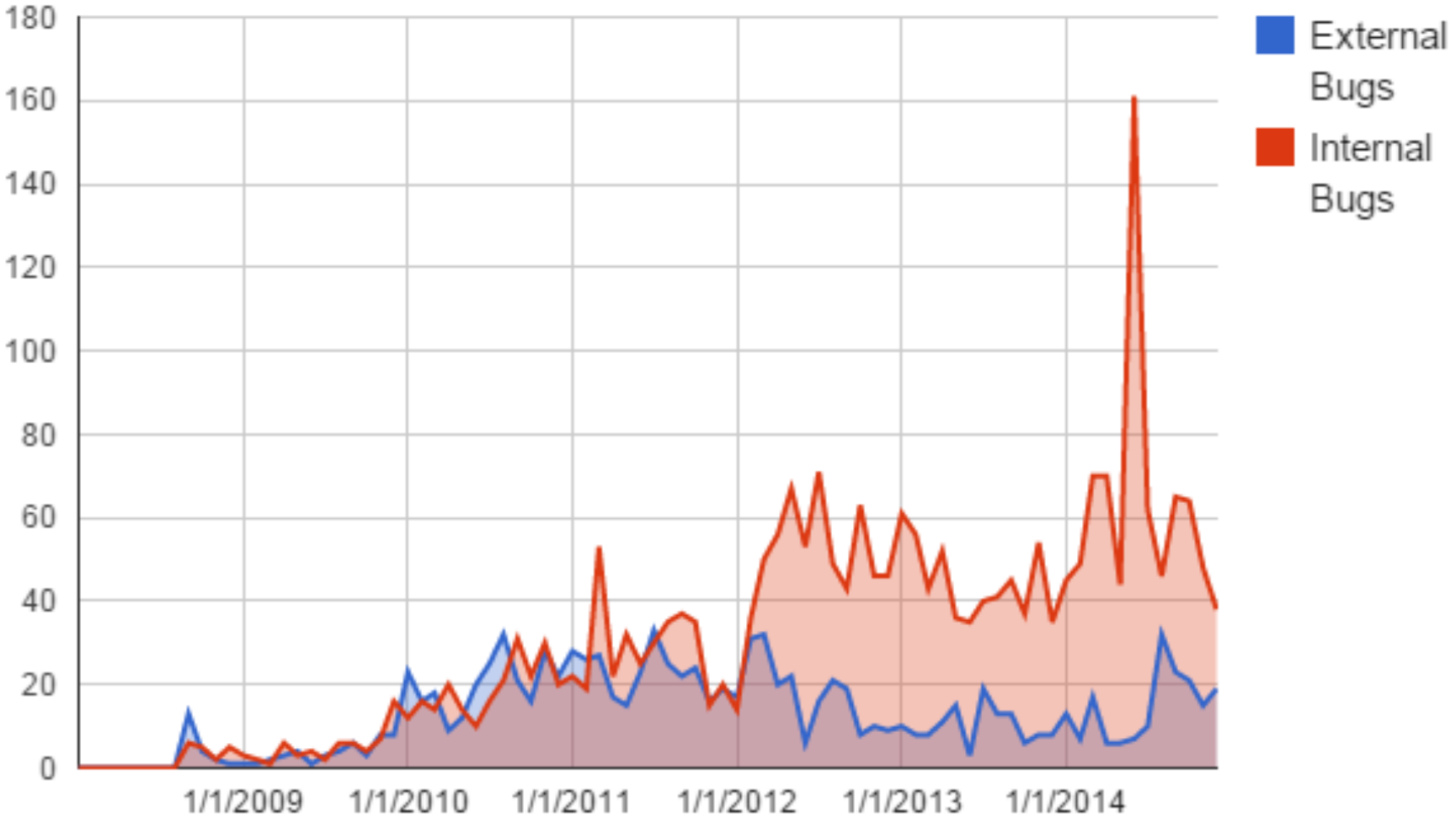
ClusterFuzz

Abhishek Arya <inferno@chromium.org>

About me!

