moz://a

The Human Component in Automated Bug Finding

Christian Holler (:decoder)
Staff Security Engineer

Firefox Browser

Get the browser that protects what's important

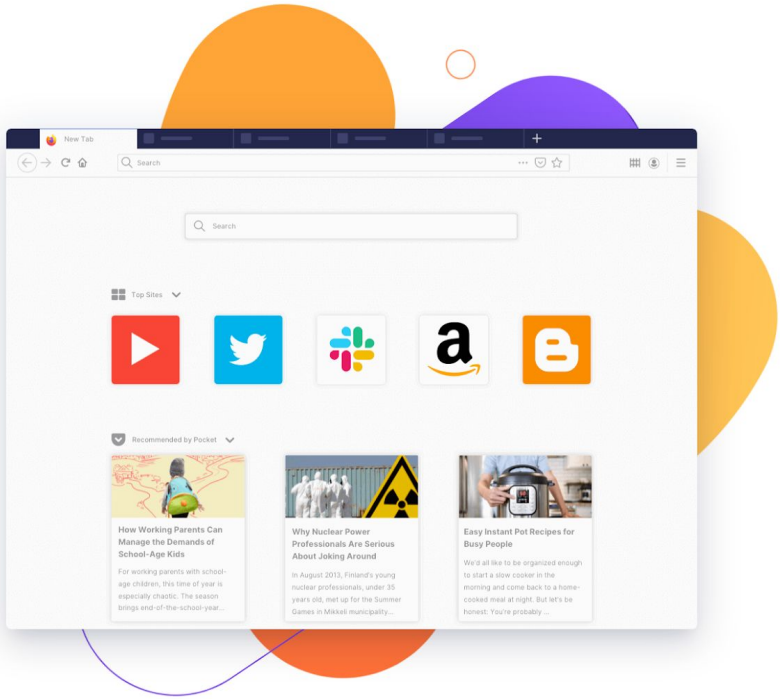
No shady privacy policies or back doors for advertisers. Just a lightning fast browser that doesn't sell you out.

Download Firefox

Firefox Privacy Notice

Download options and other languages

Firefox Browser support



Firefox Browser

Get the browser that protects what's important

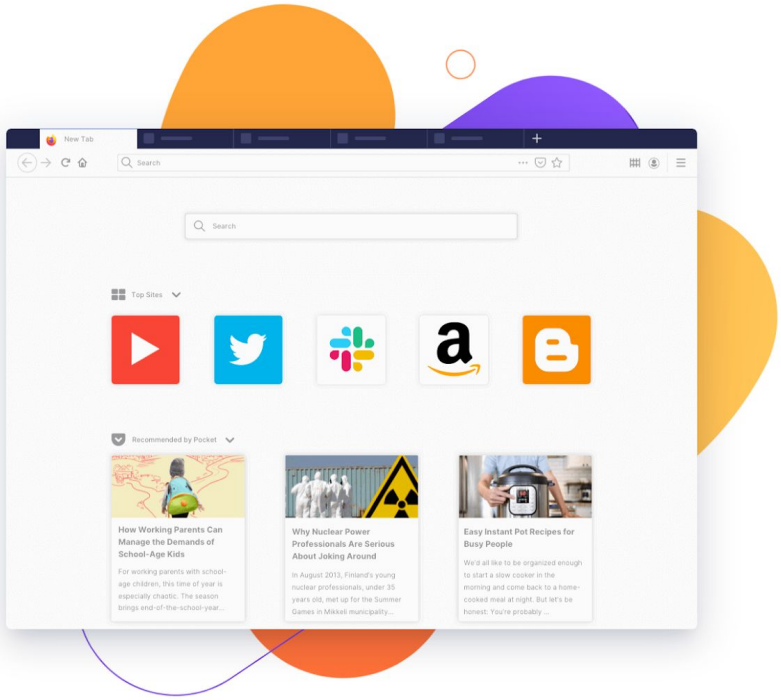
No shady privacy policies or back doors for advertisers. Just a lightning fast browser that doesn't sell you out.

