# Browser Exploitation
(Firefox Rant)

# Browser Security

JavaScript is implemented in C++

Remote-Code-Execution by default!

# Browser Security

```
struct JavaScriptStringObject {
       int len = 0;
       char *str = NULL;

       int Create(char *str) {
             self.len = len(str)
             self.str = malloc(len(str))
             strcpy(self.str, str);
       }
}
```

**JS Runtime**
**C++**

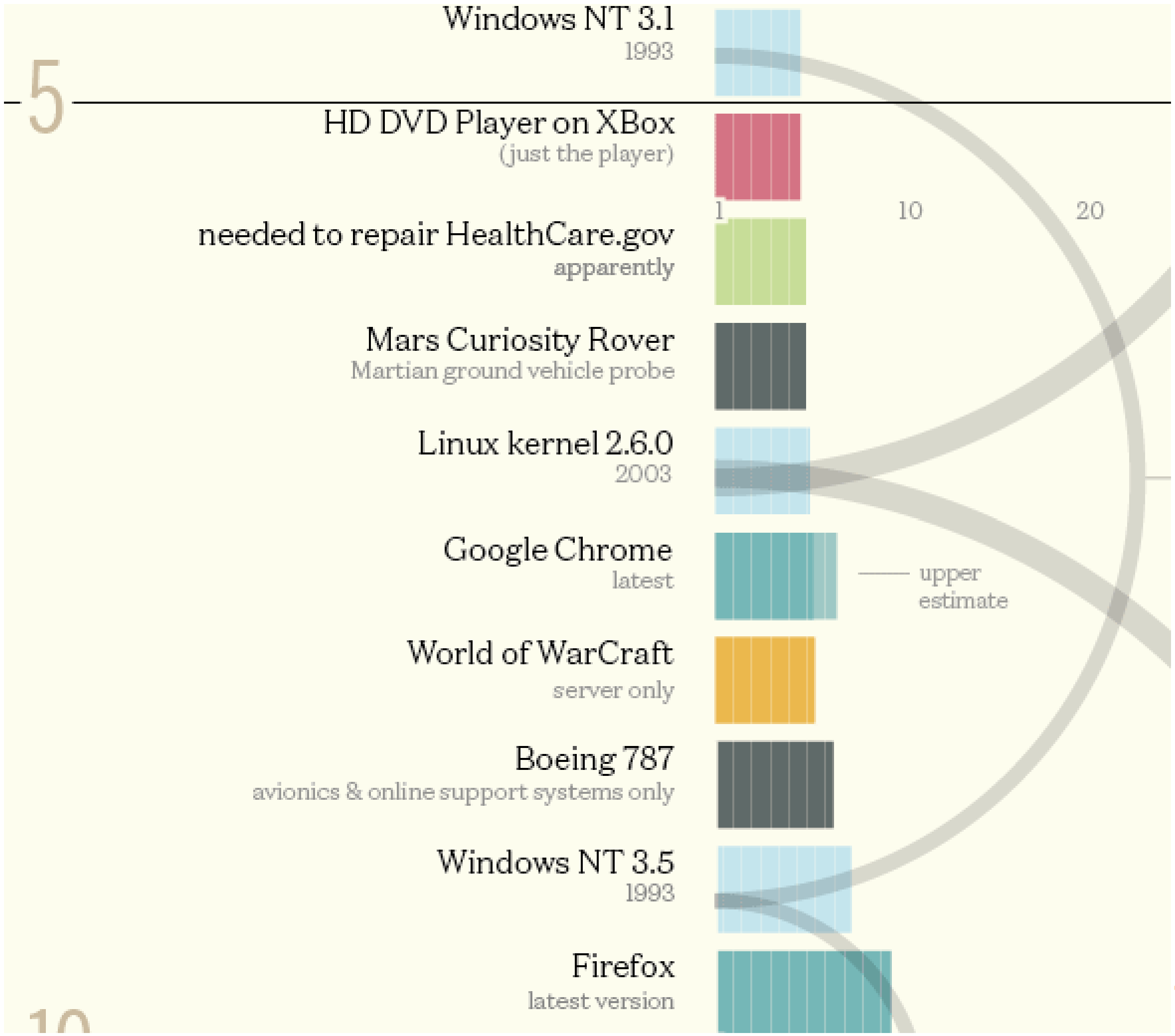
```
var jquery = “asdf”;

use_after_free(jquery, ...);

Console.log(jquery[10000]);
```

