strace: new features

Dmitry Levin

BaseALT

FOSDEM 2018

New features since FOSDEM 2017

Released

4.16	:	Syscall return value injection
4.16	:	Syscall signal injection
4.17	:	Syscall specification improvements
4.18, 4.19	:	Netlink socket parsers

Being merged

- Syscall delay injection
- Advanced syscall filtering syntax
- Advanced syscall manipulations with Lua
- Advanced syscall information tool

traditional syscall fault injection

-e fault=*set*[:error=*errno*][:when=*expr*]

strace -a28 -e trace=open -e **fault**=*open*:**when**=*3*:**error**=EACCES cat /dev/null

```
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
open("/dev/null", O_RDONLY) = -1 EACCES (Permission denied)
cat: /dev/null: Permission denied
+++ exited with 1 +++
```

Error opening /dev/urandom

```
$ strace -P /dev/urandom -e fault=open:error=ENOENT python3 < /dev/null
open("/dev/urandom", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or dir
ectory) (INJECTED)
Fatal Python error: Failed to open /dev/urandom
--- SIGSEGV si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0x50 ---
+++ killed by SIGSEGV +++
Segmentation fault
```

Error reading /dev/urandom

```
$ strace -a30 -P /dev/urandom -e fault=read:error=EIO python3 < /dev/null
open("/dev/urandom", O_RDONLY|O_CLOEXEC) = 3
fcntl(3, F_GETFD)                       = 0x1 (flags FD_CLOEXEC)
read(3, 0x8db610, 24)                   = -1 EIO (Input/output error) (INJECTED)
Fatal Python error: Failed to read bytes from /dev/urandom
--- SIGSEGV si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0x50 ---
+++ killed by SIGSEGV +++
Segmentation fault
```

glibc <= 2.25: without a proper check

```
$ strace -e mprotect -efault=mprotect:when=1:error=EPERM pwd > /dev/null
mprotect(0x7fabcd00f000, 2097152, PROT_NONE) = -1 EPERM (Operation not pe
rmitted) (INJECTED)
mprotect(0x7fabcd20f000, 16384, PROT_READ) = 0
mprotect(0x606000, 4096, PROT_READ)      = 0
mprotect(0x7fabcd441000, 4096, PROT_READ) = 0
+++ exited with 0 +++
```

glibc >= 2.26: with a proper check

```
$ strace -e mprotect -efault=mprotect:when=1:error=EPERM pwd > /dev/null
mprotect(0x7fabcd00f000, 2097152, PROT_NONE) = -1 EPERM (Operation not pe
rmitted) (INJECTED)
pwd: error while loading shared libraries: libc.so.6: cannot change memor
y protections
+++ exited with 127 +++
```

syscall tampering improvements

- new -e **inject**= option
- return value injection
- signal injection
- delay injection

strace -e **inject**=*set*[:paramN]...

Valid inject parameters:

- error=*errno* or retval=*value*
- signal=*sig*
- delay_enter=*usecs*
- delay_exit=*usecs*
- when=*expr*

syscall delay injection

strace -e **inject**=*set*:delay_enter=*usecs*
strace -e **inject**=*set*:delay_exit=*usecs*

dd if=/dev/zero of=/dev/null bs=1M count=10

```
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.00211354 s,
5.0 GB/s
```

strace -einject=write:delay_exit=100000 -ewrite -o/dev/null dd if=/dev/zero of=/dev/null bs=1M count=10

```
10+0 records in
10+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 1.10658 s,
9.5 MB/s
```

Netlink socket parsers

Result of joined efforts

GSoC 2016 : Fabien Siron, mentored by Gabriel Laskar

GSoC 2017 : JingPiao Chen

Currently supported netlink protocols

- NETLINK_AUDIT
- NETLINK_CRYPTO
- NETLINK_KOBJECT_UEVENT
- NETLINK_NETFILTER
- NETLINK_ROUTE
- NETLINK_SELINUX
- NETLINK_SOCK_DIAG
- NETLINK_XFRM
- NETLINK_GENERIC

hsh-run −− ip route list table all

```
broadcast 127.0.0.0 dev lo table local proto kernel
scope link src 127.0.0.1
local 127.0.0.0/8 dev lo table local proto kernel
scope host src 127.0.0.1
local 127.0.0.1 dev lo table local proto kernel scope
host src 127.0.0.1
broadcast 127.255.255.255 dev lo table local proto
kernel scope link src 127.0.0.1
```

