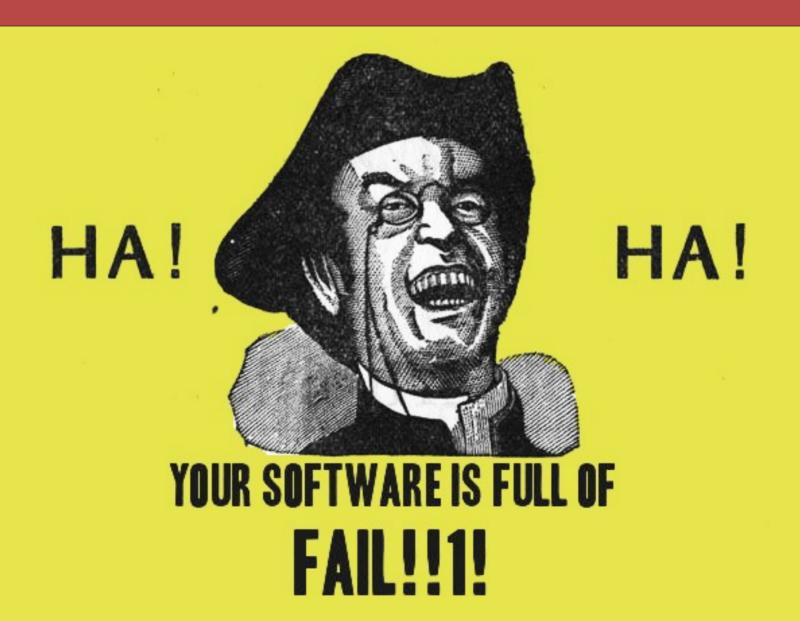
zzuf – multiple purpose fuzzer



input fuzz testing

- The idea
 - randomly alter a program's input
- Expose bugs
 - user-contributed data (intarweb, e-mail)
 - file parsers, interpreters...
 - security implications more than often
- No magical solution
 - only finds simple bugs, not all of them

what zzuf does

- Input corruption
 - randomly sets or unsets bits
 - uses a random seed
- Fully automated
 - can work on the fly
 - no costly file generation
 - works with DVDs, network, stdin
 - reproducible behaviour
 - on consecutive runs, with different programs

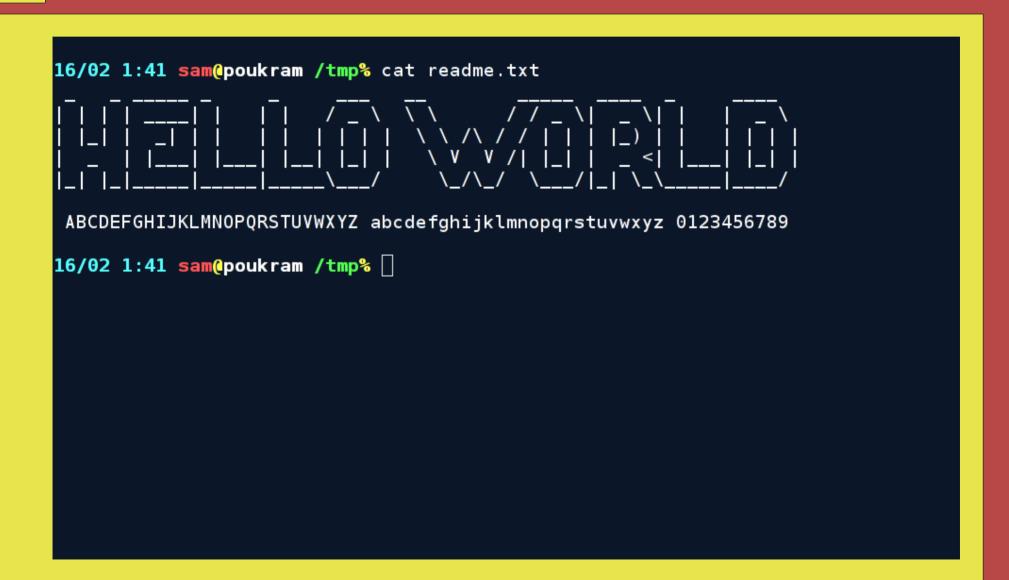
how zzuf works

- Controlling zzuf binary
 - forks tested program
 - checks **stdout**, exit value, signals...
- LD_PRELOAD mechanism
 - intercepts file reading functions
 - open(), read(), fopen(), fread()...
 - plans for zlib functions, too
 - also malloc() to check memory usage
 - more portable than ptrace, kernel land...