Download Firefox

Firefox Privacy Notice

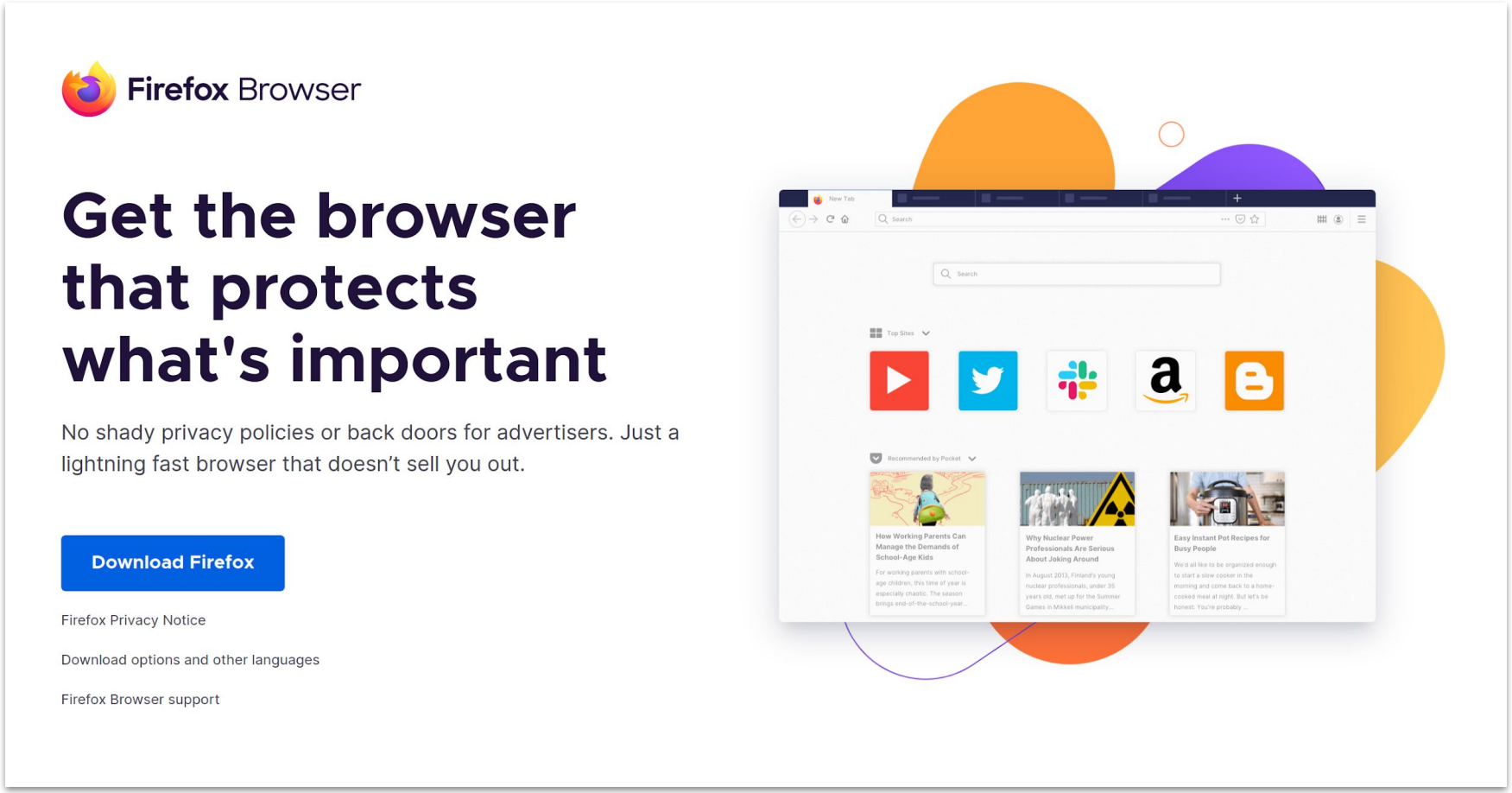
Download options and other languages

Firefox Browser support



~ **16M** lines source code

moz://a



~ **16M** lines source code

Last month: 340 authors with 2,475 commits

moz://a

Interfaces

Media Formats

Markup Languages

Fonts

JavaScript

Networking

Domain Knowledge

Developers know...

Domain Knowledge

Developers know...

... code architecture/contracts

Domain Knowledge

Developers know...

... code architecture/contracts

... expected behavior

Domain Knowledge

Developers know...

... code architecture/contracts

... expected behavior

... weaknesses

Domain Knowledge

Developers know...

... code architecture/contracts

... expected behavior

... weaknesses

"Lone warrior" approach not sustainable

Mutual Trust Relationship

The Do’s and Don’ts

Ninja Style

Ninja Style

Build fuzzer alone and in secret

Ninja Style

Build fuzzer alone and in secret

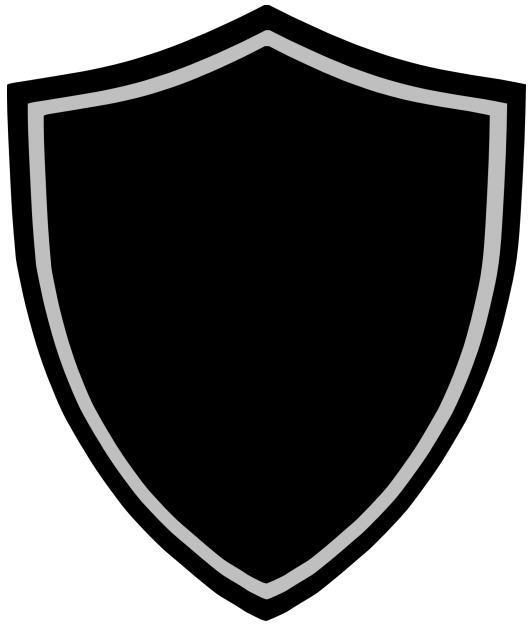
Rapid fire bugs at developers

Ninja Style

Build fuzzer alone and in secret

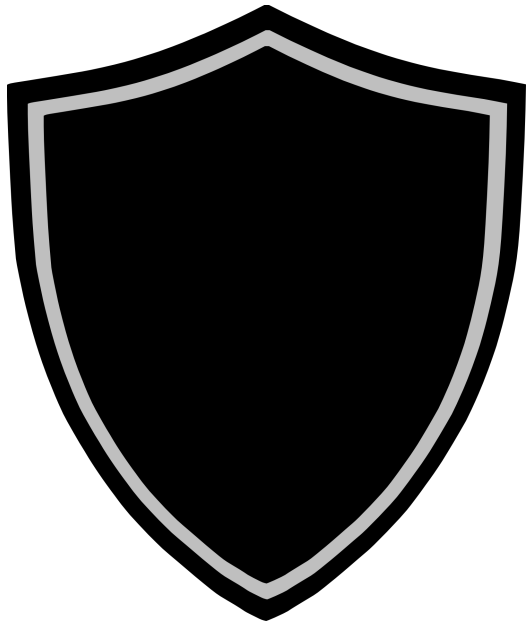
Rapid fire bugs at developers

"They'll never know what hit them. Tehehe!!11oneeleven"



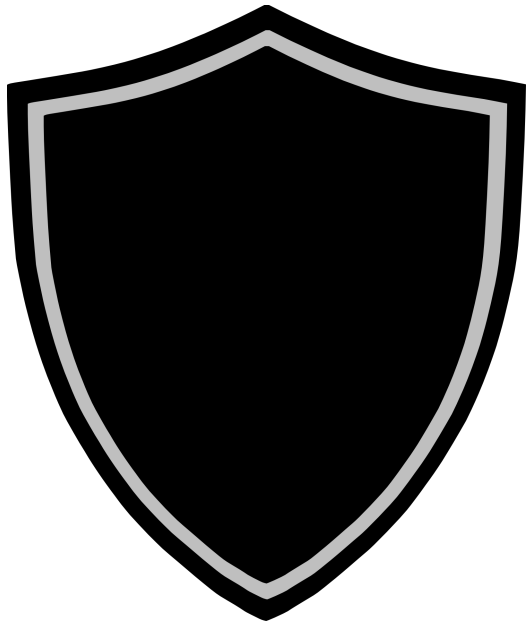
Defensive Behavior

Defensive Behavior



Overwhelmed

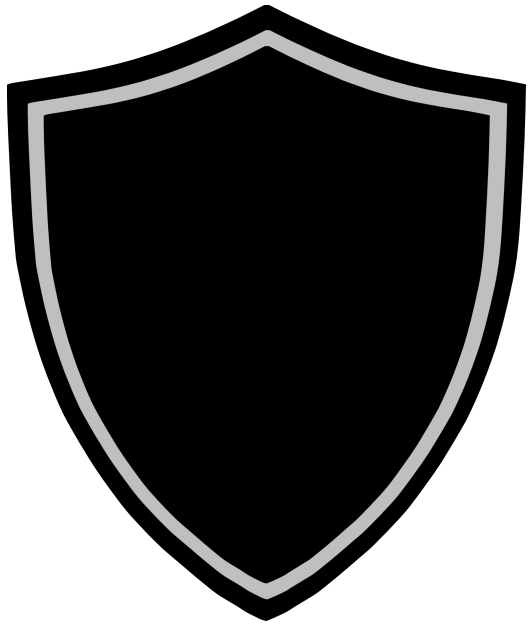
Defensive Behavior



Overwhelmed

Lack of Resources

Defensive Behavior



Overwhelmed

Lack of Resources

Code Ownership Bias

(simplify, reduce-invertible, default) Soundness bug on QF_BV formula (assert (= (bvnand a (bvnand b b)) a)) #4461