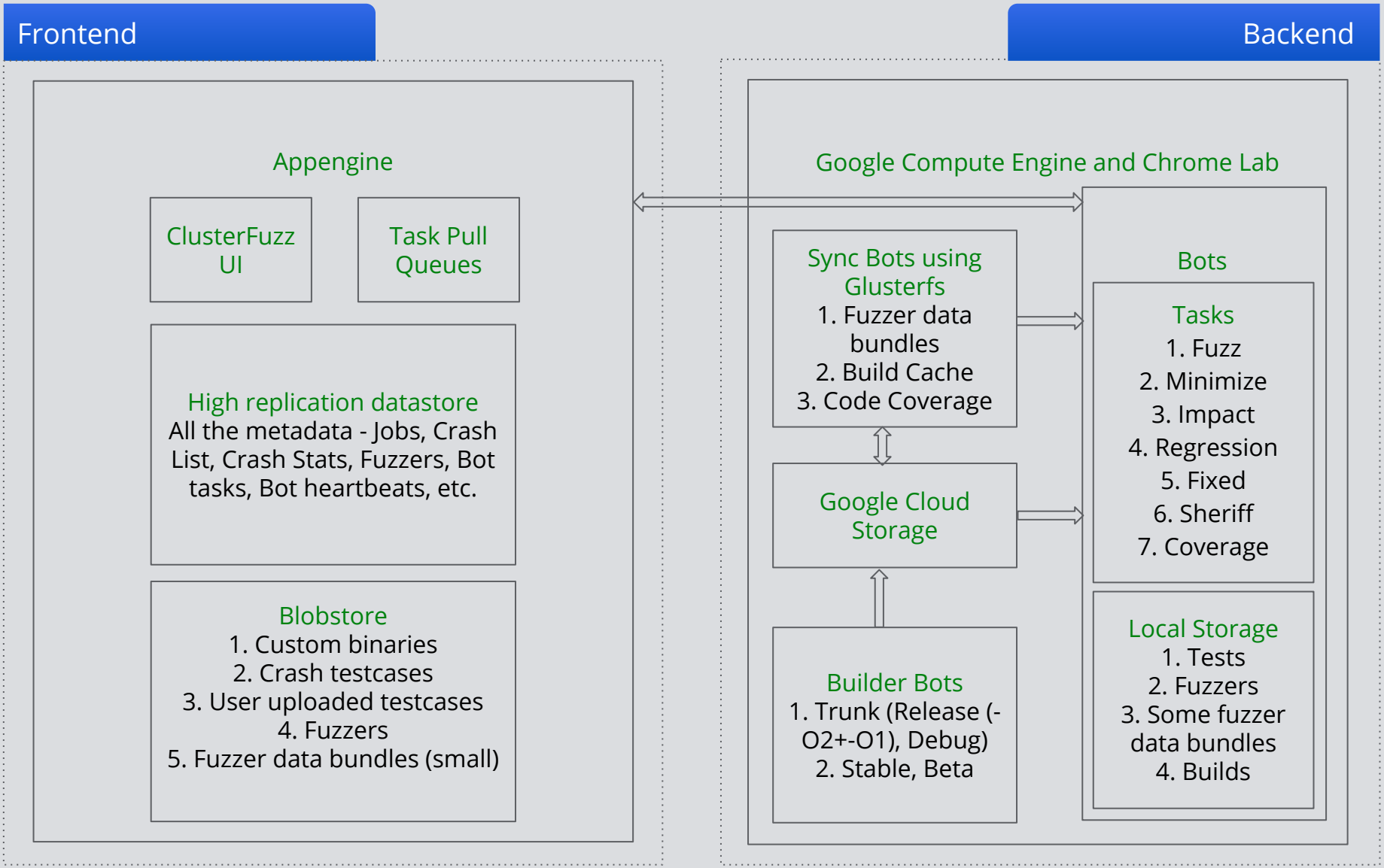
- Member, Chrome Security Team
- Tech Lead of Bugs-- FA
- ClusterFuzz architect
- Fuzzer author and facilitator
- Hack other browsers in free time

Chrome Vulns: Past trends



Solution: ClusterFuzz

ClusterFuzz: Architectural Overview



ClusterFuzz: Goals

Automated crash detection, analysis and management

Fully reproducible and minimized testcases

Real-time regression and fixed testing

Automated crash detection, analysis and management

Goal 1:Crash Automation & Management

- Fuzzers
- Code Coverage
- Memory Debugging Tools
- Scale, Scale and Scale
- Vulnerability Reward Program

Fuzzers: How to write one ?

- Common semantics, e.g.

  run.* --input_dir=A --output_dir=B --no_of_files=C
- Common prefix ‘fuzz-’ (+tags like http-, flags-)
- Cross-platform (python, perl, nodejs, etc)
- Data bundles (+shared)
- Other scripts - launcher, coverage, etc

Fuzzers: What infrastructure does?

- Setup build {trunk}, run application with test
- Choose params
  - Gestures, tool settings, timeout multiplier, window location+size, etc.
- Figure out resource dependencies
- Test for reproducibility
- Test for duplicates
- Store crash, coverage, stats, etc

Testcase Duplication Check

- Disable inline frames, e.g. -inlining in llvm-symbolizer
- crash_type - e.g. Heap-buffer-overflow READ 2
- crash_state - top 3 frames
  - +namespaces, -line_numbers
  - e.g.

    WebCore::SVGDocumentExtensions::removeAnimationElement
    WebCore::SVGSMILElement::removedFrom
    WebCore::ContainerNode::removeChild
- security_flag

Fuzzers: Types

- Generation based fuzzers
- Mutation based fuzzers
- Evolutionary fuzzers

Generation based fuzzers

- Isolated, based on specification / api
- Quick to hack and deliver
- Complex >>> simple strategies
- Finds bugs fast, fades even faster
- Good for catching regressions
- Example bug: {now fixed}

  document.body = document.createElement('iframe')

Generation based fuzzers: Examples

- CSS Parser
- Editing API
- WebAudio API
- <canvas> API
- etc

Mutation Based Fuzzers

- Initial work on test corpus (example)
- Simple >>> Complex strategies
- Steady stream of bugs over time
- Suited well for file format, protocol fuzzing
- Cautions - checksums, compression, etc
- Good value for the buck

Mutation Based Fuzzers: Example {V8}

- Pick several tests from mjsunit (V8's tests)
- Prepare the test
  - Normalize variables and function names
  - Cleanup to minimize exceptions
  - Combine multiple tests into a single test
- Mutate the test
  - Replace variables or function calls to exercise potentially untested cases
  - Trigger function optimization, gc(), ...
  - etc...

