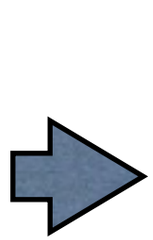
Fuzzing 101

Security Testing • Spring 2017

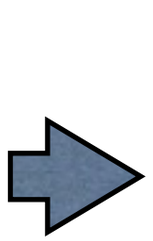
Andreas Zeller, Saarland University

Infinite Monkey Theorem

Random Testing



Program under Test



Oracle

Fuzzing

Random Testing at the System Level

“ab’d&gfdfggg”

Fuzzing

Random Testing at the System Level

1989 Paper

An Empirical Study of the Reliability of UNIX Utilities

Barton P. Miller
bart@cs.wisc.edu

Lars Fredriksen
L.Fredriksen@att.com

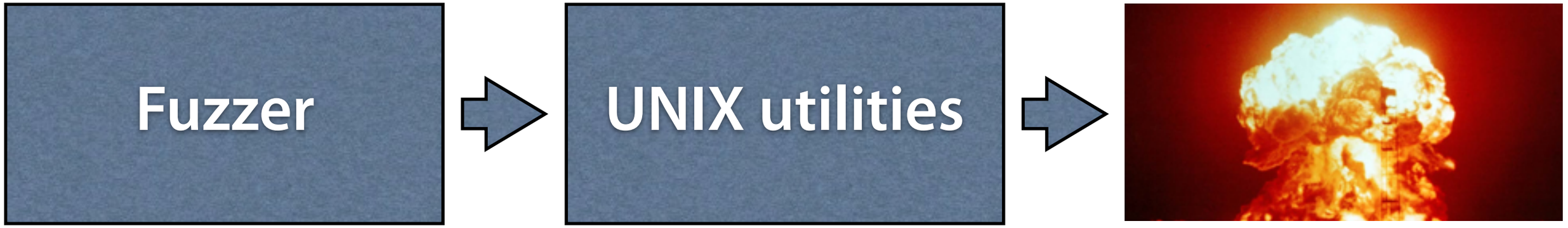
Bryan So
so@cs.wisc.edu

Summary

Operating system facilities, such as the kernel and utility programs, are typically assumed to be reliable. In our recent experiments, we have been able to crash 25-33% of the utility programs on any version of UNIX that was tested. This report describes these tests and an analysis of the program bugs that caused the crashes.

Fuzzing

Random Testing at the System Level



“ab’d&gfdfggg”

grep • sh • sed …

25%–33%

fuzzer.py

```
import random

def fuzzer():
    # Strings up to 1024 characters long
    string_length = int(random.random() * 1024)

    # Fill it with ASCII 32..128 characters
    out = ""
    for i in range(0, string_length):
        out += chr(int(random.random() * 96 + 32))
    return out

if __name__ == "__main__":
    print fuzzer()
```

Fuzzer Output

[;x1-GPZ+wcckc];,N9J+?#6^6\e?]9lu2_%'4GX"0VUB[E/r
~fApu6b8<{%siq8Zh.6{V,hr?;{Ti.r3PIxMMMv6{xS^+'Hq!
AxB"YXRS@!Kd6;wtAMefFWM(`lJ_<1~o}z3K(CCzRH JIlvHz>_*.
\>JrlU32~eGP?IR=bF3+;y$3lodQ<B89!5"W2fK*vE7v{')KC-
i,c{<[~m!]o;{.'}Gj\(X}EtYetrpbY@aGZ1{P!AZU7x#4(Rtn!
q4nCwqol^y6}0lKo=*JK~;zMKV=9Nai:wxu{J&UV#HaU)*BiC<),`
+t*gka<W=Z.%T5WGHZpI30D<Pq>&]BS6R&j?#tP7iaV}-}`\?
[_[Z^LBMPG-FKj'\xwuZ1=Q`^`5,NQ@[!CuRzJ2DlvBy!
^zkhdf3C5PAkR?V hnl3='i2Qx]D
$qs4O`1@fevnG'2\11Vf3piU37@55ap\zIyl"'f,
$ee,J4Gw:cgNKLie3nx9(`efSlg6#[K"@WjhZ}r[Scun&sBCS,T[/
vY'pduwgzDlVNy7'rnzxNwI)(ynBa>%lb`;`9fG]P_0hdG~$@6
3]KAeEnQ7lU)3Pn,0)G/6N-wyzj/MTd#A;r

Fuzzing UNIX utilities

- Use fuzzed output as a prolog prgram:
  `$ python fuzzer.py | prolog`
- Use fuzzed output as an input to grep:
  `$ python fuzzer.py | grep x`
- Use fuzzed output as a TeX document:
  `$ python fuzzer.py | tex`

Demo

Results

Utility	VAX (v)	Sun (s)	HP (h)	i386 (x)	AIX 1.1 (a)	Sequent (d)
adb	●○	●	●	○	–	–
as	●			●	●	●
awk						
bc				●○		
bib			–	–	–	–
calendar				–		
cat						
cb	●		●	●	○	●
cc						
/lib/ccom				–	–	●
checkeq				–		
checknr				–	–	
col	●○	●	●	●○	●	●
colcrt				–	–	
colrm				–	–	
comm						
compress					–	
/lib/cpp						

deroff	●	●	●		●	●
diction	●	–	●		–	●
diff						
ditroff	●○	●	–	–	–	
dtbl			–	–	–	–
emacs	●	●	○	–	–	
eqn		●	●	●		
expand					–	
f77	●		–	–	–	–
fmt						
fold					–	
ftp	●	●	●	–	●	●
graph					–	
grep						
grn			–	–	–	–
head					–	
ideal			–	–	–	–
indent	●○	●○	●	–	–	●
join		⊕				
latex			–	–	–	–
lex	●	●	●	●	●	●
lint						
lisp		–		–	–	–
look	●	○	●	●	–	●