hsh-run –mount=/proc –– strace -e trace=sendto,recvmsg ip route list

```
sendto(3, {{len=40, type=RTM_GETROUTE, flags=NLM_F_REQUEST|NLM_F_DUMP, seq=1357924680,
pid=0}, {rtm_family=AF_UNSPEC, rtm_dst_len=0, rtm_src_len=0, rtm_tos=0,
rtm_table=RT_TABLE_UNSPEC, rtm_protocol=RTPROT_UNSPEC,
rtm_scope=RT_SCOPE_UNIVERSE, rtm_type=RTN_UNSPEC, rtm_flags=0}, {nla_len=0,
nla_type=RTA_UNSPEC}}, 40, 0, NULL, 0) = 40
recvmsg(3, {msg_name={sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000}, msg_namelen=12,
msg_iov=[{iov_base=[ {{len=60, type=RTM_NEWROUTE, flags=NLM_F_MULTI, seq=1357924680,
pid=12345}, {rtm_family=AF_INET, rtm_dst_len=32, rtm_src_len=0, rtm_tos=0,
rtm_table=RT_TABLE_LOCAL, rtm_protocol=RTPROT_KERNEL, rtm_scope=RT_SCOPE_LINK,
rtm_type=RTN_BROADCAST, rtm_flags=0}, [{{nla_len=8, nla_type=RTA_TABLE},
RT_TABLE_LOCAL}, {{nla_len=8, nla_type=RTA_DST}, 127.0.0.0}, {{nla_len=8,
nla_type=RTA_PREFSRC}, 127.0.0.1}, {{nla_len=8, nla_type=RTA_OIF}, if_nametoindex("lo")}]},
{{len=60, type=RTM_NEWROUTE, flags=NLM_F_MULTI, seq=1357924680, pid=12345},
{rtm_family=AF_INET, rtm_dst_len=8, rtm_src_len=0, rtm_tos=0, rtm_table=RT_TABLE_LOCAL,
rtm_protocol=RTPROT_KERNEL, rtm_scope=RT_SCOPE_HOST, rtm_type=RTN_LOCAL,
rtm_flags=0}, [{{nla_len=8, nla_type=RTA_TABLE}, RT_TABLE_LOCAL}, {{nla_len=8,
nla_type=RTA_DST}, 127.0.0.0}, {{nla_len=8, nla_type=RTA_PREFSRC}, 127.0.0.1}, {{nla_len=8,
nla_type=RTA_OIF}, if_nametoindex("lo")}]}, {{len=60, type=RTM_NEWROUTE, flags=NLM_F_MULTI,
seq=1357924680, pid=12345}, {rtm_family=AF_INET, rtm_dst_len=32, rtm_src_len=0, rtm_tos=0,
rtm_table=RT_TABLE_LOCAL, rtm_protocol=RTPROT_KERNEL, rtm_scope=RT_SCOPE_HOST,
rtm_type=RTN_LOCAL, rtm_flags=0}, [{{nla_len=8, nla_type=RTA_TABLE}, RT_TABLE_LOCAL},
{{nla_len=8, nla_type=RTA_DST}, 127.0.0.1}, {{nla_len=8, nla_type=RTA_PREFSRC}, 127.0.0.1},
{{nla_len=8, nla_type=RTA_OIF}, if_nametoindex("lo")}]}, {{len=60, type=RTM_NEWROUTE,
flags=NLM_F_MULTI, seq=1357924680, pid=12345}, {rtm_family=AF_INET, rtm_dst_len=32,
rtm_src_len=0, rtm_tos=0, rtm_table=RT_TABLE_LOCAL, rtm_protocol=RTPROT_KERNEL,
rtm_scope=RT_SCOPE_LINK, rtm_type=RTN_BROADCAST, rtm_flags=0}, [{{nla_len=8,
nla_type=RTA_TABLE}, RT_TABLE_LOCAL}, {{nla_len=8, nla_type=RTA_DST}, 127.255.255.255},
{{nla_len=8, nla_type=RTA_PREFSRC}, 127.0.0.1}, {{nla_len=8, nla_type=RTA_OIF},
if_nametoindex("lo")}]} ], iov_len=32768}], msg_iovlen=1, msg_controllen=0, msg_flags=0}, 0) = 240
```

...

System call specification improvements

syscall classes now have % prefix

strace -e trace=%*class*

traditional syscall classes

desc	:	take or return a descriptor
file	:	take a file name
memory	:	memory mapping, memory policy
process	:	process management
signal	:	signal related
ipc	:	SysV IPC related
network	:	network related

System call specification improvements

new syscall classes

- %stat, %lstat, %fstat
- %statfs, %fstatfs
- %%stat, %%statfs

strace -y -e %%stat ls /var/empty

```
fstat(3</etc/ld.so.cache>, st_mode=S_IFREG|0644, st_size=303
...
fstat(3</proc/filesystems>, st_mode=S_IFREG|0444, st_size=0,
stat("/var/empty", st_mode=S_IFDIR|0555, st_size=40, ...) =
fstat(3</var/empty>, st_mode=S_IFDIR|0555, st_size=40, ...)
+++ exited with 0 +++
```

