SECURITY TESTING AT MOZILLA

CHRISTIAN HOLLER, M.SC.
SENIOR SECURITY ENGINEER
MOZILLA CORPORATION
Firefox

Browse Freely

Free Download
Firefox Privacy

Fly Fast
Go anywhere you want on the Web – with a quickness.

Fly Safe
Take advantage of more built-in privacy tools than any other browser.

Fly Free
Use the only browser built for freedom, not profit.
Bushfreely

Free Download
Firefox Privacy

Fly Fast
Go anywhere you want on the Web – with a quickness.

Fly Safe
Take advantage of more built-in privacy tools than any other browser.

Fly Free
Use the only browser built for freedom, not profit.

303,221 commits made by 3,957 contributors
Browse Freely

Free Download
Firefox Privacy

Fly Fast
Go anywhere you want on the Web – with a quickness.

Fly Safe
Take advantage of more built-in privacy tools than any other browser.

Fly Free
Use the only browser built for freedom, not profit.

303,221 commits made by 3,957 contributors

approx. 13.5M LoC
## Mozilla Research Projects

<table>
<thead>
<tr>
<th>Classification</th>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Foundations</strong></td>
<td>Servo</td>
<td>Rethinking the browser at every level of the technology stack</td>
</tr>
<tr>
<td><strong>Rust</strong></td>
<td>sweet.js</td>
<td>Hygienic macros for JavaScript</td>
</tr>
<tr>
<td><strong>Emscripten</strong></td>
<td>Shumway</td>
<td>The Flash multimedia platform built on the Open Web</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>LLJS</td>
<td>Bridging native low-level code with idiomatic JavaScript</td>
</tr>
<tr>
<td><strong>asm.js</strong></td>
<td>ParallelJS</td>
<td>Unlocking the full potential of modern hardware</td>
</tr>
</tbody>
</table>

- **Servo**: A programming language for developing reliable and efficient systems.
- **Emscripten**: Running native C / C++ applications on the Open Web.
- **asm.js**: A formally specified, optimizable subset of the JavaScript language, targetable by code generators like Emscripten.
Mozilla Research Projects

-**Servo**
  *Rethinking the browser at every level of the technology stack*

-**Rust**
  *A programming language for developing reliable and efficient systems*

-**Emscripten**
  *Running native C / C++ applications on the Open Web*

-**Shumway**
  *The Flash multimedia platform built on the Open Web*

-**sweet.js**
  *Hygienic macros for JavaScript*

-**LJJS**
  *Bridging native low-level code with idiomatic JavaScript*

-**Parallel JS**
  *Unlocking the full potential of modern hardware*

---

Mozilla Research

Expanding the Foundations of the Open Web

-**The Enterpreter: Run code before it can be parsed**

  Alon Zakai

  1 response

---

I'm excited to announce a new Mozilla Research experiment: the Enterpreter, a pure-JavaScript interpreter that can start running large Emscripten-compiled apps faster than JavaScript engines can, giving developers control over the latency/throughput trade-off.

An app's startup time is a precious resource. For small apps, minification and image compression are good enough to provide a smooth user onboarding experience. But when a codebase gets large enough, the JavaScript engine startup costs—in particular, parsing—can add up to noticeable startup delays.

What can we do to improve JavaScript parse time? The obvious steps are removing unneeded code and minifying, but those only get you so far. We wanted to try a more extreme experiment: what if we compressed asm.js into a bytecode format and shipped it along with a small interpreter? Read on for some interesting results!

First, let's see how we can measure the problem:
All your devices. One Firefox.

Encryption helps protect your privacy

Firefox is now available for iOS


10,554 active Mozillians worldwide

45 upcoming events around the globe

87 languages and counting, on every continent
Get Smart On the Web

The SmartOn Series is like your Internet owner’s manual: the place to learn the most useful intel and tips from Mozilla policy peeps and programmers.

Tracking

1. Ask

Ask yourself

1. Do you feel in control of your personal information online?
2. Do you know what a cookie is (no, not the delicious kind)?
3. Have you ever wondered why some content, like ads, seems to follow you around?

What is tracking?