New issue

Closed **muchang** opened this issue on May 24 · 16 comments

muchang commented on May 24

Hi,
For this case, Z3 gives an incorrect answer:

```
[527] % z3-4.8.5 small.smt2
unsat
sat
[528] % z3-4.8.6 small.smt2
unsat
sat
[529] % z3-4.8.7 small.smt2
unsat
sat
[530] % z3-4.8.8 small.smt2
unsat
sat
[531] % z3release small.smt2
unsat
sat
[532] %
[532] % cat small.smt2
(declare-fun a () (_ BitVec 1))
(declare-fun b () (_ BitVec 1))
(assert (= (bvnand a (bvnand b b)) a))
(check-sat-using (then simplify reduce-invertible default))
(check-sat)
[533] %
```

OS: Ubuntu 18.04
Commit: `d6ad371`

NikolajBjorner commented on May 24 Contributor

please stop filing fuzz bugs for the next few weeks until they can be addressed.

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

10 participants

(simplify, reduce-invertible, default) Soundness bug on QF_BV formula (assert (= (bvnand a (bvnand b b)) a)) #4461

New issue

Closed **muchang** opened this issue on May 24 · 16 comments

muchang commented on May 24

Hi,
For this case, Z3 gives an incorrect answer:

```
[527] % z3-4.8.5 small.smt2
unsat
sat
[528] % z3-4.8.6 small.smt2
unsat
sat
[529] % z3-4.8.7 small.smt2
unsat
sat
[530] % z3-4.8.8 small.smt2
unsat
sat
[531] % z3release small.smt2
unsat
sat
[532] %
[532] % cat small.smt2
(declare-fun a () (_ BitVec 1))
(declare-fun b () (_ BitVec 1))
(assert (= (bvnand a (bvnand b b)) a))
(check-sat-using (then simplify reduce-invertible default))
(check-sat)
[533] %
```

OS: Ubuntu 18.04
Commit: `d6ad371`

NikolajBjorner commented on May 24 Contributor

please stop filing fuzz bugs for the next few weeks until they can be addressed.

"please stop filing fuzz bugs for the next few weeks until they can be addressed."

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

10 participants

DON'T

Surprise your developers

DON'T

- ✗ Surprise your developers
- ✗ Act superior or adversarial

DON'T

- ✗ Surprise your developers
- ✗ Act superior or adversarial
- ✗ Assume equal priorities

DO:
Kickoff Meeting

DO:
Kickoff Meeting

\+

Developers, Fuzzing
and Management

moz://a

DO:
Kickoff Meeting

+ Developers, Fuzzing **and** Management

+ Show previous success stories

DO:
Kickoff Meeting

+ Developers, Fuzzing **and** Management

+ Show previous success stories

+ Offer your help, ask about problems

DO:
Kickoff Meeting

+ Developers, Fuzzing **and** Management

+ Show previous success stories

+ Offer your help, ask about problems

+ Define Goals - Allocate Resources

DO:
Kickoff Meeting

- Developers, Fuzzing **and** Management
- Show previous success stories
- Offer your help, ask about problems
- Define Goals - Allocate Resources
- Educate on Requirements