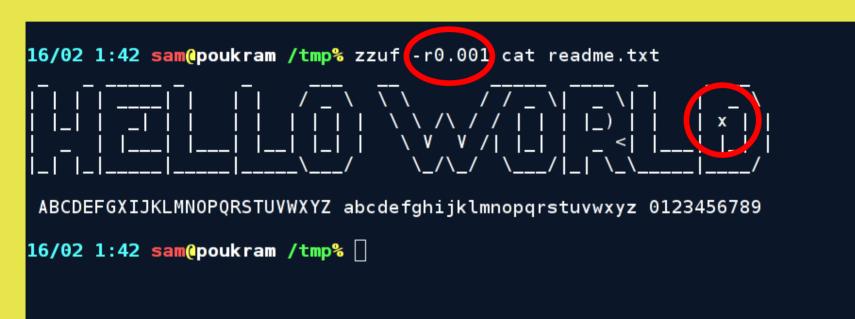
example #1

Using zzuf and cat

trivial use of cat



use cat through zzuf



-r0.001 means "fuzz 0.1% of the files we read"

fuzz more bits (1%)



ABCDEFGHIZKLMNOPYRS@U⊕WXYZ abcdefghijklmnopqrstuvwxyz 0123456789

16/02 1:41 sam@poukram /tmp% 🗌

-r0.01 means "fuzz 1% of the files we read"

fuzz even more bits (3.8%)

-r0.038 means "fuzz 3.8% of the files we read"

refuse non-printable bytes

```
16/02 1:46 sam@poukram /tmp% zzuf -r0.038 -R '\x00-\x20\x7f-\xff' cat readme.txt
$ACCDEFGHIJKD}NOPQRSTUVWXxZ abc$eggh)zklmnoPqzst}vwxyr$0123456789
16/02 1:46 sam@poukram /tmp% 🗌
   -R\x00-\x20 means "do not
   generate characters in the
         range 0x00 to 0x20"
```

example #2

• Using zzuf and file

trivial use of file

```
16/02 2:07 sam@poukram /tmp% file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux
2.4.1, dynamically linked (uses shared libs), stripped
16/02 2:07 sam@poukram /tmp% 🗌
```

Using -d (debug)

```
16/02 2:09 sam@poukram /tmp% zzuf -d -r0.001 file /bin/ls
** zzuf debug ** libzzuf initiatised for PID 27060
** zzuf debug ** fopen64("/etc/magic", "r") = [3]
** zzuf debug ** †gets(Oxhfhea6ef 8192, [3]) = 0xbfbea6ef
** zzuf debug ** fgets(0xbfbea6ef, 8192, [3]) = 0xbfbea6ef
** zzuf debug ** fgets(0xbfbea6ef, 8192, [3]) = 0xbfbea6ef
** zzuf debug ** fgets(0xbfbea6ef, 8192, [3]) = NULL
** zzuf debug ** fcloss([5]) = v
** zzuf debug ** open64("/usr/share/file/magic.mgc", 0) = 3
** zzuf debug ** mmapo+(11012224 3, 2, 3, 3) = 0xb792b008 "\x1c\x04\x1e\xf
** zzuf debug ** close(3) = 0
** zzuf debug ** open64("/bin/ls", 0) = 3
** zzuf debug ** read(3, 0xb78e3008, 262144) = 77352 "\x7fELF...
** zzuf debug ** close(3) = 0
/bin/ls: ERROR: cannot happen: invalid relation `@'
16/02 2:09 sam@poukram /tmp% 🗌
   Files in /etc and /usr/share
            should not be fuzzed
```

Using -E (exclude)

```
16/02 2:10 sam@poukram /tmp% zzuf -r0.001 -E/etc -E/usr/share file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80360, version 1 (SY3v), dynamically l
inked (uses shared libs), stripped
16/02 2:10 sam@poukram /tmp% [
```