**JavaScript**

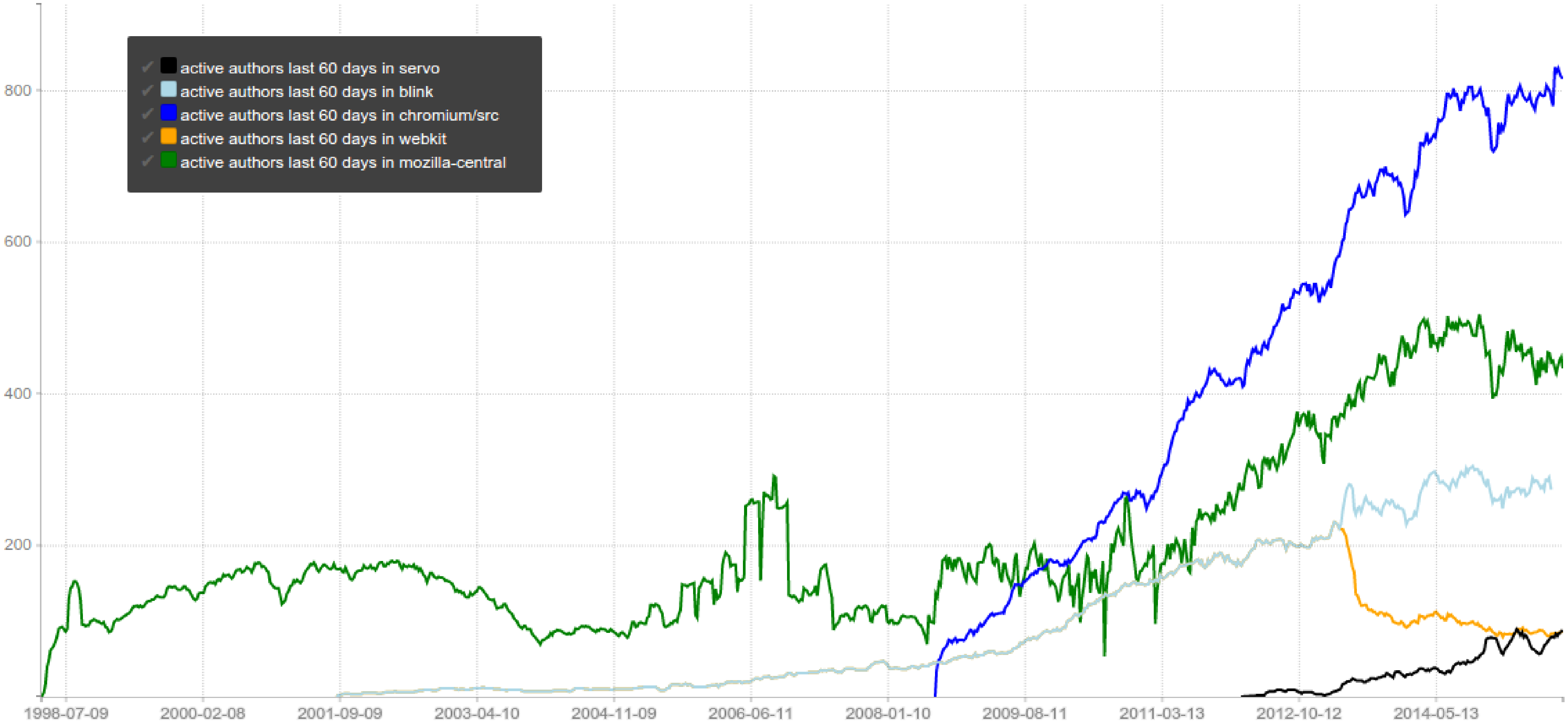
https://v8.dev/blog/sandbox

# Browser security

## Browser code size



# Developer Count (2015)



# Browser Security

Browsers:

Similar size like an OS

Support a shitload of file formats (PDF, GIF/PNG/JPEG, SVG, ...)

Can “upload” your own code (Javascript) to be executed!

# Firefox Rant

# Firefox Rant

Rant: Firefox (2016)

Good:

- Full ASLR
- (Except on OSX for 3 years… and nobody noticed)

Bad:

- No Sandbox (yet)
- No 64 bit (yet)
- No process-per-tab (yet)
- No (professional) source code auditing / SDL
- No (professional) fuzzing
- Lots of untrusted, unaudited 3rd party addons, extensions etc.

# Firefox Rant

Rant: Firefox (2017)

Good:

- Full ASLR
- (Except on OSX for 3 years… and nobody noticed)

Bad:

- No Sandbox (yet) -> “will be released soon” (since 3 years)
- No 64 bit (yet) -> 64 bit exists, but default is 32 bit
- No process-per-tab (yet) -> “will be released soon”
- No (professional) source code auditing / SDL
- No (professional) fuzzing -> More fuzzing is being done.
- Lots of untrusted, unaudited 3rd party addons, extensions etc.

But: The Firefox rendering engine (Gecko) will be replaced by Servo, written in Rust!

# Firefox Rant

Rant: Firefox (2019)

Good:

- Full ASLR
- (Except on OSX for 3 years… and nobody noticed)

Bad:

- No Sandbox (yet) -> **is there?**
- No 64 bit (yet) -> **64 bit default**
- No process-per-tab (yet) -> “will be released soon”
- No (professional) source code auditing / SDL
- No (professional) fuzzing -> More fuzzing is being done.
- Lots of untrusted, unaudited 3rd party addons, extensions etc.

But: The Firefox rendering engine (Gecko) will be replaced by Servo, written in Rust!

# Firefox Rant

Rant: Firefox (2022)

Good:

- Full ASLR
- Sandbox is there
- 64 bit default
- Process-per-tab

Bad ?:

- No (professional) source code auditing / SDL
- No (professional) fuzzing -> More fuzzing is being done.
- Lots of untrusted, unaudited 3rd party addons, extensions etc.
- Anti-ROP (CFI, Shadow Stack)?

# “Secure” Browser

The history of “secure browsers”

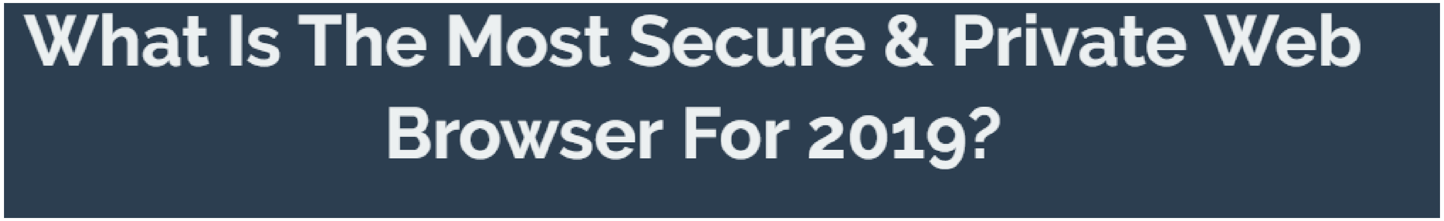
- Waterfox, brave, iridium, pale moon, epic, avg secure browser…
- Some “secure browsers” completely disabled Same-origin-policy, ASLR, DEP etc.
- Making them possibly the most **insecure** browsers

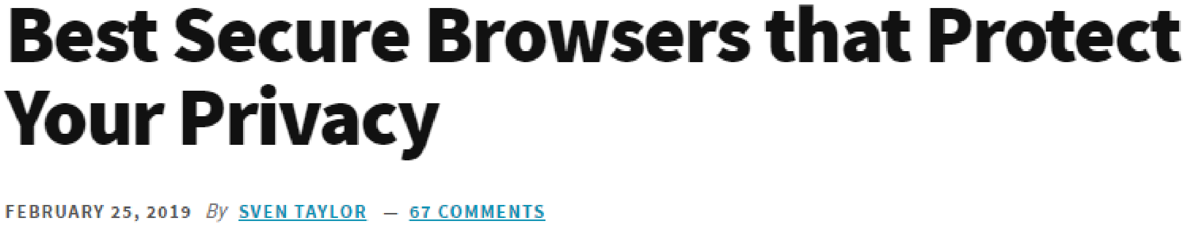
My professional opinion:

- Most secure: Chrome, Edge
- Close: IE11 (?)

