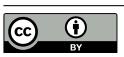
StorFuzz: Using Data Diversity to Overcome Fuzzing Plateaus Supplementary Tables

This document contains supplementary tables reporting on all experiments for StorFuzz:

List of Tables

This is supplementary material to:

L. Weiß, T. Holl, and K. Borgolte. "StorFuzz: Using Data Diversity to Overcome Fuzzing Plateaus." In: *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering (ICSE)*. Association for Computing Machinery (ACM)/Institute of Electrical and Electronics Engineers (IEEE), Apr. 2026. DOI: 10.1145/3744916.3773179



ICSE '26, Rio de Janeiro, Brazil

Table 1: Corpus size of the initial OSS-Fuzz corpus compared to the saturated corpus, and the corpora created by diversifying the saturated corpus with StorFuzz, DDFuzz, or restarting LibAFL for 24 hours. We also report the difference to the seed set size. The largest diversified corpus per benchmark is highlighted. All pairwise differences are significant.

Benchmark	OSS-Fuzz Corpus	Saturated Corpus	Diversified Corpora (Median of 10 Trials)		
			StorFuzz	LibAFL	DDFuzz
bloaty	12841	15832 (1.23x)	40776.0 (2.58x)	2479.0 (0.16x)	3953.5 (0.25x)
curl	15803	24682 (1.56x)	28818.5 (1.17x)	2469.0 (0.10x)	4890.0 (0.20x)
freetype2	21198	48954 (2.31x)	108563.5 (2.22x)	6418.5 (0.13x)	58406.0 (1.19x)
harfbuzz	29745	48105 (1.62x)	88038.0 (1.83x)	7723.5 (0.16x)	12984.5 (0.27x)
jsoncpp	3530	3635 (1.03x)	2720.0 (0.75x)	336.0 (0.09x)	567.5 (0.16x)
lcms	5188	5388 (1.04x)	7482.5 (1.39x)	562.0 (0.10x)	598.0 (0.11x)
libjpeg-turbo	6398	6755 (1.06x)	14152.0 (2.10x)	1216.0 (0.18x)	1832.5 (0.27x)
libpcap	13554	15014 (1.11x)	31085.0 (2.07x)	2246.5 (0.15x)	13908.5 (0.93x)
libpng	8296	8536 (1.03x)	17829.5 (2.09x)	771.0 (0.09x)	1101.5 (0.13x)
libxml2	17456	42708 (2.45x)	24776.5 (0.58x)	3959.0 (0.09x)	7929.5 (0.19x)
libxslt	32636	70440 (2.16x)	17214.0 (0.24x)	3533.0 (0.05x)	8484.5 (0.12x)
mbedtls	4482	4901 (1.09x)	42346.0 (8.64x)	770.0 (0.16x)	1080.0 (0.22x)
openh264	20109	25240 (1.26x)	54209.5 (2.15x)	2372.0 (0.09x)	7914.0 (0.31x)
openssl	11125	15357 (1.38x)	18318.0 (1.19x)	1057.5 (0.07x)	1198.0 (0.08x)
openthread	1368	4949 (3.62x)	15111.5 (3.05x)	698.5 (0.14x)	1219.0 (0.25x)
proj4	16360	26905 (1.64x)	24514.0 (0.91x)	5088.0 (0.19x)	10412.5 (0.39x)
re2	10738	20712 (1.93x)	9588.0 (0.46x)	934.0 (0.05x)	10366.0 (0.50x)
sqlite3	16715	40653 (2.43x)	28750.5 (0.71x)	6195.0 (0.15x)	19723.5 (0.49x)
stb	6507	7642 (1.17x)	25330.0 (3.31x)	936.5 (0.12x)	7257.0 (0.95x)
systemd	8534	10000 (1.17x)	4510.0 (0.45x)	1050.0 (0.10x)	1440.0 (0.14x)
vorbis	5655	5833 (1.03x)	12203.5 (2.09x)	403.0 (0.07x)	557.0 (0.10x)
woff2	7496	7918 (1.06x)	31144.0 (3.93x)	654.5 (0.08x)	7947.5 (1.00x)
zlib	1460	1566 (1.07x)	4118.0 (2.63x)	168.0 (0.11x)	1193.0 (0.76x)

Table 2: Corpus sizes of the corpora created by diversifying the saturated corpus with StorFuzz, DDFuzz, or restarting LibAFL for at most 24 hours. We report the median size reached by each fuzzer in the number of executions achieved in the slowest trial across all fuzzers, thereby accounting for different execution speeds. The largest diversified corpus per benchmark is highlighted.