V8 Bounds check removal bug

```
var a = new Int32Array(1024);

function test_base(a, base, condition)
{
  a[base + 1] = 1;
  a[base + 4] = 2;
  a[base + 3] = 3;
  a[base + 2] = 4;
  a[base + 4] = base + 4;
  if (condition) {
    a[base + 1] = 1;
  }
}

test_base(a, 1, true);
%OptimizeFunctionOnNextCall(test_base);
test_base(a, 3, false);
```

```
var __v_0 = new Int32Array(1024);

function __f_1(__v_28, base, condition)
{
  __v_0[base - 1] = 1;
  __v_0[base - 10] = 2;
  __v_0[base + 3] = 3;
  __v_0[base + 2] = 4;
  __v_0[base + 4] = base + 4;
  if (condition) {
    __v_0[base + 1] = 1;
  }
}

__f_1(__v_0, 11, true);
%OptimizeFunctionOnNextCall(__f_1);
__f_1(__v_1, -4, false); // Crash
```

https://crbug.com/344186

Evolutionary Fuzzers

- Lets talk about something else...

Code Coverage

```
void foo(int *a) {          [A]
    if (a)                  [B]
        *a = 0;             [C]
}                           [D]
```

- Function level coverage - [A]
- Block level coverage - [B], [C], [D]
- Edge level coverage - [B], [C], [D], [E]
  where E = dummy b/w B and D.

Code Coverage

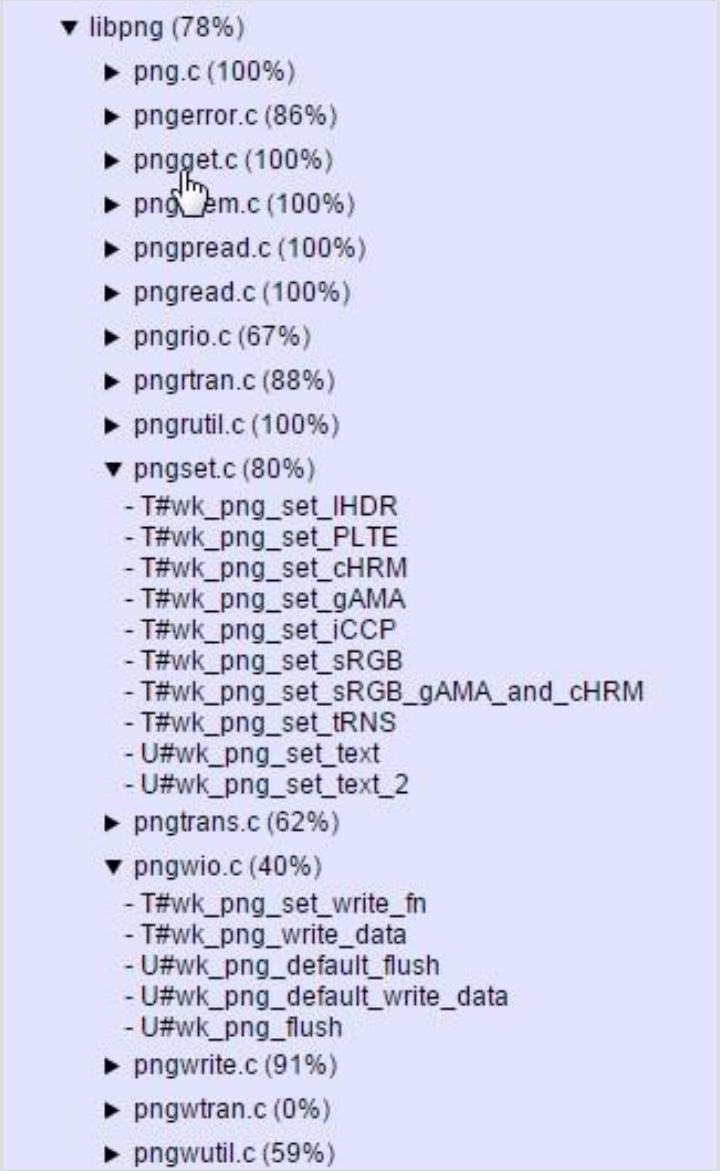
- Initially part of AddressSanitizer (ASAN)
- In-process fuzzing for speed
- Work-in-progress export to other sanitizers
- Needs latest clang >= r217106
- No contention on counters
- Negligible I/O overhead
- CPU overhead (Function: <5%, Edge: < 40%)

Code Coverage: Platforms

Android	ASAN coverage
Linux	ASAN coverage
Mac	---
Windows	---
Chrome OS	~~~
iOS	---

~ indicates partial support, --- No support

Code Coverage: Aggregate View Sample



Evolutionary Fuzzers

- Feedback-driven fuzzing
- Additional build instrumentation {+overhead}
- Per-testcase coverage {pcs vs syms}
- Shared storage for ::
  - aggregate coverage
  - optimal files list + metadata

