strace –seccomp-bpf: a look under the hood

Paul Chaignon

February 2, 2020

Overview of this talk

- How strace uses `ptrace(2)` to stop your process
- How strace uses seccomp-bpf to stop only at syscalls of interest
- How syscalls are matched in the kernel with 2 cBPF algos

strace's default behavior

strace's default behavior: strace ./tracee

```
                        user mode
                                  +--------+
                                  | strace |
                                  +--------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


           +----->-----PTRACE_SYSCALL---->----+
           |                                  |
+--------------------+               +-------------------+
| syscall-entry-stop |               | syscall-exit-stop |
+--------------------+               +-------------------+
           |                                  |
           +-----<-----PTRACE_SYSCALL----<----+
```

strace's default behavior: strace ./tracee

```
                 user mode
         +----------+     +--------+
         | ./tracee |     | strace |
         +----------+     +--------+
------------------------------------------
                kernel mode
```

```
           +----->-----PTRACE_SYSCALL---->----+
           |                                  |
+--------------------+               +-------------------+
| syscall-entry-stop |               | syscall-exit-stop |
+--------------------+               +-------------------+
           |                                  |
           +-----<-----PTRACE_SYSCALL----<----+
```

strace's default behavior: strace ./tracee

```
                        user mode
                                  +---------+
                                  | strace  |
                                  +---------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


           +----->-----PTRACE_SYSCALL---->----+
           |                                  |
+--------------------+               +-------------------+
| syscall-entry-stop |               | syscall-exit-stop |
+--------------------+               +-------------------+
           |                                  |
           +-----<-----PTRACE_SYSCALL----<----+
```

strace's default behavior: strace ./tracee

```
                        user mode
                                  +--------+
                                  | strace |
                                  +--------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


           +----->-----PTRACE_SYSCALL---->----+
           |                                  |
+--------------------+               +-------------------+
| syscall-entry-stop |               | syscall-exit-stop |
+--------------------+               +-------------------+
           |                                  |
           +-----<-----PTRACE_SYSCALL----<----+
```

strace's default behavior: strace ./tracee

```
                        user mode
                                  +--------+
                                  | strace |
                                  +--------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


           +----->-----PTRACE_SYSCALL---->----+
           |                                  |
+--------------------+               +-------------------+
| syscall-entry-stop |               | syscall-exit-stop |
+--------------------+               +-------------------+
           |                                  |
           +-----<-----PTRACE_SYSCALL----<----+
```

What's the issue?

strace -e trace=

- `strace -e trace=seccomp` => see `seccomp(2)` syscalls only
- `strace -eseccomp` => same
- `strace -e%network` => see all network-related syscalls

Unnecessary overhead

- Stops twice per syscall, at all syscalls
- Even if we want to see a single syscall!
- Stops require 2 context switches
- Very expensive!

```
$ cd linux/
$ time make -j$(nproc) > /dev/null
[...]
real        12m27,010s
$ make clean
$ time strace -f -econnect make -j$(nproc) > /dev/null
#              |  |
#              |  +----> Display connect(2) syscalls
#              +----> Trace child processes
[...]
real    24m54,473s
```

We need a way to tell the kernel at which syscalls to stop

Introducing seccomp-bpf

Introducing seccomp-bpf

- Let's use seccomp!
- Seccomp as a syscall-filtering mechanism
- seccomp-bpf to choose syscalls to filter
    - Used in Chrome's sandbox
    - Second user of BPF in Linux after socket filters (e.g., tcpdump)
    - cBPF, not eBPF!

