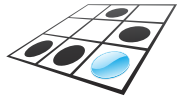
strace 2019

Dmitry Levin

September 2019



v4.20 . . . v5.3

- seccomp-assisted system call filtering: **–seccomp-bpf** option
- system call return status filtering: **-e status**=*set*, **-z**, **-Z** options
- PTRACE_GET_SYSCALL_INFO API support
- support of new system calls (32)
- elaborate syscall parsers
- long options
- copyleft license

Introduced in v5.3 (September 2019)

- Automatically generates and attaches a BPF program to filter system calls
- Makes execution of untraced system calls two orders of magnitude faster

tests/filter_seccomp-perf.c

```
static volatile sig_atomic_t stop;
static void handler(int signo) {
    stop = true;
}
int main(void) {
    signal(SIGALRM, handler);
    alarm(1);
    unsigned int i;
    for (i = 0; !stop; i++)
        chdir(".");
    printf("%d\n", i);
    return 0;
}
```

time ./filter_seccomp-perf

```
3480990
0.05user 0.94system 0:01.00elapsed 100%CPU (0avgtext+0avgdata 1324maxresider
0inputs+0outputs (0major+66minor)pagefaults 0swaps
```

time strace **–seccomp-bpf** -f -qq -e signal=none -e trace=fchdir ./filter_seccomp-perf

```
2962562
0.05user 0.94system 0:01.00elapsed 100%CPU (0avgtext+0avgdata 3280maxresider
0inputs+0outputs (0major+321minor)pagefaults 0swaps
```

time strace -f -qq -e signal=none -e trace=fchdir ./filter_seccomp-perf

```
81429
0.53user 0.73system 0:01.00elapsed 127%CPU (0avgtext+0avgdata 3156maxresider
0inputs+0outputs (0major+284minor)pagefaults 0swaps
```

$2962562/3480990 \approx 85.1\%$, $81429/3480990 \approx 2.3\%$, $81429/2962562 \approx 2.7\%$

Introduced in v5.2 (July 2019)

Print only system calls with the specified return status.
set can include the following elements:

successful : returned without an error code, alias to **-z**
failed : returned with an error code, alias to **-Z**
unfinished : did not return
detached : detached before return
unavailable : returned but failed to fetch the error status

The default is **-e status=all**.

Trace only those system calls that returned without an error code:

-z, **-e status=successful**

buildreq: rpm-utils commit 0.10.3-alt1∼2

```
--- a/rpm-utils/strace_files
+++ b/rpm-utils/strace_files
@@ -94,3 +94,3 @@ spp_output="$workdir"/spp_output
 rc=0
-strace -qq -f -e signal=none -e trace="${trace:-file}" \
+strace -qq -f -z -e signal=none -e trace="${trace:-file}" \
        -o "|/usr/share/buildreqs/spp >$spp_output" -- "$@"
```

strace -o log -f -e signal=none -e trace=execve,nanosleep \
sh -c 'sleep 0.1 & sleep 0.2 & sleep 0.3' && cat log

```
13475 execve("/bin/sh", ["sh", "-c", "sleep 0.1 & sleep 0.2 & sleep 0."...],
      0x5631be4f87a8 /* 42 vars */) = 0
13476 execve("/bin/sleep", ["sleep", "0.1"], 0xe4c4f0 /* 33 vars */ <unfinished ...>
13477 execve("/bin/sleep", ["sleep", "0.2"], 0xe4c4f0 /* 33 vars */ <unfinished ...>
13478 execve("/bin/sleep", ["sleep", "0.3"], 0xe4c4f0 /* 33 vars */ <unfinished ...>
13476 <... execve resumed>)              = 0
13477 <... execve resumed>)              = 0
13478 <... execve resumed>)              = 0
13476 nanosleep(tv_sec=0, tv_nsec=100000000,  <unfinished ...>
13477 nanosleep(tv_sec=0, tv_nsec=200000000,  <unfinished ...>
13478 nanosleep(tv_sec=0, tv_nsec=300000000,  <unfinished ...>
13476 <... nanosleep resumed>NULL)       = 0
13476 +++ exited with 0 +++
13477 <... nanosleep resumed>NULL)       = 0
13477 +++ exited with 0 +++
13478 <... nanosleep resumed>NULL)       = 0
13478 +++ exited with 0 +++
13475 +++ exited with 0 +++
```

strace -o log -z -f -e signal=none -e trace=execve,nanosleep \
sh -c 'sleep 0.1 & sleep 0.2 & sleep 0.3' && cat log

```
13475 execve("/bin/sh", ["sh", "-c", "sleep 0.1 & sleep 0.2 & sleep 0."...],
      0x5631be4f87a8 /* 42 vars */) = 0
13476 execve("/bin/sleep", ["sleep", "0.1"], 0xe4c4f0 /* 33 vars */) = 0
13477 execve("/bin/sleep", ["sleep", "0.2"], 0xe4c4f0 /* 33 vars */) = 0
13478 execve("/bin/sleep", ["sleep", "0.3"], 0xe4c4f0 /* 33 vars */) = 0
13476 nanosleep(tv_sec=0, tv_nsec=100000000, NULL) = 0
13476 +++ exited with 0 +++
13477 nanosleep(tv_sec=0, tv_nsec=200000000, NULL) = 0
13477 +++ exited with 0 +++
13478 nanosleep(tv_sec=0, tv_nsec=300000000, NULL) = 0
13478 +++ exited with 0 +++
13475 +++ exited with 0 +++
```

