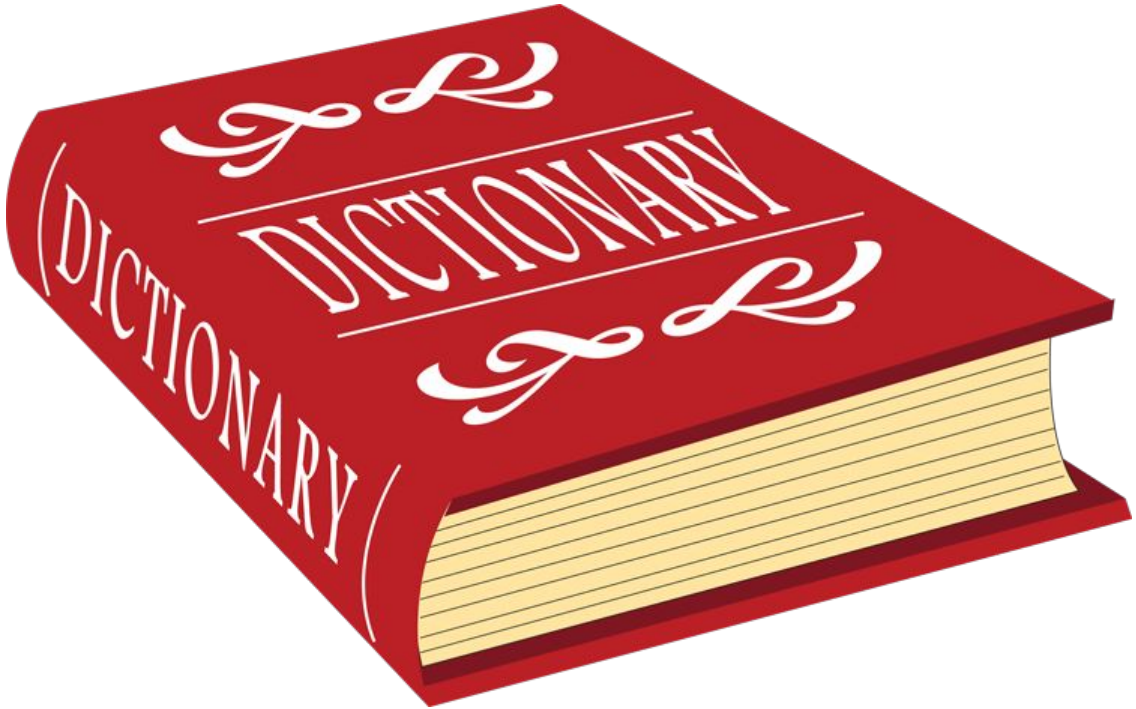
How to improve your fuzzer

- Dictionaries
- Seed corpus
- Custom options
- Fuzzing of non-raw data arguments
- Optimization

Dictionaries



Dictionaries

XML:

```
" encoding=\"1\""
" a=\"1\""
" href=\"1\""
" standalone=\"no\""
" version=\"1\""
" xml:base=\"1\""
" xml:id=\"1\""
" xml:lang=\"1\""
" xml:space=\"1\""
" xmlns=\"1\""
"&lt;"
"&#1;"
"&a;"
"&#x1;"
"ANY"
"[]"
"CDATA"
":fallback"
":include"
"--"
"EMPTY"
"\"\""
"'"
```

PNG:

```
header_png="\x89PNG\x0d\x0a\x1a\x0a
"
section_IDAT="IDAT"
section_IEND="IEND"
section_IHDR="IHDR"
section_PLTE="PLTE"
section_bKGD="bKGD"
section_cHRM="cHRM"
section_fRAc="fRAc"
section_gAMA="gAMA"
section_gIFg="gIFg"
section_gIFt="gIFt"
section_gIFx="gIFx"
section_hIST="hIST"
section_iCCP="iCCP"
section_iTXt="iTXt"
section_oFFs="oFFs"
section_pCAL="pCAL"
section_pHYs="pHYs"
section_sBIT="sBIT"
section_sCAL="sCAL"
section_sPLT="sPLT"
section_sRGB="sRGB"
```

Dictionary generation

Fuzzer	Dict. size	Corpus, %	Coverage, %	Speed, %
expat_xml_parse_fuzzer	428	-3.68	-2.72	1.59
libxml_xml_read_memory_fuzzer	780	30.81	41.14	24.13
net_ftp_directory_listing_fuzzer	842	1.84	1.03	-16.02
net_http_proxy_client_socket_fuzzer	2,006	**303.90**	26.36	-0.74
net_http_stream_parser_fuzzer	2,006	**259.34**	31.16	3.96
net_url_request_fuzzer	1,239	**127.44**	6.78	4.86
net_websocket_frame_parser_fuzzer	925	3.85	0.35	0.77
pdf_css_fuzzer	2,037	-7.56	-0.22	-1.79
pdf_xml_fuzzer	927	8.04	0.74	4.52
sqlite3_prepare_v2_fuzzer	657	**370.04**	**201.68**	-11.21
url_parse_fuzzer	650	**473.33**	**325.07**	-11.70
v8_script_parser_fuzzer	1,535	25.63	2.31	9.27

```
$ chromium/src/testing/libfuzzer/dictionary_generator.py \
        --fuzzer PATH_TO_FUZZER_BINARY \
        --spec PATH_TO_FORMAT_SPECIFICATION \
        --out GENERATED_DICTIONARY.dict
```

