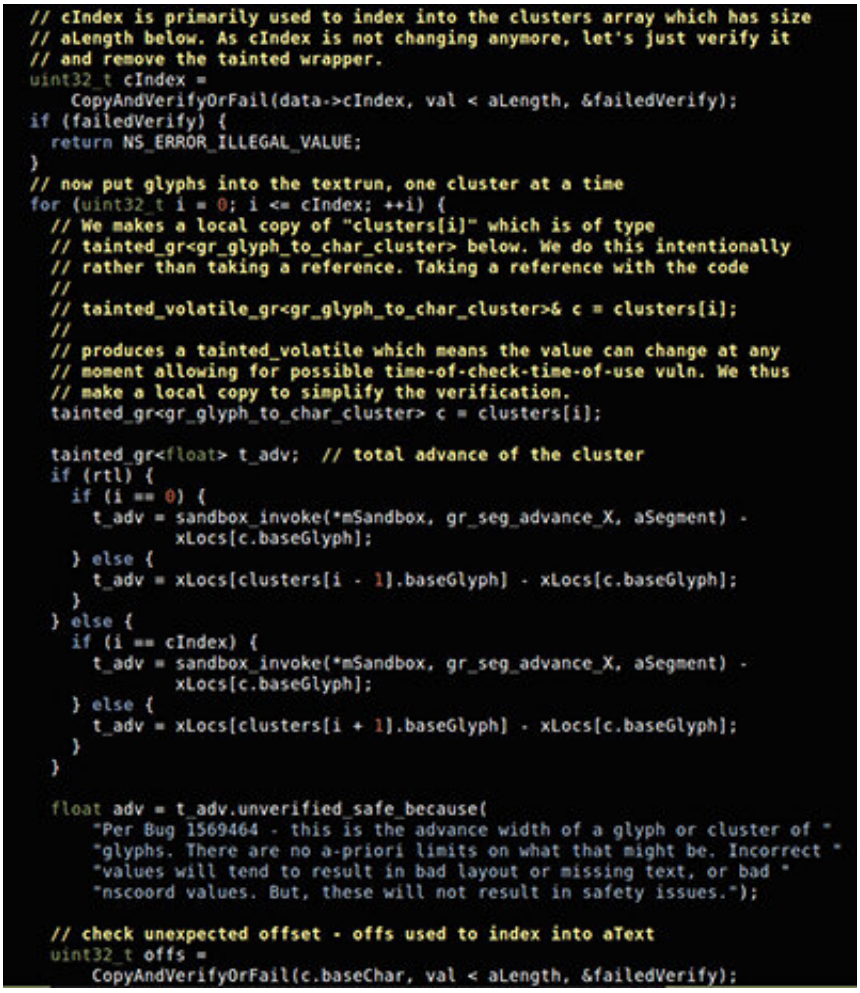
# Researchers develop framework that improves Firefox security

February 25 2020, by Joshua Baxt



```
// cIndex is primarily used to index into the clusters array which has size
// aLength below. As cIndex is not changing anymore, let's just verify it
// and remove the tainted wrapper.
uint32_t cIndex =
    CopyAndVerifyOrFail(data->cIndex, val < aLength, &failedVerify);
if (failedVerify) {
  return NS_ERROR_ILLEGAL_VALUE;
}
// now put glyphs into the textrun, one cluster at a time
for (uint32_t i = 0; i <= cIndex; ++i) {
  // We makes a local copy of "clusters[i]" which is of type
  // tainted_gr<gr_glyph_to_char_cluster> below. We do this intentionally
  // rather than taking a reference. Taking a reference with the code
  //
  // tainted_volatile_gr<gr_glyph_to_char_cluster>& c = clusters[i];
  //
  // produces a tainted_volatile which means the value can change at any
  // moment allowing for possible time-of-check-time-of-use vuln. We thus
  // make a local copy to simplify the verification.
  tainted_gr<gr_glyph_to_char_cluster> c = clusters[i];

  tainted_gr<float> t_adv;  // total advance of the cluster
  if (rtl) {
    if (i == 0) {
      t_adv = sandbox_invoke(*mSandbox, gr_seg_advance_X, aSegment) -
              xLocs[c.baseGlyph];
    } else {
      t_adv = xLocs[clusters[i - 1].baseGlyph] - xLocs[c.baseGlyph];
    }
  } else {
    if (i == cIndex) {
      t_adv = sandbox_invoke(*mSandbox, gr_seg_advance_X, aSegment) -
              xLocs[c.baseGlyph];
    } else {
      t_adv = xLocs[clusters[i + 1].baseGlyph] - xLocs[c.baseGlyph];
    }
  }

  float adv = t_adv.unverified_safe_because(
      "Per Bug 1569464 - this is the advance width of a glyph or cluster of "
      "glyphs. There are no a-priori limits on what that might be. Incorrect "
      "values will tend to result in bad layout or missing text, or bad "
      "nscoord values. But, these will not result in safety issues.");

  // check unexpected offset - offs used to index into aText
  uint32_t offs =
      CopyAndVerifyOrFail(c.baseChar, val < aLength, &failedVerify);
```