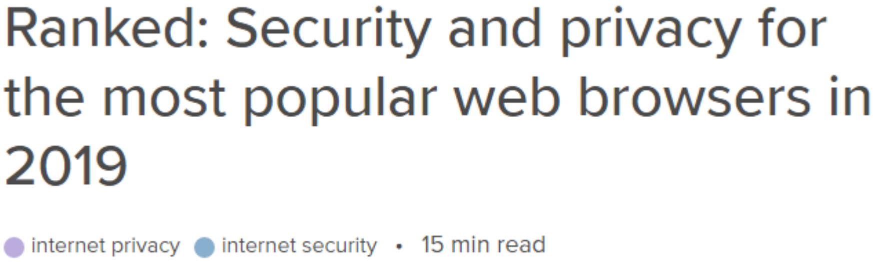
- Don’t use: Firefox (sorry), or any other browsers (Safari, IE8/9)
- Really don’t use: Torbrowser
  - Based on Firefox ESR (Long term support)
  - Every Torbrowser version therefore contains dozens, if not hundreds of publicly known exploits
  - Monocolture…

# Firefox Rant


What Is The Most Secure & Private Web Browser For 2019?


Best Secure Browsers that Protect Your Privacy

FEBRUARY 25, 2019 *By* SVEN TAYLOR — 67 COMMENTS


Ranked: Security and privacy for the most popular web browsers in 2019

● internet privacy ● internet security • 15 min read

**-> Privacy. Is. Not. Security. <-**

# Firefox Rant

2014: George Hotz (geohot, wrote first PlayStation 3 and iOS/iPhone Exploits) wrote the first Chromebook Exploit for pwnium. And:

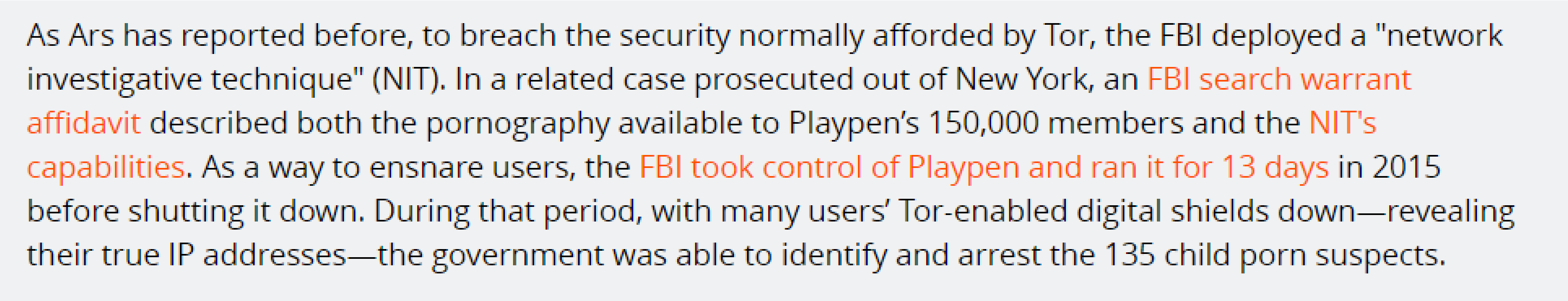
“Before pwnium, I had a few days extra, so I figured, why not try Firefox. Firefox, at least ca 2013, was about on par with a hard CTF problem. It took my 24 hours. 24 hours, full 0-day in Firefox.
A lot of people use this browser. Don’t use it. Use Chrome.”

