Ruxmon

American Fuzzy Lop - fuzzing like there’s no tomorrow

j@jspin.re

Disclaimer

- no more than 30 minutes
- speaking by myself
- generalist not a specialist
- work in progress
- no 0day here

agenda

- about me
- fuzzing
- American Fuzzy Lop (AFL)
- demo
- references

whoami

- Brazilian HueHueHue
- Penetration Tester
- OSCP & OSCE (who cares?)
- Working for SG

fuzzing

- fuzzing?
- types
- tools

fuzzing

- patience is the key
- crashes do not mean security issues
- tons of crashes != tons of security bugs

fuzzing

- targets
- corpus
- environment
  - completely budget dependent
- run
- (simplest approach)

environment

- office
    - i7
    - 32gb ram
    - 1TB
- home
    - 2 x i7
    - 16gb ram
    - 6TB

1. target
2. compile
3. corpus
4. fuzz
5. triage
6. profit

1. target
2. code review
3. wrapper
4. compile
5. corpus
6. fuzz
7. triage
8. fun

American Fuzzy Lop

- http://lcamtuf.coredump.cx/afl/
- afl-users+subscribe@googlegroups.com
  - dumb mode
  - instrumented mode
  - qemu mode
  - llvm mode - 2x faster than afl-gcc/afl-g++ http://clang.llvm.org/get_started.html
- tools
  - afl-cmin afl-tmin

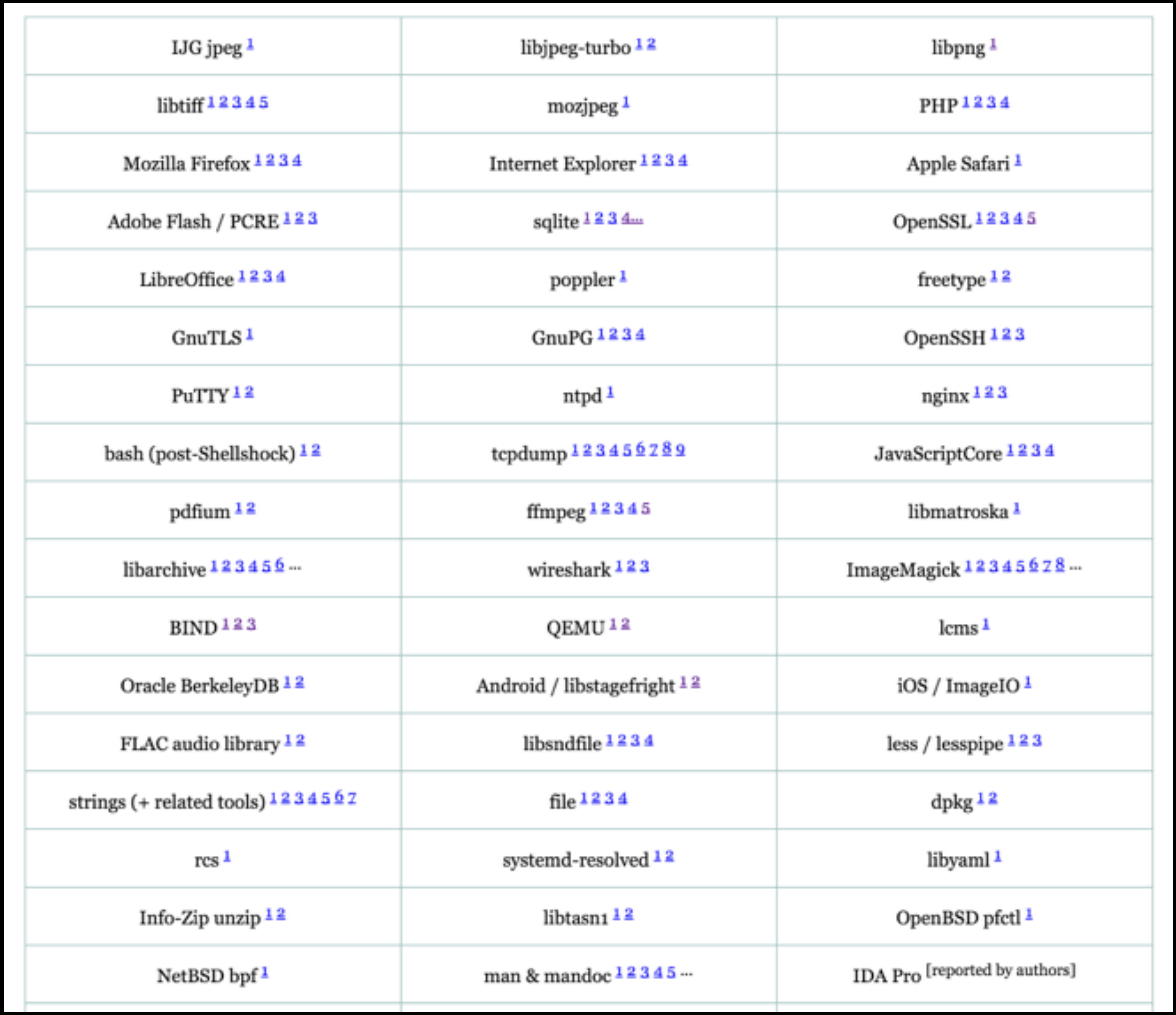
American Fuzzy Lop

- qemu mode
    - only linux
    - qemu_mode folder
    - ./build_qemu_support.sh
    - -Q option

American Fuzzy Lop

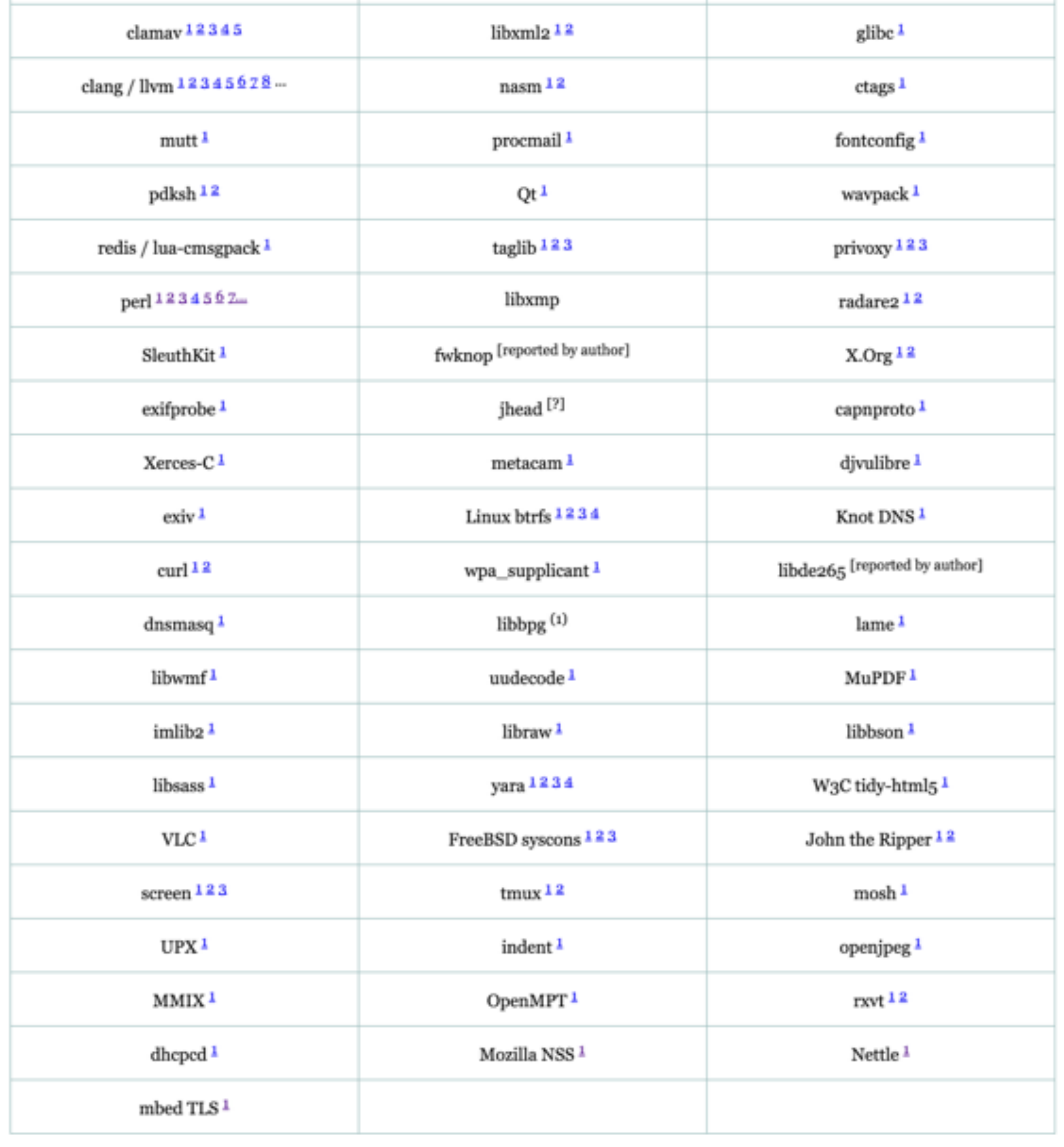
- llvm mode
  - http://clang.llvm.org/get_started.html
  - llvm_mode folder
  - afl-clang, afl-clang++, afl-clang-fast, afl-clang-fast++

American Fuzzy Lop

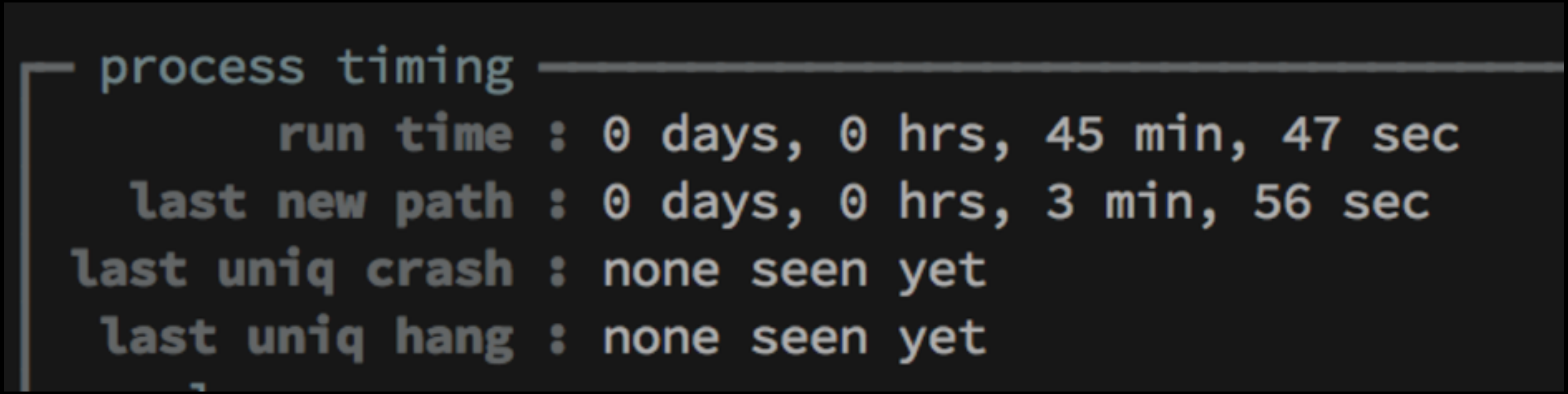


IJG jpeg[1]	libjpeg-turbo[1][2]	libpng[1]
libtiff[1][2][3][4][5]	mozjpeg[1]	PHP[1][2][3][4]
Mozilla Firefox[1][2][3][4]	Internet Explorer[1][2][3][4]	Apple Safari[1]
Adobe Flash / PCRE[1][2][3]	sqlite[1][2][3][4...]	OpenSSL[1][2][3][4][5]
LibreOffice[1][2][3][4]	poppler[1]	freetype[1][2]
GnuTLS[1]	GnuPG[1][2][3][4]	OpenSSH[1][2][3]
PuTTY[1][2]	ntpd[1]	nginx[1][2][3]
bash (post-Shellshock)[1][2]	tcpdump[1][2][3][4][5][6][7][8][9]	JavaScriptCore[1][2][3][4]
pdfium[1][2]	ffmpeg[1][2][3][4][5]	libmatroska[1]
libarchive[1][2][3][4][5][6]...	wireshark[1][2][3]	ImageMagick[1][2][3][4][5][6][7][8]...
BIND[1][2][3]	QEMU[1][2]	lcms[1]
Oracle BerkeleyDB[1][2]	Android / libstagefright[1][2]	iOS / ImageIO[1]
FLAC audio library[1][2]	libsndfile[1][2][3][4]	less / lesspipe[1][2][3]
strings (+ related tools)[1][2][3][4][5][6][7]	file[1][2][3][4]	dpkg[1][2]
rcs[1]	systemd-resolved[1][2]	libyaml[1]
Info-Zip unzip[1][2]	libtasn1[1][2]	OpenBSD pfctl[1]
NetBSD bpf[1]	man & mandoc[1][2][3][4][5]...	IDA Pro [reported by authors]

