Manticore

Yubin Hu

Content

- Features
- Installation
- Usage
- Example for WASM

Features

- 程序探索：Manticore 可以执行带有符号输入的程序并探索它可以达到的所有可能状态
- 输入生成：Manticore 可以自动生成导致给定程序状态的具体输入
- 错误发现：Manticore 可以检测二进制文件和智能合约中的崩溃和其他故障情况
- 检测：Manticore 通过事件回调和指令钩子提供对状态探索的细粒度控制
- 编程接口：Manticore 通过 Python API 公开对其分析引擎的编程访问

Installation

- Installing from PyPI
- Installing from PyPI, with extra dependencies needed to execute native binaries
- Installing a nightly development build
- Installing from the master branch
- **Install via Docker**

```
docker run -idt --name=manticore -v ~/Docker/manticore:/data/docker_share
--ulimit stack=100000000:100000000 trailofbits/manticore:latest
```

Usage

- CLI
  - 分析结果将放置在 `mcore_*` 文件夹

```
root@f1e495d82fc1:/manticore# manticore examples/linux/basic
2021-09-08 00:52:17,942: [90822] m.n.manticore:INFO: Loading program examples/linux/basic
2021-09-08 00:52:23,319: [90822] m.c.manticore:INFO: Generated testcase No. 0 - Program finished with exit status: 0
2021-09-08 00:52:23,867: [90822] m.c.manticore:INFO: Generated testcase No. 1 - Program finished with exit status: 0
2021-09-08 00:52:24,211: [90822] m.c.manticore:INFO: Results in /manticore/mcore_phzy89kd
2021-09-08 00:52:24,211: [90822] m.c.manticore:INFO: Total time: 5.977001667022705
```

- API
  - Manticore 提供了一个 Python 编程接口

```
from manticore.wasm import ManticoreWASM
from manticore.core.plugin import Plugin
```

Examply for WASM

- C -> wat/wasm -> symbolic execution
- Stub implementation of the getchar function.

```
def getchar(state):
    """ Symbolic `getchar` implementation. Returns an arbitrary single byte """
    res = state.new_symbolic_value(32, "getchar_res")
    state.constrain(0 < res)
    state.constrain(res < 256)
    return [res]
```

Examply for WASM

- Plugin

```
class PrintRetPlugin(Plugin):
    """ A plugin that looks for states that returned zero and solves for their inputs """

    def will_terminate_state_callback(self, state, *args):
        retval = state.stack.peek()
        if retval == 0:
            print("Solution found!")
            for sym in state.input_symbols:
                solved = state.solve_one(sym)
                print(f"{sym.name}: {chr(solved)} --> Return {retval}")
```

Examply for WASM

- Symbolic execution

```
# Pass our symbolic implementation of the `getchar` function into the WASM environment
# as an import.
m = ManticoreWASM("if_check.wasm", env={"getchar": getchar})

# Register our state termination callback
m.register_plugin(PrintRetPlugin())

# Run the main function, which will call getchar
m.main()

# Save a copy of the inputs to the disk
m.finalize()
```