USENIX Enigma 2016 - Timeless Debugging

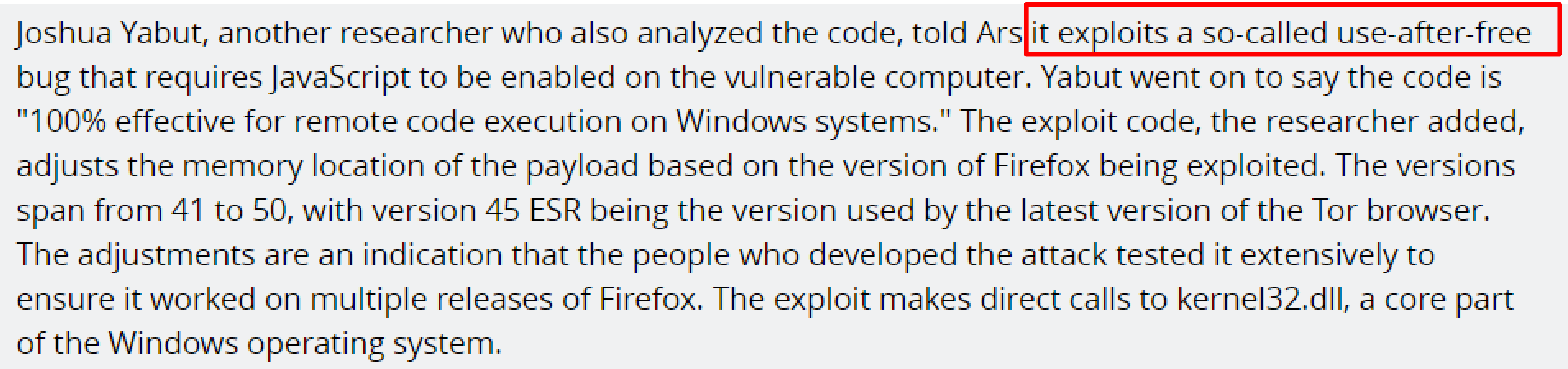
- https://youtu.be/eGl6kpSajag?t=178

# Firefox Rant

Even the FBI has Firefox Exploits…

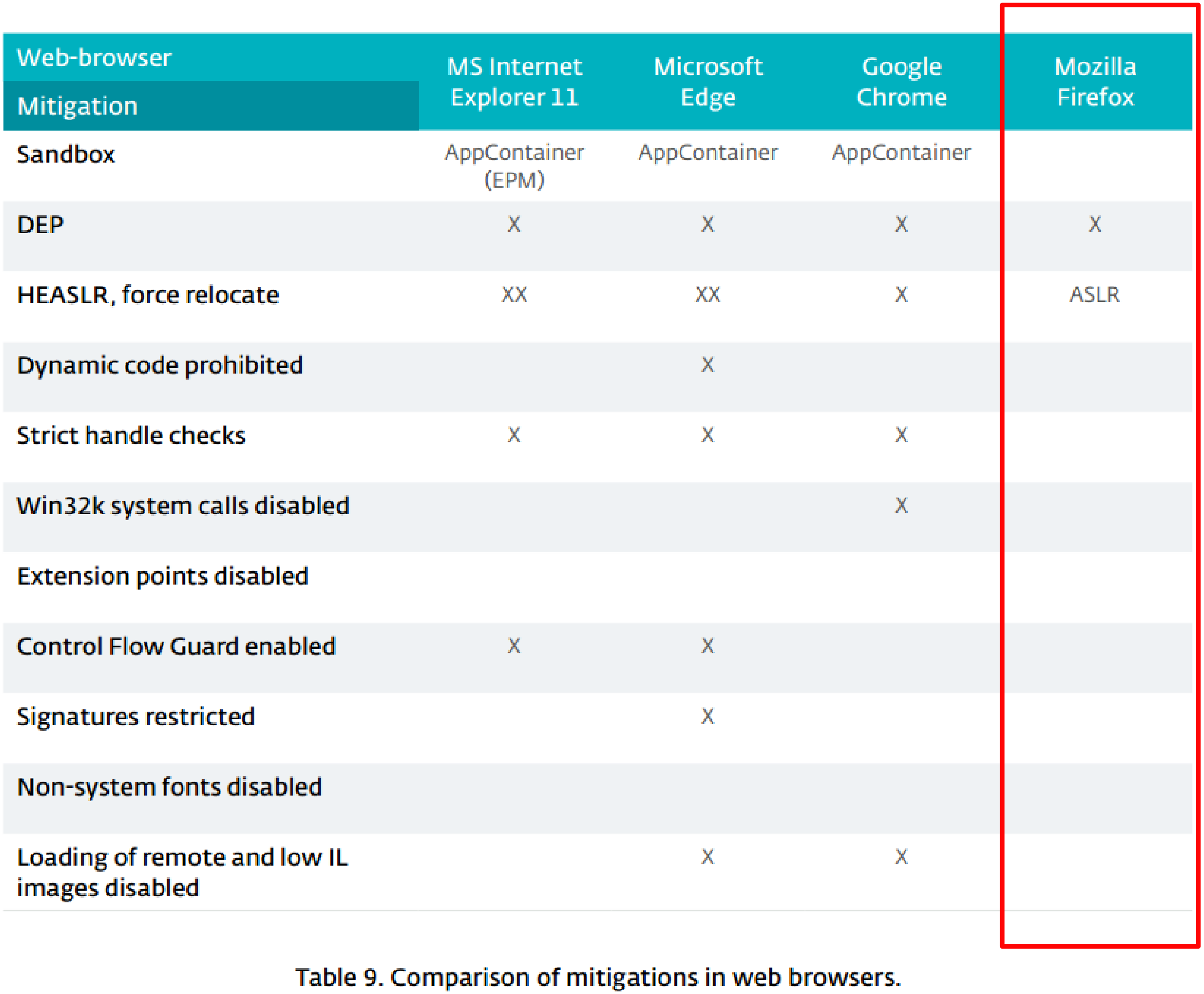


As Ars has reported before, to breach the security normally afforded by Tor, the FBI deployed a "network investigative technique" (NIT). In a related case prosecuted out of New York, an FBI search warrant affidavit described both the pornography available to Playpen's 150,000 members and the NIT's capabilities. As a way to ensnare users, the FBI took control of Playpen and ran it for 13 days in 2015 before shutting it down. During that period, with many users' Tor-enabled digital shields down—revealing their true IP addresses—the government was able to identify and arrest the 135 child porn suspects.



Joshua Yabut, another researcher who also analyzed the code, told Ars it exploits a so-called use-after-free bug that requires JavaScript to be enabled on the vulnerable computer. Yabut went on to say the code is "100% effective for remote code execution on Windows systems." The exploit code, the researcher added, adjusts the memory location of the payload based on the version of Firefox being exploited. The versions span from 41 to 50, with version 45 ESR being the version used by the latest version of the Tor browser. The adjustments are an indication that the people who developed the attack tested it extensively to ensure it worked on multiple releases of Firefox. The exploit makes direct calls to kernel32.dll, a core part of the Windows operating system.

