

Mozilla: Firefox 71 Is Now Available, TenFourFox FPR17 Also Available, Firefox Turns 15 and More

By *Roy Schestowitz*

Created *03/12/2019 - 1:28am*

Submitted by Roy Schestowitz on Tuesday 3rd of December 2019 01:28:04 AM Filed under [Moz/FF](#) [1]

•

[Mozilla Firefox 71 Is Now Available to Download for Linux, Windows, and macOS](#) [2]

The upcoming Firefox 71 web browser is now available to download for all supported platforms, including Linux, Windows, and macOS, ahead of tomorrow's official launch. Firefox 71 has entered development in late October and it promises to introduce a new "--kiosk" command-line parameter that opens the web browser in full-screen mode (a.k.a. kiosk mode), a redesigned about:config internal configuration page, as well as Picture-in-Picture (PiP) support on Windows.

"Windows users now have the ability to pop out videos on the web into an always-on-top video player using the Picture-in-Picture feature! For most videos, this can be accomplished by hovering the video with the mouse, and clicking on the Picture-in-Picture toggle," explained Mozilla in the preliminary release notes.

•

[TenFourFox FPR17 available](#) [3]

TenFourFox Feature Parity Release 17 final is now available for testing (downloads, hashes, release notes). Apologies for the delay, but I was visiting family and didn't return until a few hours ago so I could validate and perform the confidence testing on the builds. There are no other changes in this release other than a minor tweak to the ATSUI font blacklist and outstanding security patches. Assuming all is well, it will go live tomorrow evening Pacific time.

The FPR18 cycle is the first of the 4-week Mozilla development cycles. It isn't feasible for me to run multiple branches, so we'll see how much time this actually gives me for new work. As previously mentioned, FPR18 will be primarily about parity updates to Reader mode, which helps to shore up the browser's layout deficiencies and is faster to render as well. There will also be some other minor miscellaneous fixes.

- [\[Older\] Firefox at 15: its rise, fall, and privacy-first renaissance \[4\]](#)

There's a good chance you are reading this in Google's Chrome web browser, which commands 65% of the global market (and about 50% in the U.S.), according to Statcounter. Only about 4% to 5% of web surfers now go online through Firefox, the open-source browser from the California-based Mozilla foundation. But the web was much different when Firefox launched 15 years ago on November 9, 2004, and the browser began a fast rise to prominence.

When Firefox hit the scene, Internet Explorer had more than 90% market share, having felled Netscape Navigator. Given that it was the default browser on Windows, which commanded a similar share of the operating system market, its monopoly seemed like it could be permanent. But Firefox quickly caught on, and eventually grew to command about a third of the market at its height in 2009. While it's unlikely to recapture such former glory, Firefox has been experiencing something of a renaissance, not just by improving speed and features, but by putting user control over privacy front and center.

Fifteen years on, it's hard to imagine how radical Firefox was at the time of its debut. Instead of coming from a megacorporation like Microsoft (or today, Google), Firefox was built by volunteers around the world who gave their code away for free. "Open source was well known for developers," says Mitchell Baker, who cofounded the Mozilla Project back in 1998 and is today the chairwoman of the Mozilla Corporation and Mozilla Foundation. "But the common wisdom of the time was that open source was only for the geeks. You could build [tools] for developers but not consumer products out of it."

- [Help Test Firefox's built-in HTML Sanitizer to protect against UXSS bugs \[5\]](#)

I recently gave a talk at OWASP Global AppSec in Amsterdam and summarized the presentation in a blog post about how to achieve "critical"-rated code execution vulnerabilities in Firefox with user-interface XSS. The end of that blog post encourages the reader to participate the bug bounty program, but did not come with proper instructions. This blog post will describe the mitigations Firefox has in place to protect against XSS bugs and how to test them.

Our about: pages are privileged pages that control the browser (e.g., about:preferences, which contains Firefox settings). A successful XSS exploit has to bypass the Content Security Policy (CSP), which we have recently added but also our built-in XSS sanitizer to gain arbitrary code

execution. A bypass of the sanitizer without a CSP bypass is in itself a severe-enough security bug and warrants a bounty, subject to the discretion of the Bounty Committee. See the bounty pages for more information, including how to submit findings.

- [Botond Ballo: Developing Mozilla C++ code with clang and VSCode](#) [6]

I've long been a fan of smart editors which have a semantic understanding of the code you're editing, and leverage it to provide semantics-aware features such as accurate code completion (only offering completions for names that are actually in scope), go-to-definition, find references, semantic highlighting, and others.

When I joined Mozilla six years ago, my choice of editor for C++ code was Eclipse CDT, because based on experience and research, this was the most fully-featured option that was cross-platform and open-source. (Depending on who you ask, Visual Studio, XCode, and CLion have, at various times, been described as matching or exceeding Eclipse CDT in terms of editor capabilities, but the first two of these are single-platform tools, and are three all proprietary.)

This assessment was probably accurate at that time, and probably even for much of the intervening time, but in recent years Eclipse CDT has not aged well. The main reason for this is that Eclipse CDT has its own C++ parser. (For brevity, I'm using "parsing" here as an umbrella term for lexing, preprocessing, parsing, semantic analysis, and all other tasks that need to be performed to build a semantic model of code from source.) C++ is a very complex language to parse, and thus a C++ parser requires a lot of effort to write and maintain. In the early days of CDT, there was a lot of investment, mostly from commercial vendors that packaged CDT-based IDEs, in building and maintaining CDT's parser, but over time, the level of investment has faded. Meanwhile, the C++ language has been gaining new features at an increasing rate (and the Mozilla codebase adopting them - we're on the verge of switching to C++17), and CDT's parser just hasn't been able to keep up.

[Moz/FF](#)

Source URL: <http://www.tuxmachines.org/node/131207>

Links:

[1] <http://www.tuxmachines.org/taxonomy/term/118>

[2] <https://news.softpedia.com/news/mozilla-firefox-71-is-now-available-to-download-for-linux-windows-and-macos-528418.shtml>

[3] <http://tenfourfox.blogspot.com/2019/12/tenfourfox-fpr17-available.html>

[4] <https://www.fastcompany.com/90428050/firefox-at-15-its-rise-fall-and-privacy-first-renaissance>

[5] <https://blog.mozilla.org/security/2019/12/02/help-test-firefoxs-built-in-html-sanitizer-to-protect-against-uxss-bugs/>

[6] <https://botondballo.wordpress.com/2019/12/02/developing-mozilla-c-code-with-clang-and-vscode/>