL also does work

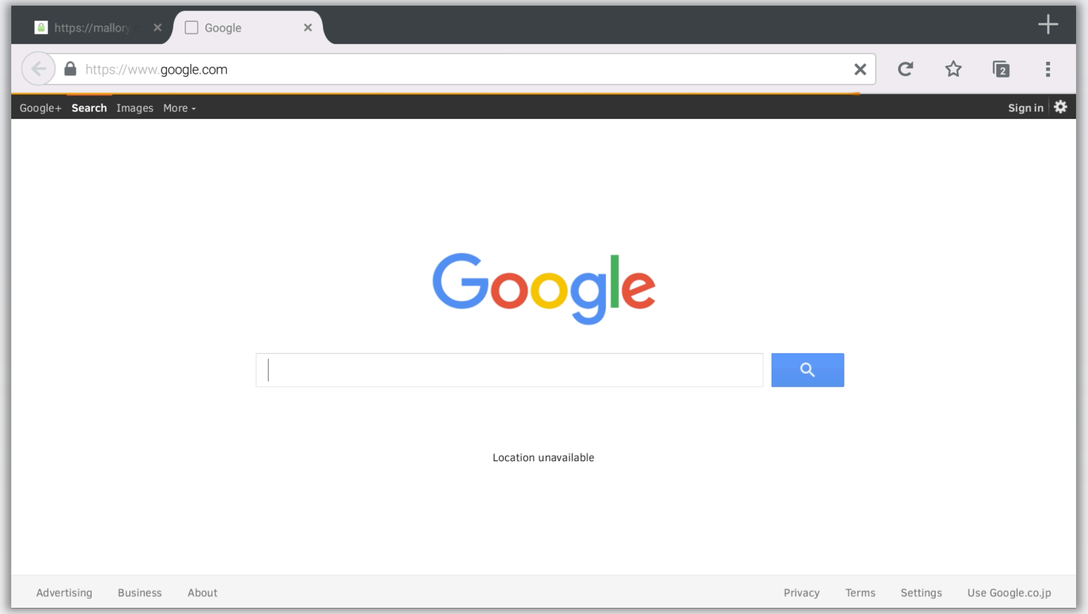
w = window.open(victim)