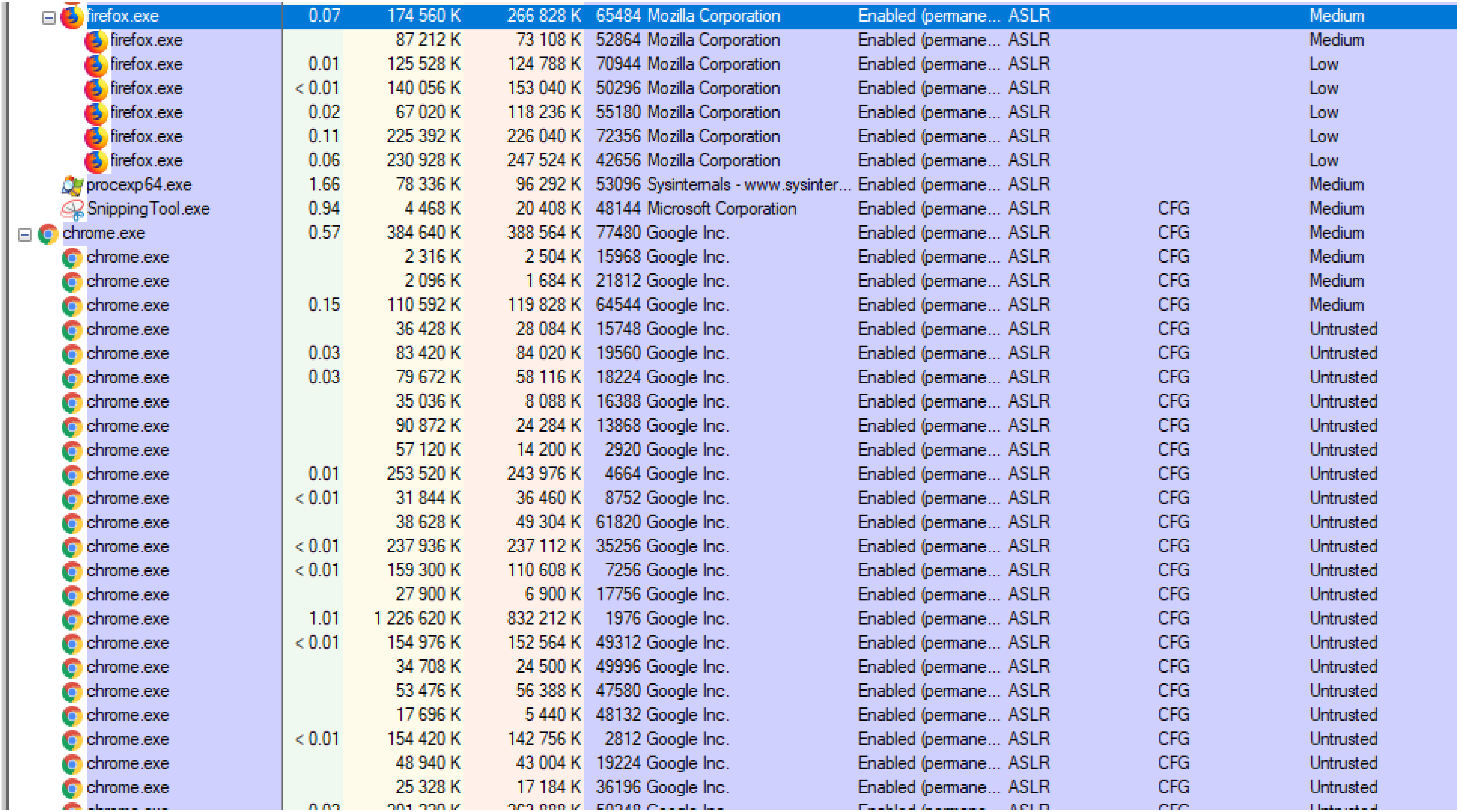
# Firefox Rant



| Web-browser<br>Mitigation | MS Internet Explorer 11 | Microsoft Edge | Google Chrome | Mozilla Firefox |
|---|---|---|---|---|
| Sandbox | AppContainer (EPM) | AppContainer | AppContainer | |
| DEP | X | X | X | X |
| HEASLR, force relocate | XX | XX | X | ASLR |
| Dynamic code prohibited | | X | | |
| Strict handle checks | X | X | X | |
| Win32k system calls disabled | | | X | |
| Extension points disabled | | | | |
| Control Flow Guard enabled | X | X | | |
| Signatures restricted | | X | | |
| Non-system fonts disabled | | | | |
| Loading of remote and low IL images disabled | | X | X | |

Table 9. Comparison of mitigations in web browsers.