Benchmark	Diversified Corpora (Median of 10 Trials)		
	StorFuzz	LibAFL	DDFuzz
bloaty	24657.0	2473.5	3949.0
curl	24094.5	2465.0	4882.5
freetype2	72462.5	6399.5	58324.0
harfbuzz	67377.0	7700.5	12958.0
jsoncpp	2480.5	336.0	567.5
lcms	6907.5	562.0	598.0
libjpeg-turbo	11943.5	1075.5	1832.5
libpcap	25542.5	2239.0	13827.5
libpng	14459.5	771.0	1101.5
libxml2	21432.0	3946.5	7915.5
libxslt	11938.0	3511.0	8477.5
mbedtls	23805.0	770.0	1078.0
openh264	51829.5	2371.0	7886.0
openssl	14707.5	1056.5	1198.0
openthread	13059.0	683.0	1219.0
proj4	17379.0	5087.0	9860.5
re2	8153.5	933.0	10359.5
sqlite3	16812.0	6076.0	19472.5
stb	20756.5	936.0	7224.0
systemd	4498.5	1049.0	1440.0
vorbis	11022.5	403.0	557.0
woff2	22299.5	654.0	7947.5
zlib	3729.5	168.0	1192.5

Table 3: *Median* number of branches covered at different points during the fuzzing campaigns starting from the saturated corpus. Two fuzzers in the configuration signal that it switched between the two fuzzers at the indicated times. We report the median coverage over 10 trials and the change relative to the previously reported point. We report only benchmarks with at least one fuzzer improving coverage. Green (□) marks the best performing fuzzing configuration at the specific point in the run, the best configuration per target is marked with (*). We omit values for improved readability, when there is no change.

Benchmark	Fuzzer Configuration	Seed Coverage (after 5x120h LibAFL)	Code Coverage after 144 h	168 h	192 h	216 h
bloaty	LibAFL	7537	7539.0 (+2.0)			7539.5 (+0.5)
	LibAFL/LibAFL		7538.5 (+1.5)	7539.0 (+0.5)	7539.5 (+0.5)	
	StorFuzz/LibAFL (*)		7556.5 (+19.5)		7638.0 (+81.5)	7639.0 (+1.0)
	DDFuzz/LibAFL		7538.0 (+1.0)			7539.0 (+1.0)
curl	LibAFL	11554	11556.5 (+2.5)	11557.0 (+0.5)		
	LibAFL/LibAFL		11557.0 (+3.0)	11558.0 (+1.0)	11558.5 (+0.5)	
	StorFuzz/LibAFL (*)		11569.5 (+15.5)	11571.0 (+1.5)	11574.5 (+3.5)	11575.0 (+0.5)
	DDFuzz/LibAFL		11557.0 (+3.0)	11558.0 (+1.0)		11558.5 (+0.5)
freetype2	LibAFL	18043	18045.0 (+2.0)		18045.5 (+0.5)	
	LibAFL/LibAFL		18044.5 (+1.5)	18045.0 (+0.5)	18046.0 (+1.0)	18046.5 (+0.5)
	StorFuzz/LibAFL (*)		18054.5 (+11.5)	18055.0 (+0.5)	18060.0 (+5.0)	18060.5 (+0.5)
	DDFuzz/LibAFL		18045.0 (+2.0)	18045.5 (+0.5)	18046.5 (+1.0)	18047.0 (+0.5)
harfbuzz	LibAFL	21846	21847.0 (+1.0)		21850.5 (+3.5)	21859.0 (+8.5)
	LibAFL/LibAFL		21847.5 (+1.5)	21852.0 (+4.5)	21867.0 (+15.0)	21877.0 (+10.0)
	StorFuzz/LibAFL (*)		21870.5 (+24.5)	21873.5 (+3.0)	21878.0 (+4.5)	21889.0 (+11.0)
	DDFuzz/LibAFL		21854.5 (+8.5)	21871.0 (+16.5)	21873.5 (+2.5)	21888.0 (+14.5)
jsoncpp	LibAFL (*)	525				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
lcms	LibAFL (*)	2476				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
libjpeg-turbo	LibAFL (*)	3417				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
libpcap	LibAFL (*)	4465				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					

