Completeness Policy Configuration

To control the completeness policy, use the configuration file syscallSwitch.config. By default, it looks like this:

```
safety switch:off,off,off
Running instructions including parameters: ./a.out
system call number need to be pass:
use brk or Illegal instruction:1
```

The first line "safety switch" controls the activation of three completeness policies. Each "off" means a specific policy is disabled; setting it to "on" enables the corresponding policy.

For example, if you want to enable only the first completeness policy, write:

```
safety switch:on,off,off
Running instructions including parameters: ./a.out
system call number need to be pass:
use brk or Illegal instruction:1
```

Our completeness policy intercepts system calls using signals—by replacing SVC instructions with either illegal instructions or the BRK instruction.
This approach introduces significant performance overhead, so it is only applied to a small subset of SVC instructions where necessary.

The first completeness policy is mainly used to handle cases where an SVC instruction is executed, but no assignment to register x8 is found within the 20 instructions preceding the SVC. We have observed that a small number of applications, such as Apache under certain versions of glibc, may require this policy to function correctly.

The second completeness policy is mainly used to inspect the target addresses of direct branch instructions (such as b and bl) to determine whether any of them fall between two replaced instructions.If such a case is detected, signal-based interception will be applied.

The third completeness policy is primarily designed to handle cases where the target address of an indirect jump falls between two replaced instructions. We intercept segmentation faults and bus errors, and perform further processing in the corresponding signal handlers to identify the problematic system call number and the possible location of the SVC instruction. For more details, please refer to the paper. The identified system call numbers will be automatically added to line 3 of the syscallSwitch.config file, as shown below:

```
1 system call number need to be pass:
```

Of course, when the third completeness policy is enabled, you may manually add any system call numbers that you believe require signal-based interception to line 3 of the syscallSwitch.config file. If you have multiple system calls to specify, separate them with commas.

The locations of SVC instructions that are identified by the third completeness policy as requiring signal-based interception—including the library name and the offset within the library—will be automatically stored in the syscallplace.config file. You can also manually edit this file to specify additional SVC instruction locations that you believe should be intercepted via signals.

```
1 Running instructions including parameters:
```

When enabling the third completeness policy, you need to fill in the above line with the command used to run the target application. This is because the policy may relaunch the application based on the contents of the syscallSwitch.config file and the information collected during its first execution.

```
1 use brk or Illegal instruction:1
```

The line above is used to specify whether signal-based interception should be triggered by an illegal instruction or by the BRK instruction. Use 0 for the default (illegal instruction), or 1 for interception via BRK.

In our current testing, enabling the first completeness policy alone has proven sufficient for all evaluated applications. The second and third policies serve as

additional safeguards.

Known Issues

The second completeness policy may cause bugs under certain versions of glibc(It seems to cause repeated triggering of the signal processing function in the signal processing function). If you encounter such issues with your version of glibc, please disable this policy. We are actively debugging the problem.

Our signal handlers for both BRK and illegal instructions are identical, but in practice, some applications may exhibit issues when intercepted using BRK, whereas illegal instructions do not have this problem. Therefore, when using our completeness policy, we recommend attempting interception via illegal instructions first. Only use BRK-based interception if the signal handler for illegal instructions is overridden by the target application.