# Per Origin Tab or not? (2019)

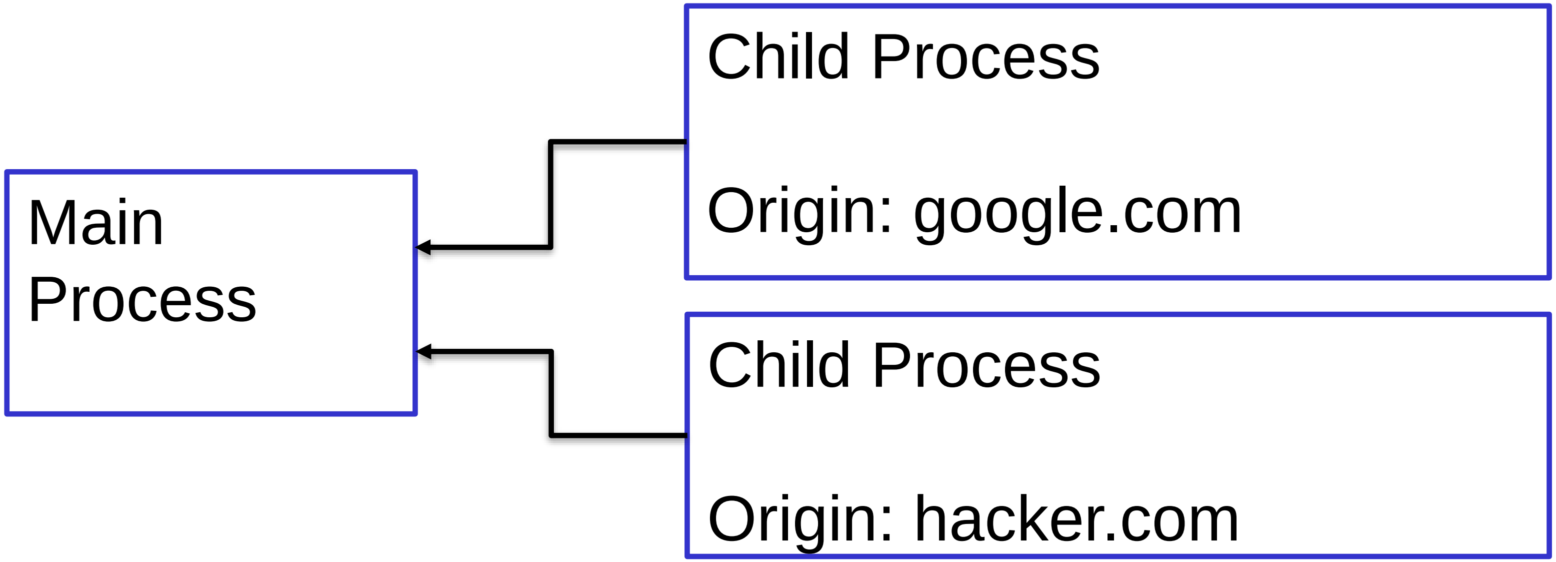


| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ⊟ firefox.exe | 0.07 | 174 560 K | 266 828 K | 65484 | Mozilla Corporation | Enabled (permane... | ASLR | | Medium |
| firefox.exe | | 87 212 K | 73 108 K | 52864 | Mozilla Corporation | Enabled (permane... | ASLR | | Medium |
| firefox.exe | 0.01 | 125 528 K | 124 788 K | 70944 | Mozilla Corporation | Enabled (permane... | ASLR | | Low |
| firefox.exe | < 0.01 | 140 056 K | 153 040 K | 50296 | Mozilla Corporation | Enabled (permane... | ASLR | | Low |
| firefox.exe | 0.02 | 67 020 K | 118 236 K | 55180 | Mozilla Corporation | Enabled (permane... | ASLR | | Low |
| firefox.exe | 0.11 | 225 392 K | 226 040 K | 72356 | Mozilla Corporation | Enabled (permane... | ASLR | | Low |
| firefox.exe | 0.06 | 230 928 K | 247 524 K | 42656 | Mozilla Corporation | Enabled (permane... | ASLR | | Low |
| procexp64.exe | 1.66 | 78 336 K | 96 292 K | 53096 | Sysinternals - www.sysinter... | Enabled (permane... | ASLR | | Medium |
| SnippingTool.exe | 0.94 | 4 468 K | 20 408 K | 48144 | Microsoft Corporation | Enabled (permane... | ASLR | CFG | Medium |
| ⊟ chrome.exe | 0.57 | 384 640 K | 388 564 K | 77480 | Google Inc. | Enabled (permane... | ASLR | CFG | Medium |
| chrome.exe | | 2 316 K | 2 504 K | 15968 | Google Inc. | Enabled (permane... | ASLR | CFG | Medium |
| chrome.exe | | 2 096 K | 1 684 K | 21812 | Google Inc. | Enabled (permane... | ASLR | CFG | Medium |
| chrome.exe | 0.15 | 110 592 K | 119 828 K | 64544 | Google Inc. | Enabled (permane... | ASLR | CFG | Medium |
| chrome.exe | | 36 428 K | 28 084 K | 15748 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | 0.03 | 83 420 K | 84 020 K | 19560 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | 0.03 | 79 672 K | 58 116 K | 18224 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 35 036 K | 8 088 K | 16388 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 90 872 K | 24 284 K | 13868 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 57 120 K | 14 200 K | 2920 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | 0.01 | 253 520 K | 243 976 K | 4664 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | < 0.01 | 31 844 K | 36 460 K | 8752 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 38 628 K | 49 304 K | 61820 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | < 0.01 | 237 936 K | 237 112 K | 35256 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | < 0.01 | 159 300 K | 110 608 K | 7256 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 27 900 K | 6 900 K | 17756 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | 1.01 | 1 226 620 K | 832 212 K | 1976 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | < 0.01 | 154 976 K | 152 564 K | 49312 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 34 708 K | 24 500 K | 49996 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 53 476 K | 56 388 K | 47580 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 17 696 K | 5 440 K | 48132 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | < 0.01 | 154 420 K | 142 756 K | 2812 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 48 940 K | 43 004 K | 19224 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | | 25 328 K | 17 184 K | 36196 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |
| chrome.exe | 0.02 | 301 320 K | 362 888 K | 50348 | Google Inc. | Enabled (permane... | ASLR | CFG | Untrusted |

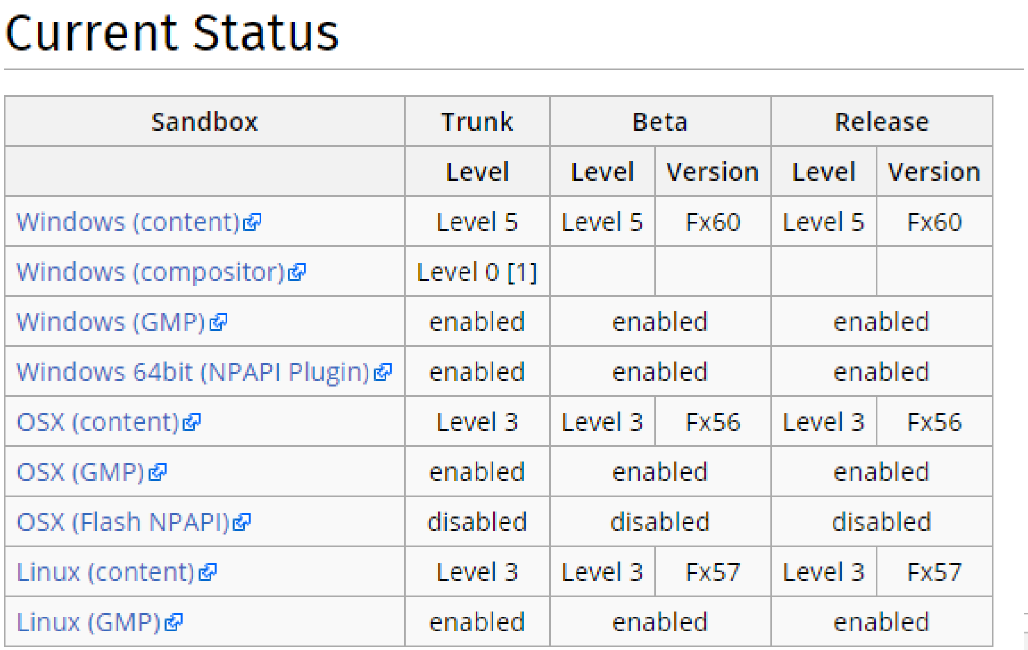
# Sandbox

Browser Sandbox?

- Isolate "dangerous" code in a separate process, low integrity level
  - Sandbox2, gvisor, nsjail, ...
- Communicate with Main Parent Process (Network, FS, Graphics, ...)
- Child code cannot access filesystem, create processes
  - RCE doesnt give access to anything important