Every time we go online, we leave behind traces of our activity. They’re often called our “digital footprints” and for good reason, because tracking is sort of like re-tracing our steps. But our digital footprints reveal more about us than where we’ve been — everything from our preferences to our identities.
01 The Internet is an integral part of modern life—a key component in education, communication, collaboration, business, entertainment and society as a whole.

02 The Internet is a global public resource that must remain open and accessible.

03 The Internet may enrich the lives of individual human beings.

04 Individuals’ security and privacy on the Internet are fundamental and must not be treated as optional.

05 Individuals must have the ability to shape the Internet and their own experiences on it.

06 The effectiveness of the Internet as a public resource depends upon interoperability (protocols, data formats, content), innovation and decentralized participation worldwide.

07 Free and open source software promotes the development of the Internet as a public resource.

08 Transparent community-based processes promote participation, accountability and trust.

09 Commercial involvement in the development of the Internet brings many benefits; a balance between commercial profit and public benefit is critical.
Individuals’ security and privacy on the Internet are fundamental and must not be treated as optional.
PLATFORM SECURITY
PLATFORM SECURITY

Gecko Layout Engine

JS Engine
PLATFORM SECURITY

Platform

Gecko Layout Engine  JS Engine  ...  Codecs and 3rd party libraries
FUZZ TESTING
FUZZ TESTING

Program
FUZZ TESTING

00110001 00110011
00110011 00110111

Program
FUZZ TESTING

00110001 00110011
00110011 00110111

Program

?
FUZZ TESTING

Program

00110001 00110011
00110011 00110111

Crashes
Assertions
Hangs
FUZZ TESTING

Generator → Program

? → Crashes
  Assertions
  Hangs
FUZZ TESTING

Generator -> Random API usage -> Program -> ?

Crashes
Assertions
Hangs
FUZZ TESTING

Mutator → Program → Crashes, Assertions, Hangs
FUZZ TESTING

Mutator -> Program -> ?

Crashes
Assertions
Hangs
FUZZ TESTING

Mutator -> Program -> ?

Crashes Assertions Hangs

Sample Test
Sample Test
Sample Test
Sample Test

Mutated Test
Fuzzing is effective: Quick to get, cheap to use, finds tricky bugs.
Grammar-based mutations for interpreter testing
Fuzzing with Code Fragments

Christian Hellwig
Mozilla Corporation
chellwig@mozilla.com

Kim Heinz
University of California, Irvine
heinz@ics.uci.edu

Andrew Zeller
University of California, Irvine
zeller@ics.uci.edu

Abstract

Fuzzing is an automated technique providing random data as input to a software system in the hope to trigger a vulnerability. In order to be effective, the fuzzed input must be constructed such to pass elementary consistency checks: a JavaScript interpreter, for instance, would only accept a syntactically valid program. On the other hand, the fuzzed input must be overwhelming enough to trigger unusual behaviors, such as a stack overflow. The JavaScript engine, therefore, triggers security bugs through incomplete or poorly implemented code paths. In this paper, we present Code Fragments, a framework to trigger security bugs in JavaScript engines. Code Fragments provide an infrastructure to automatically generate JavaScript programs that trigger security bugs in JavaScript engines. Code Fragments use an abstract syntax tree (AST) representation of JavaScript code to automatically create programs that trigger security bugs. Code Fragments are designed to be easy to use and can be integrated into any JavaScript engine. Code Fragments are also designed to be efficient, with a single AST traversal generating the entire program.

1 Introduction

Software security issues are rapidly growing. In 2004, the annual CERT Computer Security Bulletin reported 289,084 OSS vulnerabilities, with 39 vulnerabilities reported in January 2004. In the same year, the Mozilla Foundation announced an initiative to provide security services to software developers. The goal of the initiative is to reduce the number of vulnerabilities reported in software. Code Fragments are designed to be easy to use and can be integrated into any JavaScript engine. Code Fragments are also designed to be efficient, with a single AST traversal generating the entire program.

As with other fuzzers, the JavaScript engine is particularly prone to security issues. In Mozilla Firefox, for instance, there are vulnerabilities that can be exploited to trigger security bugs. Code Fragments focus on specific vulnerabilities and are designed to trigger security bugs in both the Mozilla Firefox and Google Chrome browsers.

