

Can strace make you fail?

Dmitry Levin

BaseALT

01.10.2016

What is strace?

- A diagnostic, debugging and instructional userspace utility for Linux.
- It is used to monitor interactions between processes and the Linux kernel, which include system calls, signal deliveries, and changes of process state.
- CLI and multiple filtering capabilities make it a powerful yet easy to use tracing tool.
- This year strace has been extended to tamper with traces using syscall fault injection.

sample output

```
$ strace -s 0 -P /dev/urandom python3 < /dev/null
open("/dev/urandom", O_RDONLY|O_CLOEXEC) = 3
fcntl(3, F_GETFD) = 0x1 (flags FD_CLOEXEC)
read(3, "...", 24) = 24
close(3) = 0
+++ exited with 0 +++
```

strace workflow: entering syscall

tracee

invokes a syscall

kernel

puts the tracee into SYSCALL-ENTER-STOP state

strace

- awakens
- fetches the syscall number and arguments
- applies filters
- may print something
- tells the kernel to let the tracee go on

kernel

- executes the syscall
- puts the tracee into SYSCALL-EXIT-STOP state

strace

- awakens
- may fetch the syscall return code and some arguments
- may print something
- tells the kernel to let the tracee go on

strace workflow: tampering with syscalls

- awakens
- fetches the syscall number and arguments
- applies filters
- **may tamper with the syscall number and arguments**
- may print something
- tells the kernel to let the tracee go on
- *the kernel executes the syscall*
- awakens
- may fetch the syscall return code and some arguments
- **may tamper with the syscall return code and arguments**
- may print something
- tells the kernel to let the tracee go on

What is fault injection?

- A software testing technique used for improving test coverage
- of error handling code paths that might otherwise rarely be followed
- by introducing faults

Where to place strace among other fault injection tools?

- software implemented fault injection
- runtime fault injection
- syscall interposition
- unprivileged userspace
- command-line interface

Unprivileged userspace command-line runtime syscall fault injection tool

```
strace -e trace=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
+++ exited with 0 +++
```

```
strace -e trace=open -e fault=open cat
```

```
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = -1 ENOSYS (Function not implemented) (INJECTED)
open("/lib64/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOSYS (Function not implemented) (INJECTED)
open("/lib64/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOSYS (Function not implemented) (INJECTED)
open("/lib64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOSYS (Function not implemented) (INJECTED)
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOSYS (Function not implemented) (INJECTED)
cat: error while loading shared libraries: libc.so.6: cannot open shared object file: Error 38
+++ exited with 127 +++
```

strace syscall fault injection syntax

```
strace -e trace=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
+++ exited with 0 +++
```

```
strace -e trace=open -e fault=open:error=ENOENT cat /dev/null
```

```
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJECTED)
open("/lib64/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJ
open("/lib64/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJECTED)
open("/lib64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJECTED)
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJECTED)
open("/usr/lib64/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (
open("/usr/lib64/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJECTE
open("/usr/lib64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJE
open("/usr/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) (INJECTED)
cat: error while loading shared libraries: libc.so.6: cannot open shared object file: No such file or
+++ exited with 127 +++
```



```
strace -e trace=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
+++ exited with 0 +++
```

```
strace -e trace=open -e fault=open:when=1:error=EACCES cat /dev/null
```

```
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = -1 EACCES (Permission denied) (INJECTED)
open("/lib64/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("/lib64/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("/lib64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
open("/dev/null", O_RDONLY) = 3
+++ exited with 0 +++
```

strace syscall fault injection syntax

```
strace -e trace=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
+++ exited with 0 +++
```

```
strace -e trace=open -e fault=open:when=2+:error=EPERM cat /dev/null
```

```
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 EPERM (Operation not permitted) (INJECTED)
open("/lib64/tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 EPERM (Operation not permitted) (INJECTED)
open("/lib64/tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 EPERM (Operation not permitted) (INJECTED)
open("/lib64/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 EPERM (Operation not permitted) (INJECTED)
open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 EPERM (Operation not permitted) (INJECTED)
cat: error while loading shared libraries: libc.so.6: cannot open shared object file: Operation not permitted
+++ exited with 127 +++
```

```
strace -e trace=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
+++ exited with 0 +++
```

```
strace -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) (INJECTED)
cat: /dev/null: Permission denied
+++ exited with 1 +++
```

strace syscall fault injection syntax

```
strace -e trace=open cat /dev/null{,}{,}
```

```
$ C strace -eopen 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
open("/dev/null", O_RDONLY) = 3
open("/dev/null", O_RDONLY) = 3
open("/dev/null", O_RDONLY) = 3
+++ exited with 0 +++
```

```
strace -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) (INJECTED)
cat: /dev/null: Permission denied
open("/dev/null", O_RDONLY) = 3
open("/dev/null", O_RDONLY) = 3
open("/dev/null", O_RDONLY) = -1 EACCES (Permission denied) (INJECTED)
cat: /dev/null: Permission denied
+++ exited with 1 +++
```