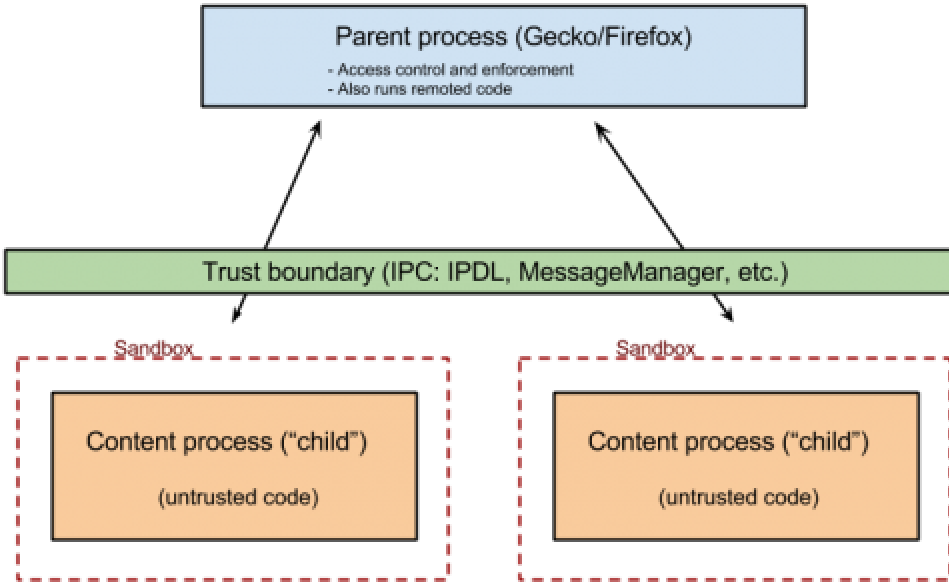


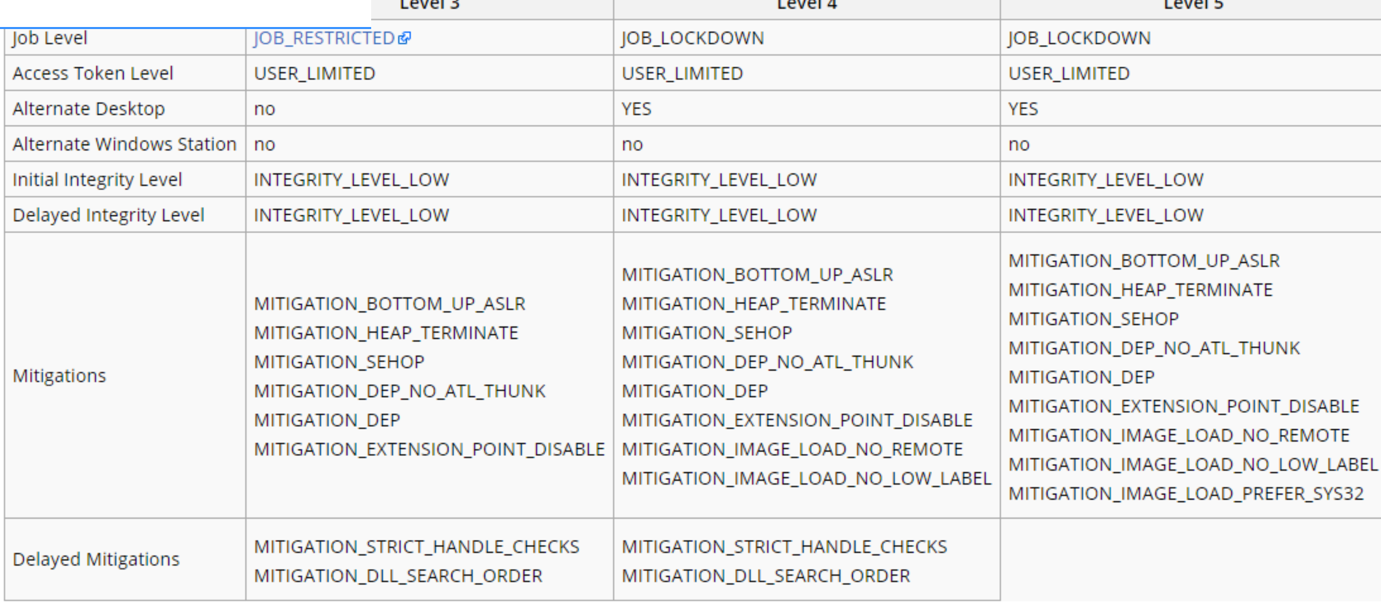
# Sandbox?



## Current Status

| Sandbox | Trunk | Beta | | Release | |
|---|---|---|---|---|---|
| | Level | Level | Version | Level | Version |
| Windows (content) | Level 5 | Level 5 | Fx60 | Level 5 | Fx60 |
| Windows (compositor) | Level 0 [1] | | | | |
| Windows (GMP) | enabled | enabled | | enabled | |
| Windows 64bit (NPAPI Plugin) | enabled | enabled | | enabled | |
| OSX (content) | Level 3 | Level 3 | Fx56 | Level 3 | Fx56 |
| OSX (GMP) | enabled | enabled | | enabled | |
| OSX (Flash NPAPI) | disabled | disabled | | disabled | |
| Linux (content) | Level 3 | Level 3 | Fx57 | Level 3 | Fx57 |
| Linux (GMP) | enabled | enabled | | enabled | |