1.1 Language Support

Code Fragments support the JavaScript language. Code Fragments are designed to be easy to use and can be integrated into any JavaScript engine. Code Fragments are also designed to be efficient, with a single AST traversal generating the entire program.

2 Experimental Setup

In this section, we describe the experimental setup used to evaluate the effectiveness of Code Fragments. Code Fragments are designed to be easy to use and can be integrated into any JavaScript engine. Code Fragments are also designed to be efficient, with a single AST traversal generating the entire program.
In use at Mozilla since 2011, 2397 bugs in the JS engine.
On Google ClusterFuzz since 2014, ~170 Bugs found.

In use at Mozilla since 2011, 2397 bugs in the JS engine.

On Google ClusterFuzz since 2014, ~170 Bugs found.
On Google ClusterFuzz since 2014, ~170 Bugs found.

LANGFUZZ

In use at Mozilla since 2011, 2397 bugs in the JS engine.

OTHER FUZZING AT MOZILLA
JSFUNFUZZ
OTHER FUZZING AT MOZILLA

JSFUNFUZZ

Generator-based JS Fuzzer, written by Jesse Ruderman
OTHER FUZZING AT MOZILLA

JSFUNFUZZ

Generator-based JS Fuzzer, written by Jesse Ruderman

Supports most ECMA features as well as Mozilla extensions

function makeBranchUnstableLoop(d, b) {
    var reps = loopCount();
    var v = uniqueVarName();
    var mod = loopModulo();
    var target = rnd(mod);
    return forLoopHead(d, b, v, reps) + " { "+
        "if (" + v + "+ % " + mod + " == " + target + ") { " +
        + makeStatement(d - 2, b) + " } " +
        "else { " +
        + makeStatement(d - 2, b) + " } " +
        "};
    }

function makeTypeUnstableLoop(d, b) {
    var a = makeMixedTypeArray(d, b);
    var v = makeNewId(d, b);
    var bv = b.concat([v]);
    return "for each (let " + v + " in " + a + ") { " +
        + makeStatement(d - 2, bv) + "}";
    }

function makeFunOnCallChain(d, b) {
    var s = "arguments.callee";
    while (rnd(2))
        s += ",.caller";
    return s;
}
OTHER FUZZING AT MOZILLA

JSFUNFUZZ

Generator-based JS Fuzzer, written by Jesse Ruderman

Supports most ECMA features as well as Mozilla extensions

2869 bugs found since 2006
Generator/Mutator DOM Fuzzer, written by Jesse Ruderman
OTHER FUZZING AT MOZILLA
DOMFUZZ

Generator/Mutator DOM Fuzzer, written by Jesse Ruderman

Modular structure, different modules for different DOM features, e.g. CSS, Events, WebAudio, SVG, Unicode ...
OTHER FUZZING AT MOZILLA DOMFUZZ

Generator/Mutator DOM Fuzzer, written by Jesse Ruderman

Modular structure, different modules for different DOM features, e.g. CSS, Events, WebAudio, SVG, Unicode ...

Open-Source: https://github.com/MozillaSecurity/funfuzz
OTHER FUZZING AT MOZILLA
DARMA

Generation-based API fuzzer using a custom language
OTHER FUZZING AT MOZILLA DHARMA

Generation-based API fuzzer using a custom language

Open-Source: https://github.com/MozillaSecurity/dharma/
A NEW APPROACH: COVERAGE
A NEW APPROACH: COVERAGE

```c
void foo(int *a) {
    if (a)
        *a = 0;
}
```
A NEW APPROACH: COVERAGE

```c
void foo(int *a) {
    if (a)
        *a = 0;
}
```
A NEW APPROACH: COVERAGE

```c
void foo(int *a) {
    if (a)
        *a = 0;
}
```
A NEW APPROACH: COVERAGE

```
void foo(int *a) {
    if (a)
        *a = 0;
}
```

Diagram:

A -> B -> C
A -> C
B
C
A NEW APPROACH: COVERAGE

Public tools: AFL (american fuzzy lop) and libFuzzer
A NEW APPROACH: COVERAGE

Public tools: AFL (american fuzzy lop) and libFuzzer
A NEW APPROACH: COVERAGE

Public tools: AFL (american fuzzy lop) and libFuzzer
HELPFUL TOOLS

FUZZ TESTING

Mutator

Program

Crashes
Assertions
Hangs

Mutated
Sample
Test

Sample
Test

Sample
Test
HELPFUL TOOLS

Fuzz Testing

- Mutator
- Program
- Mutated Test
- Sample Test

Crashes
Assertions
Hangs
HELPFUL TOOLS

FUZZ TESTING

Crashes
Assertions
Hangs

Out-of-bounds read
Use-after-free
...