strace -P /dev/null cat /dev/null

```
$ strace -P /dev/null cat /dev/null
open("/dev/null", O_RDONLY)           = 3
fstat(3, {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
read(3, "", 131072)                   = 0
close(3)                               = 0
+++ exited with 0 +++
```

strace -P /dev/null -e fault=all cat /dev/null

```
open("/dev/null", O_RDONLY)           = -1 ENOSYS (Function not implemented) (INJECTED)
cat: /dev/null: Function not implemented
+++ exited with 1 +++
```

strace -P /dev/null cat /dev/null

```
$ strace -P /dev/null cat /dev/null
open("/dev/null", O_RDONLY)           = 3
fstat(3, {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
read(3, "", 131072)                   = 0
close(3)                               = 0
+++ exited with 0 +++
```

strace -P /dev/null -e fault=fstat:error=ENOMEM cat /dev/null

```
open("/dev/null", O_RDONLY)           = 3
fstat(3, 0x7ffd970bb080)                = -1 ENOMEM (Cannot allocate memory) (INJECTED)
cat: /dev/null: Cannot allocate memory
close(3)                               = 0
+++ exited with 1 +++
```

strace -P /dev/null cat /dev/null

```
$ strace -P /dev/null cat /dev/null
open("/dev/null", O_RDONLY)           = 3
fstat(3, {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
read(3, "", 131072)                   = 0
close(3)                               = 0
+++ exited with 0 +++
```

strace -P /dev/null -e fault=fadvise64 cat /dev/null

```
open("/dev/null", O_RDONLY)           = 3
fstat(3, {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = -1 ENOSYS (Function not implemented) (INJECTED)
read(3, "", 131072)                   = 0
close(3)                               = 0
+++ exited with 0 +++
```

strace -P /dev/null cat /dev/null

```
$ strace -P /dev/null cat /dev/null
open("/dev/null", O_RDONLY)          = 3
fstat(3, {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
read(3, "", 131072)                  = 0
close(3)                              = 0
+++ exited with 0 +++
```

strace -P /dev/null -e fault=read:when=1:error=EINTR cat /dev/null

```
open("/dev/null", O_RDONLY)          = 3
fstat(3, {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
fadvise64(3, 0, 0, POSIX_FADV_SEQUENTIAL) = 0
read(3, 0x7f057432c000, 131072)      = -1 EINTR (Interrupted system call) (INJECTED)
read(3, "", 131072)                  = 0
close(3)                              = 0
+++ exited with 0 +++
```


Examples: strace syscall fault injection: python3

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 directory) (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 -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
```

First mprotect invocation: without a proper check

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

Second mprotect invocation: with a proper check

```
$ strace -e mprotect -e fault=mprotect:when=2+:error=EPERM pwd > /dev/null
mprotect(0x7fabcd00f000, 2097152, PROT_NONE) = 0
mprotect(0x7fabcd20f000, 16384, PROT_READ) = -1 EPERM (Operation not permitted) (INJECTED)
pwd: error while loading shared libraries: /lib64/libc.so.6: cannot apply additional memory
protection after relocation: Operation not permitted
+++ exited with 127 +++
```

Examples: strace access control: access by path argument

```
strace -P /etc/passwd -e fault=desc:error=EPERM cat /etc/passwd
```

```
open("/etc/passwd", 0_RDONLY) = -1 EPERM (Operation not permitted) (INJECTED)
cat: /etc/passwd: Operation not permitted
+++ exited with 1 +++
```

```
strace -P /etc/passwd -e fault=all:error=EACCES,desc:error=EPERM stat
/etc/passwd
```

```
lstat("/etc/passwd", 0x7ffd282659d0) = -1 EACCES (Permission denied) (INJECTED)
stat: cannot stat '/etc/passwd': Permission denied
+++ exited with 1 +++
```

Examples: strace access control: access by descriptor argument

```
strace -P /etc/passwd -e fault=desc:error=EPERM cat < /etc/passwd
```

```
fstat(0, 0x7ffd6c14daa0)          = -1 EPERM (Operation not permitted) (INJECTED)
cat: -: Operation not permitted
close(0)                          = -1 EPERM (Operation not permitted) (INJECTED)
cat: closing standard input: Operation not permitted
+++ exited with 1 +++
```

```
strace -P /etc/passwd -e fault=all:error=EPERM,desc:error=EACCES stat -
</etc/passwd
```

```
fstat(0, 0x7ffdb068e650)          = -1 EACCES (Permission denied) (INJECTED)
stat: cannot stat standard input: Permission denied
+++ exited with 1 +++
```

Questions?

homepage

<https://sourceforge.net/projects/strace/>
<https://strace.io>

strace.git

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

mailing list

strace-devel@lists.sourceforge.net