Files in /etc and /usr/share are now properly ignored (see also -c and -I)

Using -s (seed)

```
16/02 2:30 sam@poukram /tmp% zzuf -s0:5 -r0.01 -E/etc -E/usr/share file /bin/ls /bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), corrupted section header size /bin/ls: ELF 32-bit LSB executable, (SYSV), statically linked (uses shared libs), stripped /bin/ls: data /bin/ls: data /bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), bad note name size 0x80000061, dynamically linked, stripped 16/02 2:30 sam@poukram /tmp% []
```

-s0:5 means "use random seeds 0, 1, 2, 3 and 4"

Using -r with ranges

```
16/02 2:34 sam@poukram /tmp% zzuf -s0:5 -r0.001:0.05 -E/etc -E/usr/share file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), stripped
/bin/ls: data
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), stripped
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), corrupted program header size, stripped
/bin/ls: data
16/02 2:34 sam@poukram /tmp%
```

-r0.001:0.05 means
"choose fuzzing ratios in the 0.1% - 5% range"

example #3

Finding real bugs

giftopnm

```
16/02 0:13 samepockram /tmp% zzuf -q -s0:1000 -r0.001:0.1 giftopnm image.gif zzuf[s=19,r=0.001:0.1] signal 11 (SIGSEGV)
[1] 5328 exit 1 zzuf -q -s0:1000 -r0.001:0.1 giftopnm image.gif
16/02 9:13 sam@poukram /tmp% zzuf -s19 -r0.001:0.1 < image.gif > fuzzed.gif
16/02 9:13 sam@poukram /tmp% giftopnm ruzzed.gif
[1] 5389 segmentation fault giftopnm fuzzed.gif
16/02 9:13 sam@poukram /tmp%
```

Use -q to avoid flooding

Use the same -r and -s flags to generate a file that reproduces the behaviour

antiword

```
16/02 9:06 sam@poukram /tmp% zzuf -C10 -g -s0:10000 -r0.001:0.02 -M1000 antiword
worddocument.doc
*** glibc detected *** double free or corruption (!prev): 0x0807a020 ***
zzuf[s=19,r=0.001:0.02]: signal 6 (SIGAPRT
zzuf[s=98,r=0.001:0.02]: signal II (51032
zzuf[s=109,r=0.001:0.02]:(signal 11 (SIGSEGV))
*** glibc detected *** double free or corruption (out): 0x0807a020 ***
zzuf[s=140,r=0.001:0.02]: signal 6 (SIGABRT)
^{***} glibc detected ^{***} double free or corruption (out): 0x0807a020
zzuf[s=188,r=0.001:0.02]: signal o (SIGABRI)
zzuf[s=214,r=0.001:0.02]: signal 9 (memory exceeded?)
*** glibc detected *** double five or corruption (!prev): 0x0807a020 ***
zzuf[s=256,r=0.001:0.02]: signal 6 (SIGABRT)
zzuf[s=269,r=0.001:0.02]: signal 11 (SIGSEGV)
zzuf[s=270,r=0.001:0.02]: signal 9 (memory exceeded?)
zzuf[s=283,r=0.001:0.02]: signal 9 (memory exceeded?)
[1]
      2818 exit 1 zzuf -C10 -q -s0:10000 -r0.001:0.02 -M1000 antiword wordd
ocument.doc
16/02 9:06 sam@poukram /tmp%
Many different ways to crash
```

mplayer/ffmpeg

Use -b to only fuzz parts of the input Innocent-looking videos can cause a DoS or worse

other features

- Fork and parallelise (¬₣)
- Detect stuck processes
 - Set maximum memory allocation (-M)
 - Set maximum running time (-T)
 - Set maximum stdout output (-B)
- Fuzz network intput (-n)
- Fuzz stdin (-i)

zzuf's future

- Context-dependent fuzzing
 - ignore or recompute CRCs
 - divert the zlib library, too
- Finish the Windows® port
 - help needed
- Make it easier to test GUI applications
- Attach to a debugger

http://sam.zoy.org/zzuf/