?
HELPFUL TOOLS

FUZZ TESTING

Crashes
Assertions
Hangs

Out-of-bounds read
Use-after-free
...

Valgrind: Too slow for fuzzing
HELPFUL TOOLS

FUZZ TESTING

Crashes
Assertions
Hangs

Out-of-bounds read
Use-after-free

Valgrind: Too slow for fuzzing

Sanitizers: AddressSanitizer, ThreadSanitizer, UndefinedBehaviorSanitizer
AddressSanitizer (ASan)
AddressSanitizer (ASan)

```
==3823== ERROR: AddressSanitizer: heap-use-after-free on address 0x7fc119c68dac at pc 0x7fc1137cbf2 bp 0x7ff478572d0 sp 0x7ff478572c8

READ of size 4 at 0x7fc119c68dac thread T0
  #0 0x7fc1137cbf2 in nsINode::GetBoolFlag(nsINode::BoolFlag) const content/base/public/nsINode.h:1348
  #1 0x7fc141235a8e in nsINode::HasTextNodeDirectionalityMap() const dist/include/nsINode.h:1431
  #2 0x7fc141234277 in mozilla::nsTextNodeDirectionalityMap::RemoveElementFromMap(nsINode*, mozilla::dom::Element*) content/base/src/DirectionalityUtils.cpp:535
  #3 0x7fc141233c3a in mozilla::WalkAncestorsResetAutoDirection(mozilla::dom::Element*, bool) content/base/src/DirectionalityUtils.cpp:642
  #4 0x7fc14123b7a1 in mozilla::SetDirection(mozilla::dom::Element*, nsIContent*) content/base/src/DirectionalityUtils.cpp:927

0x7fc119c68dac is located 44 bytes inside of 128-byte region [0x7fc119c6880,0x7fc119c68df8]
freed by thread T0 here:
  #0 0x40f992 in _interceptor_free
  #1 0x7fc14d89e40 in moz_free src/memory/mozalloc/mozalloc.cpp:48
  #2 0x7fc141b953e0 in operator delete(void*) dist/include/mozilla/mozalloc.h:224
  #3 0x7fc141b953e0 in nsTextNode::nsTextNode() content/base/src/nsTextNode.cpp:117
  #4 0x7fc141a3ba07 in nsNodeUtils::LastRelease(nsINode*) content/base/src/nsNodeUtils.cpp:258

previously allocated by thread T0 here:
  #0 0x40f7a72 in malloc
  #1 0x7fc15d69d554 in moz_xmalloc src/memory/mozalloc/mozalloc.cpp:54
  #2 0x7fc141b94c06 in operator new(unsigned long) dist/include/mozilla/mozalloc.h:200
  #3 0x7fc141b94c06 in NS_NewTextNode(nsIContent**, nsNodeInfoManager*) content/base/src/nsTextNode.cpp:106
  #4 0x7fc14546ee2 in nsHTMLEditTransaction::AppendText(nsIContent*, nsITreeOperation*) parser/html/nsHTMLEditTransaction.cpp:164

Shadow bytes around the buggy address:
0xff832-shared:  fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
AddressSanitizer (ASan)

```c
==3823== ERRPR: AddressSanitizer: heap-use-after-free on address 0x7fc1196c8dac at pc 0x7fc13f7c0f2 bp 0x7fff47872d0 sp 0x7fff47872d8