DO:
Kickoff Meeting

+ Developers, Fuzzing **and** Management

+ Show previous success stories

+ Offer your help, ask about problems

+ Define Goals - Allocate Resources

+ Educate on Requirements

DO:
Kickoff Meeting

+ Developers, Fuzzing **and** Management

+ Show previous success stories

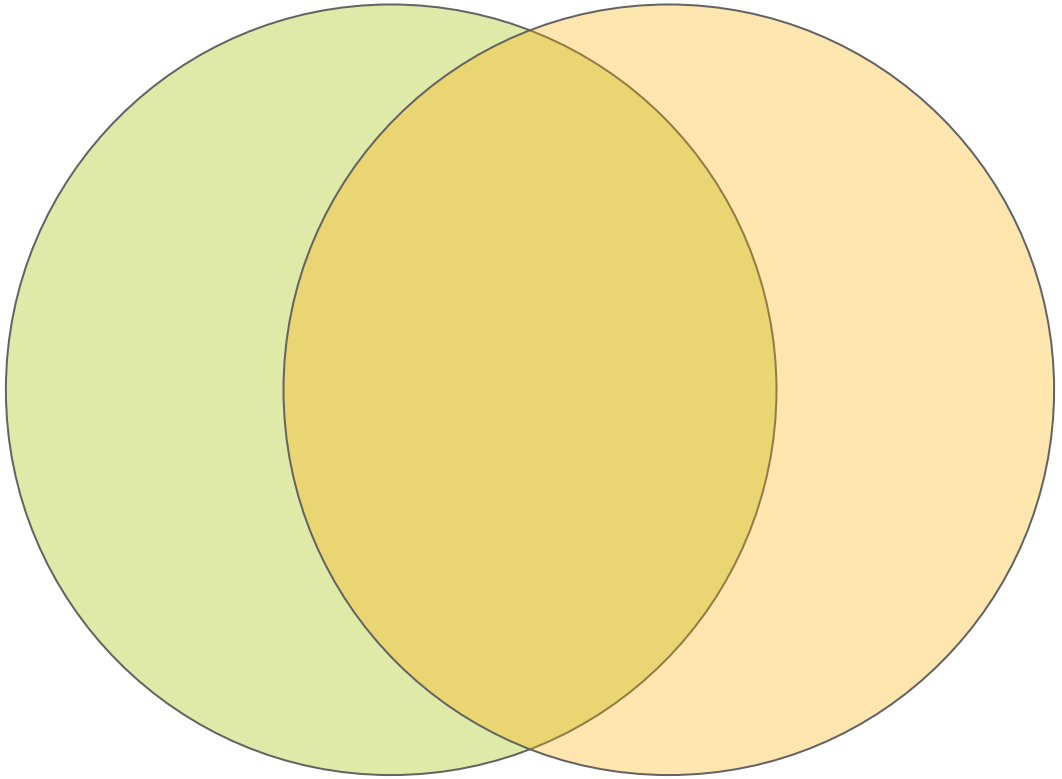
+ Offer your help, ask about problems

+ Define Goals - Allocate Resources

+ Educate on Requirements

Requirements and Goals

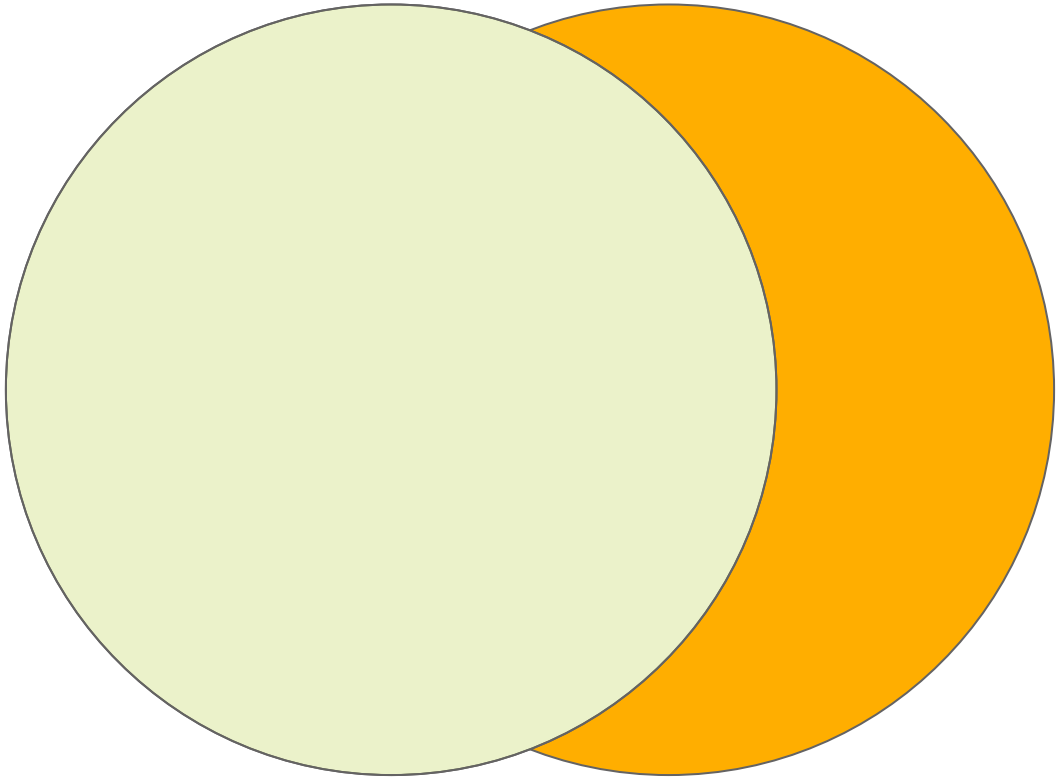
Fuzzing



Developer

Requirements and Goals

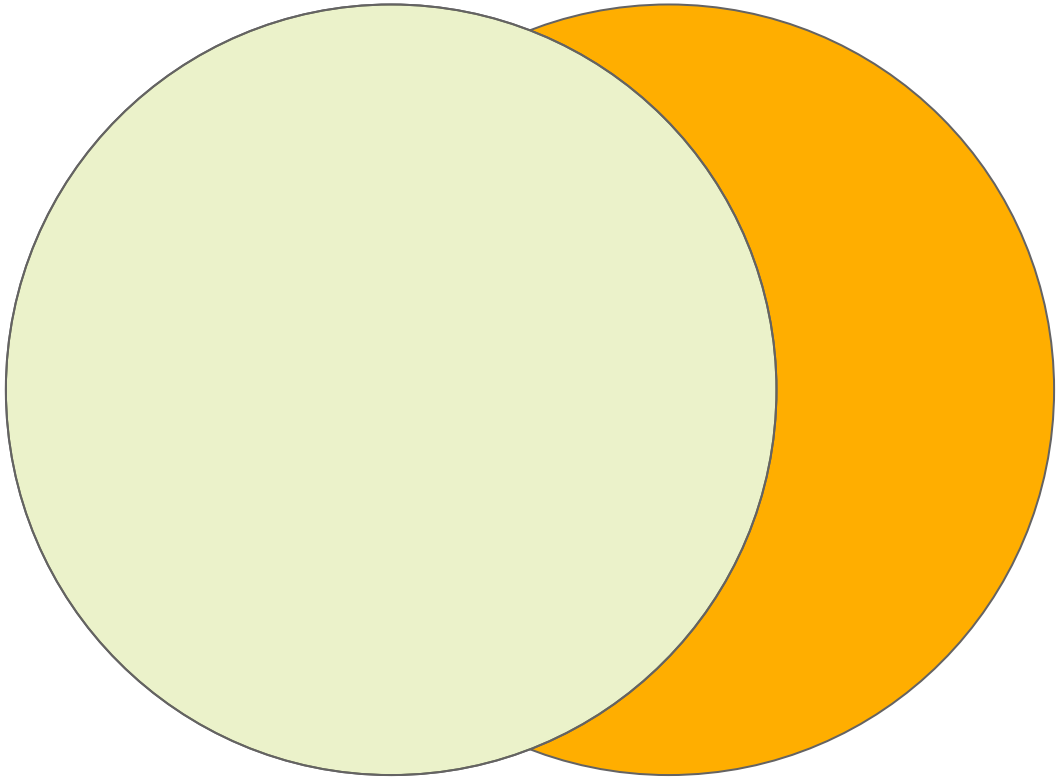
Fuzzing



Developer

Requirements and Goals

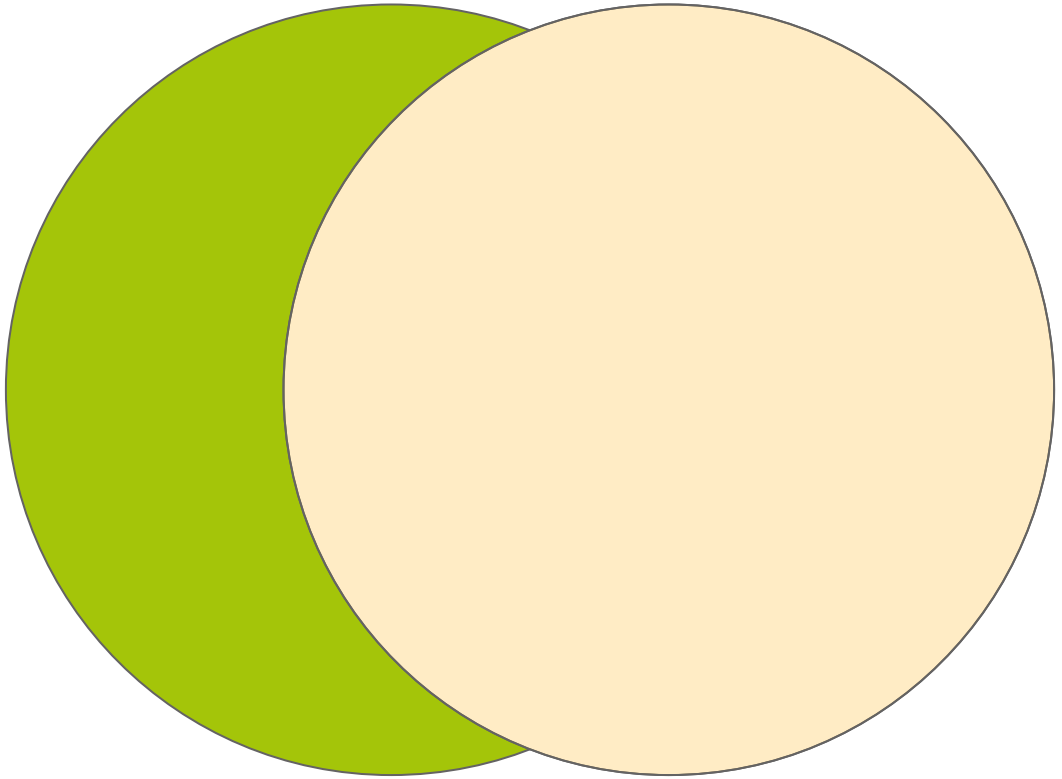
Fuzzing



Developer

Requirements and Goals

Fuzzing



Developer