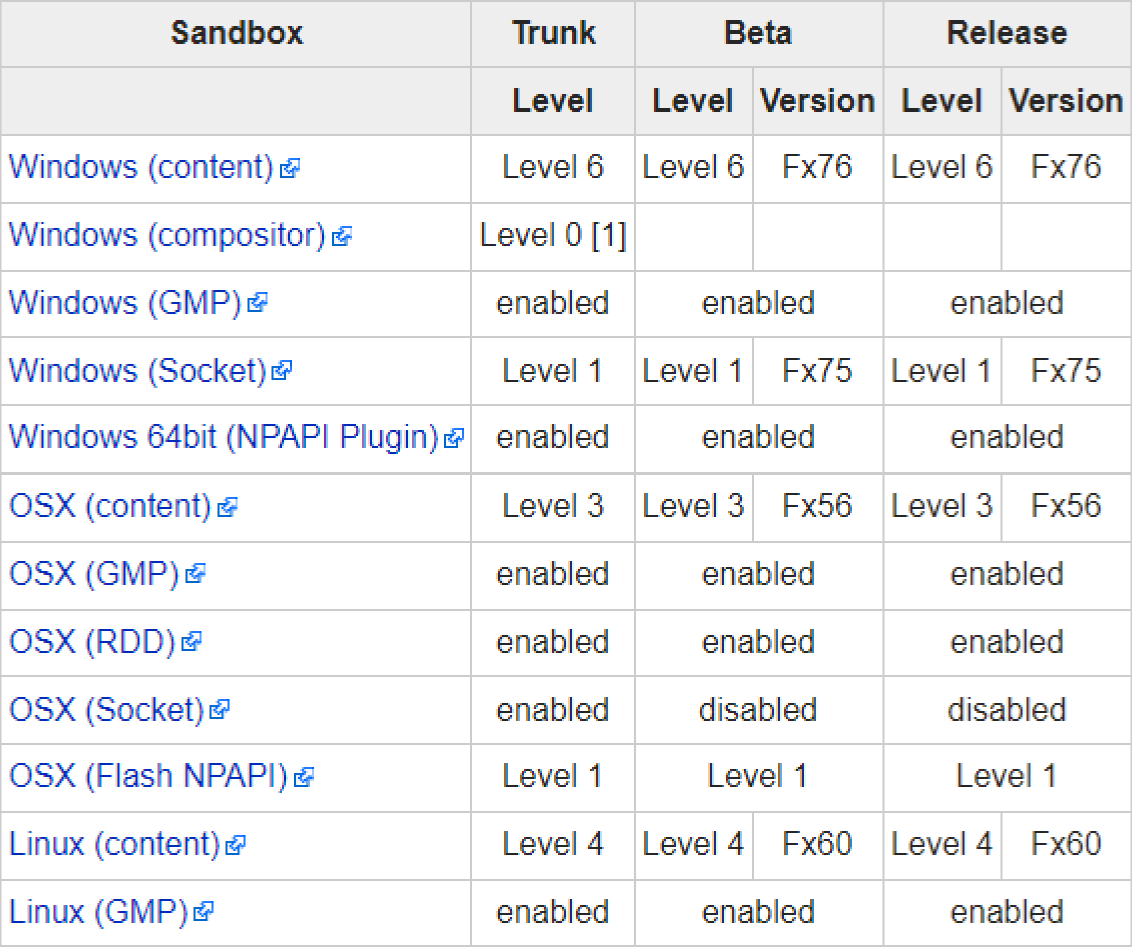

Parent process (Gecko/Firefox)
- Access control and enforcement
- Also runs remoted code
Trust boundary (IPC: IPDL, MessageManager, etc.)
Sandbox
Content process ("child")
(untrusted code)
Sandbox
Content process ("child")
(untrusted code)



| | Level 3 | Level 4 | Level 5 |
|---|---|---|---|
| Job Level | JOB_RESTRICTED | JOB_LOCKDOWN | JOB_LOCKDOWN |
| Access Token Level | USER_LIMITED | USER_LIMITED | USER_LIMITED |
| Alternate Desktop | no | YES | YES |
| Alternate Windows Station | no | no | no |
| Initial Integrity Level | INTEGRITY_LEVEL_LOW | INTEGRITY_LEVEL_LOW | INTEGRITY_LEVEL_LOW |
| Delayed Integrity Level | INTEGRITY_LEVEL_LOW | INTEGRITY_LEVEL_LOW | INTEGRITY_LEVEL_LOW |
| Mitigations | MITIGATION_BOTTOM_UP_ASLR<br>MITIGATION_HEAP_TERMINATE<br>MITIGATION_SEHOP<br>MITIGATION_DEP_NO_ATL_THUNK<br>MITIGATION_DEP<br>MITIGATION_EXTENSION_POINT_DISABLE | MITIGATION_BOTTOM_UP_ASLR<br>MITIGATION_HEAP_TERMINATE<br>MITIGATION_SEHOP<br>MITIGATION_DEP_NO_ATL_THUNK<br>MITIGATION_DEP<br>MITIGATION_EXTENSION_POINT_DISABLE<br>MITIGATION_IMAGE_LOAD_NO_REMOTE<br>MITIGATION_IMAGE_LOAD_NO_LOW_LABEL | MITIGATION_BOTTOM_UP_ASLR<br>MITIGATION_HEAP_TERMINATE<br>MITIGATION_SEHOP<br>MITIGATION_DEP_NO_ATL_THUNK<br>MITIGATION_DEP<br>MITIGATION_EXTENSION_POINT_DISABLE<br>MITIGATION_IMAGE_LOAD_NO_REMOTE<br>MITIGATION_IMAGE_LOAD_NO_LOW_LABEL<br>MITIGATION_IMAGE_LOAD_PREFER_SYS32 |
| Delayed Mitigations | MITIGATION_STRICT_HANDLE_CHECKS<br>MITIGATION_DLL_SEARCH_ORDER | MITIGATION_STRICT_HANDLE_CHECKS<br>MITIGATION_DLL_SEARCH_ORDER | |

# Sandbox?

https://wiki.mozilla.org/Security/Sandbox



| Sandbox | Trunk | Beta | | Release | |
|---|---|---|---|---|---|
| | Level | Level | Version | Level | Version |
| Windows (content) | Level 6 | Level 6 | Fx76 | Level 6 | Fx76 |
| Windows (compositor) | Level 0 [1] | | | | |
| Windows (GMP) | enabled | enabled | | enabled | |
| Windows (Socket) | Level 1 | Level 1 | Fx75 | Level 1 | Fx75 |
| Windows 64bit (NPAPI Plugin) | enabled | enabled | | enabled | |
| OSX (content) | Level 3 | Level 3 | Fx56 | Level 3 | Fx56 |
| OSX (GMP) | enabled | enabled | | enabled | |
| OSX (RDD) | enabled | enabled | | enabled | |
| OSX (Socket) | enabled | disabled | | disabled | |
| OSX (Flash NPAPI) | Level 1 | Level 1 | | Level 1 | |
| Linux (content) | Level 4 | Level 4 | Fx60 | Level 4 | Fx60 |
| Linux (GMP) | enabled | enabled | | enabled | |

# Chrome / Edge

# Chrome, Edge

- CFG: Control flow guard
- ACG: Code cannot be dynamically generated or modified
- CIG: only allow properly signed images to load
- No-child
- Arbitrary code guard: no X pages
- EAF: Export Address Filtering
- IAF: Import Address Filtering
- Force randomization (for ASLR)

https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/exploit-protection-reference
https://blogs.windows.com/msedgedev/2017/02/23/mitigating-arbitrary-native-code-execution/

# References

Look Mom, I don’t use Shellcode

- Browser Exploitation Case Study for IE11
- Moritz Jodeit
- EKO12 (Ekoparty Security Conference)
- https://www.youtube.com/watch?v=PbIpd89efX8&index=14&list=PLdgOScViw-omMZQymL2SWKh5BLfMhDijB