READ of size 4 at 0x7fc1196c8dac thread T0
  #0 0x7fc13f7c0f2 in nsINode::GetBoolFlag(nsINode::BooleanFlag) const content/base/public/nsINode.h:1348
  #1 0x7fc141235a8e in nsINode::HasTextNodeDirectionalityMap() const dist/include/nsINode.h:1431
  #2 0x7fc141234277 in mozilla::nsTextNodeDirectionalityMap::RemoveElementFromMap(nsINode*, mozilla::dom::Element*) content/base/src/DirectionalityUtils.cpp:535
  #3 0x7fc141233c3a in mozilla::WalkAncestorsResetAutoDirection(mozilla::dom::Element*, bool) content/base/src/DirectionalityUtils.cpp:642
  #4 0x7fc14123b7a1 in mozilla::SetDirOnBind(mozilla::dom::Element*, nsIContent*) content/base/src/DirectionalityUtils.cpp:927

0x7fc1196c8dac is located 44 bytes inside of 120-byte region [0x7fc1196c8d08,0x7fc1196c8df8)
freed by thread T0 here:
  #0 0x40f992 in moze_00c::free_interceptor_free
  #1 0x7fc15d65d490 in moze_00c::memory/mozalloc/mozalloc.cpp:48
  #2 0x7fc141b953e0 in operator delete(void*) dist/include/mozalloc.h:224
  #3 0x7fc141b953e0 in mozilla::nsTextNode::nsTextNode() content/base/src/nsTextNode.cpp:117
  #4 0x7fc141a3ba07 in mozilla::nsNodeUtils::LastRelease(nsINode*) content/base/src/nsNodeUtils.cpp:258

previously allocated by thread T0 here:
  #0 0x40f972 in malloc
  #1 0x7fc15d65d554 in moze_00c::memory/mozalloc/mozalloc.cpp:54
  #2 0x7fc141b94c00 in operator new(unsigned long) dist/include/mozalloc.h:200
  #3 0x7fc141b94c00 in NS_NewTextNode(nsIContent**, nsIContentManager*) content/base/src/nsTextNode.cpp:106
  #4 0x7fc14f54bd2e in nsHtml5TreeOperation::AppendText(nsIContent*, nsHtml5TreeOpExecutor*) parser/html/nsHtml5TreeOperation.cpp:164

Shadow bytes around the buggy address:
0x1ff8232d9160: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x1ff8232d9170: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x1ff8232d9180: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x1ff8232d9190: 00 00 00 00 tb fb fb fb fb fb fb fb fb fb fb
0x1ff8232d91a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x1ff8232d91b0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
0x1ff8232d91c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x1ff8232d91d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1ff8232d91e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x1ff8232d91f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1ff8232d9200: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa

Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Heap left redzone: fa
Heap right redzone: fb
Freed Heap region: fd

==30234== ABORTING
```
AddressSanitizer (ASan)

```
==3823== ERROR: AddressSanitizer: heap-use-after-free on address 0x7fc13f7cbf2 in nsINode::GetBoolFlag(nsINode::BooleanFlag) const content/base/public/nsINode.h:1348
==3823==    at 0x7fc13f7cbf2 in nsINode::GetBoolFlag(nsINode::BooleanFlag) const content/base/public/nsINode.h:1348
==3823==    by 0x7fc141234277 in mozilla::nsTextNode::DirectionalityMap::RemoveElementFromMap(nsINode*, mozilla::dom::Element*) content/base/src/DirectionalityUtils.cpp:535
==3823==    by 0x7fc141233e3a in mozilla::WalkAncestorsResetAutoDirection(mozilla::dom::Element*, bool) content/base/src/DirectionalityUtils.cpp:642
==3823==    by 0x7fc14123b7a1 in mozilla::SetDirStart(mozilla::dom::Element*, nsIContent*) content/base/src/DirectionalityUtils.cpp:927
...
0x7fc1196c8dac is located 44 bytes inside of 128-byte region [0x7fc1196c8d00,0x7fc1196c8df8]
freed by thread 70 here:
    by 0x40f6572 in interceptor_free
    by 0x7fc141b953e0 in operator delete(void*) dist/include/mozilla/mozalloc.h:224
    by 0x7fc141b953e0 in mozilla::nsTextNode::~nsTextNode() content/base/src/nsINode.h:117
    by 0x7fc141a5ba07 in mozilla::nsNodeUtils::LastRelease(nsINode*) content/base/src/nsNodeUtils.cpp:258
... 
previously allocated by thread 70 here:
    by 0x40f6572 in malloc
    by 0x7fc141b94c00 in operator new(unsigned long) dist/include/mozilla/mozalloc.h:200
    by 0x7fc141b94c00 in NS_NewTextNode(nsIContent**, nsIContentManager*) content/base/src/nsTextNode.cpp:106
    by 0x7fc1454bd2e in nsHtml5TreeOperation::AppendText(nsIContent*, nsHtml5TreeOpExecutor*) parser/html/nsHtml5TreeOperation.cpp:164
... 
Shadow bytes around the buggy address:
0x1ff823d29160: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x1ff823d29170: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd f...
AddressSanitizer (ASan)