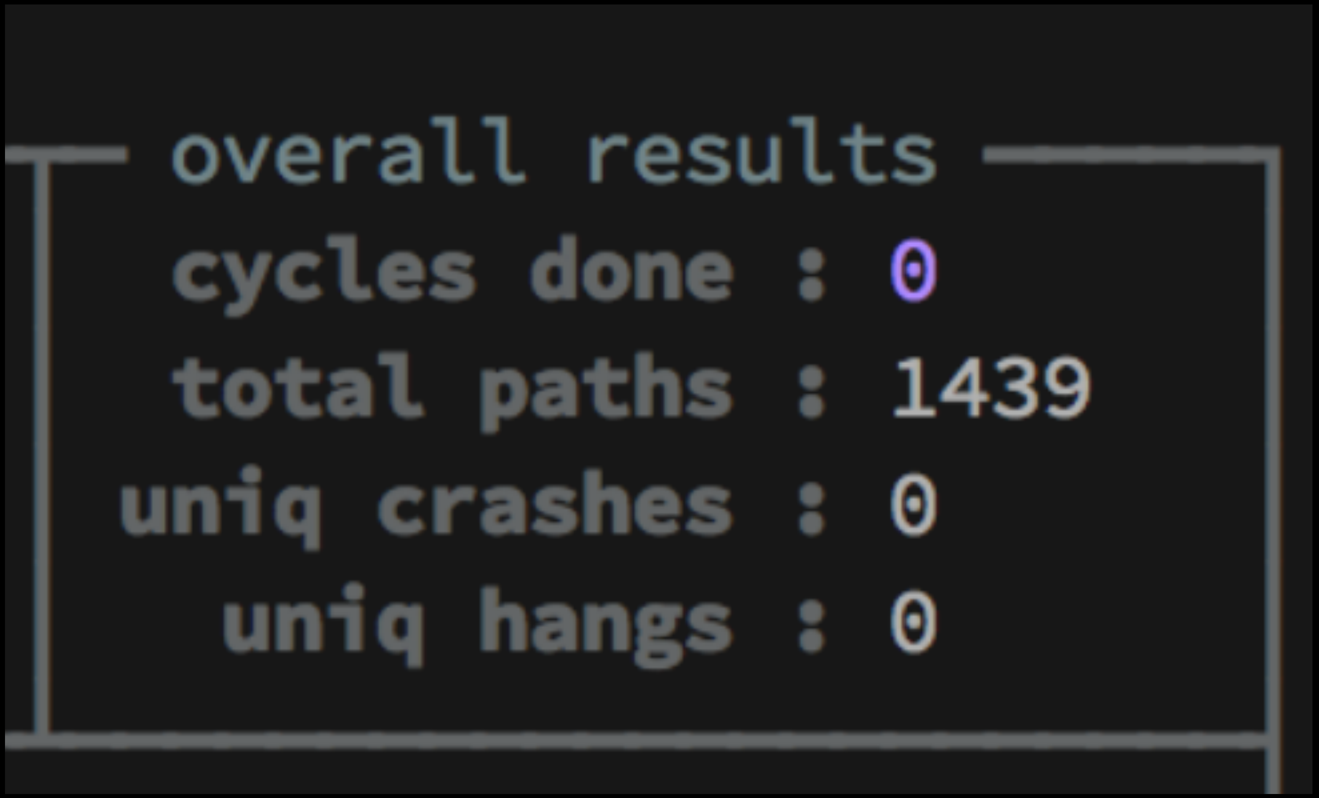
American Fuzzy Lop



clamav [1] [2] [3] [4] [5]	libxml2 [1] [2]	glibc [1]
clang / llvm [1] [2] [3] [4] [5] [6] [7] [8] ...	nasm [1] [2]	ctags [1]
mutt [1]	procmail [1]	fontconfig [1]
pdksh [1] [2]	Qt [1]	wavpack [1]
redis / lua-cmsgpack [1]	taglib [1] [2] [3]	privoxy [1] [2] [3]
perl [1] [2] [3] [4] [5] [6] [7]...	libxmp	radare2 [1] [2]
SleuthKit [1]	fwknop [reported by author]	X.Org [1] [2]
exifprobe [1]	jhead [?]	capnproto [1]
Xerces-C [1]	metacam [1]	djvulibre [1]
exiv [1]	Linux btrfs [1] [2] [3] [4]	Knot DNS [1]
curl [1] [2]	wpa_supplicant [1]	libde265 [reported by author]
dnsmasq [1]	libbpg (1)	lame [1]
libwmf [1]	uudecode [1]	MuPDF [1]
imlib2 [1]	libraw [1]	libbson [1]
libsass [1]	yara [1] [2] [3] [4]	W3C tidy-html5 [1]
VLC [1]	FreeBSD syscons [1] [2] [3]	John the Ripper [1] [2]
screen [1] [2] [3]	tmux [1] [2]	mosh [1]
UPX [1]	indent [1]	openjpeg [1]
MMIX [1]	OpenMPT [1]	rxvt [1] [2]
dhcpcd [1]	Mozilla NSS [1]	Nettle [1]
mbed TLS [1]		



```
┌─ process timing ─────────────────────────────────────
│        run time : 0 days, 0 hrs, 45 min, 47 sec
│   last new path : 0 days, 0 hrs, 3 min, 56 sec
│ last uniq crash : none seen yet
│  last uniq hang : none seen yet
```



```
┌─ overall results ───┐
│ cycles done : 0     │
│ total paths : 1439  │
│uniq crashes : 0     │
│  uniq hangs : 0     │
├─────────────────────┤
```

status_screen.txt

dumb mode

http://blog.techorganic.com/2015/04/10/64-bit-linux-stack-smashing-tutorial-part-1/