Code Coverage: fuzzer_utils api

```
fuzzer_utils.createCoverageFile(
  fuzz_testcase_file_path,
  original_file_path=None,
  testcase_is_modified=True,
  clear_testcase_with_zero_coverage=True,
  store_modified_testcases=True)
```

Code Coverage: fuzzer_utils api

- testcase_is_original
  - new coverage -> add to optimal files list
  - no new coverage -> delete if clear_testcase_with_zero_testcase set
- testcase_is_modified
  - new coverage -> add to corpus+optimal file list if store_modified_testcases set
  - no new coverage -> ignore

Memory Debugging Tools

- Why not Valgrind ??
  - cpu: 10-300x
  - heap bugs only
  - slow boot

- Why not a single tool ??
  - Slowdowns will add up
  - Memory overheads will multiply
  - Non-trivial to implement

Memory Debugging Tools

- AddressSanitizer (aka ASan)
  - detects use-after-free, buffer overflows (heap, stack, globals), stack-use-after-return, container-overflow.
  - cpu: 2x, memory: 1.5x-3x

- ThreadSanitizer (aka TSan)
  - detects data races, esp on use-after-frees, object vptr.
  - cpu: 4x-10x, memory: 5x-8x

Memory Debugging Tools

- MemorySanitizer (aka MSan)
  - detects uninitialized memory reads
  - cpu: 3x, memory: 2x
  - special mode: origins

- UndefinedBehaviorSanitizer (aka UBSan)
  - detects several classes of bugs(19), esp on type confusion, etc.
  - cpu: unavailable
  - memory: ~1x (no allocator, no shadow).

Memory Debugging Tools

- Others
  - SyzyASAN
  - DrMemory

Memory Debugging Tools: Examples

- Type confusion (UBsan vptr)

```
../../third_party/WebKit/Source/core/rendering/RenderTable.h:366:1: runtime error:
downcast of address 0x3e5988411f38 which does not point to an object of type blink::
RenderTable
0x3e5988411f38: note: object is of type blink::RenderBlockFlow
 00 00 00 00  d8 94 4b 4f 33 7f 00 00  60 f0 47 eb 8b 06 00 00  00 40 60 b8 68 3c 00 00
48 81 45 88
              ^~~~~~~~~~~~~~~~~~~~~~
              vptr for blink::RenderBlockFlow
    #0 0x7f33475e5867 in blink::RenderTableSection::table() const
third_party/WebKit/Source/core/rendering/RenderTable.h:366:1
    #1 0x7f3347606aae in blink::RenderTableSection::setNeedsCellRecalc()
third_party/WebKit/Source/core/rendering/RenderTableSection.cpp:1433:26
```

Memory Debugging Tools: Examples

- Container-overflow (ASAN)

```
#include <vector>
#include <assert.h>
typedef long T;
int main() {
  std::vector<T> v;
  v.push_back(0);
  v.push_back(1);
  v.push_back(2);
  assert(v.capacity() >= 4);
  assert(v.size() == 3);
  T *p = &v[0];
  // Here the memory is accessed inside a heap-allocated buffer
  // but outside of the region `[v.begin(), v.end())`.
  return p[3];  // OOPS.
  }
```

Memory Debugging Tools

	Buffer overflow	Use after free	Type confusion	Uninitialized value	Same origin bypass
Android	ASAN, ~Sec-asserts~	ASAN	~Sec-asserts~	---	Site Isolation Project
Linux	ASAN, ~Sec-asserts~	ASAN, TSAN	UBSAN vptr, ~Sec-asserts~	MSAN	
Mac	ASAN, ~Sec-asserts~	ASAN	~Sec-asserts~	---	
Windows	ASAN, SyzyASAN, ~DrMemory~, ~Sec-asserts~	ASAN, SyzyASAN, ~DrMemory~	~Sec-asserts~	~DrMemory~	
ChromeOS	~WIP~	~WIP~	~WIP~	~WIP~	
iOS	---	---	---	---	

~ indicates partial support, --- No support

Scale, Scale and Scale

- Linux - 3000
- Windows - 1000
- Android - 100
- Mac - 20
- ChromeOS - ~WIP~
- iOS - ~WIP~

ClusterFuzz + Cash

- Chrome Vulnerability Reward Program (VRP)
  - Started in late January 2010.
  - ~$1.6 million rewarded to date.
  - 27 critical, 641 high, 205 medium, 9 low paid.
- ClusterFuzz + VRP
  - 99 bugs from “Fuzzer Contribution Program”.
  - Rewards upgraded in Aug 2013, Sep 2014 {based on clarity, exploitability, patch}.

Fully reproducible and minimized testcases

Goal 2: Fully reproducible & minimized tests

- Same bot configuration as crash
- Multi-threaded minimization based on Delta Debugging
- Custom minimizers for some file types
- +Gesture minimization
- +Resource minimization
- +Tool settings correction (redzone {asan}, history size {tsan}, origins {msan}, etc).

How does the minimizer work ?

