5. Zapoznaj się z dokumentacją polecenia strace na Linuxie. Opisz najważniejsze według siebie opcje.

Polecenie strace jest narzędziem służącym do śledzenia sygnałów oraz wywołań funkcyjnych danego programu. Możemy z niego skorzystać na dwa sposoby. Pierwszym z nich jest wywołanie programu bezpośrednio z poleceniem strace, drugim natomiast jest podpięcie się pod już istniejący w systemie proces.

Wybrane opcje:

a) strace -e trace=

Powyższe polecenie pozwala ograniczyć otrzymane informacje do tych, które nas interesują np. wywołanie „strace -e trace=open" wyświetla wszystkie logi, które posiadają słowo kluczowe 'open' czyli pliki, które są otwierane.

b) strace -p PID

Realizacja drugiego z przypadku Za pomocą przełącznika '-p' możemy podpiąć się pod istniejący proces w systemie.

c) strace -f

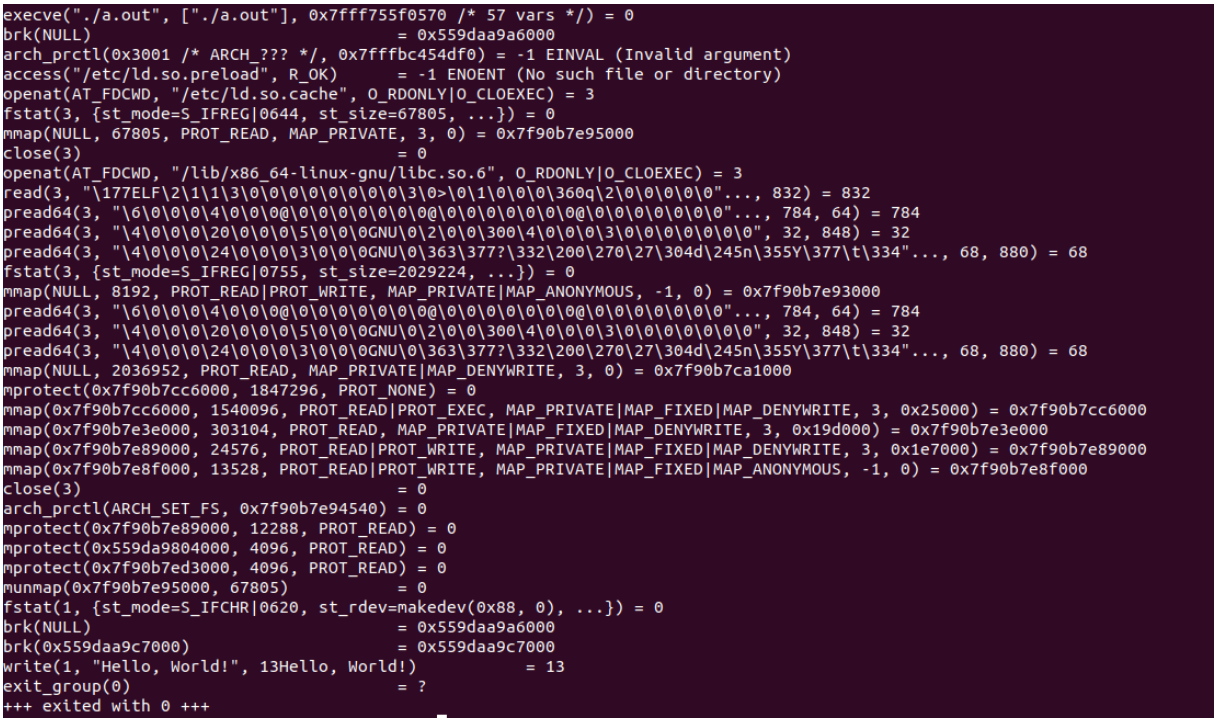
Polecenie strace śledzi tylko i wyłącznie pojedynczy proces. Jeżeli chcemy śledzić wszystkie procesy potomne naszego procesu należy użyć przełącznika '-f'. Użycie tej flagi pozwala nam na śledzenie procesu głównego jak i wszystkich jego dzieci.

d) stace -c

Wyświetla czas procesu spędzony na poszczególne operacje, liczbę ich wywołań oraz informacje o błędach napotkanych. Na samym dole widnieje sumaryczne podsumowanie wyświetlonych informacji.