Demo

demo - libexif

- https://android.googlesource.com/platform/external/libexif/

https://android.googlesource.com/platform/external/

libcxxrt

libdaemon

libdivsufsort

libdrm

libedit

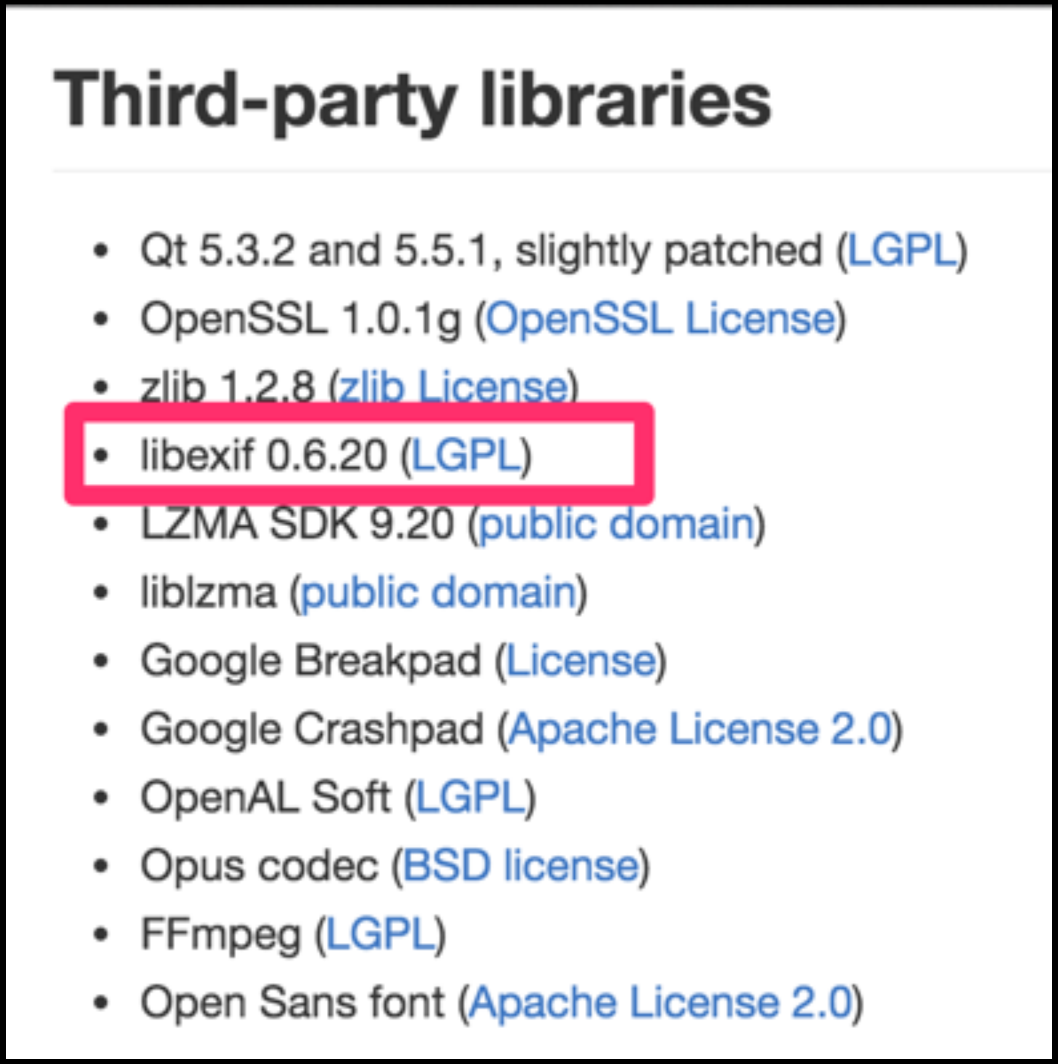
libevent

libexif

libffi

demo - libexif

- https://github.com/telegramdesktop/tdesktop

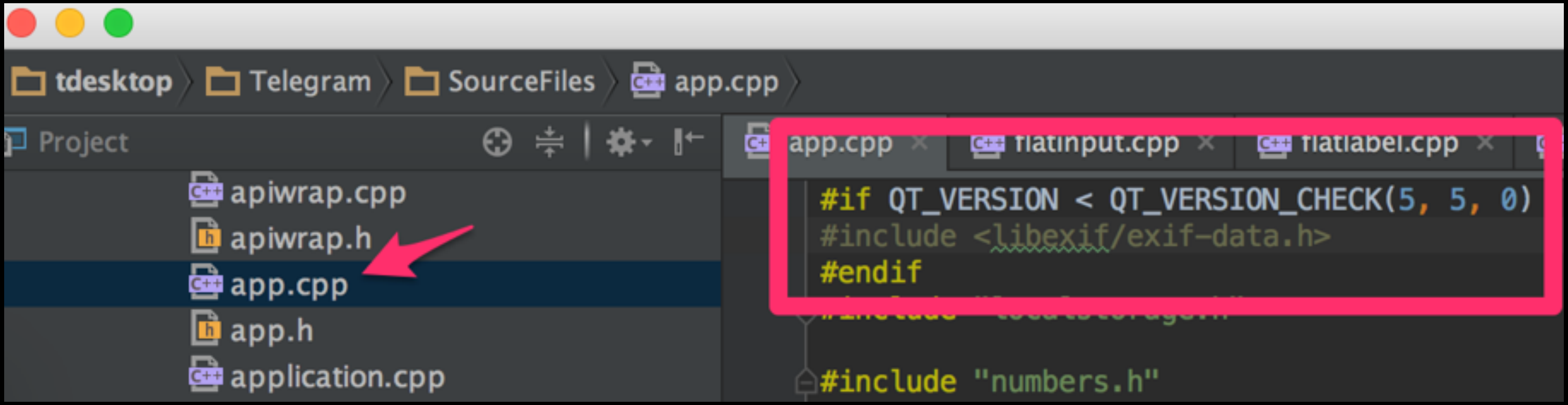


Third-party libraries

- Qt 5.3.2 and 5.5.1, slightly patched (LGPL)
- OpenSSL 1.0.1g (OpenSSL License)
- zlib 1.2.8 (zlib License)
- libexif 0.6.20 (LGPL)
- LZMA SDK 9.20 (public domain)
- liblzma (public domain)
- Google Breakpad (License)
- Google Crashpad (Apache License 2.0)
- OpenAL Soft (LGPL)
- Opus codec (BSD license)
- FFmpeg (LGPL)
- Open Sans font (Apache License 2.0)

demo - libexif

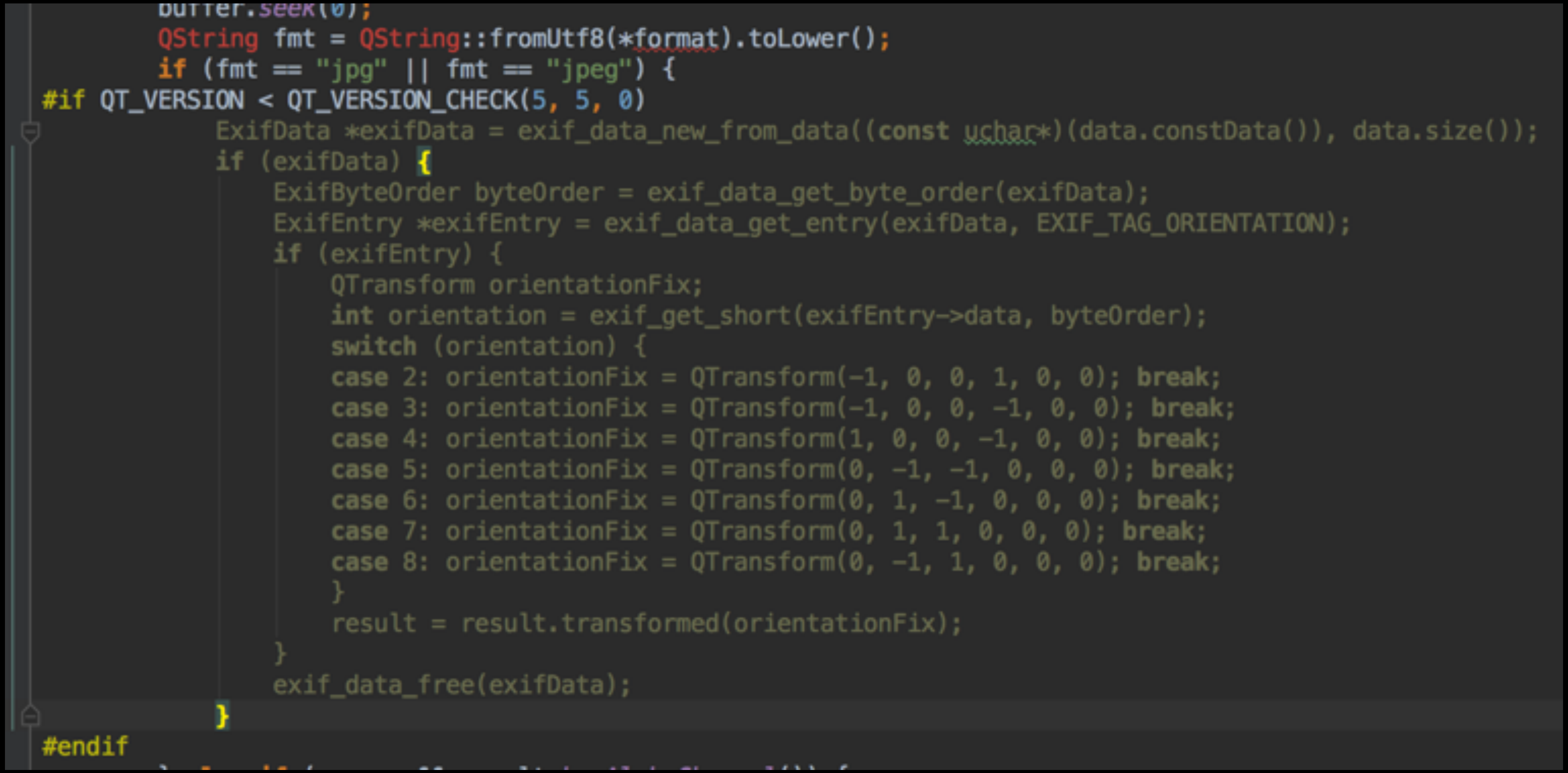
- https://github.com/telegramdesktop/tdesktop



```
#if QT_VERSION < QT_VERSION_CHECK(5, 5, 0)
#include <libexif/exif-data.h>
#endif
#include "localstorage.h"

#include "numbers.h"
```

demo - libexif

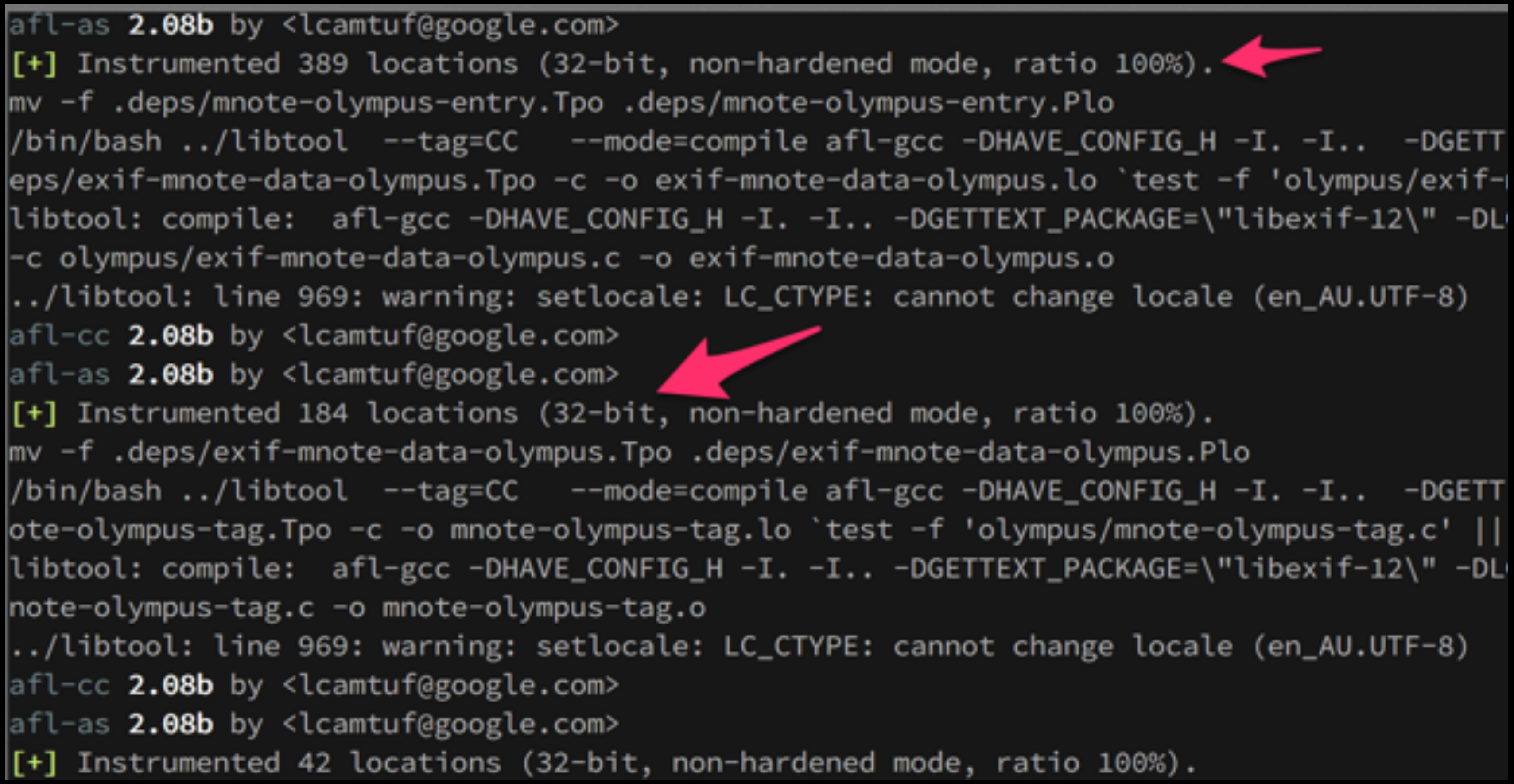
- https://github.com/telegramdesktop/tdesktop



```
        buffer.seek(0);
        QString fmt = QString::fromUtf8(*format).toLower();
        if (fmt == "jpg" || fmt == "jpeg") {
#if QT_VERSION < QT_VERSION_CHECK(5, 5, 0)
            ExifData *exifData = exif_data_new_from_data((const uchar*)(data.constData()), data.size());
            if (exifData) {
                ExifByteOrder byteOrder = exif_data_get_byte_order(exifData);
                ExifEntry *exifEntry = exif_data_get_entry(exifData, EXIF_TAG_ORIENTATION);
                if (exifEntry) {
                    QTransform orientationFix;
                    int orientation = exif_get_short(exifEntry->data, byteOrder);
                    switch (orientation) {
                    case 2: orientationFix = QTransform(-1, 0, 0, 1, 0, 0); break;
                    case 3: orientationFix = QTransform(-1, 0, 0, -1, 0, 0); break;
                    case 4: orientationFix = QTransform(1, 0, 0, -1, 0, 0); break;
                    case 5: orientationFix = QTransform(0, -1, -1, 0, 0, 0); break;
                    case 6: orientationFix = QTransform(0, 1, -1, 0, 0, 0); break;
                    case 7: orientationFix = QTransform(0, 1, 1, 0, 0, 0); break;
                    case 8: orientationFix = QTransform(0, -1, 1, 0, 0, 0); break;
                    }
                    result = result.transformed(orientationFix);
                }
                exif_data_free(exifData);
            }
#endif
```