A collaborative team of computer scientists have developed a framework to improve the security of Mozilla Firefox, a popular web browser. The framework, called RLBox, will ultimately make millions of users' browsers more secure. Credit: University of California - San Diego

Researchers from the University of California San Diego, University of Texas at Austin, Stanford University and Mozilla have developed a new framework to improve web browser security. The framework, called RLBox, has been integrated into Firefox to complement Firefox's other security-hardening efforts.

RLBox increases browser security by separating third-party libraries that are vulnerable to attacks from the rest of the browser to contain potential damage—a practice called sandboxing. The study will be published in the proceedings of the USENIX Security Symposium.

Browsers, like Firefox, rely on third-party libraries to support media decoding (e.g., rendering images or playing audio files) among many other functionalities. These libraries are often written in low-level programming languages, like C, and highly optimized for performance.

"Unfortunately, bugs in C code are often security vulnerabilities—security vulnerabilities that attackers are really good at exploiting," noted senior author Deian Stefan, an assistant professor with UC San Diego's Department of Computer Science and Engineering.

RLBox allows browsers to continue to use off-the-shelf, highly tuned libraries without worrying about the security impact of these libraries. "By isolating libraries we can ensure that attackers can't exploit bugs in these libraries to compromise the rest of the browser," said the lead Ph.D. student on the project, Shravan Narayan.

A key piece of RLBox is the underlying sandboxing mechanism, which keeps a buggy library from interfering with the rest of the browser. The study investigates various sandboxing techniques with different trade-offs. But the team ultimately partnered with the engineering team at San Francisco-based Fastly to adopt a sandboxing technique based on WebAssembly, a new intermediate language designed with sandboxing

in mind. The team believes that WebAssembly will be a key part of future secure browsers and secure systems more broadly. The WebAssembly sandboxing effort is detailed in a recent Mozilla Hacks blog post.

"Unfortunately, it's not enough to put a library in a sandbox, you need to carefully check all the data that comes out of the sandbox—otherwise a sophisticated attacker can trick the browser into doing the wrong thing and render the sandboxing effort useless, " said Stefan. RLBox eliminates these classes of attacks by tagging everything that crosses the boundary and ensuring that all such tagged data is validated before it is used.

RLBox has been integrated into Mozilla's Firefox and will be shipping to Linux users in Firefox 74 and Mac users in Firefox 75, with plans to implement in other platforms.

"This is a big deal," says Bobby Holley, principal engineer at Mozilla. "Security is a top priority for us, and it's just too easy to make dangerous mistakes in C/C++. We're writing a lot of new code in Rust, but Firefox is a huge codebase with millions of lines of C/C++ that aren't going away any time soon. RLBox makes it quick and easy to isolate existing chunks of code at a granularity that hasn't been possible with the process-level sandboxing used in browsers today."

In the study, the team isolated half a dozen libraries using RLBox. To start, Firefox will ship with their sandboxed Graphite font shaping library. Mozilla plans to apply the sandboxing more broadly in the future, ultimately making millions of users' browsers more secure.

Provided by University of California - San Diego

Citation: Researchers develop framework that improves Firefox security (2020, February 25) retrieved 25 April 2024 from https://techxplore.com/news/2020-02-framework-firefox.html

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.