Table 3: *Continued from previous page*

Benchmark	Fuzzer Configuration	Seed Coverage (after 5x120h LibAFL)	Code Coverage after 144 h	168 h	192 h	216 h
libpng	LibAFL (*)	2116				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
libxml2	LibAFL	16311		16312.0 (+1.0)		16313.0 (+1.0)
	LibAFL/LibAFL		16311.5 (+0.5)	16314.0 (+2.5)	16315.0 (+1.0)	16315.5 (+0.5)
	StorFuzz/LibAFL (*)		16314.0 (+3.0)	16314.5 (+0.5)	16319.0 (+4.5)	16321.0 (+2.0)
	DDFuzz/LibAFL		16313.0 (+2.0)	16314.5 (+1.5)	16315.0 (+0.5)	
libxslt	LibAFL	12135	12140.0 (+5.0)	12141.0 (+1.0)	12142.5 (+1.5)	12144.5 (+2.0)
	LibAFL/LibAFL		12143.5 (+8.5)	12144.5 (+1.0)		12145.0 (+0.5)
	StorFuzz/LibAFL (*)		12158.5 (+23.5)	12160.0 (+1.5)	12162.0 (+2.0)	12163.0 (+1.0)
	DDFuzz/LibAFL		12139.0 (+4.0)	12146.0 (+7.0)	12148.0 (+2.0)	12148.5 (+0.5)
mbedtls	LibAFL	4418				
	LibAFL/LibAFL					
	StorFuzz/LibAFL (*)		4419.0 (+1.0)			
	DDFuzz/LibAFL					
openh264	LibAFL	9839				
	LibAFL/LibAFL					
	StorFuzz/LibAFL (*)		9841.0 (+2.0)			
	DDFuzz/LibAFL					
openssl	LibAFL	5974	5977.0 (+3.0)			
	LibAFL/LibAFL		5977.0 (+3.0)			
	StorFuzz/LibAFL (*)		5977.0 (+3.0)	5977.5 (+0.5)	5979.0 (+1.5)	
	DDFuzz/LibAFL		5977.0 (+3.0)			5978.0 (+1.0)
openthread	LibAFL	4581	4635.5 (+54.5)	4708.0 (+72.5)	4710.5 (+2.5)	
	LibAFL/LibAFL (*)		4637.5 (+56.5)	4712.0 (+74.5)	4720.0 (+8.0)	4721.5 (+1.5)
	StorFuzz/LibAFL		4651.0 (+70.0)	4658.5 (+7.5)	4671.5 (+13.0)	4684.0 (+12.5)
	DDFuzz/LibAFL		4583.5 (+2.5)	4684.0 (+100.5)	4690.5 (+6.5)	4720.0 (+29.5)
proj4	LibAFL	11110	11112.0 (+2.0)	11112.5 (+0.5)	11115.0 (+2.5)	11116.5 (+1.5)
	LibAFL/LibAFL		11112.0 (+2.0)	11113.0 (+1.0)		11114.0 (+1.0)
	StorFuzz/LibAFL (*)		11119.0 (+9.0)	11120.0 (+1.0)	11125.5 (+5.5)	11127.5 (+2.0)
	DDFuzz/LibAFL		11111.0 (+1.0)	11112.0 (+1.0)	11113.0 (+1.0)	11115.0 (+2.0)
re2	LibAFL	2937		2938.0 (+1.0)		
	LibAFL/LibAFL		2938.0 (+1.0)			
	StorFuzz/LibAFL		2938.0 (+1.0)			
	DDFuzz/LibAFL (*)		2938.0 (+1.0)			2939.0 (+1.0)