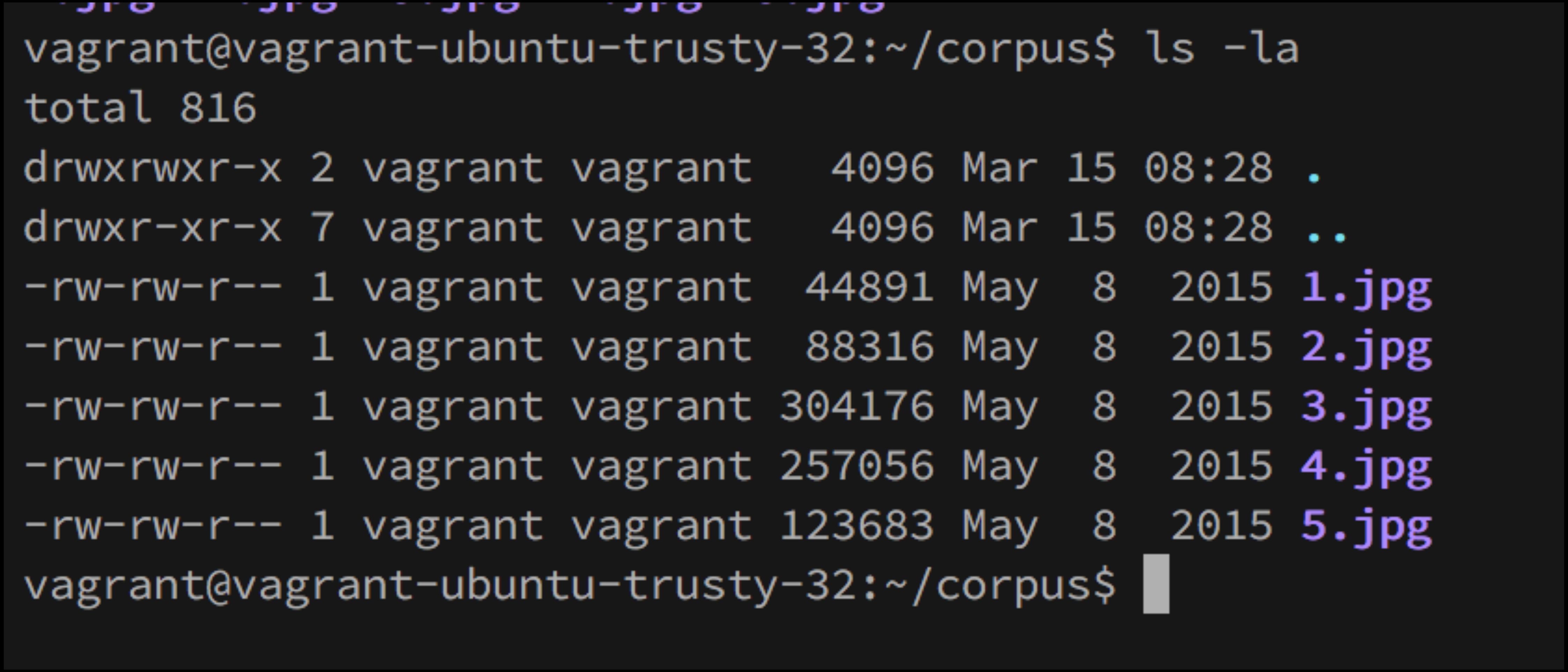
demo - libexif

- Compile - afl-gcc



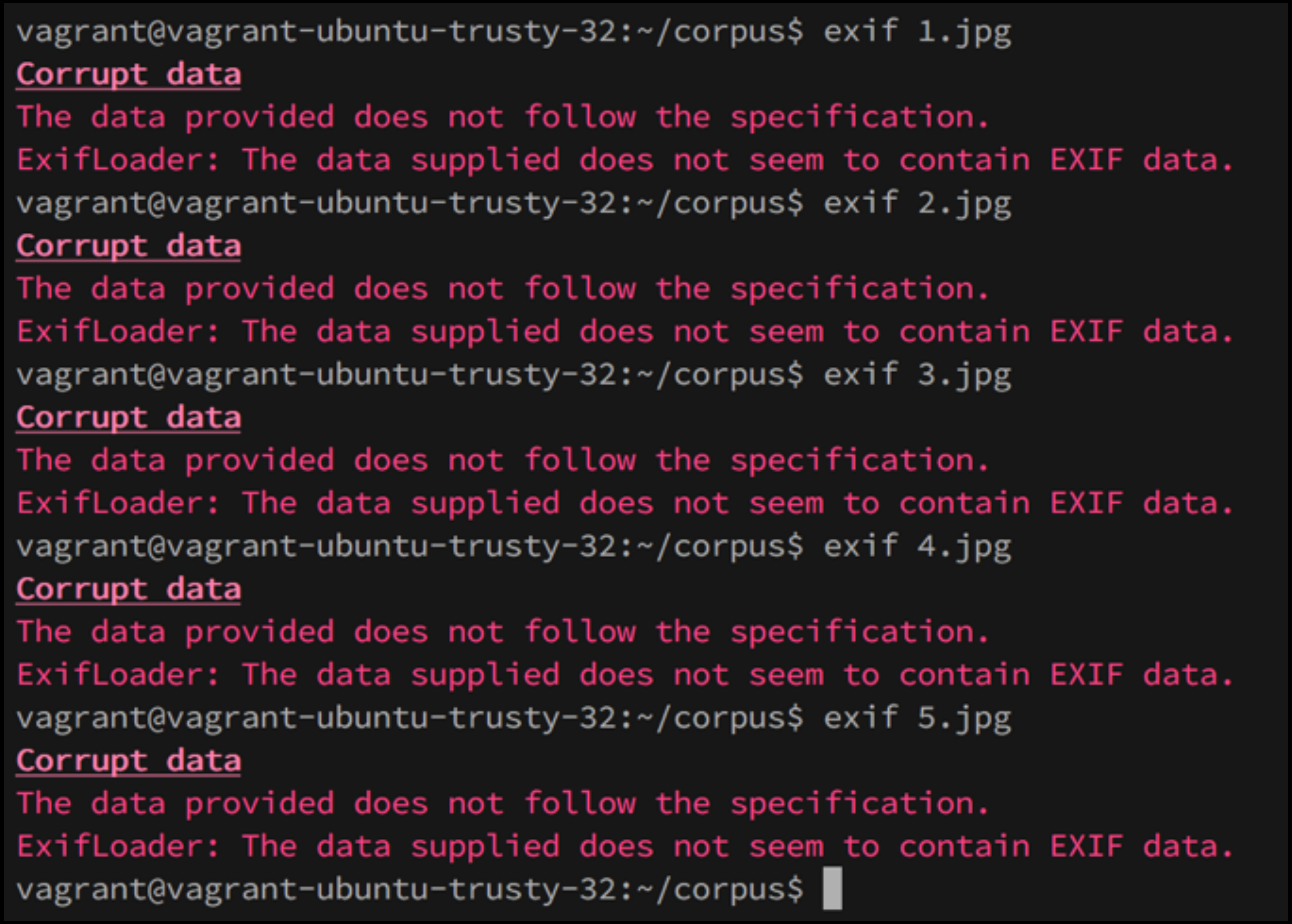
```
afl-as 2.08b by <lcamtuf@google.com>
[+] Instrumented 389 locations (32-bit, non-hardened mode, ratio 100%).
mv -f .deps/mnote-olympus-entry.Tpo .deps/mnote-olympus-entry.Plo
/bin/bash ../libtool  --tag=CC   --mode=compile afl-gcc -DHAVE_CONFIG_H -I. -I..  -DGETT
eps/exif-mnote-data-olympus.Tpo -c -o exif-mnote-data-olympus.lo `test -f 'olympus/exif-
libtool: compile:  afl-gcc -DHAVE_CONFIG_H -I. -I.. -DGETTEXT_PACKAGE=\"libexif-12\" -DL
-c olympus/exif-mnote-data-olympus.c -o exif-mnote-data-olympus.o
../libtool: line 969: warning: setlocale: LC_CTYPE: cannot change locale (en_AU.UTF-8)
afl-cc 2.08b by <lcamtuf@google.com>
afl-as 2.08b by <lcamtuf@google.com>
[+] Instrumented 184 locations (32-bit, non-hardened mode, ratio 100%).
mv -f .deps/exif-mnote-data-olympus.Tpo .deps/exif-mnote-data-olympus.Plo
/bin/bash ../libtool  --tag=CC   --mode=compile afl-gcc -DHAVE_CONFIG_H -I. -I..  -DGETT
ote-olympus-tag.Tpo -c -o mnote-olympus-tag.lo `test -f 'olympus/mnote-olympus-tag.c' ||
libtool: compile:  afl-gcc -DHAVE_CONFIG_H -I. -I.. -DGETTEXT_PACKAGE=\"libexif-12\" -DL
note-olympus-tag.c -o mnote-olympus-tag.o
../libtool: line 969: warning: setlocale: LC_CTYPE: cannot change locale (en_AU.UTF-8)
afl-cc 2.08b by <lcamtuf@google.com>
afl-as 2.08b by <lcamtuf@google.com>
[+] Instrumented 42 locations (32-bit, non-hardened mode, ratio 100%).
```

demo - libexif



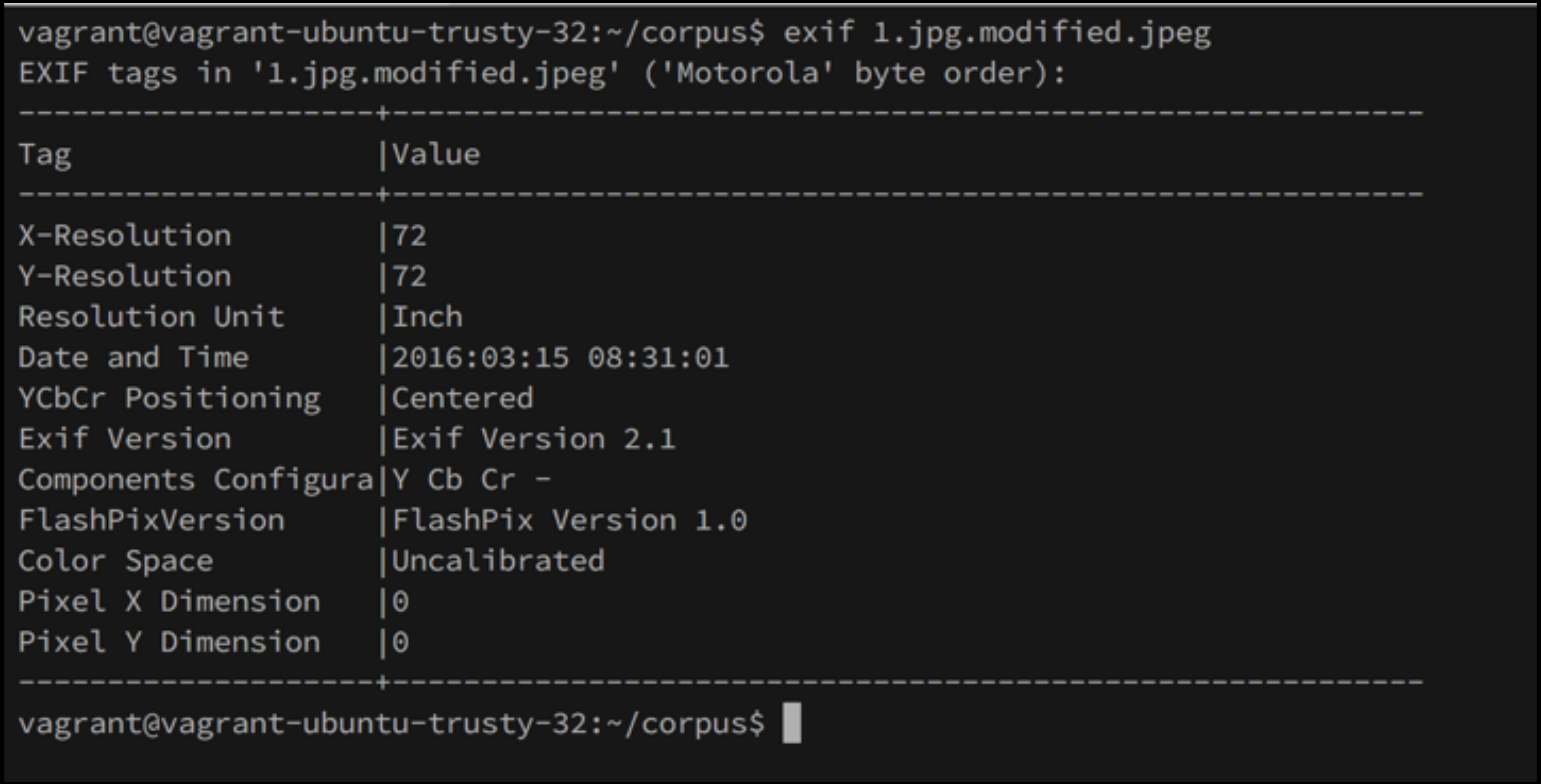
```
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ ls -la
total 816
drwxrwxr-x 2 vagrant vagrant   4096 Mar 15 08:28 .
drwxr-xr-x 7 vagrant vagrant   4096 Mar 15 08:28 ..
-rw-rw-r-- 1 vagrant vagrant  44891 May  8  2015 1.jpg
-rw-rw-r-- 1 vagrant vagrant  88316 May  8  2015 2.jpg
-rw-rw-r-- 1 vagrant vagrant 304176 May  8  2015 3.jpg
-rw-rw-r-- 1 vagrant vagrant 257056 May  8  2015 4.jpg
-rw-rw-r-- 1 vagrant vagrant 123683 May  8  2015 5.jpg
vagrant@vagrant-ubuntu-trusty-32:~/corpus$
```