Compile-Time Instrumentation
Average Slowdown: 1.93x
A SIMPLE EXAMPLE
A SIMPLE EXAMPLE

CVE-2014-0160 (Heartbleed)
Critical out-of-bounds read in OpenSSL
A SIMPLE EXAMPLE

CVE-2014-0160 (Heartbleed)
Critical out-of-bounds read in OpenSSL

Hanno Böck combined:
OpenSSL library
AddressSanitizer
Coverage Guided Fuzzer
wget https://www.openssl.org/source/openssl-1.0.1f.tar.gz
Tar xf openssl-1.0.1f.tar.gz
CCFLAGS="-fsanitize-coverage=edge,indirect-calls" # -fsanitize-coverage=bit-counters
(cd openssl-1.0.1f/44; ./config 44
make -j 32 CC="clang -g -fsanitize=address $CC_flags"
# Get and build libFuzzer
svn co http://llvm.org/svn/llvm-project/llvm/trunk/lib/Fuzzer
clang -c -g -02 -std=c++11 Fuzzers/*.cpp -o Fuzzers
# Get examples of key/pem files.
git clone https://github.com/hannob/selftest
cp selftest/server.* .
cat << EOF > handshake-fuzz.cc
#include <openssl/ssl.h>
#include <openssl/err.h>
#include <assert.h>
#include <stdio.h>
#include <etdes.h>

SSL_CTX *ctx;
int Init()
{    
SSL_library_init();
SSL_load_error_strings();
ERR_load_BIO_strings();
openssl_add_all_algorithms();
assert (ctx = SSL_CTX_new(TLSv1_method()));
assert (SSL_CTX_use_certificate_file(ctx, "server.pem", SSL_FILETYPE_PEM));
assert (SSL_CTX_use_PrivateKey_file(ctx, "server.key", SSL_FILETYPE_PEM));
return 0;
}
extern "C" int LLVMFuzzerTestOneInput(const uint8_t *Data, size_t Size)
{
    static int unused = Init();
    SSL *server = SSL_new(ctx);
    BIO *s_bio = BIO_new(BIO_s_mem());
    BIO *r_bio = BIO_new(BIO_s_mem());
    BIO *out_bio = BIO_new(BIO_s_mem());
    SSL_set_bio(server, s_bio, r_bio);
    SSL_set_accept_state(server);
    BIO_write(s_bio, Data, Size);
    BIO_free_all(server);
    SSL_free(server);
    return 0;
}
EOF
# Build the fuzzer.
clang++ -g -fsanitize=address \openssl-1.0.1f/libfuzzels.a openssl-1.0.1f/libcrypto.a Fuzzer.o
# Run 20 independent fuzzer jobs,
./a.out -jobs=20 -workers=20
A SIMPLE EXAMPLE

Found Heartbleed **in only 6 hours**, no TLS-specific fuzzer required
SCALING FUZZING
SCALING FUZZING

More CPU time = More bugs found
SCALING FUZZING

More CPU time = More bugs found

Deployment?
SCALING FUZZING

More CPU time = More bugs found

Deployment?  Management?
SCALING FUZZING

More CPU time = More bugs found

Deployment?

Management?

Results?
Scaling Fuzzing

More CPU time = More bugs found

Deployment?  Management?  Results?

A set of *generic, open-source* tools would be nice
There are 26 unbucketed entries in the database.