Table 3: *Continued from previous page*

Benchmark	Fuzzer Configuration	Seed Coverage (after 5x120h LibAFL)	Code Coverage after 144 h	168 h	192 h	216 h
sqlite3	LibAFL	21545	21583.0 (+38.0)	21587.0 (+4.0)	21592.0 (+5.0)	21598.5 (+6.5)
	LibAFL/LibAFL		21578.5 (+33.5)	21601.5 (+23.0)	21614.0 (+12.5)	21622.5 (+8.5)
	StorFuzz/LibAFL (*)		21592.0 (+47.0)	21609.0 (+17.0)	21623.5 (+14.5)	21630.5 (+7.0)
	DDFuzz/LibAFL		21582.0 (+37.0)	21601.5 (+19.5)	21613.5 (+12.0)	21626.0 (+12.5)
stb	LibAFL (*)	2322				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
systemd	LibAFL (*)	271				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
vorbis	LibAFL (*)	1381				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
woff2	LibAFL (*)	1258				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					
zlib	LibAFL (*)	473				
	LibAFL/LibAFL (*)					
	StorFuzz/LibAFL (*)					
	DDFuzz/LibAFL (*)					

Table 4: *Median* number of branches covered at different points during the fuzzing campaigns starting from the corpus saturated with both LibAFL and libFuzzer. Two fuzzers in the configuration signal that it switched between the two fuzzers at the indicated times. We report the median coverage over 10 trials and the change relative to the previously reported point. We report only benchmarks with at least one fuzzer improving coverage. Green (□) marks the best performing fuzzing configuration at the specific point in the run, the best configuration per target is marked with (*). We omit values for improved readability, when there is no change.

Benchmark	Fuzzer Configuration	Seed Coverage (after 5x120h LibAFL)	Code Coverage after 144 h	168 h	192 h	216 h
bloaty	StorFuzz/LibAFL	7550	7565.5 (+15.5)	7566.5 (+1.0)	7596.5 (+30.0)	
	WingFuzz/libFuzzer (*)		7628.5 (+78.5)	7629.0 (+0.5)	7651.0 (+22.0)	7651.5 (+0.5)
curl	StorFuzz/LibAFL	11623	11633.5 (+10.5)	11634.5 (+1.0)		11635.0 (+0.5)
	WingFuzz/libFuzzer (*)		11636.0 (+13.0)	11637.5 (+1.5)		11639.0 (+1.5)
freetype2	StorFuzz/LibAFL	18107	18126.0 (+19.0)	18129.0 (+3.0)	18129.5 (+0.5)	
	WingFuzz/libFuzzer (*)		18134.0 (+27.0)	18135.0 (+1.0)	18136.0 (+1.0)	
harfbuzz	StorFuzz/LibAFL (*)	21879	21908.0 (+29.0)	21914.5 (+6.5)	21922.0 (+7.5)	21927.5 (+5.5)
	WingFuzz/libFuzzer		21881.0 (+2.0)	21904.0 (+23.0)	21910.5 (+6.5)	21913.5 (+3.0)
jsoncpp	StorFuzz/LibAFL (*)	525				
	WingFuzz/libFuzzer (*)					
lcms	StorFuzz/LibAFL (*)	2476		2480.0 (+4.0)		
	WingFuzz/libFuzzer			2476.5 (+0.5)	2477.0 (+0.5)	
libjpeg-turbo	StorFuzz/LibAFL (*)	3417				
	WingFuzz/libFuzzer (*)					
libpcap	StorFuzz/LibAFL	4469				
	WingFuzz/libFuzzer (*)			4469.5 (+0.5)	4470.0 (+0.5)	
libpng	StorFuzz/LibAFL (*)	2116				
	WingFuzz/libFuzzer (*)					
libxml2	StorFuzz/LibAFL	16394		16395.0 (+1.0)	16396.0 (+1.0)	16397.5 (+1.5)
	WingFuzz/libFuzzer (*)		16396.5 (+2.5)		16400.0 (+3.5)	16401.0 (+1.0)
libxslt	StorFuzz/LibAFL (*)	12217	12231.0 (+14.0)	12236.5 (+5.5)	12237.0 (+0.5)	12239.0 (+2.0)
	WingFuzz/libFuzzer		12222.0 (+5.0)	12226.5 (+4.5)	12231.0 (+4.5)	12234.5 (+3.5)
mbedtls	StorFuzz/LibAFL (*)	4420				
	WingFuzz/libFuzzer (*)					
openh264	StorFuzz/LibAFL (*)	9841				
	WingFuzz/libFuzzer (*)					
openssl	StorFuzz/LibAFL	5982	5985.0 (+3.0)			
	WingFuzz/libFuzzer (*)		5985.5 (+3.5)	5986.0 (+0.5)	5986.5 (+0.5)	5987.5 (+1.0)
openthread	StorFuzz/LibAFL (*)	4778	4791.0 (+13.0)	4795.0 (+4.0)	4799.0 (+4.0)	4800.5 (+1.5)
	WingFuzz/libFuzzer		4780.0 (+2.0)	4781.0 (+1.0)		4781.5 (+0.5)
proj4	StorFuzz/LibAFL	11351	11356.0 (+5.0)	11362.0 (+6.0)	11369.5 (+7.5)	11370.0 (+0.5)
	WingFuzz/libFuzzer (*)		11371.5 (+20.5)	11377.0 (+5.5)	11391.5 (+14.5)	11397.0 (+5.5)
re2	StorFuzz/LibAFL	2963				
	WingFuzz/libFuzzer (*)		2965.0 (+2.0)			2966.0 (+1.0)