Recommended dictionary

```
Command: ['<...>/boringssl_read_pem_fuzzer', <...>]
Bot: clusterfuzz-linux-pre-0234
Time ran: 3251.046254

INFO: Seed: 1441206595
#0   READ   units: 1166 exec/s: 0
#1167     INITED cov: 204 bits: 366 indir: 16 units: 119 exec/s: 0
#1168     NEW    cov: 277 bits: 366 indir: 22 units: 120 exec/s: 0 L: 129
MS: 1 InsertByte-

<...>

#147891067     DONE   cov: 278 bits: 370 indir: 22 units: 123 exec/s:
45490
###### Recommended dictionary. ######
"(:(\xe7\x9d/!;!;O" # Uses: 8451959
"-----BEGIN " # Uses: 8454876
###### End of recommended dictionary. ######
Done 147891067 runs in 3251 second(s)
stat::number_of_executed_units: 147891067
stat::average_exec_per_sec:     45490
<...>
```

Recommended dictionary

```
Command: ['<...>/net_url_request_fuzzer, <...>]
Bot: clusterfuzz-linux-high-end-pre-0034
Time ran: 3251.478762

Dictionary: 128 entries
INFO: Seed: 964796221
#0   READ   units: 10055 exec/s: 0
#128 pulse  cov: 5322 bits: 7463 indir: 415 units: 10055 exec/s: 64


<...>

#366810   DONE   cov: 12167 bits: 55024 indir: 756 units: 10093 exec/s:
112
###### Recommended dictionary. ######
"blob" # Uses: 3996
"http" # Uses: 2420
"file" # Uses: 1536
"foo" # Uses: 1105
"ntlm" # Uses: 922
###### End of recommended dictionary. ######
<...>
```

Recommended dictionary

```
Command: ['<...>/net_parse_cookie_line_fuzzer', <...>]
Bot: clusterfuzz-linux-high-end-pre-0080
Time ran: 3251.114852

INFO: Seed: 1831825350
#0   READ   units: 1142 exec/s: 0
#1142     INITED cov: 318 bits: 1041 indir: 35 units: 332 exec/s: 0
#1143     NEW    cov: 427 bits: 1041 indir: 41 units: 333 exec/s: 0 L: 29
MS: 1 ChangeByte-
<...>
#6808457  NEW    cov: 428 bits: 1045 indir: 41 units: 337 exec/s: 11272
L: 698 MS: 5
ChangeByte-AddFromTempAutoDict-ShuffleBytes-AddFromTempAutoDict-CrossOver
- DE: "httponly"-"path"-
<...>
#36643517 DONE   cov: 428 bits: 1045 indir: 41 units: 337 exec/s: 11271
###### Recommended dictionary. ######
"httponly" # Uses: 1917525
"path" # Uses: 1942187
###### End of recommended dictionary. ######
<...>
```

Seed corpus

```
$ ls testing/libfuzzer/fuzzers/woff2_corpus/
AhemSpaceLigature.woff2     DejaVuSerif.woff2     MEgalopolisExtra.woff2
Ahem.woff2                  EzraSIL.woff2         mplus-1p-regular.woff2
DejaVuSerif-webfont.woff2   LinLibertineO.woff2   OpenSans.woff2
tcu-font.woff2
```

Seed corpus

```
$ ls testing/libfuzzer/fuzzers/woff2_corpus/
AhemSpaceLigature.woff2    DejaVuSerif.woff2    MEgalopolisExtra.woff2
Ahem.woff2                 EzraSIL.woff2        mplus-1p-regular.woff2
DejaVuSerif-webfont.woff2  LinLibertineO.woff2  OpenSans.woff2
tcu-font.woff2
```

```
fuzzer_test("convert_woff2ttf_fuzzer") {
 sources = [
  "convert_woff2ttf_fuzzer.cc",
 ]
 deps = [
  "//third_party/woff2:woff2_dec",
 ]
 seed_corpus = "//testing/libfuzzer/fuzzers/woff2_corpus"
 libfuzzer_options = [ "max_len=803500" ]
}
```

Custom options

```
fuzzer_test("convert_woff2ttf_fuzzer") {
  sources = [
    "convert_woff2ttf_fuzzer.cc",
  ]
  deps = [
    "//third_party/woff2:woff2_dec",
  ]
  seed_corpus = "//testing/libfuzzer/fuzzers/woff2_corpus"
  libfuzzer_options = [ "max_len=803500" ]
}
```