demo - libexif



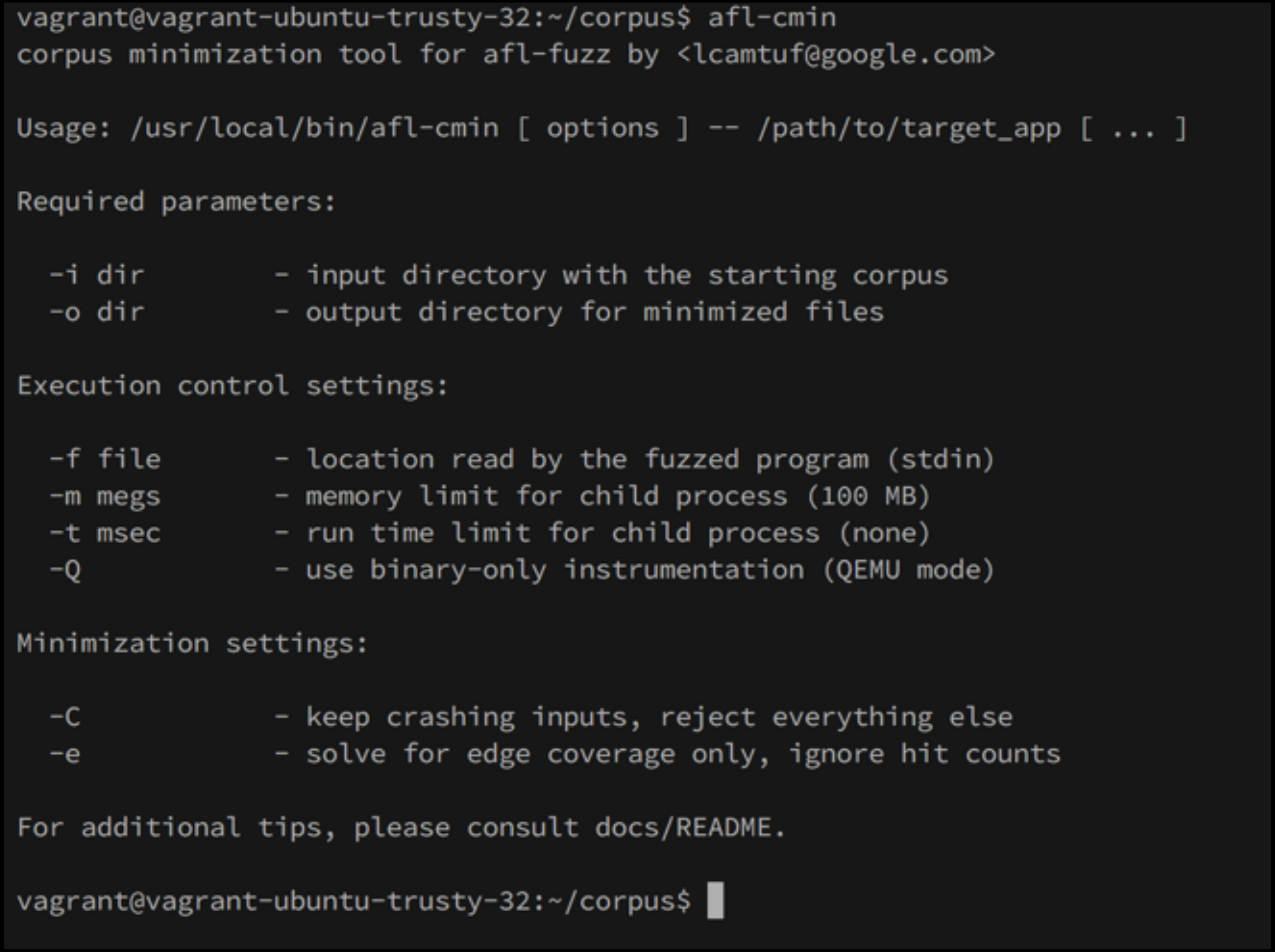
```
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ exif 1.jpg
Corrupt data
The data provided does not follow the specification.
ExifLoader: The data supplied does not seem to contain EXIF data.
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ exif 2.jpg
Corrupt data
The data provided does not follow the specification.
ExifLoader: The data supplied does not seem to contain EXIF data.
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ exif 3.jpg
Corrupt data
The data provided does not follow the specification.
ExifLoader: The data supplied does not seem to contain EXIF data.
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ exif 4.jpg
Corrupt data
The data provided does not follow the specification.
ExifLoader: The data supplied does not seem to contain EXIF data.
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ exif 5.jpg
Corrupt data
The data provided does not follow the specification.
ExifLoader: The data supplied does not seem to contain EXIF data.
vagrant@vagrant-ubuntu-trusty-32:~/corpus$
```

demo - libexif



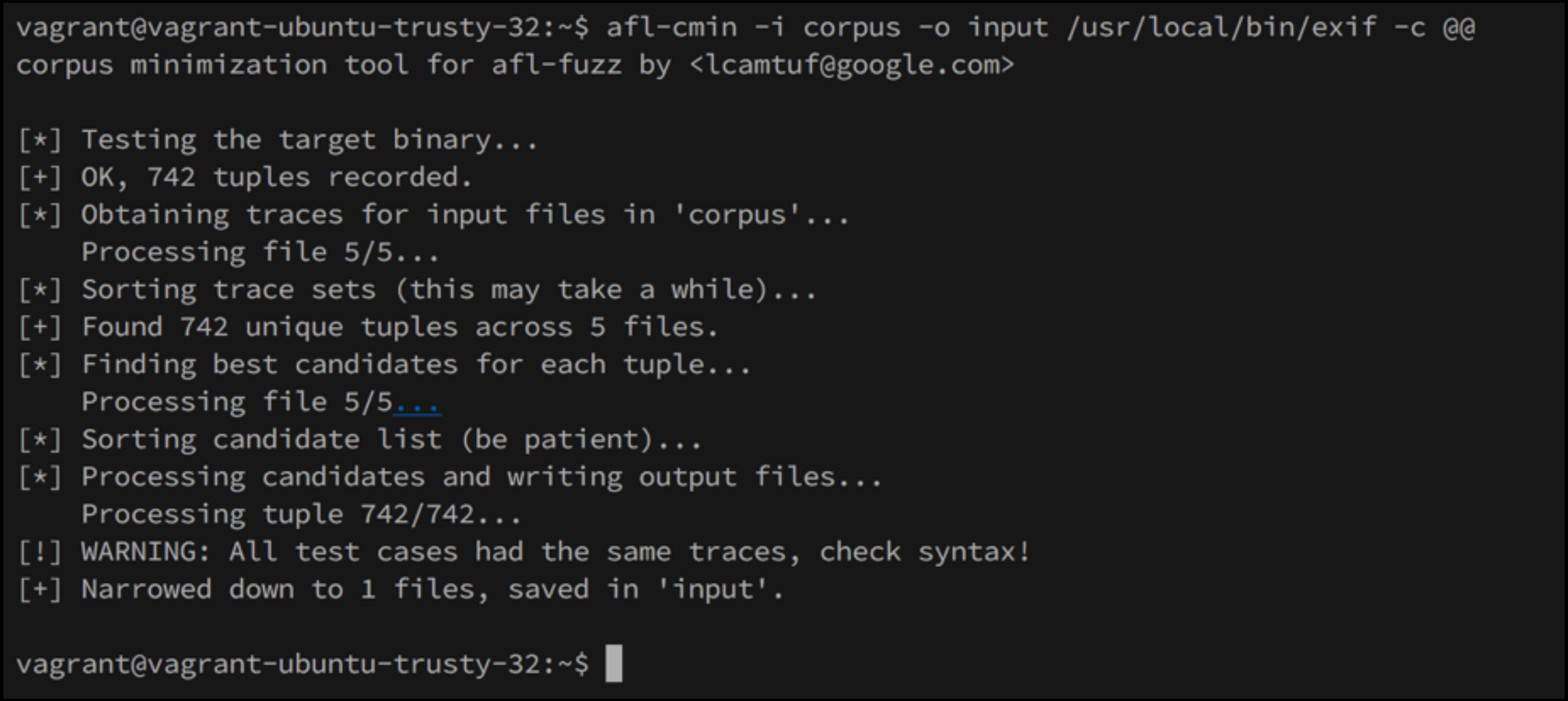
```
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ exif 1.jpg.modified.jpeg
EXIF tags in '1.jpg.modified.jpeg' ('Motorola' byte order):
--------------------+----------------------------------------------------------
Tag                 |Value
--------------------+----------------------------------------------------------
X-Resolution        |72
Y-Resolution        |72
Resolution Unit     |Inch
Date and Time       |2016:03:15 08:31:01
YCbCr Positioning   |Centered
Exif Version        |Exif Version 2.1
Components Configura|Y Cb Cr -
FlashPixVersion     |FlashPix Version 1.0
Color Space         |Uncalibrated
Pixel X Dimension   |0
Pixel Y Dimension   |0
--------------------+----------------------------------------------------------
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ 
```

