Report 2: File-read System Call testing

Explicit System Call Testing

The test application would trigger all Syscalls one by one, evaluating that the audit record contains all the expected parameters, e.g the arguments, valid argument types, return values etc. The testing will be done for various success and failure modes, with cross checking for appropriate error codes in case of failure mode.

Repository

AuditTestSuite

Directory Structure

Source contains following significant files

src/filesystem

- **readlink.c** : Source for triggering `readlink(2)` and `readlinkat(2)`, used for following symbolic links.
- **open.c** : Source for triggering `open(2)` and `openat(2)`. Note: `syscall(2)` is used for calling open as libc converts 'open(2)' to 'openat(2)'.
- **test** : A POSIX compliant shell script which does all the hard work. From firing off the network binary to extracting the data from resulting trail and analysing the result. Detailed functioning of the script is described later.

Approach

Here is how I attempted to test the file-read (fr) syscalls.

1) Get all audit_events with file-read(fr) audit class.

```
$ cat /etc/security/audit_event | grep ":fr"
22:AUE_READLINK:readlink(2):fr
72:AUE_OPEN_R:open(2) - read:fr
80:AUE_OPEN_RW:open(2) - read,write:fr,fw
270:AUE_OPENAT_R:openat(2) - read:fr
278:AUE_OPENAT_RW:openat(2) - read,write:fr,fw
347:AUE_DARWIN_LOADSHFILE:load_shared_file():fr
361:AUE_DARWIN_COPYFILE:copyfile():fr,fw
43037:AUE_LOADSHFILE:load_shared_file():fr
43051:AUE_COPYFILE:copyfile(2):fr,fw
43151:AUE_READLINKAT:readlinkat(2):fr
43170:AUE_OPEN_EXTENDED_R:open_extended(2) - read:fr
43178:AUE_OPEN_EXTENDED_RW:open_extended(2) - read,write:fr,fw
```

Note: From the obtained results, it is noticed that some Darwin supported system calls are deprecated, these will be ignored.

Description of test

2) Set the audit flag:fr (file-read)

```
sed -i "" '/\<flags:/s/\(.*\)/flags:fr/' /etc/security/audit_control
```

3) Start the audit daemon and set a new trail for recording the syscalls

```
$ service auditd start; audit -n
```

4) Create a temporary file and its symbolic link in the `/tmp` directory. These disposable files will be used to trigger the `open(2)` and `openat(2)` syscalls in read-write only (no-create) mode and `readlink(2)`, `readlinkat(2)` for symlink following.

5) Fire off the syscalls, one after the other, they will read and close the file/symbolic link.

```
./open &
./readlink &
```

6) Rest of the steps are same as in the testing of network socket system calls, i.e Report1

Troubles encountered

The GNU libc converts `open(2)` to `openat(2)` to optimize the resource usage in case no file descriptor is passed to `open(2)`. With no other option, I presented this issue to the #freebsd-security IRC channel and a member resolved it by suggesting the use of `syscall(2)` for calling `open(2)`. i.e

```
syscall(SYS_open, "/tmp/templog", O_RDWR)
```

With this approach, I was finally able to audit `open(2)` :smiley:

Result

Audit of all concerned tests (in this case too) was successful (Yay! x2)

```
$ ./test
```

```
Audit Directory: /var/audit .. □
Starting auditd.
Audit daemon and new trail started .. □
Launching system calls .. □
Audit daemon stopped .. □

Trigger sent.
===============================================
Success mode passed: open(2) .. □
Failure mode passed: open(2) .. □
===============================================
Success mode passed: openat(2) .. □
Failure mode passed: openat(2) .. □
===============================================
Success mode passed: readlink(2) .. □
Failure mode passed: readlink(2) .. □
===============================================
Success mode passed: readlinkat(2) .. □
Failure mode passed: readlinkat(2) .. □


------------------Statistics-------------------
Tests evaluated: 8
Tests passed: 8
```

Further plan and Improvements

- Add tests for the remaining file-read syscalls, e.g `copyfile(2)` and `open_extended(2)`. Might need to use `syscall(2)` for calling both of them as it is possible libc modifies them too.

Bugs

- Sometimes, setting audit flag as `fr` (file-read) continuously logs `openat(2)` and it renders the audit system useless.

Edit

Added tests for a lot more file-create (fc) system calls * symlink(2) & symlinkat(2) * mkdir(2) & mkdirat(2) * mkfifo(2) & mkfifoat(2) * mknod(2) & mknodat(2) * link(2) & linkat(2)
Although it remains to automate them!