- Tokenize the input
- Generate hypotheses that certain groups of tokens are not required for the crash
- Test hypotheses by running the test with the tokens from the hypothesis removed
  - If it crashes, removing them was fine
  - If not, try breaking it into smaller groups

Custom Minimizers

- Specialized tokenizers
- Generate hypothesis for groups of tokens that can be removed together
  - `assertTrue(crash()) → crash()`
  - `try { crash(); } catch(e) {} → crash()`
  - `if (i_am_true) { crash(); } → crash()`

Resource Minimization

- Problems
  - Tons of tests {all over the web (ie, moz, etc.)}
  - Dev complaints with local reproduction
  - Testcase flakiness issues
  - Platform issues with file system monitors

Resource Minimization

- Solution
  - --log-net-log command line switch
  - Platform-independent resource dump at runtime
  - Captures both local (file://) and remote loads (e.g. http GET, POST).
  - Example:: {"params":{"load_flags":67125248,"method":"GET","priority":"LOW","url":"file:///Z:/test.js"},"phase":1,"source":{"id":178,"type":1},"time":"278568790","type":91},

Tool Settings Correction: ASAN example crbug.com/118662

- Stack trace with default redzone (128)

```
==9485== ERROR: AddressSanitizer heap-use-after-free on address 0x7f8f653ff11e at
pc 0x7f8f849fbb10 bp 0x7f8f5514a0a0 sp 0x7f8f5514a098
READ of size 2 at 0x7f8f653ff11e thread T14
    #0 0x7f8f849fbb10 in WTF::charactersToIntStrict(unsigned short const*,
unsigned long, bool*, int)
    #1 0x7f8f8589d863 in WebCore::InlineTextBox::isLineBreak() const
    #2 0x7f8f858a771d in WebCore::InlineTextBox::containsCaretOffset(int)
```

- Actual stack trace with bigger redzone (1024/2048)

```
==14334== ERROR: AddressSanitizer heap-buffer-overflow on address 0x7f7e42b9b81c
at pc 0x7f7e8f79a6ca bp 0x7f7e3cc30040 sp 0x7f7e3cc30038
READ of size 2 at 0x7f7e42b9b81c thread T15
#0 0x7f7e8f79a6ca in WebCore::InlineTextBox::isLineBreak() const
#1 0x7f7e8f7abddd in WebCore::InlineTextBox::containsCaretOffset(int) const
#2 0x7f7e8e06b19d in WebCore::Position::inRenderedText() const
```

Real-time regression and fixed testing

Goal 3: Real-time regression and fixed testing

- Use LKGR builds archived on Google Cloud
- Account for bad builds / startup crashes
- Use a LOOK_BEHIND_WINDOW
- If previous step failed, then use binary bisect
- Use FindIt to find culprit changelist

FindIt: Culprit CL Finder

- Manual owner triage is usually ....
  - Slow
  - Inaccurate
  - Complex
    - from factors such as refactorings, size of regression range, etc

FindIt: How it works

1. Parse the stacktrace

```
#0 0x7fbba6d3827f in WebCore::ThreadState::visitStack(WebCore::Visitor*) src/third_party/WebKit/Source/heap/ThreadState.cpp:343
#1 0x7fbba6d380bd in WebCore::ThreadState::visitRoots(WebCore::Visitor*) src/third_party/WebKit/Source/heap/ThreadState.cpp:357
#2 0x7fbba6d31abb in WebCore::Heap::collectGarbage(WebCore::ThreadState::StackState, WebCore::Heap::GCType) src/third_party/WebKit/Source/heap/Heap.cpp:1291
#3 0x7fbba6d3713d in WebCore::ThreadState::cleanup() src/third_party/WebKit/Source/heap/ThreadState.cpp:301
#4 0x7fbba8d65e41 in WebCore::WorkerThread::workerThread() src/third_party/WebKit/Source/core/workers/WorkerThread.cpp:147
#5 0x7fbba69d8833 in WTF::wtfThreadEntryPoint(void*) src/third_party/WebKit/Source/wtf/ThreadingPthreads.cpp:175
#6 0x7fbb9b48ce99 in start_thread /build/buildd/eglibc-2.15/nptl/pthread_create.c:308
#7 0x7fbb991e63fc in ?? /build/buildd/eglibc-2.15/misc/../sysdeps/unix/sysv/linux/x86_64/clone.S:112
```

ThreadState.cpp	(Index: 0, crashed line: 343) (Index: 1, crashed line: 357) (Index: 3, crashed line: 301)	Component: blink
Heap.cpp	(Index: 2, crashed line: 1291)	Component: blink
…	…	…

FindIt: How it works

2. Parse the changelog(s) in the regression range

166619	fmalita@chromium.org	Mark Skia suppressions for rebaseline BUG=339090,339806,339852 TBR=robertphillips@google.com Review URL: https://codereview.chromium.org/134193018 M - /trunk/LayoutTests/TestExpectations
166622	jochen@chromium.org	Add assert that the referrer header is modified via the correct method BUG=none R=abarth@chromium.org Review URL: https://codereview.chromium.org/153643002 M - /trunk/Source/web/tests/AssociatedURLLoaderTest.cpp M - /trunk/public/platform/WebURLRequest.h M - /trunk/Source/platform/exported/WebURLRequest.cpp
166623	ch.dumez@samsung.com	Update HTMLCollection's named property getter to behave according to spec Update HTMLCollection's named property getter to behave according to spec: http://dom.spec.whatwg.org/#htmlcollection 1. namedItem() should be marked as named getter in IDL as per the spec. This also means we can get rid of the anonymous named getter in our IDL. 2. The named getter should be enumerable as per Web IDL, meaning that elements id / names should be enumerated. The supported property names are spec'd at: http://dom.spec.whatwg.org/#htmlcollection Firefox 26 and IE11 both show the names while enumerating. However, IE11 does not show the ids when enumerating, which is incorrect. 3. The argument to namedItem() named getter should be mandatory. This is consistent with the spec, Firefox 26 and IE11.