Trace only those system calls that returned with an error code:

-Z, **-e status=failed**

eu-elflint no-such-file1 no-such-file2

```
eu-elflint: cannot open input file: No such file or directory
eu-elflint: cannot open input file: No such file or directory
```

strace -qq -Z -efile eu-elflint no-such-file1 no-such-file2

```
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "no-such-file1", O_RDONLY) = -1 ENOENT (No such file or directory
eu-elflint: cannot open input file: No such file or directory
openat(AT_FDCWD, "no-such-file2", O_RDONLY) = -1 ENOENT (No such file or directory
eu-elflint: cannot open input file: No such file or directory
```

Example based on Debian bug report #459820 submitted in 2008

```
#include <stdio.h>
#include <unistd.h>
int main() {
    setlinebuf(stdout);
    puts("------------");
    __asm__("movl $2, %eax; int $0x80");
    printf("[I am %d]\n", getpid());
    return 0;
}
```

Regular invocation: ./debbug459820

```
------------
[I am 23450]
[I am 23451]
```

strace -f -z ./debbug459820 > /dev/null

```
...
write(1, "------------\n", 13)              = 13
strace: Process 23451 attached
open(NULL, O_RDONLY|O_CREAT|O_TRUNC|O_DSYNC|O_DIRECT
|O_NOATIME|O_CLOEXEC|O_PATH|O_TMPFILE|0x1000020,
0134300) = 23451
[pid 23450] getpid()                          = 23450
[pid 23451] getpid()                          = 23451
[pid 23450] write(1, "[I am 23450]\n", 13) = 13
[pid 23451] write(1, "[I am 23451]\n", 13) = 13
[pid 23450] +++ exited with 0 +++
+++ exited with 0 +++
```

PTRACE_GET_SYSCALL_INFO: strace >= v4.26, linux >= v5.3-rc1

```
...
write(1, "------------\n", 13)                = 13
strace: [ Process PID=23450 runs in 32 bit mode. ]
strace: Process 23451 attached
fork()                                        = 23451
strace: [ Process PID=23450 runs in 64 bit mode. ]
[pid 23450] getpid()                          = 23450
strace: [ Process PID=23451 runs in 64 bit mode. ]
[pid 23451] getpid()                          = 23451
[pid 23450] write(1, "[I am 23450]\n", 13) = 13
[pid 23451] write(1, "[I am 23451]\n", 13) = 13
[pid 23450] +++ exited with 0 +++
+++ exited with 0 +++
```

Introduced in v5.3 (September 2019)

pidfd_open, clone3

Introduced in v5.2 (July 2019)

open_tree, move_mount, fsopen, fsconfig, fsmount, fspick

Introduced in v5.1 (May 2019)

clock_gettime64, clock_settime64, clock_adjtime64, clock_getres_time64, clock_nanosleep_time64, timer_gettime64, timer_settime64, timerfd_gettime64, timerfd_settime64, utimensat_time64, pselect6_time64, ppoll_time64, io_pgetevents_time64, recvmmsg_time64, mq_timedsend_time64, mq_timedreceive_time64, semtimedop_time64, rt_sigtimedwait_time64, futex_time64, sched_rr_get_interval_time64, pidfd_send_signal, io_uring_setup, io_uring_enter, io_uring_register

1991 . . . 2019: short options only

a:Ab:cCdDe:E:fFhiI:ko:O:p:P:qrs:S:tTu:vVwxX:yzZ

Introduced in v5.3 (September 2019)

–help : alias to **-h**

–version : alias to **-V**

–seccomp-bpf : seccomp-assisted system call filtering

1991 . . . 2018: permissive license

strace was released under a Berkeley-style license
at the request of Paul Kranenburg.

Since v4.26∼52 (December 2018): copyleft license

- test suite: GNU General Public License v2+
- the rest of strace: GNU Lesser General Public License v2.1+

The first major strace feature implemented after this change is PTRACE_GET_SYSCALL_INFO API support.

Questions?

homepage

https://strace.io

strace.git

https://github.com/strace/strace.git
https://gitlab.com/strace/strace.git

mailing list

strace-devel@lists.strace.io

IRC channel

#strace@freenode