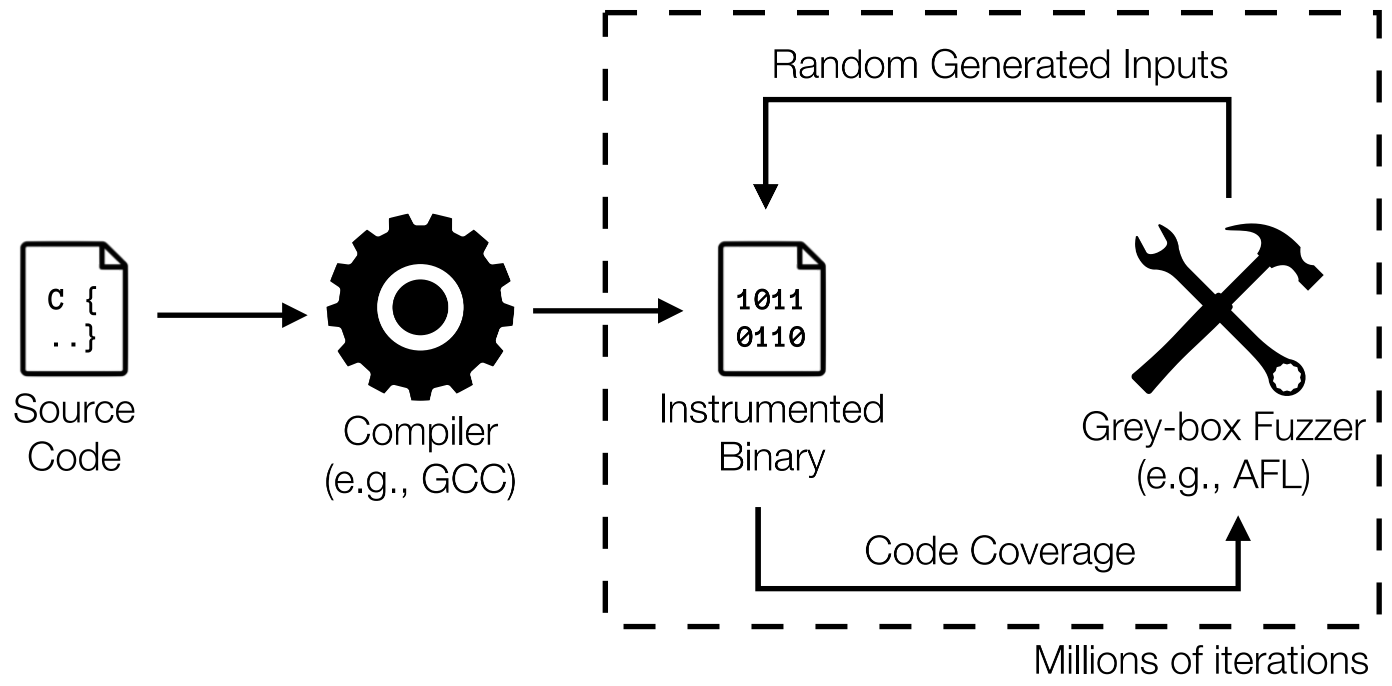
STOCHFUZZ: Sound and Cost-effective Fuzzing of Stripped Binaries by Incremental and Stochastic Rewriting

PURDUE UNIVERSITY

Zhuo Zhang, Wei You, Guanhong Tao, Yousra Aafer, Xuwei Liu, Xiangyu Zhang

Grey-box Fuzzing



Grey-box fuzzing found:

- More than 21,000 bugs in the Chromium projects
- More than 16,000 bugs in other open source projects

What if the source code is not available (e.g., close-sourced programs)?

Existing Solutions for Binary-only Fuzzing

QEMU: *Dynamic Binary Translation* (afl-qemu): translate a subject binary during its execution. It is sound but expensive (high overhead >600%).

intel: *Hardware-Assisted Tracing* (ptfuzzer): make use of advanced hardware support such as Intel PT to collect runtime traces that can be post-processed (relatively high overhead and only coverage-based feedback).