File Name	CL
ThreadState.cpp	166624, 166634
Heap.cpp	166624
WorkerThread.cpp	166624
ScrollbarTheme.cpp	166630
...	...

FindIt: How it works

3. Generate a list of suspected CLs, and sort / filter the results

- From the parsed stacktrace and changelogs

CL	Reason
166624	Changes line 147 of WorkerThread.cpp Changes line 301 of ThreadState.cpp Changes file Heap.cpp
166634	Changes line 343 of TheadState.cpp

- 166634 is the first result because:
  - Both CLs change the crashed lines.
  - 166634 changes line 343 of ThreadState.cpp, and the crash on the line 343 happens at the top of the stack (0)
- Real culprit CL is.....166634!

FindIt: How it works

4. Show blame information if no results are available

```
    #0 0x7f133d26e929 in WebCore::toRenderBox(WebCore::RenderObject*) src/third_party/WebKit/Source/core/rendering/RenderBox.h:695
    #1 0x7f13435a48ad in WebCore::RenderMedia::layout() src/third_party/WebKit/Source/core/rendering/RenderMedia.cpp:65
    #2 0x7f13429fe027 in WebCore::RenderObject::layoutIfNeeded() src/third_party/WebKit/Source/core/rendering/RenderObject.h:683
    #3 0x7f1342e9e21f in WebCore::RenderBlock::layoutInlineChildren(bool, WebCore::LayoutUnit&, WebCore::LayoutUnit&)
src/third_party/WebKit/Source/core/rendering/RenderBlockLineLayout.cpp:1996
    #4 0x7f1342bd1d87 in WebCore::RenderBlock::layoutBlock(bool, WebCore::LayoutUnit) src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:1636
    #5 0x7f1342bcb344 in WebCore::RenderBlock::layout() src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:1445
    #6 0x7f1342c2011f in WebCore::RenderBlock::layoutBlockChild(WebCore::RenderBox*, WebCore::RenderBlock::MarginInfo&, WebCore::LayoutUnit&, WebCore::LayoutUnit&)
src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:2623
    #7 0x7f1342bde1aa in WebCore::RenderBlock::layoutBlockChildren(bool, WebCore::LayoutUnit&) src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:2560
    #8 0x7f1342bd1dff in WebCore::RenderBlock::layoutBlock(bool, WebCore::LayoutUnit) src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:1638
    #9 0x7f1342bcb344 in WebCore::RenderBlock::layout() src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:1445
    #10 0x7f1342c2011f in WebCore::RenderBlock::layoutBlockChild(WebCore::RenderBox*, WebCore::RenderBlock::MarginInfo&, WebCore::LayoutUnit&, WebCore::LayoutUnit&)
src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:2623
    #11 0x7f1342bde1aa in WebCore::RenderBlock::layoutBlockChildren(bool, WebCore::LayoutUnit&) src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:2560
    #12 0x7f1342bd1dff in WebCore::RenderBlock::layoutBlock(bool, WebCore::LayoutUnit) src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:1638
    #13 0x7f1342bcb344 in WebCore::RenderBlock::layout() src/third_party/WebKit/Source/core/rendering/RenderBlock.cpp:1445
    #14 0x7f134399a1c7 in WebCore::RenderView::layoutContent(WebCore::LayoutState const&) src/third_party/WebKit/Source/core/rendering/RenderView.cpp:130
    #15 0x7f134399f086 in WebCore::RenderView::layout() src/third_party/WebKit/Source/core/rendering/RenderView.cpp:290
    #16 0x7f1346edc6e3 in WebCore::FrameView::layout(bool) src/third_party/WebKit/Source/core/page/FrameView.cpp:1014
    #17 0x7f1340cdb0bf in WebCore::Document::implicitClose() src/third_party/WebKit/Source/core/dom/Document.cpp:2269
    #18 0x7f13469f80a7 in WebCore::FrameLoader::checkCallImplicitClose() src/third_party/WebKit/Source/core/loader/FrameLoader.cpp:713
    #19 0x7f13469f6eff in WebCore::FrameLoader::checkCompleted() src/third_party/WebKit/Source/core/loader/FrameLoader.cpp:656
```

Stack frame index	File	Line	Last revision
0	RenderBox.h	695	140640
1	RenderMedia.cpp	65	83397
…	…	…	…