Attacker

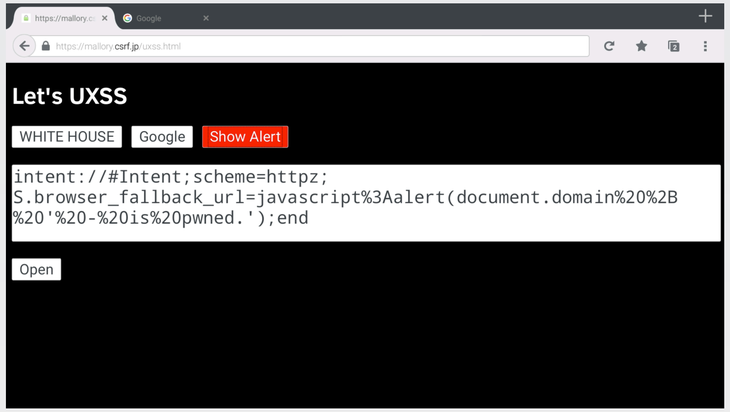


Attacker

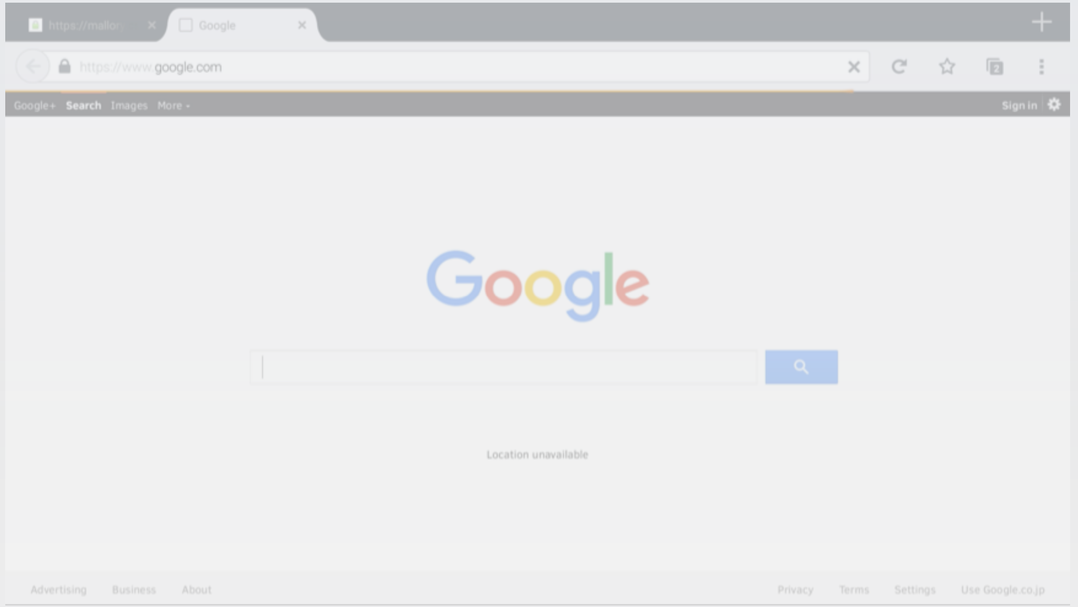


Victim

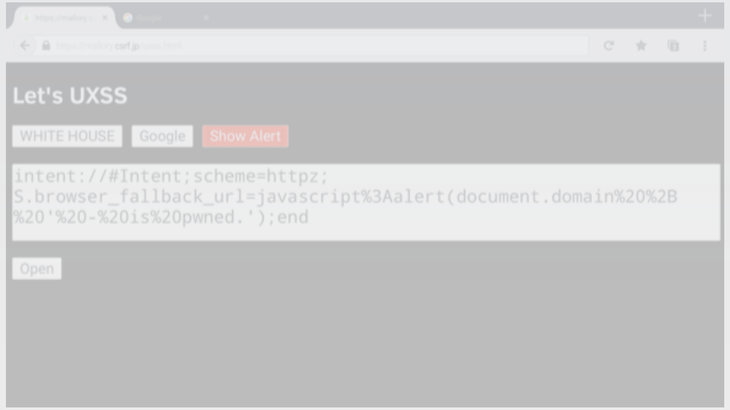
w.location = 'intent:...'



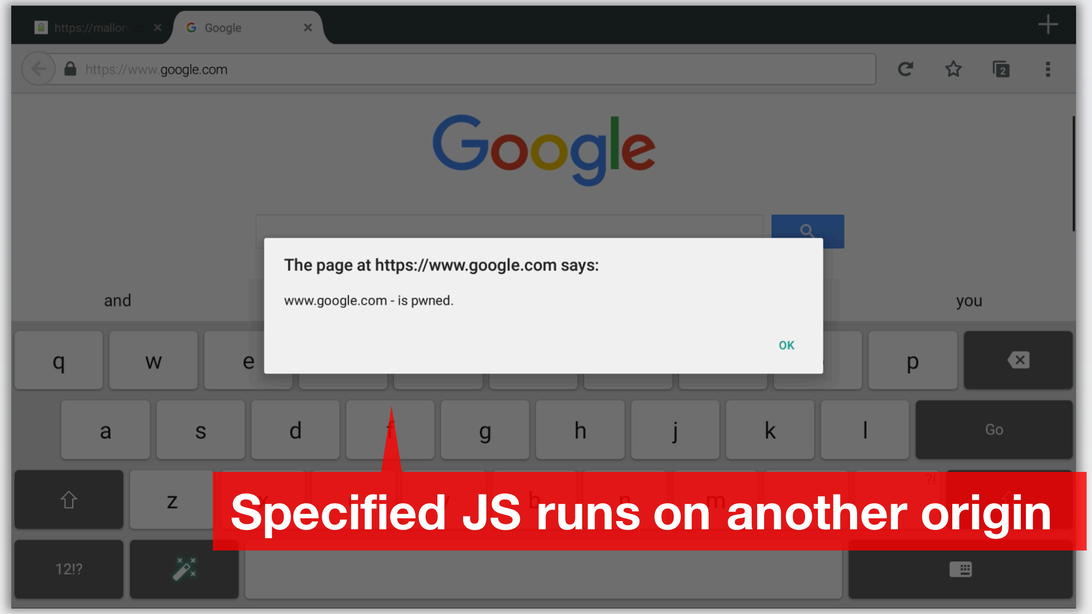
Attacker



Victim



Attacker



Victim

Firefox for **iOS**

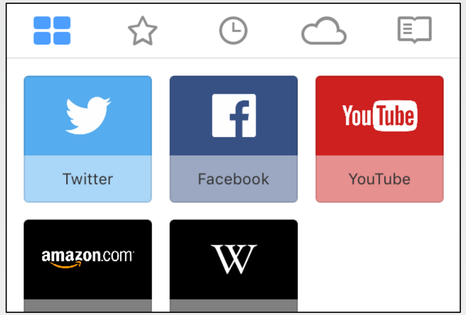
- Firefox for iOS is **eligible for a bounty** but not officially announced
- Due to Apple's restriction, Firefox for iOS uses **WKWebView** for loading and rendering web contents
- Flaw in **WKWebView is ineligible** since it's out of control of Mozilla