demo - libexif



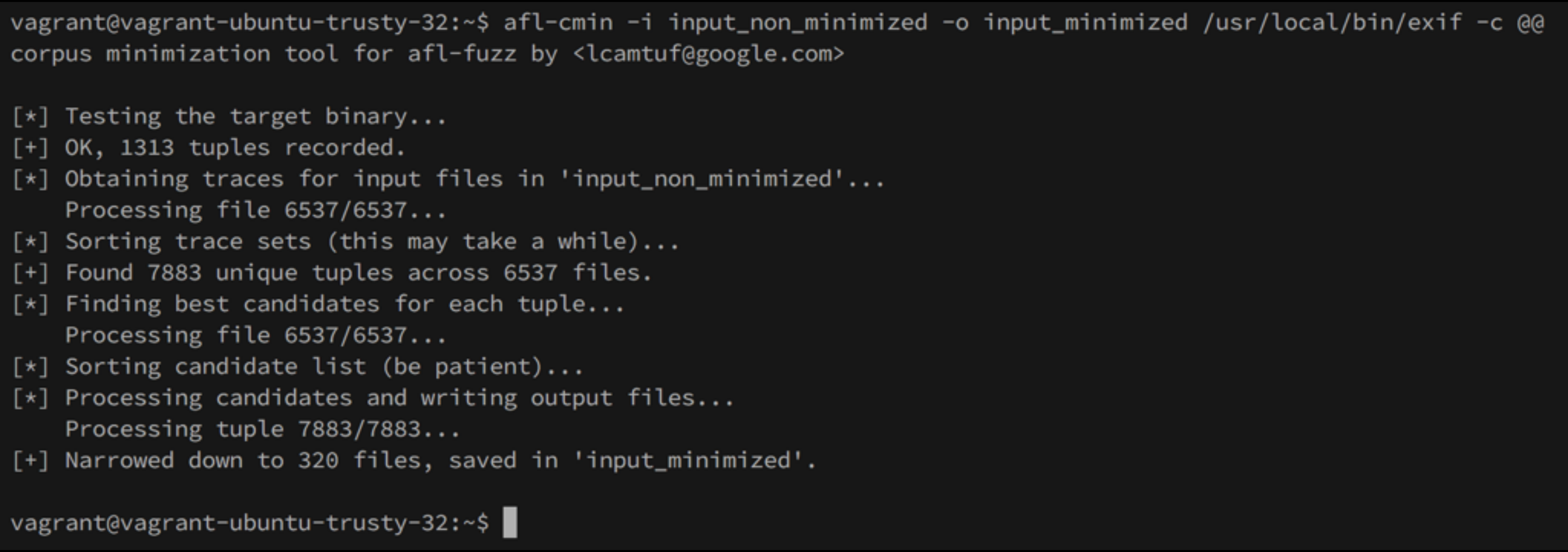
```
vagrant@vagrant-ubuntu-trusty-32:~/corpus$ afl-cmin
corpus minimization tool for afl-fuzz by <lcamtuf@google.com>

Usage: /usr/local/bin/afl-cmin [ options ] -- /path/to/target_app [ ... ]

Required parameters:

  -i dir        - input directory with the starting corpus
  -o dir        - output directory for minimized files

Execution control settings:

  -f file       - location read by the fuzzed program (stdin)
  -m megs       - memory limit for child process (100 MB)
  -t msec       - run time limit for child process (none)
  -Q            - use binary-only instrumentation (QEMU mode)

Minimization settings:

  -C            - keep crashing inputs, reject everything else
  -e            - solve for edge coverage only, ignore hit counts

For additional tips, please consult docs/README.

vagrant@vagrant-ubuntu-trusty-32:~/corpus$ 
```

demo - libexif



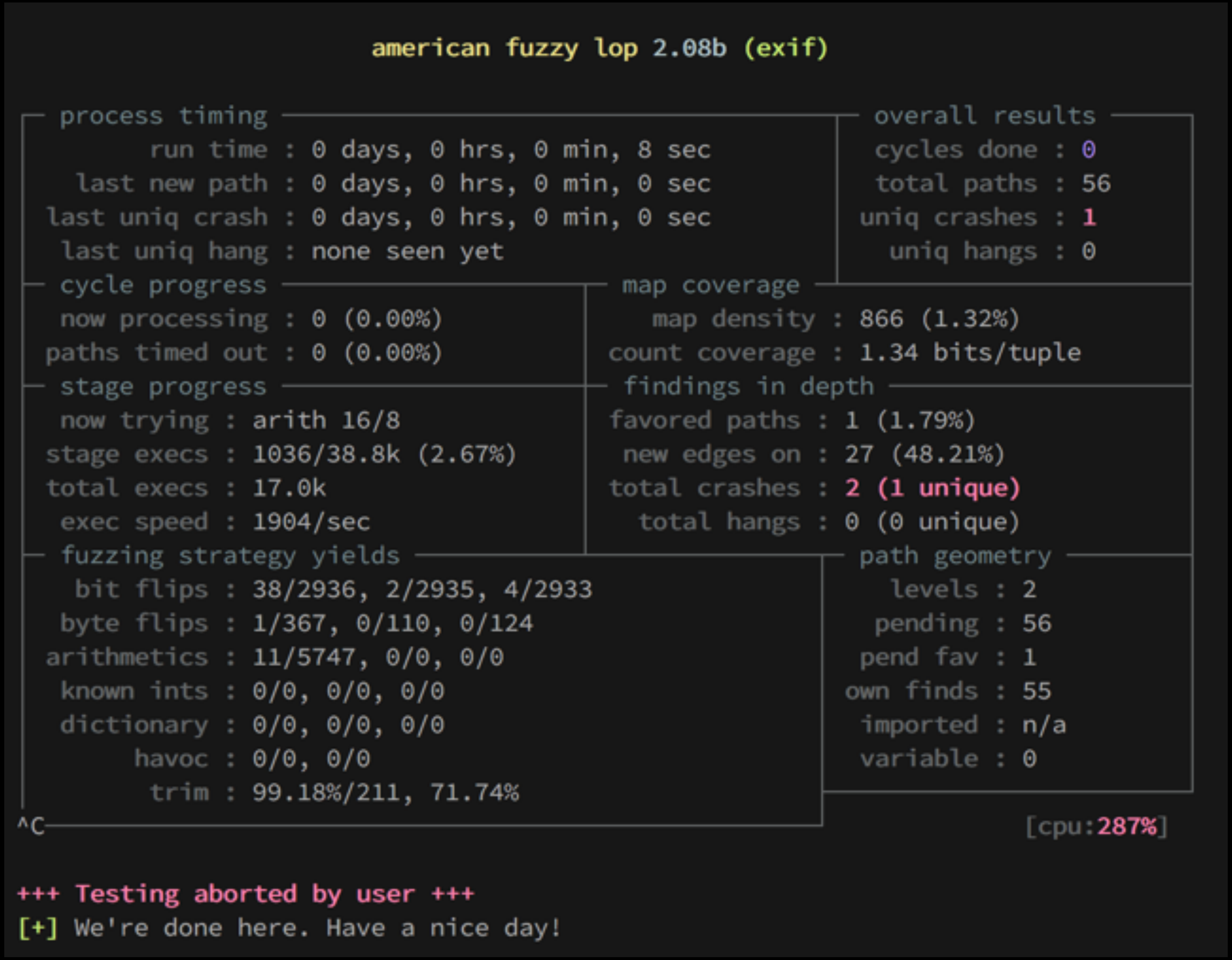
```
vagrant@vagrant-ubuntu-trusty-32:~$ afl-cmin -i corpus -o input /usr/local/bin/exif -c @@
corpus minimization tool for afl-fuzz by <lcamtuf@google.com>

[*] Testing the target binary...
[+] OK, 742 tuples recorded.
[*] Obtaining traces for input files in 'corpus'...
    Processing file 5/5...
[*] Sorting trace sets (this may take a while)...
[+] Found 742 unique tuples across 5 files.
[*] Finding best candidates for each tuple...
    Processing file 5/5...
[*] Sorting candidate list (be patient)...
[*] Processing candidates and writing output files...
    Processing tuple 742/742...
[!] WARNING: All test cases had the same traces, check syntax!
[+] Narrowed down to 1 files, saved in 'input'.

vagrant@vagrant-ubuntu-trusty-32:~$ 
```

demo - libexif



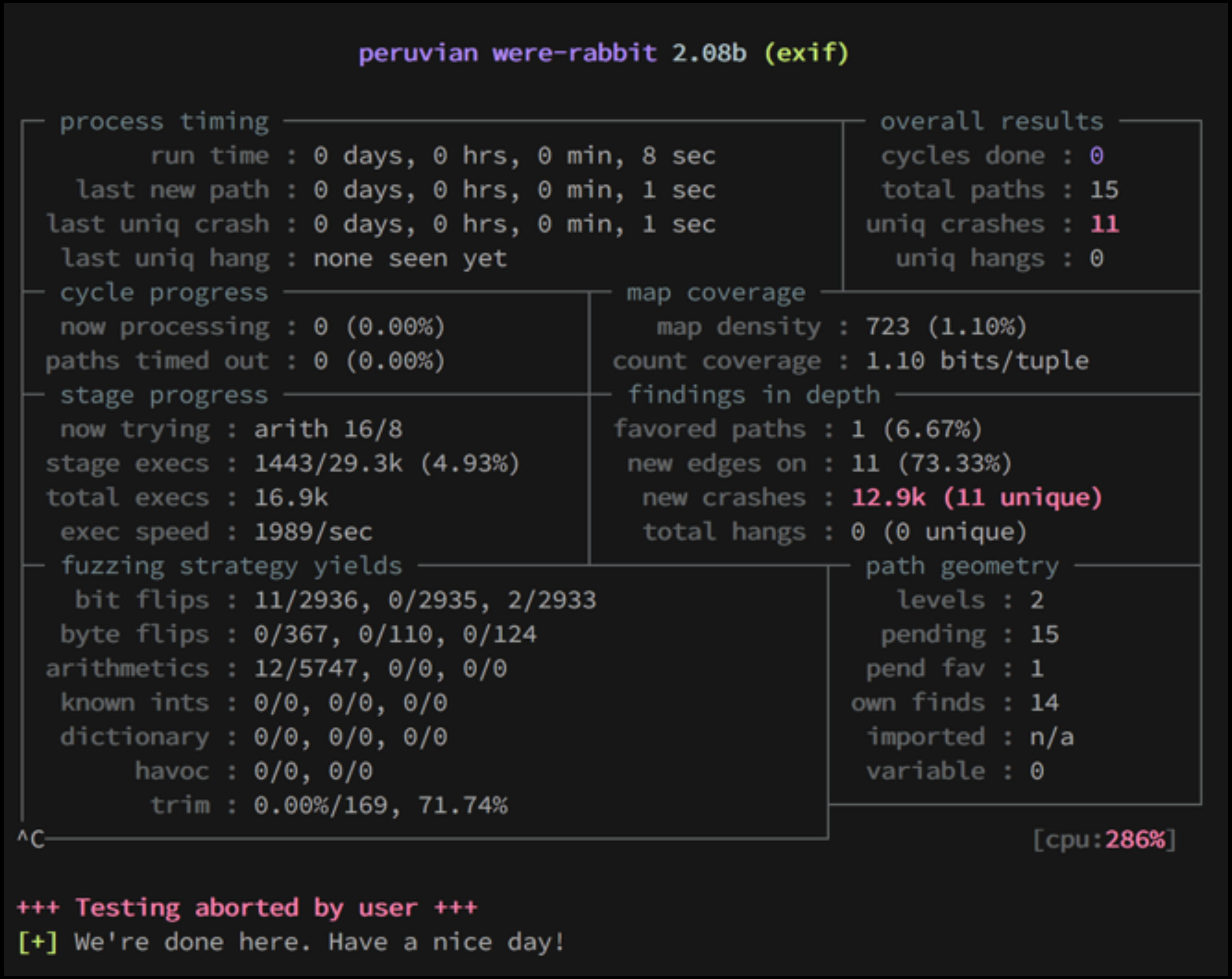
```
vagrant@vagrant-ubuntu-trusty-32:~$ afl-cmin -i input_non_minimized -o input_minimized /usr/local/bin/exif -c @@
corpus minimization tool for afl-fuzz by <lcamtuf@google.com>

[*] Testing the target binary...
[+] OK, 1313 tuples recorded.
[*] Obtaining traces for input files in 'input_non_minimized'...
    Processing file 6537/6537...
[*] Sorting trace sets (this may take a while)...
[+] Found 7883 unique tuples across 6537 files.
[*] Finding best candidates for each tuple...
    Processing file 6537/6537...
[*] Sorting candidate list (be patient)...
[*] Processing candidates and writing output files...
    Processing tuple 7883/7883...
[+] Narrowed down to 320 files, saved in 'input_minimized'.

vagrant@vagrant-ubuntu-trusty-32:~$ 
```