Custom options

Summary + Labels ▾
Crash in ReconstructTransformedHmtx Reproducible ClusterFuzz
Timeout and memory leaks in woff2::ConvertWOFF2ToTTF()
Heap-buffer-overflow in Read Reproducible Clusterfuzz

```
fuzzer_test("convert_woff2ttf_fuzzer") {
 sources = [
  "convert_woff2ttf_fuzzer.cc",
 ]
 deps = [
  "//third_party/woff2:woff2_dec",
 ]
 seed_corpus = "//testing/libfuzzer/fuzzers/woff2_corpus"
 libfuzzer_options = [ "max_len=803500" ]
}
```

Optimization [initial version]

```
#include "third_party/zlib/zlib.h"

// Entry point for LibFuzzer.
extern "C" int LLVMFuzzerTestOneInput(const unsigned char *data, size_t size) {
    uint8_t buffer[1024 * 1024] = { 0 };
    size_t buffer_length = sizeof(buffer);

    if (Z_OK != uncompress(buffer, &buffer_length, data, size)) {
        return 0;
    }

    return 0;
}
```

Optimization [initial version]

```
#include "third_party/zlib/zlib.h"

// Entry point for LibFuzzer.
extern "C" int LLVMFuzzerTestOneInput(const unsigned char *data, size_t size) {
    uint8_t buffer[1024 * 1024] = { 0 };
    size_t buffer_length = sizeof(buffer);

    if (Z_OK != uncompress(buffer, &buffer_length, data, size)) {
        return 0;
    }

    return 0;
}
```

Optimization [why you did that?]

```
#include "third_party/zlib/zlib.h"

// Entry point for LibFuzzer.
extern "C" int LLVMFuzzerTestOneInput(const unsigned char *data, size_t size) {
    const int NUM_ITEMS = 1024 * 1024;
    const int BUF_SIZE = NUM_ITEMS * sizeof(uint8_t);
    uint8_t *buffer = new uint8_t[NUM_ITEMS];
    uLongf buffer_length = (uLongf)BUF_SIZE;
    memset(buffer, 0, BUF_SIZE);

    if (Z_OK != uncompress(buffer, &buffer_length, data, static_cast<uLong>(size))) {
        delete[] buffer;
        return 0;
    }

    delete[] buffer;
    return 0;
}
```

Optimization [reanimation]

```
#include "third_party/zlib/zlib.h"

// Entry point for LibFuzzer.
extern "C" int LLVMFuzzerTestOneInput(const unsigned char *data, size_t size) {
    const int NUM_ITEMS = 1024 * 1024;
    const int BUF_SIZE = NUM_ITEMS * sizeof(uint8_t);
    uint8_t *buffer = new uint8_t[NUM_ITEMS];
    uLongf buffer_length = (uLongf)BUF_SIZE;
    memset(buffer, 0, BUF_SIZE);

    if (Z_OK != uncompress(buffer, &buffer_length, data, static_cast<uLong>(size))) {
        delete[] buffer;
        return 0;
    }

    delete[] buffer;
    return 0;
}
```

Optimization [reanimation]

```
#include "third_party/zlib/zlib.h"
static Bytef buffer[256 * 1024] = { 0 };
// Entry point for LibFuzzer.
extern "C" int LLVMFuzzerTestOneInput(const unsigned char *data, size_t size) {
    uLongf buffer_length = static_cast<uLongf>(sizeof(buffer));
    const int BUF_SIZE = NUM_ITEMS * sizeof(uint8_t);
    uint8_t *buffer = new uint8_t[NUM_ITEMS];
    uLongf buffer_length = (uLongf)BUF_SIZE;
    memset(buffer, 0, BUF_SIZE);

    if (Z_OK != uncompress(buffer, &buffer_length, data, static_cast<uLong>(size))) {
        delete[] buffer;
        return 0;
    }

    delete[] buffer;
    return 0;
}
```

Optimization [final version]

```
#include "third_party/zlib/zlib.h"

static Bytef buffer[256 * 1024] = { 0 };

// Entry point for LibFuzzer.
extern "C" int LLVMFuzzerTestOneInput(const uint8_t* data, size_t size) {
    uLongf buffer_length = static_cast<uLongf>(sizeof(buffer));

    if (Z_OK != uncompress(buffer, &buffer_length, data, static_cast<uLong>(size))) {
        return 0;
    }

    return 0;
}
```

Guess performance difference?

Guess performance difference?

- remove memset() for 1MB heap buffer -> **~5x**

Guess performance difference?

- remove memset() for 1MB heap buffer -> **~5x**
- move 1MB heap buffer to the stack -> **~2x**

Guess performance difference?

- remove memset() for 1MB heap buffer -> **~5x**
- move 1MB heap buffer to the stack -> **~2x**
- use 256KB instead of 1Mb on the stack -> **~3x**

Guess performance difference?

- remove memset() for 1MB heap buffer -> **~5x**
- move 1MB heap buffer to the stack -> **~2x**
- use 256KB instead of 1Mb on the stack -> **~3x**
- move 256KB buffer from stack to global -> **~2x**

Guess performance difference?

- remove memset() for 1MB heap buffer -> **~5x**
- move 1MB heap buffer to the stack -> **~2x**
- use 256KB instead of 1Mb on the stack -> **~3x**
- move 256KB buffer from stack to global -> **~2x**

53x