seccomp-bpf examples

- Allow process to `open(2)` and `openat(2)` only
- Kill it if it tries anything else

```
ld [4]                  /* load seccomp_data->arch */
jne #0xc000003e, bad    /* is AUDIT_ARCH_X86_64? */
ld [0]                  /* load seccomp_data->nr */
jeq #257, good          /* is openat(2)? */
jeq #2, good            /* is open(2)? */
bad: ret #0             /* return RET_KILL_THREAD */
good: ret #0x7fff0000   /* return RET_ALLOW */
```

seccomp-bpf examples

- Allow process to open specific files only
- Need help from userspace

```
ld [4]                  /* load seccomp_data->arch */
jne #0xc000003e, bad    /* is AUDIT_ARCH_X86_64? */
ld [0]                  /* load seccomp_data->nr */
jeq #257, good          /* is openat(2)? */
jeq #2, good            /* is open(2)? */
bad: ret #0             /* return RET_KILL_THREAD */
good: ret #0x7ff00000   /* return RET_TRACE */
```

strace --seccomp-bpf

--seccomp-bpf's behavior

```
                        user mode
                                  +---------+
                                  | strace  |
                                  +---------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


        +------->-------PTRACE_SYSCALL------->------+
        |                                           |
+--------------+                                    |
| seccomp-stop |----<----+                          |
+--------------+         |                          |
                         |                          |
+--------------------+   |                  +-------------------+
| syscall-entry-stop |   +-<-PTRACE_CONT-<--| syscall-exit-stop |
+--------------------+                      +-------------------+
```

--seccomp-bpf's behavior

```
                        user mode
                +----------+      +--------+
                | ./tracee |      | strace |
                +----------+      +--------+
        ------------------------------------------
                       kernel mode




        +------->-------PTRACE_SYSCALL------->------+
        |                                           |
+--------------+                                    |
| seccomp-stop |----<----+                          |
+--------------+         |                          |
                         |                          |
+--------------------+   |                  +------------------+
| syscall-entry-stop |   +-<-PTRACE_CONT-<--| syscall-exit-stop |
+--------------------+                      +------------------+
```

--seccomp-bpf's behavior

```
                        user mode
                                  +---------+
                                  | strace  |
                                  +---------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


        +------->-------PTRACE_SYSCALL------->------+
        |                                           |
+--------------+                                    |
| seccomp-stop |----<----+                          |
+--------------+         |                          |
                         |                          |
+--------------------+   |                  +------------------+
| syscall-entry-stop |   +-<-PTRACE_CONT-<--| syscall-exit-stop |
+--------------------+                      +------------------+
```

--seccomp-bpf's behavior

```
                        user mode
                +----------+      +--------+
                | ./tracee |      | strace |
                +----------+      +--------+
        ------------------------------------------
                       kernel mode




        +------->-------PTRACE_SYSCALL------->------+
        |                                           |
+--------------+                                    |
| seccomp-stop |----<----+                          |
+--------------+         |                          |
                         |                          |
+--------------------+   |                  +-------------------+
| syscall-entry-stop |   +-<-PTRACE_CONT-<--| syscall-exit-stop |
+--------------------+                      +-------------------+
```

--seccomp-bpf's behavior

```
                        user mode
                                  +---------+
                                  | strace |
                                  +---------+
        -------------------------------------------
                       kernel mode
                +-----------+
                | ./tracee |
                +-----------+


        +------->-------PTRACE_SYSCALL------->------+
        |                                            |
+---------------+                                    |
| seccomp-stop |----<-----+                          |
+---------------+          |                         |
                           |                         |
+---------------------+    |                +-------------------+
| syscall-entry-stop |    +-<-PTRACE_CONT-<--| syscall-exit-stop |
+---------------------+                     +-------------------+
```

--seccomp-bpf's behavior

```
                        user mode
                                  +---------+
                                  | strace  |
                                  +---------+
        ------------------------------------------
                       kernel mode
                +----------+
                | ./tracee |
                +----------+


        +------->-------PTRACE_SYSCALL------->------+
        |                                           |
+--------------+                                    |
| seccomp-stop |----<----+                          |
+--------------+         |                          |
                         |                          |
+--------------------+   |                  +-------------------+
| syscall-entry-stop |   +-<-PTRACE_CONT-<--| syscall-exit-stop |
+--------------------+                      +-------------------+
```