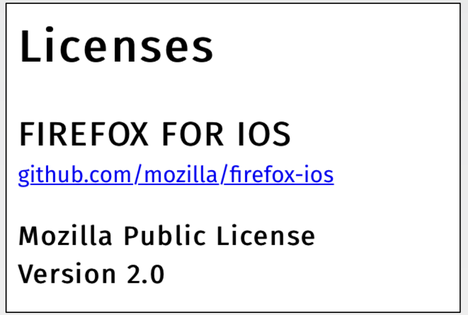
Example

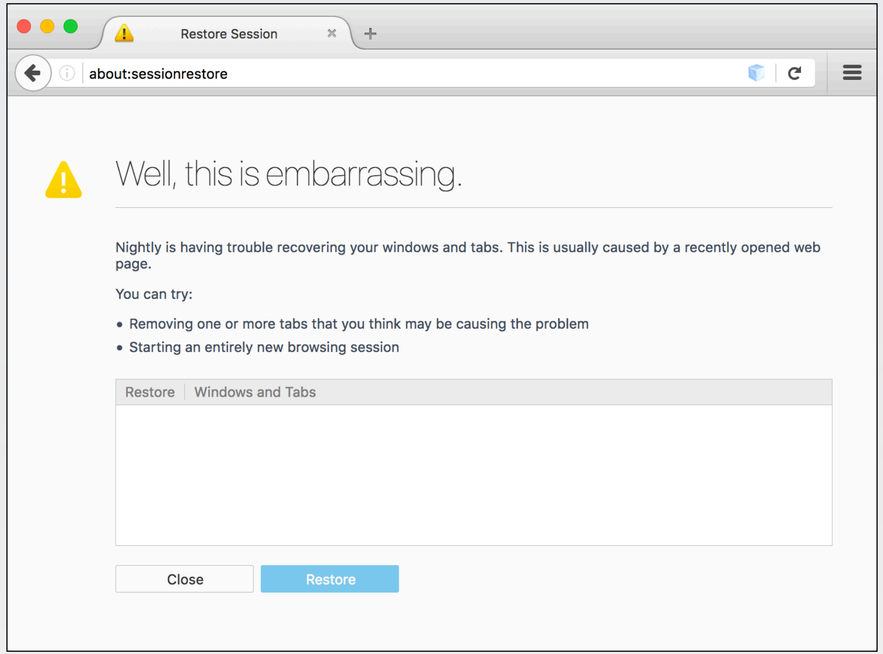
XSS in Browser Internal Page

Firefox for iOS distributes browser internal pages from **local web server**

Firefox for **Desktop**	about:home	about:license
Firefox for **iOS**	http://**localhost**:6571 /about/home	http://**localhost**:6571 /about/license







about:sessionrestore

Feature of Firefox for restoring previous browsing session after crash

- Firefox for iOS hosts session restoration feature on http://**localhost**:6571/about/sessionrestore

- Restoring URL can be set by query parameter "**history**"

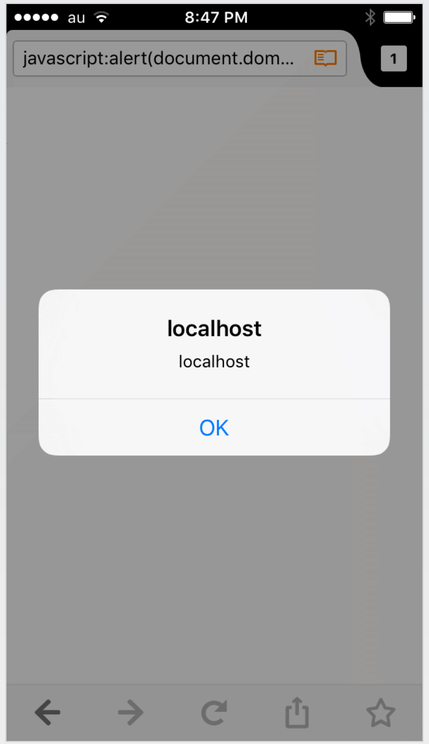
SessionRestore.html

"**history**" is parsed as JSON

```
19      var dataIndex = document.location.href.search("history");
20      var sessionRestoreComponents = JSON.parse(unescape(document.location.
21      var urlList = sessionRestoreComponents['history'];
22      var currentPage = sessionRestoreComponents['currentPage'];
23
24      // Find the index of the last selected page and restore it.
25      var restoreIndex = urlList.length + currentPage - 1;
26      if (restoreIndex < 0 || restoreIndex >= urlList.length) {
27          restoreIndex = urlList.length - 1;
28      }
29
30      window.location = urlList[restoreIndex];
```

and finally set to **window.location**

mozilla/firefox-ios/SessionRestore.html, Github
https://github.com/mozilla/firefox-ios/blob/6ab27d75e0c3365b1decffff678072a9224f149f/Client/Assets/SessionRestore.html



Any page can trigger XSS on localhost

http://localhost:6571/about/sessionrestore?

history= {"history":[" javascript:alert(document.domain) "]}

Conclusion

To avoid contested targets
try to find bugs in…

- Web Platform
- Unstable Features
- Sub Products

THANKS

乌云 WooYun

乌云白帽大会 · 2016
不插电