Table 4: *Continued from previous page*

Benchmark	Fuzzer Configuration	Seed Coverage (after 5x120h LibAFL)	Code Coverage after 144 h	168 h	192 h	216 h
sqlite3	StorFuzz/LibAFL (*)	21585	21682.5 (+97.5)	21694.5 (+12.0)	21707.0 (+12.5)	21715.0 (+8.0)
	WingFuzz/libFuzzer		21665.5 (+80.5)	21678.5 (+13.0)	21695.0 (+16.5)	21703.0 (+8.0)
stb	StorFuzz/LibAFL (*)	2322				
	WingFuzz/libFuzzer (*)					
systemd	StorFuzz/LibAFL (*)	271				
	WingFuzz/libFuzzer (*)					
vorbis	StorFuzz/LibAFL	1381				
	WingFuzz/libFuzzer (*)		1384.0 (+3.0)			
woff2	StorFuzz/LibAFL (*)	1263				
	WingFuzz/libFuzzer (*)					
zlib	StorFuzz/LibAFL (*)	473				
	WingFuzz/libFuzzer (*)					

Table 5: Bugs discovered by StorFuzz. In some cases the bugs were publicly reported by others while we were in the process of responsible disclosure. We have anonymized the references where necessary.

Identifier	Description	Public Reference	CVE
ImageMagick-1	(Rediscovered) UAF in `ReadTIFFImage`	30f7a3d	(n.a.)
ImageMagick-2 / LibRaw-4	Heap OOB Read due to a Bug in LibRaw `open_datastream`	PR #679	(requested)
ImageMagick-3	Heap OOB Reads	81ac8a0 & bac413a	CVE-2025-43965
ImageMagick-4 / libheif-3	SEGV due to NULL-Pointer Dereference in libheif `ImageItem_Grid::get_decoder`	Issue #1473	
ImageMagick-5	SEGV due to NULL-Pointer Dereference in libheif `ImageItem:: get_coded_imgate_colorspace`	Issue #1455	
ImageMagick-6 / LibRaw-1	Heap OOB Read in LibRaw `parse_tiff_ifd`	66fe663	CVE-2025-43961
ImageMagick-7 / LibRaw-2	Heap OOB Read in LibRaw `phase_one_correct`	66fe663	CVE-2025-43962 & CVE-2025-43964
ImageMagick-8 / LibRaw-3	SEGV due to Heap OOB Read in LibRaw `parse_one_correct`	be26e76	CVE-2025-43963
UPX-1	SEGV due to OOB Read in `elf_lookup("JNI_OnLoad")`	Issue #871	†
UPX-2	SEGV due to OOB Read in `invert_pt_dynamic`	Issue #872 (d4bc364)	†
UPX-3	SEGV due to OOB Read in `invert_pt_dynamic` due to integer overflow	Issue #872 (4aa92d8)	†
UPX-4	OOB Read in `get_dynsym_name`	Issue #871	†
UPX-5	OOB Read in `PackMachBase::canUnpack() 1682`	Issue #874	†
UPX-6	OOB Read in `PackMachBase::canUnpack() 1862 / 1866`	Issue #875	†
libheif-1	SEGV due to NULL-Pointer Dereference in `ImageItem_iden::get_luma_bits_per_pixe`	Issue #1455	CVE-2025-43967
libheif-2	SEGV due to NULL-Pointer Dereference in `ImageItem::get_coded_image_colorspace`	b385553	CVE-2025-43966
VLC-1	Heap OOB Write in mp4.c `FragCreateTrunIndex` (MP4 demuxer)	(pending)	
VLC-2	Heap OOB Write in spudec/parse.c `ParseRLE` (SPU decoder)	(pending)	
VLC-3	Heap OOB Write in svcdsub.c `SVCDSubRenderImage` (SVCD decoder)	(pending)	
VLC-4	Limited Heap OOB Write in substx3g.c `Decode` (tx3g decoder)	(pending)	
VLC-5	Double Free in libmp4.c `MP4_ReadBox_sgpd` and `MP4_FreeBox_sgpd` (MP4 demuxer)	(pending)	
VLC-6	Possible Stack-based Buffer Overflow in `aout_ChannelReorder` (audio output)	(pending)	
VLC-7	Heap OOB Read in wav.c `ChunkParseFmt` (WAV demuxer)	(pending)	
VLC-8	Limited Heap OOB Read in meta.c `iTUNTripletCallback` (MP4 demuxer)	(pending)	
VLC-9	Heap OOB Read in `aout_ChannelReorder` (audio output)	(pending)	
VLC-10	Limited Heap OOB Read in ty.c `find_es_header` (TY demuxer)	(pending)	
VLC-11	Arbitrary Read via `es_format_Copy` due to Use of Uninitialized Memory (via MP4 demuxer)	(pending)	
VLC-12	Division by Zero in libmp4.c `MP4_ReadBox_iloc` (MP4 demuxer)	(pending)	