Example Requirement

Bugs must be fixed

Example Requirement

Bugs must be fixed

"That bug isn't interesting, please ignore it."

Example Requirement

Bugs must be fixed

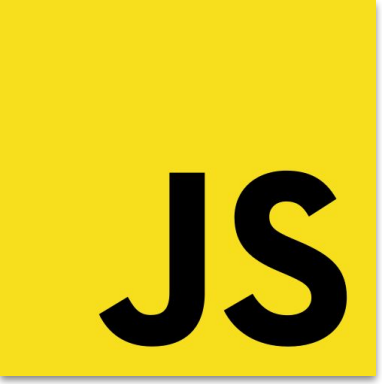
"... but that's not a bug."

Example Requirement

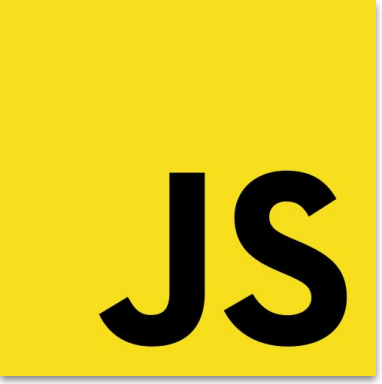
Bugs must be fixed

"... but that's not a bug."

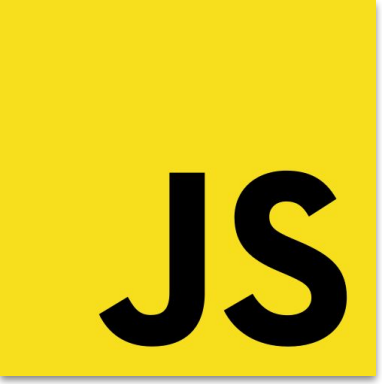
"Contract" about what constitutes a bug



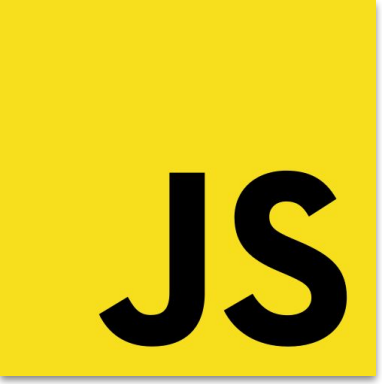

```
$ js
js>
```



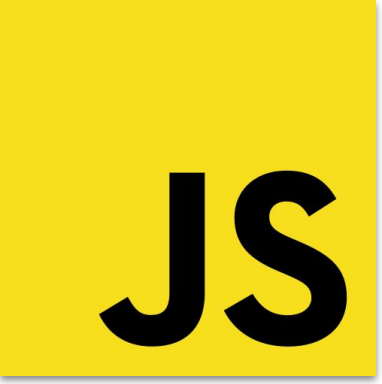

```
$ js
js> print("Hello watman")
Hello watman
js>
```




```
$ js
js> print("Hello watman")
Hello watman
js> crash();
Hit MOZ_CRASH(forced crash) at shell/js.cpp:3700
Segmentation fault
```