6. Wykorzystaj program strace na Linuxie do śledzenia wykonywania się programu:

a) przeanalizuj wykonanie się programu wyświetlającego napis Hello world na ekranie



```
execve("./a.out", ["./a.out"], 0x7fff755f0570 /* 57 vars */) = 0
brk(NULL)                               = 0x559daa9a6000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffbc454df0) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=67805, ...}) = 0
mmap(NULL, 67805, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f90b7e95000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f90b7e93000
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f90b7ca1000
mprotect(0x7f90b7cc6000, 1847296, PROT_NONE) = 0
mmap(0x7f90b7cc6000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f90b7cc6000
mmap(0x7f90b7e3e000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f90b7e3e000
mmap(0x7f90b7e89000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f90b7e89000
mmap(0x7f90b7e8f000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f90b7e8f000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f90b7e94540) = 0
mprotect(0x7f90b7e89000, 12288, PROT_READ) = 0
mprotect(0x559da9804000, 4096, PROT_READ) = 0
mprotect(0x7f90b7ed3000, 4096, PROT_READ) = 0
munmap(0x7f90b7e95000, 67805)           = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL)                               = 0x559daa9a6000
brk(0x559daa9c7000)                     = 0x559daa9c7000
write(1, "Hello, World!", 13Hello, World!)          = 13
exit_group(0)                           = ?
+++ exited with 0 +++
```

Polecenia:

- execve – rozpoczęcie wykonywania programu,
- access – sprawdzanie praw dostępu do pliku,
- openat – otwarcie pliku poprzez stworzenie nowego ‘file descriptora’ w aktualnym katalogu roboczym,
- fstat – zwraca informacje o statusie pliku,
- mmap – odwzorowanie danej części wybranego pliku w przestrzeni adresowej procesu,
- read – czytanie z ‘file descriptora’,
- pread – czytanie z ‘file descriptora’, rozpoczęcie czytania od określonej pozycji,
- write – pisanie do ‘file descriptora’,
- mprotect – zmiana (ustawienie)zabezpieczeń dostępu do określonego regionu pamięci,

b) wykorzystaj program strace do znalezienia wszystkich plików konfiguracyjnych, jakie powłoka próbuje odczytać przy starcie,

strace -e trace=open,access bash

c) sprawdź czy plik edytowany w programie pico jest stale otwarty,

pico test.txt

sudo strace -p 5543 -e trace=open,openat,close

Plik nie jest otwarty cały czas, w trakcie pracy programu mamy otwarte tylko 3 fd (0,1,2), w momencie zapisu czegoś do pliku mamy informacje o otwarciu nowego fd i automatycznym jego zamknięciu.

```
strace: Process 5543 attached
openat(AT_FDCWD, "test.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
close(3)                                = 0
openat(AT_FDCWD, "/etc/passwd", O_RDONLY|O_CLOEXEC) = 3
close(3)                                = 0
openat(AT_FDCWD, "./.test.txt.swp", O_WRONLY|O_CREAT|O_EXCL|O_APPEND, 0666) = 3
close(3)                                = 0
openat(AT_FDCWD, "test.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
close(3)                                = 0
```

d) odczytaj jakie file deskryptory posiada uruchomiona aplikacja wyświetlająca napis Hello world na ekranie,

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

Dodatkowo 3 standardowe (standard input, output, error). W jednym momencie otwarte max 4 fd.

7. Wykorzystaj program strace do znalezienia błędu w programie program.c. Jaki sygnał zabił program? Jak można wykorzystać strace do pomiaru czasu wykonania poszczególnych elementów programu?

- Sygnał, który zabił program:

--- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0x5583b232b000} ---

+++ killed by SIGSEGV (core dumped) +++

- Czas wykonania poszczególnych elementów uzyskujemy za pomocą ‘strace -T’

- Znalezione źródło błędu

mmap(0x7f10d63f2000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f10d63f2000 <0.000054>

fd równe -1