<table>
<thead>
<tr>
<th>ID</th>
<th>Date Added</th>
<th>Bucket</th>
<th>Short Signature</th>
<th>Crash Address</th>
<th>Test Status</th>
<th>Product</th>
<th>Version</th>
<th>Platform</th>
<th>OS</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>373340</td>
<td>Wed, 13 Jul 2016 12:55:23 +0000</td>
<td>CL</td>
<td>Assertion failure: shadowZone.needsIncrementalBarrier(), at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/heap.h:1205</td>
<td>0x0L</td>
<td>Q7: 37279 (binary)</td>
<td>mozilla-central</td>
<td>asea8f7d245</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>373348</td>
<td>Wed, 13 Jul 2016 12:55:12 +0000</td>
<td>CL</td>
<td>Assertion failure: shadowZone.needsIncrementalBarrier(), at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/heap.h:1205</td>
<td>0x0L</td>
<td>Q7: 280</td>
<td>mozilla-central</td>
<td>asea8f7d245</td>
<td>x86</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>384098</td>
<td>Thu, 12 Jul 2016 08:41:22 +0000</td>
<td>CL</td>
<td>Assertion failure: DB::елеScope_ (LOG4J) Cannot allocate a new chunk in an inflated space., at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/dbmalloc.cpp:105</td>
<td>0x0L</td>
<td>Q7: 332</td>
<td>mozilla-central</td>
<td>1b0e8f2d238e</td>
<td>x86</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>363138</td>
<td>Sun, 10 Jul 2016 07:35:59 +0000</td>
<td>CL</td>
<td>[0x:Activator::set]</td>
<td>0x06L</td>
<td>Q7: 37114 (binary)</td>
<td>mozilla-central</td>
<td>b0e2d9e6f57d</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>359018</td>
<td>Wed, 6 Jul 2016 18:45:36 +0000</td>
<td>CL</td>
<td>[0x:Tracer2::TracerRange]::Value()</td>
<td>0x07:132b2000000L</td>
<td>Q7: 22983 (binary)</td>
<td>mozilla-central</td>
<td>7bbd5f832e8</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>358833</td>
<td>Sun, 3 Jul 2016 22:36:47 +0000</td>
<td>CL</td>
<td>Assertion failure: alink &lt; 0, is length, at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/heap.h:458</td>
<td>0x0L</td>
<td>Q7: 32698 (binary)</td>
<td>mozilla-central</td>
<td>39df8f5b7642</td>
<td>arm</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>354879</td>
<td>Sun, 3 Jul 2016 09:29:55 +0000</td>
<td>CL</td>
<td>Assertion failure: 53::Exception::Pendulous(), at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/heap.h:114</td>
<td>0x0L</td>
<td>Q7: 18113 (binary)</td>
<td>mozilla-central</td>
<td>fdece57b4e4f</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>354465</td>
<td>Sat, 2 Jul 2016 17:35:53 +0000</td>
<td>CL</td>
<td>Assertion failure: (nullifying)</td>
<td></td>
<td>isDessing(), at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/heap.h:114</td>
<td>0x0L</td>
<td>Q7: 34385 (binary)</td>
<td>mozilla-central</td>
<td>fdece57b4e4f</td>
<td>x86</td>
</tr>
<tr>
<td>353761</td>
<td>Fri, 1 Jul 2016 14:09:27 +0000</td>
<td>CL</td>
<td>[0x:script]</td>
<td>0x0b7e360673cL</td>
<td>Q7: 17250 (binary)</td>
<td>mozilla-central</td>
<td>82e1f319e05</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>341444</td>
<td>Mon, 27 Jun 2016 11:52:51 +0000</td>
<td>CL</td>
<td>Assertion failure: generation == table, generation(), at threadrs/qqtb/mobile-central-clone/workspace/jsr/src/heap.h:608</td>
<td>0x0L</td>
<td>Q7: 30878 (binary)</td>
<td>mozilla-central</td>
<td>620e8f39647e</td>
<td>arm</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>339575</td>
<td>Sun, 26 Jun 2016 02:36:52 +0000</td>
<td>CL</td>
<td>[0x:77]</td>
<td>0x7b0b21f02dL</td>
<td>Q7: 190264 (binary)</td>
<td>mozilla-central</td>
<td>811026652b10</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>339853</td>
<td>Sat, 25 Jun 2016 06:06:24 +0000</td>
<td>CL</td>
<td>[0x:77]</td>
<td>0x76800b3d976L</td>
<td>Q7: 41983 (binary)</td>
<td>mozilla-central</td>
<td>7ee3e03f191f</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>329692</td>
<td>Wed, 22 Jun 2016 14:03:47 +0000</td>
<td>CL</td>
<td>[0x:77]</td>
<td>0x769b3f6c300L</td>
<td>Q7: 23383 (binary)</td>
<td>mozilla-central</td>
<td>3e552986b56</td>
<td>x86-64</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>327322</td>
<td>Sun, 19 Jun 2016 01:35:05 +0000</td>
<td>CL</td>
<td>[0x::HasUncompiledScript]</td>
<td>0x02:1L</td>
<td>Q7: 36444 (binary)</td>
<td>mozilla-central</td>
<td>3e035b1e252b</td>
<td>arm</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
<tr>
<td>327098</td>
<td>Sat, 18 Jun 2016 15:44:28 +0000</td>
<td>CL</td>
<td>[0x:77]</td>
<td>0x0f99000f0cL</td>
<td>Q7: 219229 (binary)</td>
<td>mozilla-central</td>
<td>59859d88f0df</td>
<td>x86</td>
<td>LangFuzz-didl</td>
<td></td>
</tr>
</tbody>
</table>
FUZZMANAGER