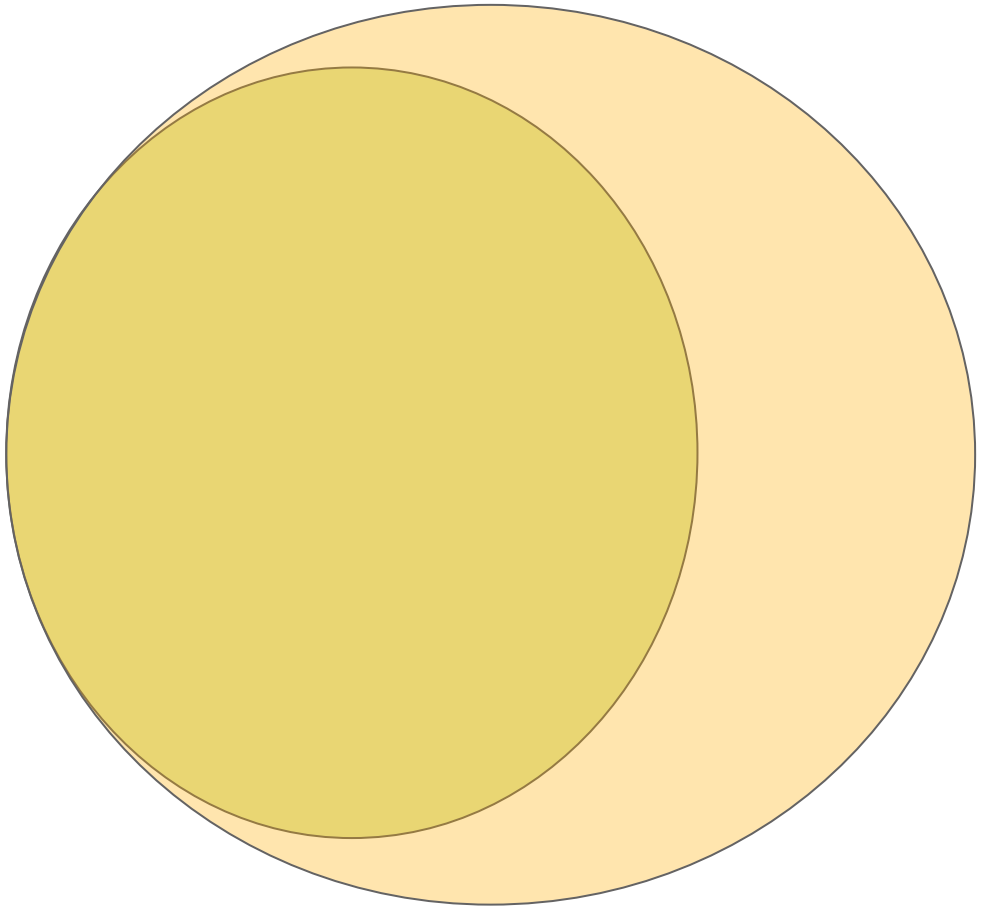

```
$ js --fuzzing-safe
js>
```




```
$ js --fuzzing-safe
js> crash();
typein:1:1 ReferenceError: crash is not defined
```

Requirements vs. Goals

Fuzzing



Developer

Fuzzblockers

Fuzzblockers

Disruptive effect on fuzzing operations

Fuzzblockers

Disruptive effect on fuzzing operations

(e.g. highly frequent, resource intensive)

\+ Hard to avoid

Fuzzblockers

Disruptive effect on fuzzing operations

(e.g. highly frequent, resource intensive)

\+ Hard to avoid

Highest Priority for Fuzzing

(Usually) low priority for developers

Fuzzblockers

Disruptive effect on fuzzing operations

(e.g. highly frequent, resource intensive)

\+ Hard to avoid

Highest Priority for Fuzzing

(Usually) low priority for developers

Try writing a fix **yourself**!

moz://a

I want **you** to fix a bug

I want **you** to fix a bug

You learn something
about the code

I want **you** to fix a bug

You learn something
about the code

You learn something
about development

I want **you** to fix a bug

You learn something
about the code

You learn something
about development

You can progress faster

I want **you** to fix a bug

You learn something
about the code

You learn something
about development

You can progress faster

Developers will be happy

When?

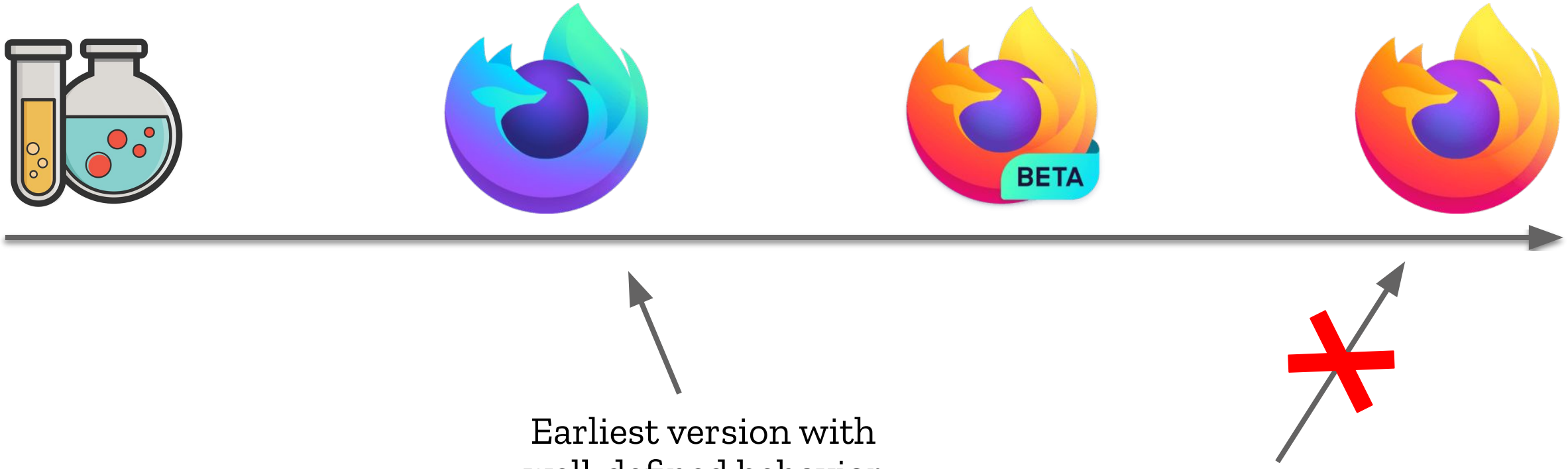
When to Fuzz?

When to Fuzz?

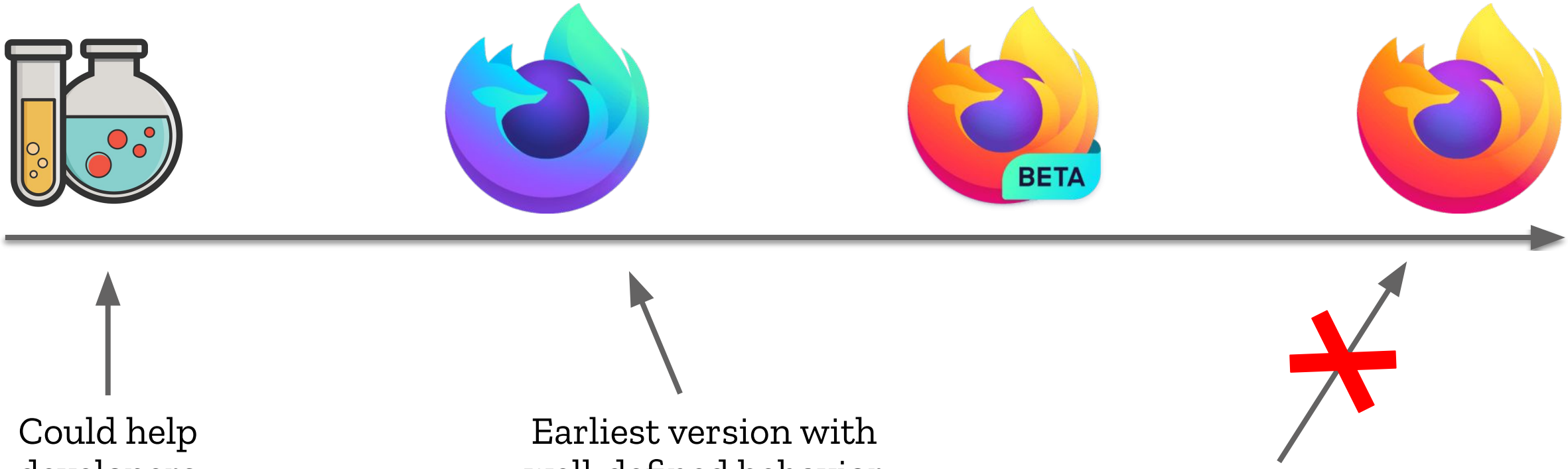
When to Fuzz?