%fstat syscall class

- fstat, fstat64
- fstatat64, newfstatat
- oldfstat

strace -y -e %fstat find /var/empty -mindepth 1

```
fstat(3</etc/ld.so.cache>, st_mode=S_IFREG|0644, ...) = 0
...
fstat(3</lib64/libdl-2.26.so>, st_mode=S_IFREG|0644, ...) =
fstat(3</proc/filesystems>, st_mode=S_IFREG|0444, ...) = 0
newfstatat(AT_FDCWD, "/var/empty", st_mode=S_IFDIR|0555, ..
fstat(4</var/empty>, st_mode=S_IFDIR|0555, ...) = 0
newfstatat(AT_FDCWD, "/var/empty", st_mode=S_IFDIR|0555, ..
```

added support of regular expressions

strace -e trace=/*regexp*

regexp is an extended regular expression

strace -e '/fstat(at)?(64)?$' find /var/empty -mindepth 1

```
fstat(3</etc/ld.so.cache>, st_mode=S_IFREG|0644, ...) = 0
...
fstat(3</lib64/libdl-2.26.so>, st_mode=S_IFREG|0644, ...) =
fstat(3</proc/filesystems>, st_mode=S_IFREG|0444, ...) = 0
newfstatat(AT_FDCWD, "/var/empty", st_mode=S_IFDIR|0555, ..
fstat(4</var/empty>, st_mode=S_IFDIR|0555, ...) = 0
newfstatat(AT_FDCWD, "/var/empty", st_mode=S_IFDIR|0555, ..
```

System call specification improvements

added support of conditional descriptions

strace -e trace=?*set*

strace -e trace=?statx ./tests/statx

```
statx(AT_FDCWD, "/dev/full", AT_STATX_SYNC_AS_STAT,
STATX_ALL, stx_mask=STATX_BASIC_STATS, stx_attributes=0,
stx_mode=S_IFCHR|0666, stx_size=0, ...) = 0
```

strace -e trace=/statx might not work

```
strace: invalid system call '/statx'
```

System call specification improvements

glibc: open vs openat

```
glibc-2.25$ strace -qq -e open cat /dev/null
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
open("/dev/null", O_RDONLY)                  = 3
glibc-2.26$ strace -qq -e open cat /dev/null
glibc-2.26$ strace -qq -e openat cat /dev/null
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/dev/null", O_RDONLY) = 3
```

strace -e **open**,**openat** is not portable

portable syscall specifications

regexp : /^open(at)?$

condition : ?open,openat

Advanced syscall filtering syntax

new syntax

[*action*(]*filter_expression*[;*arg1*[;*arg2*...]][)]

action is one of **trace**, **abbrev**, **verbose**, **raw**, **read**, **write**, **fault**, **inject**, or **stacktrace**;

argN are arguments of ***action***;

filter_expression is a combination of filters.

supported filters

syscall *set* : set of syscalls described by ***set***;

fd *fd1*... : set of syscalls operating with descriptor numbers described by ***fd1***...;

path *path* : set of syscalls operating with paths described by ***path***.

Advanced syscall filtering syntax

echo -n foo | strace -e 'trace(fd 1)' cat >/dev/null

```
fstat(1, st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...) =
write(1, "foo", 3)                              = 3
close(1)                                        = 0
+++ exited with 0 +++
```

strace -y -s4 -e 'trace(syscall read)' -e 'read(path /dev/zero)' head -c5 /dev/zero

```
read(3</lib64/libc-2.26.so>, "\177ELF"..., 832) = 832
read(3</dev/zero>, "\0\0\0\0"..., 5)    = 5
 | 00000  00 00 00 00 00                     .....                |
+++ exited with 0 +++
```

Advanced syscall filtering syntax

strace -ve 'syscall %file and not syscall %desc' cat /dev/null

```
execve("/usr/bin/cat", ["/usr/bin/cat", "/dev/null"], []) =
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file
+++ exited with 0 +++
```

strace -ve 'syscall %file and !(syscall %desc || path /usr/bin/cat)' /usr/bin/cat /dev/null

```
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file
+++ exited with 0 +++
```

Advanced syscall filtering syntax

strace -e 'fd 1' -e 'stacktrace(syscall close)' cat /dev/null

```
fstat(1, st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...)
close(1)                                = 0
 > /lib64/libc-2.26.so(_IO_file_close+0xb) [0x77a2b]
 > /lib64/libc-2.26.so(_IO_file_close_it+0x13c) [0x791fc]
 > /lib64/libc-2.26.so(fclose+0x1bf) [0x6c4df]
 > /bin/cat() [0x4cea]
 > /bin/cat() [0x2692]
 > /lib64/libc-2.26.so(__locale_getenv+0x140) [0x37680]
 > /lib64/libc-2.26.so(exit+0x1a) [0x376da]
 > /lib64/libc-2.26.so(__libc_start_main+0xf8) [0x21128]
 > /bin/cat() [0x217a]
+++ exited with 0 +++
```

Questions?

homepage

https://strace.io

strace.git

https://github.com/strace/strace.git
https://gitlab.com/strace/strace.git
git://git.code.sf.net/p/strace/code.git

mailing list

strace-devel@lists.sourceforge.net

IRC channel

#strace@freenode