demo - libexif

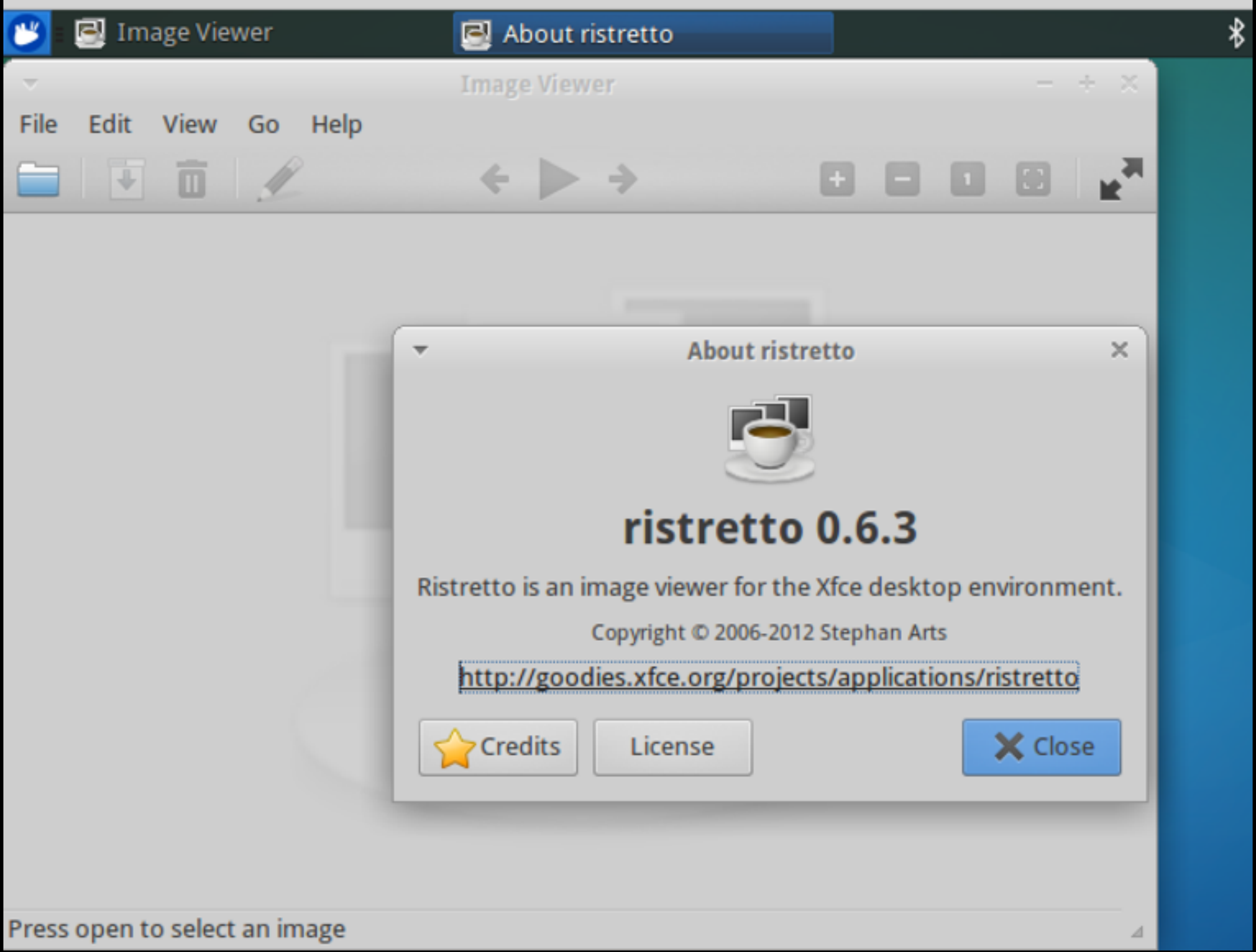


```
                   american fuzzy lop 2.08b (exif)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 0 min, 8 sec        │  cycles done : 0      │
│   last new path : 0 days, 0 hrs, 0 min, 0 sec        │  total paths : 56     │
│ last uniq crash : 0 days, 0 hrs, 0 min, 0 sec        │ uniq crashes : 1      │
│  last uniq hang : none seen yet                      │   uniq hangs : 0      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 0 (0.00%)         │    map density : 866 (1.32%)          │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.34 bits/tuple      │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : arith 16/8            │ favored paths : 1 (1.79%)              │
│ stage execs : 1036/38.8k (2.67%)    │  new edges on : 27 (48.21%)            │
│ total execs : 17.0k                 │ total crashes : 2 (1 unique)           │
│  exec speed : 1904/sec              │   total hangs : 0 (0 unique)           │
├─ fuzzing strategy yields ───────────┴───────────────┬─ path geometry ────────┤
│   bit flips : 38/2936, 2/2935, 4/2933               │    levels : 2          │
│  byte flips : 1/367, 0/110, 0/124                   │   pending : 56         │
│ arithmetics : 11/5747, 0/0, 0/0                     │  pend fav : 1          │
│  known ints : 0/0, 0/0, 0/0                         │ own finds : 55         │
│  dictionary : 0/0, 0/0, 0/0                         │  imported : n/a        │
│       havoc : 0/0, 0/0                              │  variable : 0          │
│        trim : 99.18%/211, 71.74%                    ├────────────────────────┘
^C────────────────────────────────────────────────────┘             [cpu:287%]

+++ Testing aborted by user +++
[+] We're done here. Have a nice day!
```

demo - libexif



```
                  peruvian were-rabbit 2.08b (exif)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 0 min, 8 sec        │  cycles done : 0      │
│   last new path : 0 days, 0 hrs, 0 min, 1 sec        │  total paths : 15     │
│ last uniq crash : 0 days, 0 hrs, 0 min, 1 sec        │ uniq crashes : 11     │
│  last uniq hang : none seen yet                      │   uniq hangs : 0      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 0 (0.00%)         │    map density : 723 (1.10%)           │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.10 bits/tuple       │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : arith 16/8            │ favored paths : 1 (6.67%)              │
│ stage execs : 1443/29.3k (4.93%)    │  new edges on : 11 (73.33%)            │
│ total execs : 16.9k                 │ new crashes : 12.9k (11 unique)        │
│  exec speed : 1989/sec              │  total hangs : 0 (0 unique)            │
├─ fuzzing strategy yields ───────────┴───────────────┬─ path geometry ────────┤
│   bit flips : 11/2936, 0/2935, 2/2933               │    levels : 2          │
│  byte flips : 0/367, 0/110, 0/124                   │   pending : 15         │
│ arithmetics : 12/5747, 0/0, 0/0                     │  pend fav : 1          │
│  known ints : 0/0, 0/0, 0/0                         │ own finds : 14         │
│  dictionary : 0/0, 0/0, 0/0                         │  imported : n/a        │
│       havoc : 0/0, 0/0                              │  variable : 0          │
│        trim : 0.00%/169, 71.74%                     ├────────────────────────┘
^C────────────────────────────────────────────────────┘             [cpu:286%]

+++ Testing aborted by user +++
[+] We're done here. Have a nice day!
```



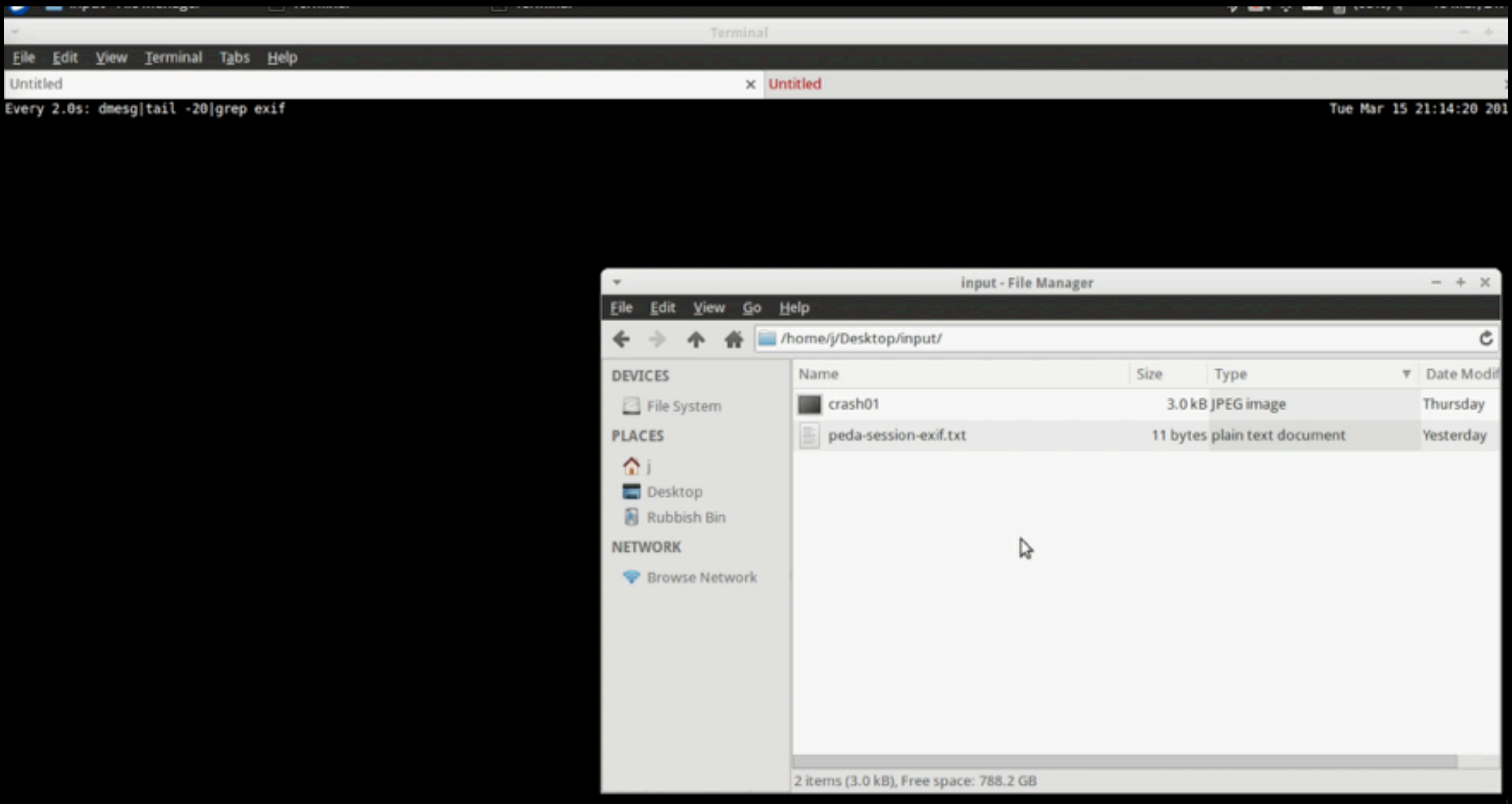
ristretto 0.6.3

Ristretto is an image viewer for the Xfce desktop environment.