When to Fuzz?

When to Fuzz?



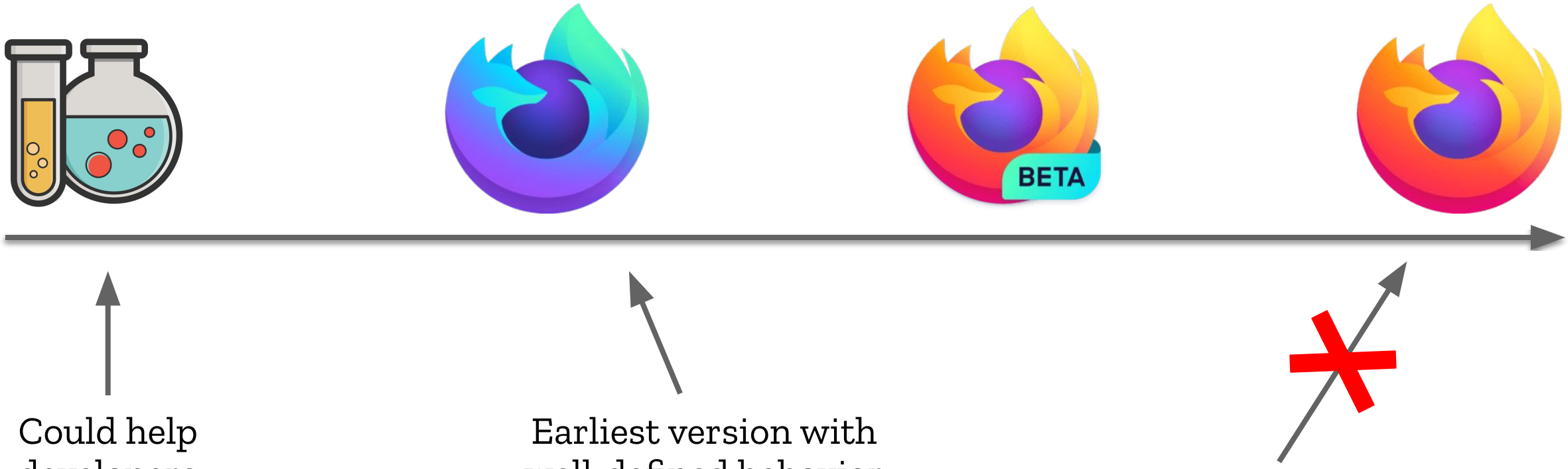
When to Fuzz?



Could help developers

Earliest version with well-defined behavior

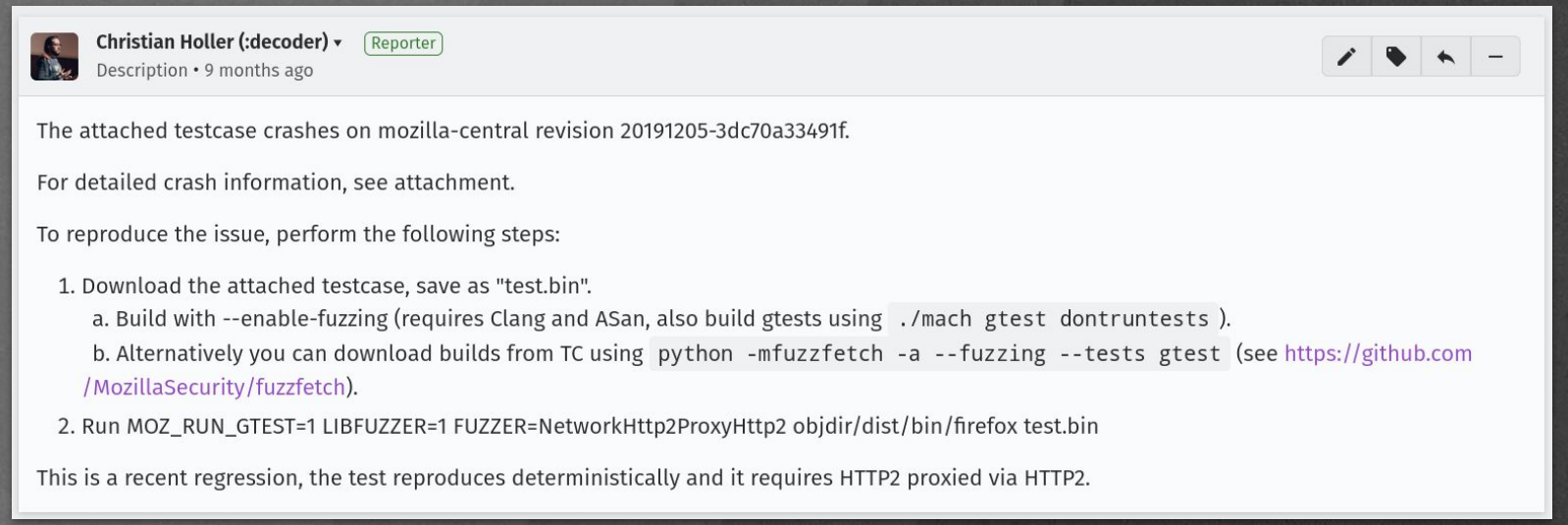
When to Fuzz?



As early as possible (*)

DO: Simple Steps to reproduce

DO: Simple Steps to reproduce



Christian Holler (:decoder) ▾ Reporter
Description • 9 months ago

The attached testcase crashes on mozilla-central revision 20191205-3dc70a33491f.

For detailed crash information, see attachment.