Static Binary Instrumentation (e9patch and ddisasm): leverage advanced binary analysis to directly instrument binaries (cost-effective but usually unsound).

Observation

Fuzzing is a highly repetitive process that provides *a large number* of opportunities for *trial-and-error*.

We can try different data and code separations, which lead to *different instrumented executables*, in different fuzzing runs.

Static Binary Instrumentation with Error Awareness

original space

```
0 :  mov rbx, [14] // rbx=14
8 :  add rbx, 14    // rbx=22
12:  jmp rbx
14:  .qword 8
22:  syscall
```

(a) example code

original space / *shadow space*

```
0 :  jmp 90      ->  90 :  [AFL Tracing]
8 :  hlt             100:  mov rbx, [14]
12:  hlt             108:  add rbx, 14
14:  .qword 8        112:  jmp rbx
22:  jmp 114     ->  114:  [AFL Tracing]
                     124:  syscall
```

(b) ideal case

```
0 :  jmp 90
8 :  hlt
12 : hlt
14 : .qword 8
22 : hlt
. . .
90 : [AFL Tracing]
100: mov rbx, [14]
108: add rbx, 14
112: jmp rbx
```

FNs are detected by intentional crashes which are triggered by inserted hlt instructions.

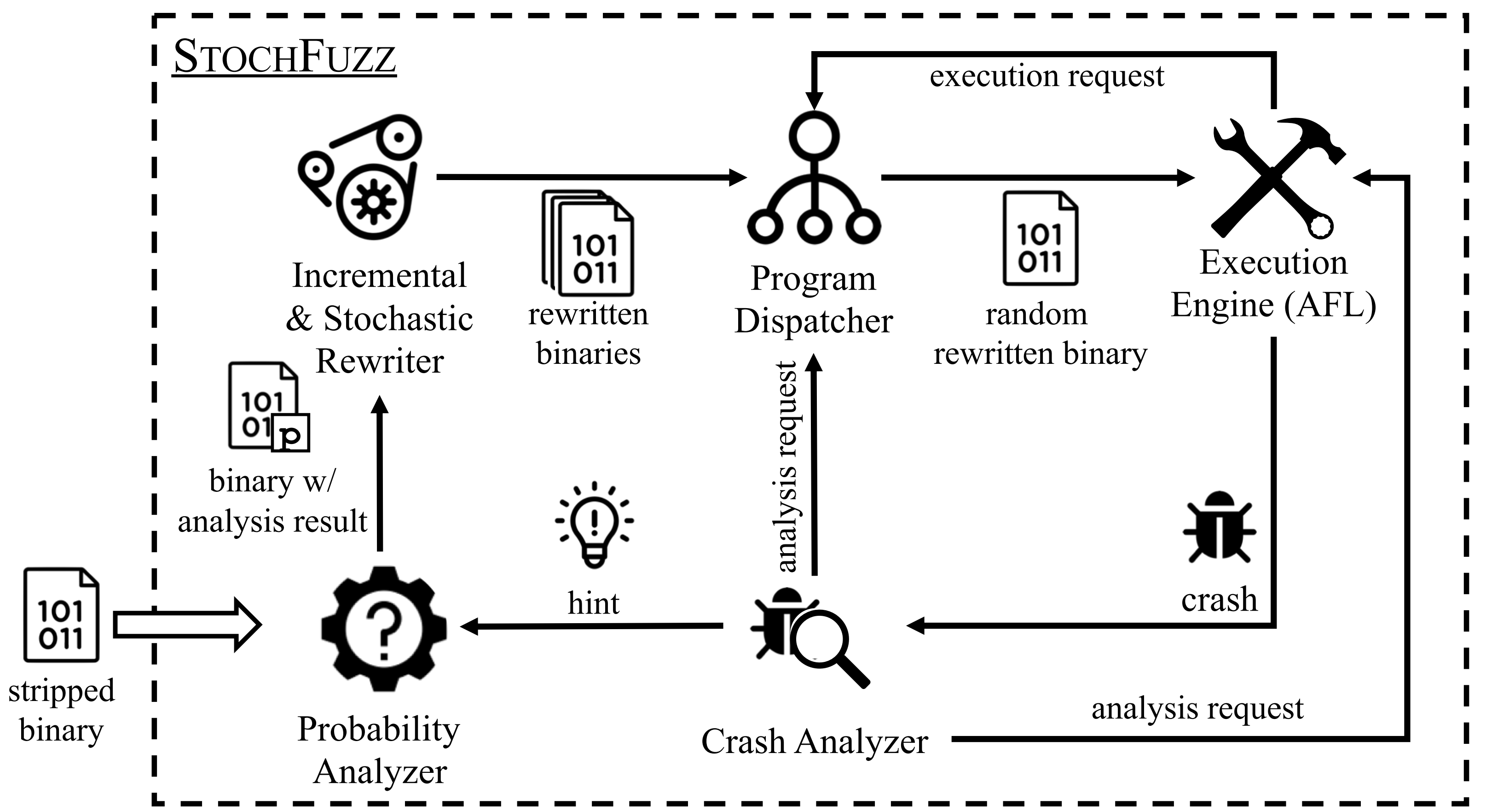
(c) false negative (regarding code as data)

```
0 :  jmp 90
8 :  hlt
12 : hlt
14 : hlt
22 : hlt
. . .
90 : [AFL Tracing]
100: mov rbx, [14]
108: add rbx, 14
112: jmp rbx
114: [AFL Tracing]
124: syscall
```

