strace: new features

Dmitry Levin

OSSDEVCONF 2017

New features since 4.13

Released

4.16	:	Syscall tampering and fault injection
4.17	:	Syscall specification improvements
4.18, 4.19	:	Netlink socket parsers

GSoC 2017 projects

- Advanced syscall filtering syntax
- Advanced syscall tampering and filtering with Lua
- Advanced syscall information tool

System call tampering and fault injection

traditional syscall fault injection

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

strace -a28 -e trace=open -e **fault**=*open*:**when**=*3*:**error**=EACCES ca t /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 +++
```

System call tampering and fault injection

syscall tampering improvements

- return value injection
- signal injection

-e inject=*set*[:error=*errno*|:retval=*value*][:signal=*sig*][:when=*expr*]

strace -e trace=open -**e fault**=*open*:**when**=*3*:**retval**=42 ca t /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)                  = 42 (INJECTED)
cat: /dev/null: Bad file descriptor
cat: /dev/null: Bad file descriptor
+++ exited with 1 +++
```

System call tampering and fault injection

syscall tampering improvements

- return value injection
- signal injection

-e inject=*set*[:error=*errno*|:retval=*value*][:signal=*sig*][:when=*expr*]

strace -a20 -P precious.txt -e fault=unlink:error=EACCES:signal=ABRT unlink precious.txt

```
unlink("precious.txt") = -1 EACCES (Permission denied)
 (INJECTED)
--- SIGABRT si_signo=SIGABRT, si_code=SI_KERNEL ---
+++ killed by SIGABRT (core dumped) +++
```

System call specification improvements

syscall classes now have % prefix

strace -e trace=%*class*

added new syscall classes

%stat, %lstat, %fstat, %%stat, %statfs, %fstatfs, %%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 +++
```

System call specification improvements

added support of regular expressions

strace -e trace=/*regexp*

strace -e 'trace=/^(.*_)?statv?fs' df / >/dev/null

```
statfs("/", f_type=TMPFS_MAGIC, f_bsize=4096, f_blocks=2904
+++ exited with 0 +++
```

strace -e %statfs df / >/dev/null

```
statfs("/", f_type=TMPFS_MAGIC, f_bsize=4096, f_blocks=2904
+++ exited with 0 +++
```

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 does not work

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

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 –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

...
```

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**;

arnN 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 '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 '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

```
strace: Requested path '/usr/bin/cat' resolved into '/bin/ca
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file
+++ exited with 0 +++
```

Advanced syscall filtering syntax

strace -k -e 'fd 1' cat /dev/null

```
fstat(1, st_mode=S_IFCHR|0620, st_rdev=makedev(136, 5), ...)
 > /lib64/libc-2.24.so(__fxstat64+0x14) [0xdab54]
 > /bin/cat() [0x1bb9]
 > /lib64/libc-2.24.so(__libc_start_main+0xf0) [0x20400]
 > /bin/cat() [0x258b]
close(1)                                = 0
 > /lib64/libc-2.24.so(_IO_file_close+0xb) [0x7195b]
 > /lib64/libc-2.24.so(_IO_file_close_it+0x13c) [0x7302c]
 > /lib64/libc-2.24.so(fclose+0x1a3) [0x669a3]
 > /bin/cat() [0x5daa]
 > /bin/cat() [0x2a92]
 > /lib64/libc-2.24.so(__locale_getenv+0x140) [0x35c60]
 > /lib64/libc-2.24.so(exit+0x1a) [0x35cba]
 > /lib64/libc-2.24.so(__libc_start_main+0xf7) [0x20407]
 > /bin/cat() [0x258b]
+++ exited with 0 +++
```

Questions?

homepage

https://strace.io

strace.git

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

mailing list

strace-devel@lists.sourceforge.net

IRC channel

#strace@freenode