To reproduce the issue, perform the following steps:

1. Download the attached testcase, save as "test.bin".
   a. Build with --enable-fuzzing (requires Clang and ASan, also build gtests using `./mach gtest dontruntests`).
   b. Alternatively you can download builds from TC using `python -mfuzzfetch -a --fuzzing --tests gtest` (see https://github.com/MozillaSecurity/fuzzfetch).
2. Run MOZ_RUN_GTEST=1 LIBFUZZER=1 FUZZER=NetworkHttp2ProxyHttp2 objdir/dist/bin/firefox test.bin

This is a recent regression, the test reproduces deterministically and it requires HTTP2 proxied via HTTP2.

DO: Measure Code Coverage

```
2432                    case SCTAG_BOOLEAN:
2433                    case SCTAG_BOOLEAN_OBJECT:
2434  537,423,996         vp.setBoolean(!!data);
2435  537,423,996         if (tag == SCTAG_BOOLEAN_OBJECT && !PrimitiveToObject(context(), vp)) {
2436                        return false;
2437                      }
2438                      break;
2439
2440                    case SCTAG_STRING:
2441                    case SCTAG_STRING_OBJECT: {
2442  6,939,712,734       JSString* str = readString(data);
2443  6,939,712,707       if (!str) {
2444                        return false;
2445                      }
2446                      vp.setString(str);
2447  6,939,712,707       if (tag == SCTAG_STRING_OBJECT && !PrimitiveToObject(context(), vp)) {
2448                        return false;
2449                      }
2450                      break;
2451                    }
2452
2453                    case SCTAG_NUMBER_OBJECT: {
2454                      double d;
2455                      if (!in.readDouble(&d)) {
2456                        return false;
2457                      }
2458                      vp.setDouble(CanonicalizeNaN(d));
2459                      if (!PrimitiveToObject(context(), vp)) {
2460                        return false;
2461                      }
2462                      break;
2463                    }
2464
2465                    case SCTAG_BIGINT:
2466                    case SCTAG_BIGINT_OBJECT: {
2467                      RootedBigInt bi(context(), readBigInt(data));
2468                      if (!bi) {
2469                        return false;
2470                      }
2471                      vp.setBigInt(bi);
2472                      if (tag == SCTAG_BIGINT_OBJECT && !PrimitiveToObject(context(), vp)) {
2473                        return false;
2474                      }
2475                      break;
2476                    }
```

moz://a

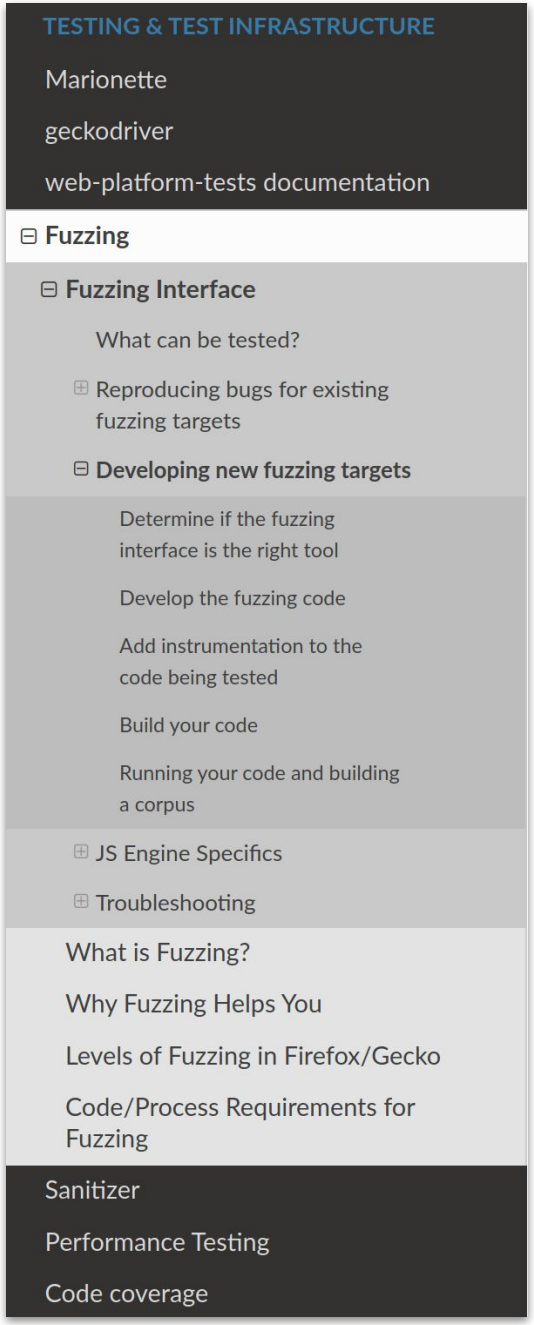
DO: Measure Code Coverage

```
2432                  case SCTAG_BOOLEAN:
2433                  case SCTAG_BOOLEAN_OBJECT:
2434  537,423,996       vp.setBoolean(!!data);
2435  537,423,996       if (tag == SCTAG_BOOLEAN_OBJECT && !PrimitiveToObject(context(), vp)) {
2436                      return false;
2437                    }
2438                    break;
2439
2440                  case SCTAG_STRING:
2441                  case SCTAG_STRING_OBJECT: {
2442  6,939,712,734     JSString* str = readString(data);
2443  6,939,712,707     if (!str) {
2444                      return false;
2445                    }
2446                    vp.setString(str);
2447  6,939,712,707     if (tag == SCTAG_STRING_OBJECT && !PrimitiveToObject(context(), vp)) {
2448                      return false;
2449                    }
2450                    break;
2451                  }
2452
2453                  case SCTAG_NUMBER_OBJECT: {
2454                    double d;
2455                    if (!in
2456                      re    fa
2457                    }
2458                    vp.set ble(CanonicalizeNaN(d));
2459                    if (!PrimitiveToObject(context(), vp)) {
2460                      return false;
2461                    }
2462                    break;
2463                  }
2464
2465                  case SCTAG_BIGINT:
2466                  case SCTAG_BIGINT_OBJECT: {
2467                    RootedBigInt bi(context(), readBigInt(data));
2468                    if (!bi) {
2469                      return false;
2470                    }
2471                    vp.setBigInt(bi);
2472                    if (tag == SCTAG_BIGINT_OBJECT && !PrimitiveToObject(context(), vp)) {
2473                      return false;
2474                    }
2475                    break;
2476                  }
```

SHARE!

moz://a

DO: Educate



moz://a

Fuzzing is Teamwork

moz://a

Thank You