Mozilla Firefox: Firefox 68, Charsets and Grizzly Browser Fuzzing Framework

By *Roy Schestowitz*
Created *11/07/2019 - 11:30am*
Submitted by Roy Schestowitz on Thursday 11th of July 2019 11:30:03 AM Filed under Moz/FF [1]

- **Mike Hommey: Reproducing the Linux builds of Firefox 68 [2]**

  Starting with Firefox 68, the Linux builds shipped by Mozilla should be reproducible (it is not currently automatically validated that it definitely is, but 68.0 is). These builds are optimized with Profile Guided Optimization, and the profile data was not kept and published until recently, which is why they weren?t reproducible until now.

  The following instructions require running Docker on a Linux host (this may or may not work on a non-Linux host, I don?t know what e.g. Docker for Mac does, and if the docker support in the mach command works with it). I?ll try to make them generic enough that they may apply to any subsequent release of Firefox.

- **Mozilla Releases Firefox 68 as the Next ESR Series with Cryptomining Protection [3]**

  Mozilla officially released today the Firefox 68 web browser for all supported platforms, including Linux, Mac, and Windows, making it an ESR (Extended Support Release) version. The popular open-source and cross-platform Firefox web browser from Mozilla has been updated to version 68.0, a major release that expands the dark mode in the reader view to make the controls, toolbars, and sidebars on windows dark too. Additionally, Firefox 68 introduces new cryptomining and fingerprinting protections to strict content blocking settings.

  Firefox 68 also improves add-on security and discovery by introducing a Recommended Extensions program in about:addons to help users easily find high quality and secure add-ons and themes, a new reporting feature in about:addons to let users quickly report security and

performance issues with add-ons, and revamp the extensions dashboard in about:addons.

- **Dave Townsend: Please watch your character encodings [4]**

  I started writing this as a newsgroup post for one of Mozilla?s mailing lists, but it turned out to be too long and since this part was mainly aimed at folks who either didn?t know about or wanted a quick refresher on character encodings I decided to blog it instead. Please let me know if there are errors in here, I am by no means an expert on this stuff either and I do get caught out sometimes!

  Text is tricky. Unicode supports the notion of 1,114,112 distinct characters, slightly more than a byte of memory can hold. So to store a character we have to use a way of encoding its value into bytes in memory. A straightforward encoding would just use three bytes per character. But (roughly) the larger the character value the less often it is used, and memory is precious, so often variable length encodings are used. These will use fewer bytes in memory for characters earlier in the range at the cost of using a little more memory for the rarer characters. Common encodings include UTF-8 (one byte for ASCII characters, up to four bytes for other characters) and UTF-16 (two bytes for most characters, four bytes for less used ones).

  What does this mean?

- **Grizzly Browser Fuzzing Framework [5]**

  At Mozilla, we rely heavily on automation to increase our ability to fuzz Firefox and the components from which it is built. Our fuzzing team is constantly developing tools to help integrate new and existing capabilities into our workflow with a heavy emphasis on scaling. Today we would like to share Grizzly ? a browser fuzzing framework that has enabled us to quickly and effectively deploy fuzzers at scale.

  Grizzly was designed to allow fuzzer developers to focus solely on writing fuzzers and not worry about the overhead of creating tools and scripts to run them. It was created as a platform for our team to run internal and external fuzzers in a common way using shared tools. It is cross-platform and supports running multiple instances in parallel.

Moz/FF

Source URL: http://www.tuxmachines.org/node/125771

Links:
[1] http://www.tuxmachines.org/taxonomy/term/118
[2] https://glandium.org/blog/?p=3923
[3] https://news.softpedia.com/news/mozilla-releases-firefox-68-as-the-next-esr-series-with-cryptomining-protection-

526670.shtml
[4] https://www.oxymoronical.com/blog/2019/07/Please-watch-your-character-encodings
[5] https://blog.mozilla.org/security/2019/07/10/grizzly/