oracle / **kernel-fuzzing**

Fuzzers for the Linux kernel

#kernel

2 commits **1** branch **1** release **2** contributors

Branch: **master** New pull request Create new file Upload files Find file Clone or download

brunoborges Update README.md Latest commit 9536d44 on 15 Oct 2016

bin	Public release.	a year ago
fuzzers	Public release.	a year ago
include	Public release.	a year ago
python	Public release.	a year ago
src	Public release.	a year ago
templates	Public release.	a year ago
.gitignore	Public release.	a year ago
CONTRIBUTING.md	Public release.	a year ago
LICENSE	Public release.	a year ago
README.md	Update README.md	a year ago
config-example.yml	Public release.	a year ago
satconfig.common	Public release.	a year ago
satconfig.kvm	Public release.	a year ago
satconfig.uml	Public release.	a year ago

README.md

Introduction

kernel-fuzzing is a repository of fuzzers for the Linux kernel. Each fuzzer usually targets a specific subsystem and knows how to turn a small binary "testcase" (usually a few kilobytes or less in size) into a sequence of syscalls and arguments that will trigger some interesting code path in the kernel.

The fuzzers were originally meant to be used with American Fuzzy Lop (AFL), but could in theory be used with any framework with a small amount of glue code.

Getting started

Prerequisites

- afl.git: You will need our patches for reading the kernel instrumentation feedback from /dev/afl or /dev/kcov.
- linux.git: You will need our branch with the AFL patches and satconfig patches:

```
$ git pull https://github.com/vegard/linux-2.6.git v4.6+kconfig-sat
```

- kvm/qemu: KVM is required for certain fuzzers or if you just want to use KVM instead of UML.
- gcc 5: gcc 5 or later is needed for building the kernel using our instrumentation plugin. On Ubuntu you can do:

```
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install gcc-5
```

- Python packages: jinja2 and yaml. On Ubuntu you can do:

```
$ sudo apt-get install python-jinja2
$ sudo apt-get install python-yaml
```

Prepare the kernel branch

1. Check out the kernel version that you want to test
2. Merge the kernel branch into AFL.
3. Merge the satconfig branch:

```
$ git pull https://github.com/vegard/linux-2.6.git v4.6+kconfig-sat
```

4. Set the 'linux_afl_rev' variable in config.yml to point to the resulting commit (can be a branch, a sha1, or even just 'HEAD')

Set up config.yml

Have a look at config-example.yml.

Launch a fuzzer

1. Start screen
2. Run the following command:

```
$ bin/start --master ext4 0
```

This should start a single ext4 fuzzer in the current screen window.

3. If the fuzzer stops or crashes, you can resume it later with:

```
$ bin/start --resume --master ext4 0
```

4. If you want to start additional fuzzers, create a new screen window and run e.g.:

```
$ bin/start ext4 1
```

5. If you want to run many fuzzers in parallel, it is advised that you bind each fuzzer to a single, specific CPU:

```
$ taskset -c 0 bin/start --master ext4 0
$ taskset -c 1 bin/start ext4 1
...
```

How to contribute

Please see CONTRIBUTING.