Results

Utility	VAX (v)	Sun (s)	HP (h)	i386 (x)	AIX 1.1 (a)	Sequent (d)
adb	●○	●	●	○	–	–
as	●			●	●	●
awk						
bc				●○		
bib			–	–	–	–
calendar				–		
cat						
cb	●		●	●	○	●
cc						
/lib/ccom				–	–	●
checkeq				–		
checknr				–	–	
col	●○	●	●	●○	●	●
colcrt				–	–	
colrm				–	–	
comm						
compress					–	
/lib/cpp						
csh	●○	○	○	–	○	○
dbx		*	–	–		
dc				○		
deqn		●	–	–	–	–
deroff	●	●	●		●	●
diction	●	–	●		–	●
diff						
ditroff	●○	●	–	–	–	
dtbl			–	–	–	–
emacs	●	●	○	–	–	
eqn		●	●	●		
expand					–	
f77	●		–	–	–	–
fmt						
fold					–	
ftp	●	●	●	–	●	●
graph					–	
grep						
grn			–	–	–	–
head					–	
ideal			–	–	–	–
indent	●○	●○	●	–	–	●
join		⊕				
latex			–	–	–	–
lex	●	●	●	●	●	●
lint						
lisp		–		–	–	–
look	●	○	●	●	–	●

Table 2: List of Utilities Tested and the Systems on which They Were Tested (part 1)

*● = utility crashed, ○ = utility hung, * = crashed on SunOS 3.2 but not on SunOS 4.0, ⊕ = crashed only on SunOS 4.0, not 3.2. – = utility unavailable on that system. ! = utility caused the operating system to crash.*

Utility	VAX (v)	Sun (s)	HP (h)	i386 (x)	AIX 1.1 (a)	Sequent (d)
m4				●		
mail						
make			●			
more					–	
nm						
nroff				●		
pc				–	–	–
pic			–	–	–	–
plot	–	○	●	–	–	
pr						–
prolog	●○	●○	●○	–	–	–
psdit				–	–	
ptx	–	●	●	○		○
refer	●	*	●	–	–	!●
rev				–	–	
sed						
sh				–		
soelim					–	
sort						
spell	●○	●	●	○	●	●
spline					–	
split						
sql		–			–	–
strings					–	
strip						
style	●	–	●		–	●
sum						
tail						
tbl						
tee						
telnet	●	●	●	–	●	○
tex			–	–	–	–
tr						
troff	–	–	–			
tsort	●	*	●	●	●	●
ul	●	●	●	–	–	●
uniq	●	●	●	●	●	●
units	●○	●	●	●	●	●
vgrind	●		–	–	–	
vi	●		●	–		
wc						
yacc						
# tested	85	83	75	55	49	73
# crashed/hung	25	21	25	16	12	19
%	29.4%	25.3%	33.3%	29.1%	24.5%	26.0%

Table 2: List of Utilities Tested and the Systems on which They Were Tested (part 2)

*● = utility crashed, ○ = utility hung, * = crashed on SunOS 3.2 but not on SunOS 4.0, ⊕ = crashed only on SunOS 4.0, not 3.2. – = utility unavailable on that system. ! = utility caused the operating system to crash.*

Reasons for Crashes

- Pointers and arrays
- Not checking return codes
- And more…

Pointers and Arrays

```
while ((cc = getch()) != c)
{
        string[j++] = cc;
        ...
}
```

Not checking Return Codes

```
char rdc()
{
     char lastc;

     do {
          lastc = getchar();
     } while (lastc != ' ' ||
              lastc != '\t');

     return (lastc);
}
```

And more…

- Send "!o%888888888f" as command to the csh command-line shell
- Invoke this with string = "%888888888f":

```
char *string = ...
printf(string);
```

Safe Coding

- Check all array references for valid bounds
- Apply bounds on all inputs
- Check all system call return values
- Never trust third-party inputs

…all of which is supported by modern languages

…but there are newbie programmers born every minute

Controlling Fuzzing

We want to control our fuzzing script from the **command line:**

- Setting character ranges
- Setting maximal line lengths

Example

```
$ python ./fuzzer-getopt.py -h
Usage: ./fuzzer-getopt.py [-h] [-l MAX_LENGTH]
[-s RANGE_START] [-w RANGE_WIDTH]
```

```
$ python ./fuzzer-getopt.py -s 65 -w 26
KYWEVMRHEDUEIZKZYYVAVTMOIJHZPPEXWZMNCSTJVHGGBJP
FKSYUAMKVUXZKBNYSWERMZECYLVNZCYFWJYKJTJIWEVZMHE
WESCUUDWFKANNNJXKCHPWDLUQYPJEDGRXPCCLMVJMBREHKF
AZSHRIHMNOOAPDKCYPIPZYTVEXCFQTIOPTDDLJUJGYXSOHA
IEDHRVCAOUBDOEECMKZTQLQVLBMDSNYCRIQVFICTJCISRAV
LWFVTGGBAXEJEFPDGHIPFJZVUIJKZUQUQTXWZBSSLGJNALE
KRYSEVTTUERUTPPDHWHRLDQNGAMWVKJVTDSETZQQWEHJNQW
TAKLBFROWYBSES
```

Demo

More Extensions

- Control the number of lines to produce
- Control whether control characters (such as NUL, Ctrl-D, Ctrl-Z) should be included
- Control whether high-ASCII (128..255) should be included
- See Miller's paper for inspirations

Lab Practice

- Labs all across the 2nd floor
- Git repo for storage and submissions
- Python info: Reference, Zen of Python
- Generic info: Google, Stack Overflow
- Exchange ideas, not code
- Questions?