--seccomp-bpf's behavior

- Before Linux 4.8, seccomp-stop happens before syscall-entry-stop
- We need to restart with PTRACE_SYSCALL twice :(

```
        +------->-------PTRACE_SYSCALL---->----+
        |                                      |
+--------------------+                         |
| syscall-entry-stop |                         |
+--------------------+                         |
        | PTRACE_SYSCALL                       |
+--------------+                     +-------------------+
| seccomp-stop |--<--PTRACE_CONT--<--| syscall-exit-stop |
+--------------+                     +-------------------+
```

cBPF algorithms: linear

```
[...]
ld [0]                  /* load seccomp_data->nr */
jeq #0, trace           /* is read(2)? */
jeq #1, trace           /* is write(2)? */
jeq #2, trace           /* is open(2)? */
jeq #3, trace           /* is close(2)? */
jeq #4, trace           /* is stat(2)? */
jeq #5, trace           /* is fstat(2)? */
[...]
skip: ret #0x7fff0000   /* return RET_ALLOW */
trace: ret #0x7ff00000  /* return RET_TRACE */
```

cBPF algorithms: linear

```
[...]
ld [0]                    /* load seccomp_data->nr */
jlt #0, skip              /* is < read(2)? */
jle #5, trace             /* is <= fstat(2)? */
[...]
skip: ret #0x7fff0000     /* return RET_ALLOW */
trace: ret #0x7ff00000    /* return RET_TRACE */
```

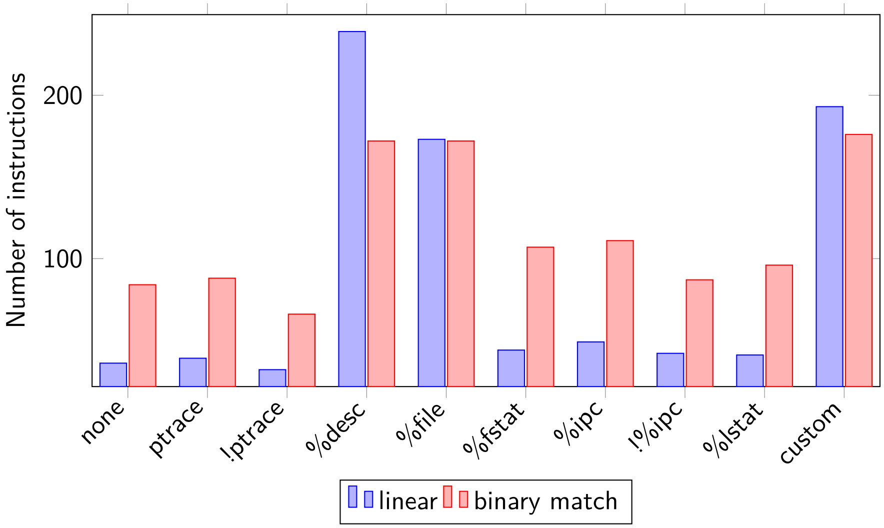
cBPF algorithms: binary match

- cBPF has 32-bit bitwise operations
- Encode all syscalls of interest as 32-bit bit arrays

```
select(2) >--------------+
ioctl(2) >--------+      |
                  |      |
00000000 00000000 10000001 00000000
```

- cBPF doesn't have indirect jumps => no jump tables
- switch over bit arrays implemented as linear search

cBPF algorithms



custom = %memory,%ipc,%pure,%signal,%network

Limitations

--seccomp-bpf implies -f

- `strace -f` to trace children processes
- `--seccomp-bpf` implies `-f`
- In kernel, children inherit seccomp filter chain of parent

- Kernel doesn't copy filters but keeps a reference count
- To inherit only some filters in the chain, we need to make copies :(

strace –seccomp-bpf -p [pid]

- `strace -p [pid]` to trace an existing process
- But no way to attach seccomp-bpf filters to existing processes

Conclusion

To sum up

- strace stops at all syscalls by default (expensive!)
- `strace --seccomp-bpf -e...` to stop only at syscalls of interest
- Uses 2 seccomp-bpf algorithms

Future work

- `socketcall(2)` and `ipc(2)`'s subcalls not supported!
- cBPF program would have to match on first syscall argument

- `strace -c` to print summary of traced syscalls
- Perfect use case for eBPF!
- Statistics can be aggregated in the kernel and summary only sent to strace

Thanks!