Table 5: *Continued from previous page*

Identifier	Description	Public Reference	CVE
VLC-13	Division by Zero in avi.c `AVI_Rescale` (AVI demuxer)	(pending)	
VLC-14	Assertion Failure in `AVI_IndexLoad` (AVI demuxer)	(pending)	
VLC-15	Assertion Failure in `vlc_meta_SetWithPriority` (multiple code paths)	Issue #28976	(n.a.)
VLC-16	Assertion Failure in `date_Increment` (multiple code paths)	(pending)	
VLC-17	Assertion Failure in `date_Increment` (multiple code paths)	Issue #28978	(n.a.)
VLC-18	Assertion Failure in `aout_ChannelReorder` (e.g. via WAV demuxing)	(pending)	
VLC-19	Assertion Failure in `picture_Setup`	Issue #28980	(n.a.)
VLC-20	Assertion Failure in SPU subtitle rendering (`Render`)	(pending)	
VLC-21	Assertion Failure in `webvtt_region_Reduce` (WebVTT processing)	(pending)	
VLC-22	Assertion Failure in `TrackUpdateStarttimes` (MP4 demuxing)	(pending)	
VLC-23	Assertion Failure in `MP4_GetAudioFrameInfo` (MP4 demuxing)	(pending)	
VLC-24	Assertion Failure in `SetupAudioES` (MP4 demuxing / ES setup)	Issue #28985	(n.a.)
VLC-25	NULL-Pointer Dereference in `text_segment_ruby_New` (via WebVTT)	(pending)	
VLC-26	NULL-Pointer Dereference in `webvtt_FillStyleFromCssDeclaration` (via `CSSGrammar.y`)	(pending)	
VLC-27	NULL-Pointer Dereference in CSS parsing (`CSSGrammar.y`)	(pending)	
assimp-1	SEGV: Heap OOB Write in `ParseLV4MeshBonesVertices`	Issue #6024[$]	CVE-2025-3159
assimp-2	Heap OOB Write in `CSMImporter::InternReadFile`	PR #6138[$]	CVE-2025-2592
assimp-3	Stack OOB Write in `GetNextLine<char>`	Issue #6016[$]	CVE-2025-2151
assimp-4	SIGSEGV: Heap OOB Read in `MDCImporter::ValidateSurfaceHeader`	Issue #6167	CVE-2025-5165
assimp-5	SIGSEGV: Heap OOB Read in `MDCImporter::InternReadFile`	Issue #6168	CVE-2025-5166
assimp-6	SIGSEGV: Constant-Pointer Dereference in `NDOImporter::InternReadFile`	PR #6055[$]	
assimp-7	SIGSEGV: Constant-Pointer Dereference in `CSMImporter::InternReadFile`	Issue #6012[$]	CVE-2025-2751
assimp-8	Heap OOB Read in `LWOImporter::GetS0`	Issue #6169	CVE-2025-5167
assimp-9	Heap OOB Read in `MDLImporter::ImportUVCoordinate_3DGS_MDL345`	Issue #6170	CVE-2025-5168
assimp-10	Heap OOB Read in `MDLImporter::InternReadFile_3DGS_MDL345`	Issue #6171	CVE-2025-5169
assimp-11	Heap OOB Read in `MDLImporter::InternReadFile_Quake1`	Issue #6172	CVE-2025-5200