FindIt: How it works

Suspected CLs	No CL in the regression range changes the crashed files. The result is the blame information. Author: inferno@chromium.org Component: blink Changelist: https://chromium.googlesource.com/chromium/blink.git/+/0d872bd8726bea3e30e8f4c315540e75ecaf625a Time: Thu Jan 24 04:14:26 2013 The CL last changed line 708 of file RenderBox.h, which is stack frame 0. Author: dglazkov@chromium.org Component: blink Changelist: https://chromium.googlesource.com/chromium/blink.git/+/93798a55e44820d9c07913a6f19ee72614e2f067 Time: Sun Apr 10 14:57:13 2011 The CL last changed line 65 of file RenderMedia.cpp, which is stack frame 1. Author: antti@apple.com Component: blink Changelist: https://chromium.googlesource.com/chromium/blink.git/+/61923612b7e0b67cff94284a8dec5c7d0db0e0c6 Time: Tue Dec 04 19:21:09 2007 The CL last changed line 216 of file RenderVideo.cpp, which is stack frame 2. Author: inferno@chromium.org Component: blink Changelist: https://chromium.googlesource.com/chromium/blink.git/+/ec5f95f4822bf4eb61ae690ce8556c44d7a98b7f Time: Tue Aug 21 00:09:47 2012 The CL last changed line 2122 of file RenderBlockLineLayout.cpp, which is stack frame 3. Author: hyatt@apple.com Component: blink Changelist: https://chromium.googlesource.com/chromium/blink.git/+/57927284b48643998ca3f57b4931ee5267386ab4 Time: Wed Oct 06 21:44:02 2010 The CL last changed line 1594 of file RenderBlock.cpp, which is stack frame 4. Author: hyatt Component: blink Changelist: https://chromium.googlesource.com/chromium/blink.git/+/29f11130cbd8dadd0cc78df51bd44b2687a6b06f Time: Tue Apr 29 23:32:54 2003 The CL last changed line 1388 of file RenderBlock.cpp, which is stack frame 5.

ClusterFuzz: Sample

Testcase Report

Google Chrome Canary

Redo + Delete

Timestamp	2014-11-05 23:35:44
Fuzzer	Phoglund_webrtc_peerconnection
Job Type	Linux_asan_chrome_mp
Crash type	Heap-use-after-free READ 8
Crash address	0x604000136c90
Crash state	webrtc::PeerConnection::OnSessionStateChange cricket::BaseSession::SetState webrtc::WebRtcSession::UpdateSessionState
Redzone	16 bytes
Bug Information	Bug 430925 [Remove]
Reproducible	Yes [Mark Unconfirmed]
Security	Yes
Requires	HTTP
Crash revision	Chromium: 302861 Angle: 4de44cb67e9e36966fb1993c0be35659a47182ef Blink: 3ea7c10ceb44449d5d1c8403a5953d4f24bcd8a9 FFmpeg: 4bc3dc1af71c98bb257fba5e442303f68b4ff8fc ICU: d8b2a9d7b0039a4950ee008c5b1d998902c44c60 Libjingle: 19c8d5c35a3e7a7341124f3865a3a117985e7c08 Libvpx: 2e5ced5fd62a73f4f5687ab19520b3aad1c53f6f NaCl: 9cd80947288ebca86dd07b55171089b84553a7c7 Pdfium: 4dc95e74e1acc75f4eab08bc771874cd2a9c3a9b Skia: 76c4fa6e300d08fb1a17cc446735710f2233e15c V8: e806ce15cea9a8706e2457216d6736f386dc6bdf WebRTC: 9a8c28f10f329c5ce91e77057933e60224000627
Regressed	Chromium: 302775:302793 Blink: 62ba796b1b0c462746e0362ccdf08d92f8ff1e34:d53f9960670a7589b8b808c3d5693b92ae4a2dd1
Fixed	Chromium: 302968:303000 Blink: 97cb3f8ea7c73636fdb662a8fe6aef86344e350a:b4bebb047f35d98a11b55053b6619b71f71356d7 Libjingle: 19c8d5c35a3e7a7341124f3865a3a117985e7c08:7d974c11e23898cd59838c79751b96c45b09ec4b Skia: 334355742137cc238887d4f85856e6c4d9ec8aa8:cdccd75925185b25972b8b80f8492bd9d263028d V8: e806ce15cea9a8706e2457216d6736f386dc6bdf:566c2b28c24717da1474a7d32104a81c50433f09 WebRTC: 9a8c28f10f329c5ce91e77057933e60224000627:b831a9e3d5f9f0563d249b726cffa8a070e58aee
Impacts	Head

FindIt: Culprit CL Finder

Suspected CLs	The result is a list of CLs that change the crashed files. Author: tommi Component: chromium Changelist: https://chromium.googlesource.com/chromium/src//+/72eaef099698d09dbf9f3fe3cf5de2e98259239e Time: Wed Nov 05 11:46:18 2014 File rtc_peer_connection_handler.cc is changed in this cl (and is part of stack frame #2, "content::RTCPeerConnectionHandler::~RTCPeerConnectionHandler") File peer_connection_dependency_factory.cc is changed in this cl (and is part of stack frame #4, "content::PeerConnectionDependencyFactory::CreatePeerCon urce"; frame #7, "content::PeerConnectionDependencyFactory::InitializeMediaStreamAudioSource") File rtc_peer_connection_handler.cc is changed in this cl (and is part of stack frame #1, "content::RTCPeerConnectionHandler::initialize") Minimum distance from crash line to modified line: 8. (file: rtc_peer_connection_handler.cc, crashed on: 659, modified: 651).