FPs are detected by unintentional crashes which are not triggered by inserted hlt instructions.

(d) false positive (regarding data as code)

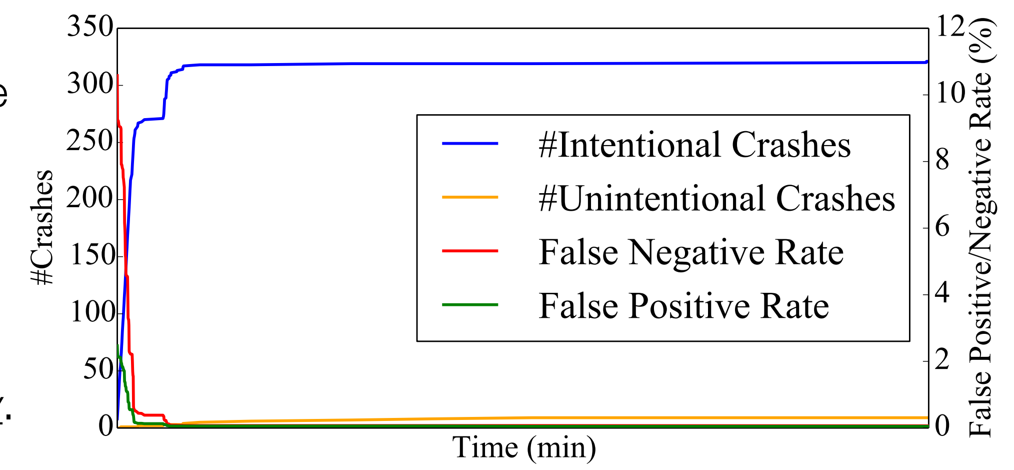
Architecture of STOCHFUZZ



Progress of Incremental and Stochastic Rewriting

Over time, an increasing number of samples can be collected, allowing us to achieve the *precise* separation and *correct* rewriting.

Observe the rewriting process *converges quickly*.