Description
Assertion failure: failibleScope_ (DOM) Cannot allocate a new chunk in an infallible scope, at /svn/repos/mozilla-central/ju/src/js/LifoAlloc.cpp:105

External Bug Status
Reported as bug 1285218.

Unlink

Crashes covered by this signature
4

Best Crash Entry
367343 (Size: 121)

Signature
{
  "symptoms": [  
    {  
      "src": "stderr",  
      "type": "output",  
      "value": "Assertion failure: failibleScope_ (DOM) Cannot allocate a new chunk in an infallible scope\"s\", at \(\{a-zA-Z\}\)js/LifoAlloc.cpp:105/\""  
    },  
    {  
      "type": "stackFrames",  
      "functionNames": [  
        "js::LifoAlloc::getOrCreateChunk",  
        "LifoAlloc\",  
        "js::LifoAlloc::allocInfallible",  
        "LifoAlloc::Infallible",  
        "operator new",  
        "++",  
        "++",  
        "jit::EmBuilder::initializeArrayElement"  
      ],  
      "type": "crashAddress",  
      "address": "0x8000000000000000\"  
    }  
  ]
}
LIFE OF A BUG
LIFE OF A BUG

deploy machines

FuzzManager
LIFE OF A BUG

deploy machines

FuzzManager

provide signatures
LIFE OF A BUG

deploy machines

provide signatures

submit crashes

FuzzManager
LIFE OF A BUG

deploy machines

FuzzManager

provide signatures

Triage

submit crashes
LIFE OF A BUG

1. File Bug Report
2. Bug Database (e.g., Bugzilla)
3. Triage
4. Deploy machines
5. Provide signatures
6. Submit crashes

FuzzManager
GETTING INVOLVED
GETTING INVOLVED

https://github.com/MozillaSecurity
GETTING INVOLVED

https://github.com/MozillaSecurity

irc://irc.mozilla.org/#security
GETTING INVOLVED

https://github.com/MozillaSecurity
irc://irc.mozilla.org/#security
https://www.mozilla.org/security/bug-bounty/
GETTING INVOLVED

https://github.com/MozillaSecurity

irc://irc.mozilla.org/#security

https://www.mozilla.org/security/bug-bounty/

3,000 USD for every critical/high security vulnerability

(... and a t-shirt)
THE FUTURE
THE FUTURE
THE FUTURE
THE FUTURE

Experiment: Come up with hypotheses, confirm them
Experiment: Come up with hypotheses, confirm them

Derive a model of how things work
THE FUTURE

Experiment: Come up with hypotheses, confirm them

Derive a model of how things work

Fuzzing tools should learn and evolve
THANKS