Filed bug: crbug.com/430925

Issue 430925: Heap-use-after-free in webrtc::PeerConnection::OnSessionStateChange

Status: Fixed
Owner: tommi@chromium.org
Closed: Nov 6

ClusterFuzz
OS-All
Type-Bug-Security
Securit...Severity-High
Security_Impact-Head
Stability-Memory-AddressSanitizer
Merge-NA
M-40

Add a comment and make changes below

Project Member Reported by clusterfuzz@chromium.org, Nov 6, 2014

```
Detailed report: https://cluster-fuzz.appspot.com/testcase?key=6639266019934208

Fuzzer: Phoglund_webrtc_peerconnection
Job Type: Linux_asan_chrome_mp

Crash Type: Heap-use-after-free READ 8
Crash Address: 0x604000136c90
Crash State:
  webrtc::PeerConnection::OnSessionStateChange
  cricket::BaseSession::SetState
  webrtc::WebRtcSession::UpdateSessionState

Regressed: https://cluster-fuzz.appspot.com/revisions?job=linux_asan_chrome_mp&range=302775:302793

Minimized Testcase (15.47 Kb): https://cluster-fuzz.appspot.com/download/AMIfv95uZ8g8bbrFg8ZKudy-
9UQnA8bPYE2gJ2gufyGgUofgHqfpCqhN4ydRuT2SVmq2_v2g7FtbzBYkyjh_7w-W7lQn2MtlY-jWRRZXVYGSA_O8OnKQU6fKX426dnDK-4vkAREtNYqvFBvp84J4UnNLeRB2DLc8cA

Additional requirements: Requires HTTP

Filer: inferno
```

#1 inferno@chromium.org

```
Author: tommi
Component: chromium
Changelist: https://chromium.googlesource.com/chromium/src//+/72eaef099698d09dbf9f3fe3cf5de2e98259239e
Time: Wed Nov 05 11:46:18 2014
File rtc_peer_connection_handler.cc is changed in this cl (and is part of stack frame #2,
"content::RTCPeerConnectionHandler::~RTCPeerConnectionHandler")
File peer_connection_dependency_factory.cc is changed in this cl (and is part of stack frame #4,
"content::PeerConnectionDependencyFactory::CreatePeerConnectionFactory"; frame #5, "content::PeerConnectionDependencyFactory::GetPcFactory";
frame #6, "content::PeerConnectionDependencyFactory::CreateLocalAudioSource"; frame #7,
"content::PeerConnectionDependencyFactory::InitializeMediaStreamAudioSource")
File rtc_peer_connection_handler.cc is changed in this cl (and is part of stack frame #1, "content::RTCPeerConnectionHandler::initialize")
Minimum distance from crash line to modified line: 8. (file: rtc_peer_connection_handler.cc, crashed on: 659, modified: 651).
```

Status: Assigned
Owner: tommi@chromium.org

Patched + Verified: < 1 day

Project Member #5 clusterfuzz@chromium.org Nov 6, 2014 Delete comment

ClusterFuzz has detected this issue as fixed in range 302968:303000.

Detailed report: https://cluster-fuzz.appspot.com/testcase?key=6639266019934208

Fuzzer: Phoglund_webrtc_peerconnection
Job Type: Linux_asan_chrome_mp

Crash Type: Heap-use-after-free READ 8
Crash Address: 0x604000136c90
Crash State:
  webrtc::PeerConnection::OnSessionStateChange
  cricket::BaseSession::SetState
  webrtc::WebRtcSession::UpdateSessionState

Regressed: https://cluster-fuzz.appspot.com/revisions?job=linux_asan_chrome_mp&range=302775:302793
Fixed: https://cluster-fuzz.appspot.com/revisions?job=linux_asan_chrome_mp&range=302968:303000

Minimized Testcase (15.47 Kb): https://cluster-fuzz.appspot.com/download/AMIfv95uZ8g8bbrFg8ZKudy-9UQnA8bPYE2gJ2gufyGgUofgHqfpCqhN4ydRuT2SVmq2_v2g7FtbzBYkyjh_7w-W7lQn2MtlY-jWRRZXVYGSA_O8OnKQU6fKX426dnDK-4vkAREtNYqvFBvp84J4UnNLeRB2DLc8cA

Additional requirements: Requires HTTP

If you suspect that the result above is incorrect, try re-doing that job on the testcase report page.

Project Member #6 clusterfuzz@chromium.org Nov 6, 2014 Delete comment

Adding Merge-Triage label for tracking purposes.

Once your fix had sufficient bake time (on canary, dev as appropriate), please nominate your fix for merge by adding the Merge-Requested label.

When your merge is approved by the release manager, please start merging with higher milestone label first. Make sure to re-request merge for every milestone in the label list. You can get branch information on omahaproxy.appspot.com.

Your fix is very close to the branch point. After the branch happens, please make sure to check if your fix is in.

- Your friendly ClusterFuzz

Labels: -Restrict-View-SecurityTeam Merge-Triage M-40 Restrict-View-SecurityNotify

Thank you