assimp-12	Heap OOB Reads in `LWOImporter::CountVertsAndFacesLWO2`	Issue #6173	CVE-2025-5201
assimp-13	Heap OOB Read in `MDLImporter::ParseSkinLump_3DGS_MDL7` / `SkipSkinLump`	Issue #6176	CVE-2025-5204
assimp-14	Heap OOB Read in `HL1MDLLoader::validate_header`	Issue #6174	CVE-2025-5202

Table 5: *Continued from previous page*

Identifier	Description	Public Reference	CVE
assimp-15	Heap OOB Read in `SkipSpaces<char>`	Issue #6175	CVE-2025-5203
assimp-16	Multiple Out of Memory issues		(n.a.)
assimp-17	Multiple Issues where Requested Allocation Size Exceeds Maximum		(n.a.)
PHP-1	NULL-Pointer Dereference when using `register_tick_function` in destructor	Issue #18033	(n.a.)*

† UPX maintainers do not consider memory access violations security-relevant to UPX. They consider UPX to run in the same security context as its input.

* This bug does not constitute a security issue according to the project's security policy.

$ The bug was publicly disclosed by others during the period of coordinated disclosure.

Table 6: Edges covered by different fuzzers in 24 hours starting from a saturated seed set. We can see that DDFuzz performs better than datAFLow and WingFuzz outperforms SGFuzz. The presented numbers are median result of 5 runs. The best performing libFuzzer/AFL-based fuzzer(s) for each benchmark are highlighted. Note that these coverage values cannot be compared to the results in the main text directly as they were obtained using a different machine.

	AFL-based		**libFuzzer-based**	
Benchmark	**datAFLow**	**DDFuzz**	**SGFuzz**	**WingFuzz**
bloaty	6944	6946	6950	7032
curl	11513	11514	11526	11552
freetype2	18044	18050	18054	18080
harfbuzz	11355	11366	11354	11371
jsoncpp	525	525	525	525
lcms	2426	2426	2426	2426
libjpeg-turbo	3089	3177	3089	3089
libpcap	4465	4465	4466	4466
libpng	2116	2116	2116	2116
libxml2	16201	16212	16214	16248
libxslt	11492	11500	11527	11505
mbedtls	4332	4332	4332	4333
openh264	9683	9683	9684	9683
openssl	5961	5959	5961	5966
openthread	4561	4569	4525	4585
proj4	10529	10529	10526	10589
re2	2882	2885	2884	2892
sqlite3	21527	21529	21532	21571
stb	2322	2322	2322	2322
systemd	(failed to build)	243	243	243
vorbis	1388	1389	1383	1383
woff2	1255	1255	1255	1255
zlib	473	473	473	473