Fuzzing curl

Max Dymond

2018-04-15

metaswitch

Agenda

- Introduction
- How does fuzzing work?
- How does curl_fuzzer work?
- oss-fuzz
- Successes
- The future

Introduction

- Max Dymond
  - Software developer from England
  - Working for Metaswitch (a telecoms provider)
- How did I get started with curl?
- How did I get started with curl_fuzzer?

How does fuzzing work?

- Fuzz targets
  - extern "C" int LLVMFuzzerTestOneInput(const uint8_t* data, size_t size)
- Tools
  - AFL
  - Libfuzzer
- Special compilation
  - Static libraries
  - Special flags

How does curl_fuzzer work?

- Fuzz targets work on binary input
- curl_fuzzer works with TLV data to map binary input to:
  - CURLOPT option setting
  - Responses to requests
- Tries to be fast!
  - Most simple protocols complete transfer in a few ms
  - Some protocols are harder (FTP)

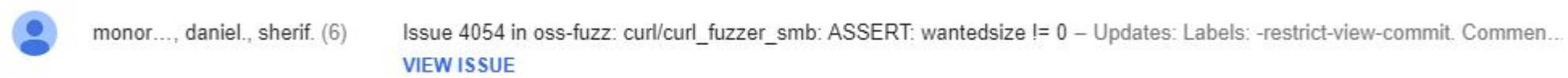
oss-fuzz (1)

- oss-fuzz is an initiative started by Google in Dec 2016
- Designed to find bugs in internet-critical projects
- Uses Google's infrastructure to run billions of fuzzing test cases per night
  - Billions just for curl…

fuzzer	perf_report	logs	tests_executed	total_crashes	new_crashes	known_crashes	edge_cov	func_cov	corpus_size	avg_exec_per_sec	new_tests_added	avg_edge_coverage
libFuzzer_curl_fuzzer	Performance	Logs	7,625,540	0	0	0	36.66% (12030/32812)	42.12% (1318/3129)	5058 (22 MB)	148	149	6,004.7
libFuzzer_curl_fuzzer_dict	Performance	Logs	2,714,897,399	0	0	0	20.99% (4588/21854)	35.48% (781/2201)	704 (746 KB)	2,910.3	684	2,383.2
libFuzzer_curl_fuzzer_file	Performance	Logs	6,603,222,459	0	0	0	18.10% (3924/21683)	32.88% (721/2193)	616 (1 MB)	8,549.5	1,103	2,063.6
libFuzzer_curl_fuzzer_fnmatch	Performance	Logs	1,729,679,185	417	0	1	99.62% (261/262)	100.00% (16/16)	567 (8 KB)	5,641.2	165	121.7
libFuzzer_curl_fuzzer_ftp	Performance	Logs	122,009,936	4	0	1	25.05% (7006/27970)	34.18% (960/2809)	1640 (2 MB)	167.9	971	3,615.6
libFuzzer_curl_fuzzer_gopher	Performance	Logs	2,031,865,674	0	0	0	20.77% (4530/21806)	35.50% (781/2200)	644 (723 KB)	2,998.3	1,596	2,353.6
libFuzzer_curl_fuzzer_http	Performance	Logs	957,403,709	0	0	0	29.11% (8027/27573)	36.08% (1002/2777)	3361 (16 MB)	1,331	1,920	4,094
libFuzzer_curl_fuzzer_imap	Performance	Logs	529,603,117	0	0	0	23.49% (6298/26808)	33.51% (932/2781)	1959 (2 MB)	814.2	749	3,248
libFuzzer_curl_fuzzer_ldap	Performance	Logs	9,458,429,838	0	0	0	14.93% (3133/20984)	29.93% (645/2155)	525 (87 KB)	11,844.7	731	1,679.3

oss-fuzz (2)

- When they find a bug, you get an email:



- Standard procedure:
  - Download testcase
  - Reproduce failure using local versions of tools
  - Diagnose problem
  - Fix it!

Successes (1)