Copyright © 2006-2012 Stephan Arts

http://goodies.xfce.org/projects/applications/ristretto

Press open to select an image



```
Every 2.0s: dmesg|tail -20|grep exif                                                          Tue Mar 15 21:14:20 201
```

/home/j/Desktop/input/

Name	Size	Type	Date Modif
crash01	3.0 kB	JPEG image	Thursday
peda-session-exif.txt	11 bytes	plain text document	Yesterday

2 items (3.0 kB), Free space: 788.2 GB

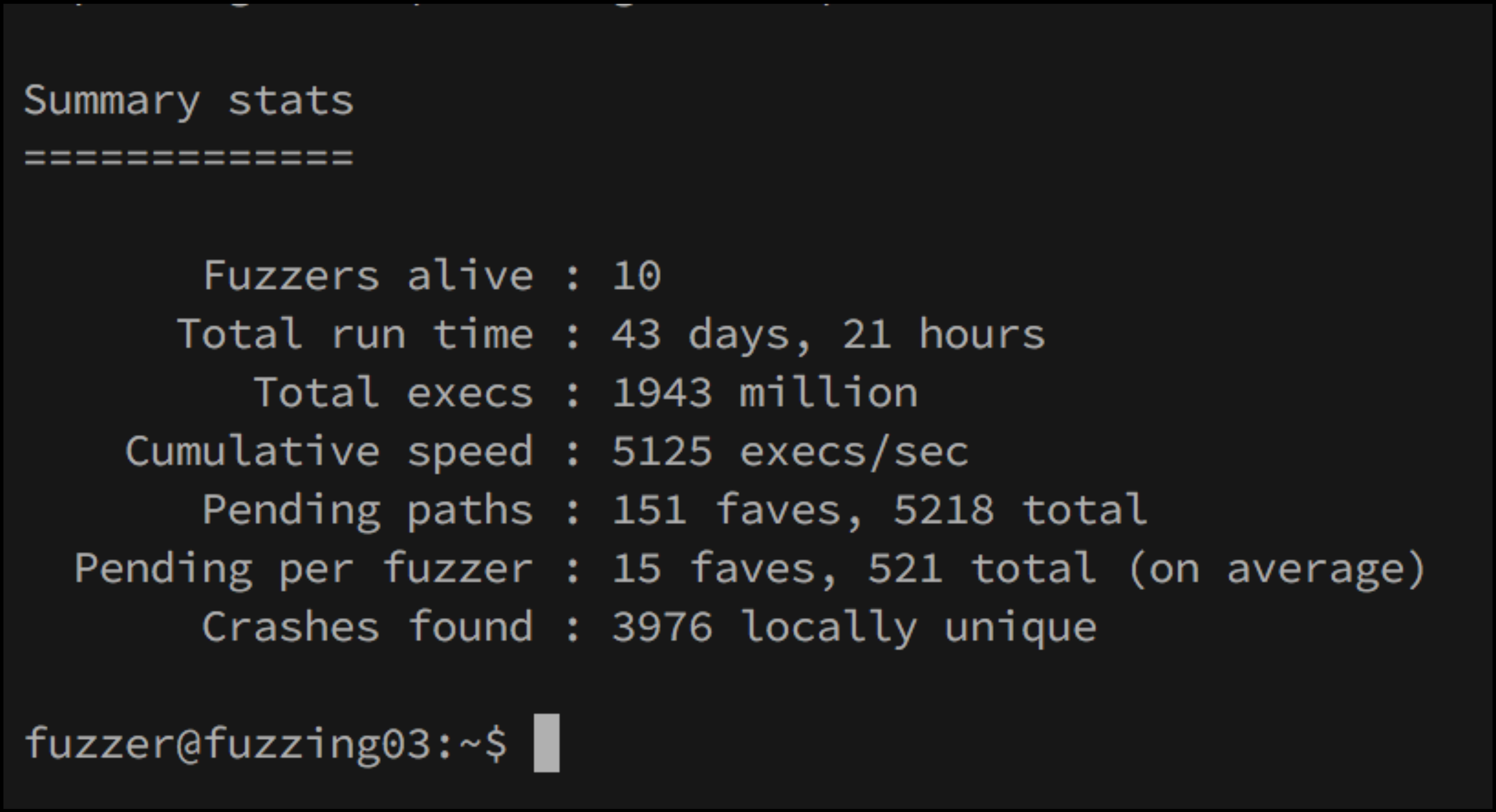
Demo

http://libexif.sourceforge.net/

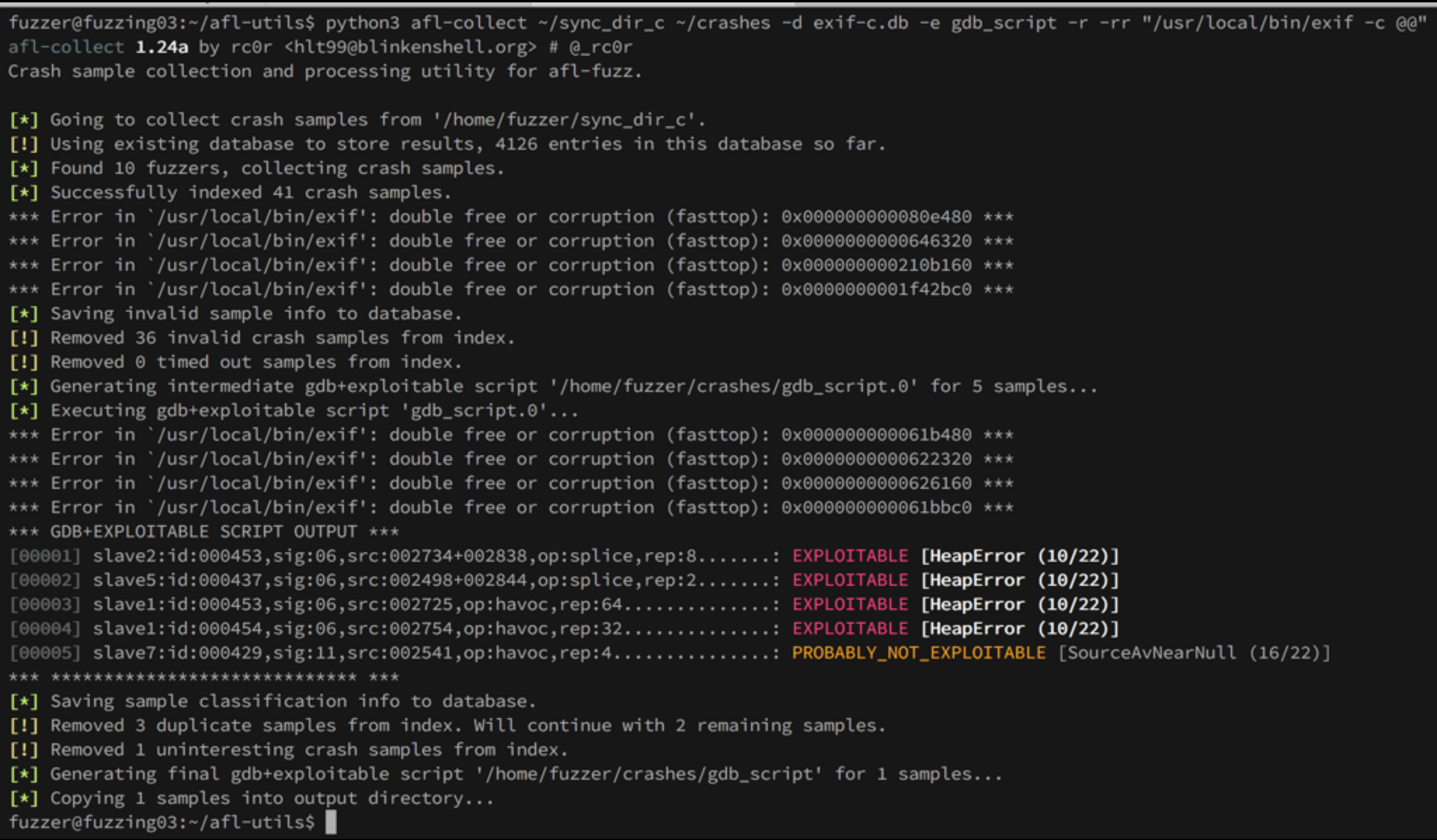
my numbers

1. 4 weeks running
2. tons of crash
    1. libexif
    2. binutils
    3. perl
    4. otool
    5. poppler
    6. websocketpp
3. until now 4 security bug (I guess)

demo - libexif



```
Summary stats
=============

       Fuzzers alive : 10
      Total run time : 43 days, 21 hours
         Total execs : 1943 million
    Cumulative speed : 5125 execs/sec
       Pending paths : 151 faves, 5218 total
  Pending per fuzzer : 15 faves, 521 total (on average)
       Crashes found : 3976 locally unique

fuzzer@fuzzing03:~$ █
```



```
fuzzer@fuzzing03:~/afl-utils$ python3 afl-collect ~/sync_dir_c ~/crashes -d exif-c.db -e gdb_script -r -rr "/usr/local/bin/exif -c @@"
afl-collect 1.24a by rc0r <hlt99@blinkenshell.org> # @_rc0r
Crash sample collection and processing utility for afl-fuzz.

[*] Going to collect crash samples from '/home/fuzzer/sync_dir_c'.
[!] Using existing database to store results, 4126 entries in this database so far.
[*] Found 10 fuzzers, collecting crash samples.
[*] Successfully indexed 41 crash samples.
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x000000000080e480 ***
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x0000000000646320 ***
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x000000000210b160 ***
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x0000000001f42bc0 ***
[*] Saving invalid sample info to database.
[!] Removed 36 invalid crash samples from index.
[!] Removed 0 timed out samples from index.
[*] Generating intermediate gdb+exploitable script '/home/fuzzer/crashes/gdb_script.0' for 5 samples...
[*] Executing gdb+exploitable script 'gdb_script.0'...
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x000000000061b480 ***
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x0000000000622320 ***
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x0000000000626160 ***
*** Error in `/usr/local/bin/exif': double free or corruption (fasttop): 0x000000000061bbc0 ***
*** GDB+EXPLOITABLE SCRIPT OUTPUT ***
[00001] slave2:id:000453,sig:06,src:002734+002838,op:splice,rep:8.......: EXPLOITABLE [HeapError (10/22)]
[00002] slave5:id:000437,sig:06,src:002498+002844,op:splice,rep:2.......: EXPLOITABLE [HeapError (10/22)]
[00003] slave1:id:000453,sig:06,src:002725,op:havoc,rep:64..............: EXPLOITABLE [HeapError (10/22)]
[00004] slave1:id:000454,sig:06,src:002754,op:havoc,rep:32..............: EXPLOITABLE [HeapError (10/22)]
[00005] slave7:id:000429,sig:11,src:002541,op:havoc,rep:4...............: PROBABLY_NOT_EXPLOITABLE [SourceAvNearNull (16/22)]
*** ***************************** ***
[*] Saving sample classification info to database.
[!] Removed 3 duplicate samples from index. Will continue with 2 remaining samples.
[!] Removed 1 uninteresting crash samples from index.
[*] Generating final gdb+exploitable script '/home/fuzzer/crashes/gdb_script' for 1 samples...
[*] Copying 1 samples into output directory...
fuzzer@fuzzing03:~/afl-utils$
```

third party projects

- alf-utils https://github.com/rc0r/afl-utils
  - triage
- preeny https://github.com/zardus/preeny
  - network fuzzing

bnagy party projects

- https://github.com/bnagy/

“Done”

https://github.com/jespinhara/ruxmon

@jespinhara

References

- http://s1m0n.dft-labs.eu/files/alligator_2015.pdf
- http://foxglovesecurity.com/2016/03/15/fuzzing-workflows-a-fuzz-job-from-start-to-finish/
- https://github.com/bnagy/
- https://vimeo.com/129701495 Fuzzing OSX at Scale
- https://www.fastly.com/blog/how-fuzz-server-american-fuzzy-lop
- https://blog.hboeck.de/archives/868-How-Heartbleed-couldve-been-found.html
- https://ricochet.im/files/ricochet-ncc-audit-2016-01.pdf