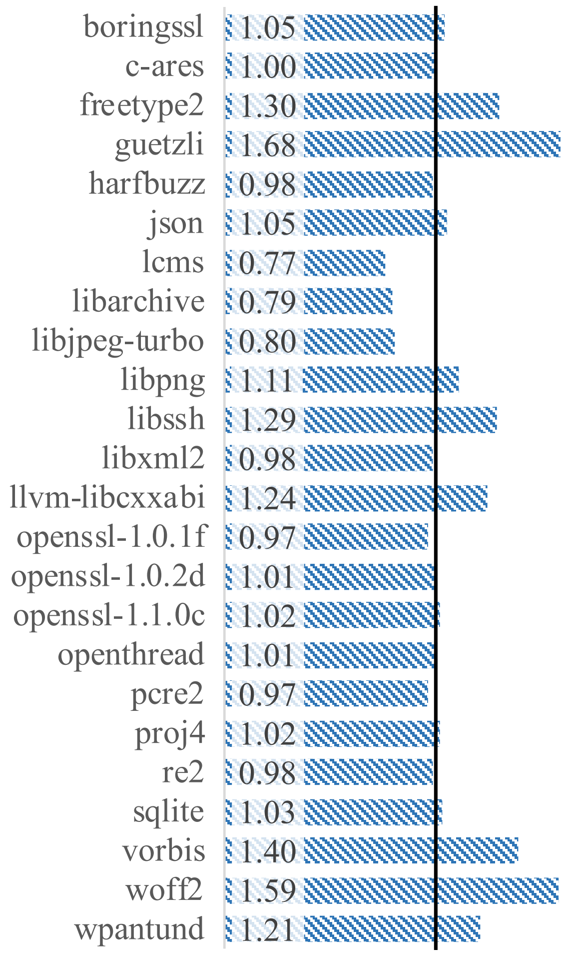


Evaluation

(a) Mean and standard deviation of time-to-discover (in minutes) for bugs in Google FTS

Tool	guetzli	json	llvm-libcxxabi	pcre2	re2	woff2
afl-gcc	513.25 ± 114.84	0.85 ± 0.63	0.08 ± 0.00	763.61 ± 40.44	2.21 ± 2.14	12.89 ± 0.44
afl-clang-fast	539.56 ± 240.83	0.18 ± 0.17	0.08 ± 0.00	461.73 ± 219.89	3.08 ± 3.93	12.09 ± 4.91
afl-qemu	$+\infty$	2.64 ± 3.56	0.23 ± 0.05	$+\infty$	$+\infty$	67.23 ± 26.94
ptfuzzer	$+\infty$	49.08 ± 82.35	0.79 ± 0.25	$+\infty$	42.92 ± 68.08	29.18 ± 0.19
e9patch	$+\infty$	21.87 ± 36.21	0.35 ± 0.00	$+\infty$	$+\infty$	30.73 ± 0.28
ddisasm	505.22 ± 93.45	N/A	0.08 ± 0.00	913.90 ± 495.42	N/A	14.60 ± 0.25
STOCHFUZZ	363.37 ± 120.14	0.67 ± 1.02	0.08 ± 0.00	768.91 ± 264.82	2.32 ± 0.54	7.43 ± 0.27

(b) Evaluation on Google FTS w/ intentional data Inlining (by mixing .text and .rodata)



Program	# Data Bytes	*e9patch*	*ddisasm*	STOCHFUZZ
boringssl	263,539	✗	✗	✓
c-ares	7	✗		✓
freetype2	91,960	✗	✗	✓
guetzli	18,543	✗	✗	✓
harfbuzz	63,061	✗	✗	✓
json	0			✓
lcms	22,576	✗	✗	✓
libarchive	55,698	✗	✗	✓
libjpeg-turbo	79,329	✗	✗	✓
libpng	9,054	✗	✗	✓
libssh	141,943	✗	✗	✓
libxml2	128,007	✗	✗	✓
llvm-libcxxabi	0			✓
openssl-1.0.1f	169,787	✗	✗	✓
openssl-1.0.2d	43,796	✗	✗	✓
openssl-1.1.0c	369,397	✗	✗	✓
openthread	32,691	✗	✗	✓
pcre2	95,763	✗	✗	✓
proj4	30,978	✗	✗	✓
re2	35,336	✗	✗	✓
sqlite	35,467	✗	✗	✓
vorbis	59,986	✗	✗	✓
woff2	494,994	✗	✗	✓
wpantund	89,203	✗	✗	✓

Program	Size	MD5	Status
CUDA	33M	EDAF12B5	Bug ID 3190151 (FIXED)
PNGOUT	89K	64F6899D	CVE-2020-29384

(c) Zero-day vulnerabilities in close-sourced softwares