ID ▾	Type ▾	Component ▾	Status ▾	Proj ▾	Reported ▾	Owner ▾	Summary + Labels ▾
3327	Bug	----	Verified	curl	2017-09-07	----	curl: Null-dereference READ in fwrite ClusterFuzz Reproducible
3465	Bug	----	Verified	curl	2017-09-22	----	curl: Integer-overflow in file_range ClusterFuzz Reproducible
3513	Bug	----	Verified	curl	2017-09-29	----	curl: Direct-leak in curl_domalloc ClusterFuzz Reproducible
3557	Bug	----	Verified	curl	2017-10-05	----	curl: Direct-leak in curl_domalloc ClusterFuzz Reproducible
3567	Bug	----	Verified	curl	2017-10-06	----	curl: Direct-leak in curl_domalloc ClusterFuzz Reproducible
3574	Bug	----	Verified	curl	2017-10-08	----	curl: Integer-overflow in Curl_http_readwrite_headers ClusterFuzz Reproducible
3586	Bug-Security	----	Verified	curl	2017-10-08	----	curl: Heap-buffer-overflow in Curl_client_write ClusterFuzz Reproducible
3600	Bug	----	Verified	curl	2017-10-10	----	curl: Direct-leak in curl_domalloc ClusterFuzz Reproducible
3682	Bug	----	Verified	curl	2017-10-19	----	curl/curl_fuzzer: Direct-leak in curl_docalloc ClusterFuzz Reproducible
3694	Bug	----	Verified	curl	2017-10-19	----	curl/curl_fuzzer: Undefined-shift in ftp_state_pasv_resp ClusterFuzz Reproducible
3956	Bug-Security	----	Verified	curl	2017-10-31	----	curl/curl_fuzzer_http: Index-out-of-bounds in fuzz_send_next_response ClusterFuzz Reproducible
4054	Bug	----	Verified	curl	2017-11-04	----	curl/curl_fuzzer_smb: ASSERT: wantedsize != 0 ClusterFuzz Reproducible
4161	Bug-Security	----	Verified	curl	2017-11-10	----	curl/curl_fuzzer_ftp: Heap-buffer-overflow in setcharset ClusterFuzz Reproducible
4785	Bug-Security	----	Verified	curl	2017-12-25	----	curl/curl_fuzzer_imap: Heap-buffer-overflow in fuzz_read_callback ClusterFuzz Reproducible
4853	Bug	----	Verified	curl	2017-12-31	----	curl/curl_fuzzer_ftp: Integer-overflow in ftp_range ClusterFuzz Reproducible
5206	Bug-Security	----	Verified	curl	2018-01-10	----	curl/curl_fuzzer_pop3: Heap-buffer-overflow in pop3_get_message ClusterFuzz Reproducible
5251	Bug-Security	----	Verified	curl	2018-01-11	----	curl/curl_fuzzer_ftp: Heap-buffer-overflow in loop ClusterFuzz Reproducible
5397	Bug-Security	----	Verified	curl	2018-01-16	----	curl/curl_fuzzer_fnmatch: Heap-buffer-overflow in loop ClusterFuzz Reproducible
5421	Bug-Security	----	Verified	curl	2018-01-16	----	curl/curl_fuzzer_fnmatch: Heap-buffer-overflow in loop ClusterFuzz Reproducible
5507	Bug-Security	----	WontFix	curl	2018-01-20	----	curl/curl_fuzzer_http: Index-out-of-bounds in fuzz_handle_transfer ClusterFuzz Unreproducible
5794	Bug	----	New	curl	2018-01-29	----	curl/curl_fuzzer_fnmatch: Timeout in curl_fuzzer_fnmatch ClusterFuzz Unreproducible
5908	Bug	----	Verified	curl	2018-02-02	----	curl/curl_fuzzer_fnmatch: NULL ClusterFuzz Reproducible
6483	Bug-Security	----	Verified	curl	2018-02-20	----	curl/curl_fuzzer_rtsp: Heap-buffer-overflow in rtsp_rtp_readwrite ClusterFuzz Reproducible
6937	Bug	----	WontFix	curl	2018-03-15	----	curl/curl_fuzzer_rtsp: Timeout in curl_fuzzer_rtsp ClusterFuzz Unreproducible
6991	Bug-Security	----	WontFix	curl	2018-03-17	----	curl/curl_fuzzer_http: Stack-buffer-overflow in fuzz_handle_transfer ClusterFuzz Unreproducible
7105	Bug-Security	----	New	curl	2018-03-24	----	curl/curl_fuzzer_[redacted]

Successes (2)

- Stats:
  - Since Sun Aug 27 15:57:05 2017 +0100:
    - Found 18 actual curl bugs
    - 3 flaky bugs (which don’t reproduce)
    - 3 fuzzer bugs (my bad!)
    - 2 duplicates
  - Most buggy areas so far:
    - ftp + fnmatch
    - rtsp

The future

- More fuzzing targets
  - Need to compile targets in static mode; this makes some targets complicated
    - In the future this restriction might be lifted
- More speed!
  - Possibly reworking slow areas like FTP to use cleverer servers
- More coverage
  - Some tests may need a bit more help to give useful results – e.g. SSL is currently not really being fuzzed

Questions?

- Twitter: @cmeister_2
- Email: cmeister2@gmail.com
- cmeister2 on curl IRC