Manticore

Information Security Inc.

iSEC
information security inc.

Contents

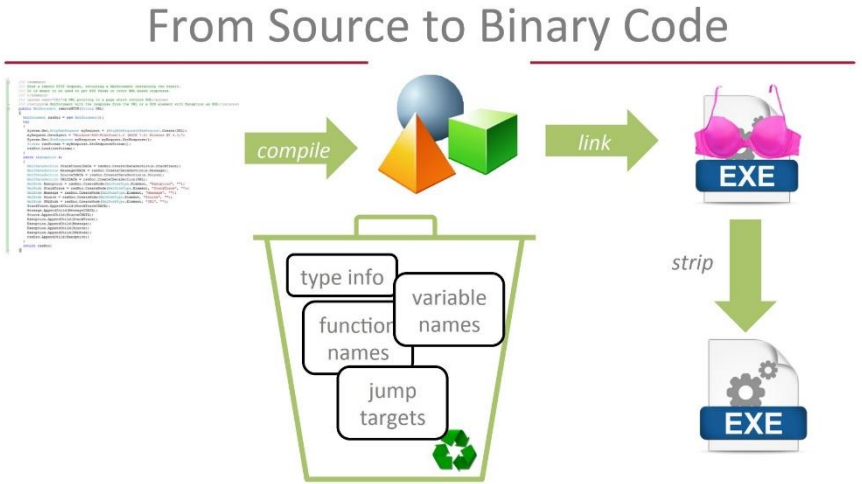
- About Manticore
- From Source to Binary Code
- Features
- Demo Setup
- Installing Manticore
- Using Manticore
- References

About Manticore

- Manticore is a prototyping tool for dynamic binary analysis, with support for symbolic execution, taint analysis, and binary instrumentation

From Source to Binary Code

- Binaries lack significant information present in source



Features

- Input Generation: Manticore automatically generates inputs that trigger unique code paths
- Crash Discovery: Manticore discovers inputs that crash programs via memory safety violations
- Execution Tracing: Manticore records an instruction-level trace of execution for each generated input
- Programmatic Interface: Manticore exposes programmatic access to its analysis engine via a Python API

Features

- Manticore supports binaries of the following formats, operating systems, and architectures. It has been primarily used on binaries compiled from C and C++
- OS/Formats: Linux ELF
- Architectures: x86, x86_64, ARMv7

Demo Setup

- Setup
- Ubuntu 16.04.3 LTS

```
root@admin1-virtual-machine:~# cat /etc/*rel*
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.3 LTS"
NAME="Ubuntu"
VERSION="16.04.3 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.3 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
```

Installing Manticore

- Create a python virtual environment

```
root@admin1-virtual-machine:~# virtualenv manticore
Running virtualenv with interpreter /usr/bin/python2
New python executable in /root/manticore/bin/python2
Also creating executable in /root/manticore/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
root@admin1-virtual-machine:~# . manticore/bin/activate
```

Installing Manticore

- Install z3 dependency

```
(manticore) root@admin1-virtual-machine:~# apt install z3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-image-4.10.0-32-generic linux-image-extra-4.10.0-32-generic
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  z3
0 upgraded, 1 newly installed, 0 to remove and 51 not upgraded.
Need to get 5,239 kB of archives.
After this operation, 16.7 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 z3 amd64 4.4.0-5 [5,239 kB]
Fetched 5,239 kB in 3s (1,545 kB/s)
Selecting previously unselected package z3.
(Reading database ... 293772 files and directories currently installed.)
Preparing to unpack .../archives/z3_4.4.0-5_amd64.deb ...
Unpacking z3 (4.4.0-5) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up z3 (4.4.0-5) ...
```

Installing Manticore

- Using pip to install Manticore

```
(manticore) root@admin1-virtual-machine:~# pip install manticore
Collecting manticore
  Downloading manticore-0.1.4.tar.gz (23.5MB)
     100% |████████████████████████████████| 23.5MB 22kB/s
Collecting capstone>=3.0.5rc2 (from manticore)
  Downloading capstone-3.0.5rc2-py2-none-manylinux1_x86_64.whl (1.6MB)
     100% |████████████████████████████████| 1.6MB 977kB/s
Collecting pyelftools (from manticore)
  Downloading pyelftools-0.24.tar.gz (411kB)
     100% |████████████████████████████████| 419kB 3.5MB/s
Collecting unicorn (from manticore)
  Downloading unicorn-1.0.1-py2.py3-none-manylinux1_x86_64.whl (18.2MB)
     100% |████████████████████████████████| 18.2MB 85kB/s
Collecting ply (from manticore)
  Downloading ply-3.10.tar.gz (150kB)
     100% |████████████████████████████████| 153kB 9.1MB/s
Building wheels for collected packages: manticore, pyelftools, ply
  Running setup.py bdist_wheel for manticore ... done
  Stored in directory: /root/.cache/pip/wheels/16/3f/fe/8e68f3bccfd25b777be233d0db0ba82468d80b2de30fe8d9de
  Running setup.py bdist_wheel for pyelftools ... done
  Stored in directory: /root/.cache/pip/wheels/f0/bd/5e/ced1af292575c94420c85843a251b0c4a43546e68ff801134c
  Running setup.py bdist_wheel for ply ... done
  Stored in directory: /root/.cache/pip/wheels/ad/dd/ad/8ce1991a7b380dfe23d6cc81a4de5c2775bc728b5a0a7721aa
Successfully built manticore pyelftools ply
Installing collected packages: capstone, pyelftools, unicorn, ply, manticore
Successfully installed capstone-3.0.5rc2 manticore-0.1.4 ply-3.10 pyelftools-0.24 unicorn-1.0.1
```

Using Manticore

- Manticore help menu

```
(manticore) root@admin1-virtual-machine:~# manticore -h
usage: manticore [-h] [--assertions ASSERTIONS] [--buffer BUFFER]
                 [--context CONTEXT] [--coverage COVERAGE] [--data DATA]
                 [--dumpafter DUMPAFTER] [--env ENV] [--errorfile ERRORFILE]
                 [--file FILES] [--maxstorage MAXSTORAGE]
                 [--maxstates MAXSTATES] [--maxsymb MAXSYMB] [--names NAMES]
                 [--offset OFFSET] [--policy POLICY] [--profile]
                 [--procs PROCS] [--replay REPLAY] [--size SIZE]
                 [--timeout TIMEOUT] [-v] [--workspace WORKSPACE]
                 PROGRAM [PROGRAM ...]

Symbolically analyze a program

positional arguments:
  PROGRAM                Programs to analyze (arguments after ?)

optional arguments:
  -h, --help             show this help message and exit
  --assertions ASSERTIONS
                         File with additional assertions
  --buffer BUFFER        Specify buffer to make symbolic
  --context CONTEXT      path to file with additional context
  --coverage COVERAGE    where to write the coverage data
  --data DATA            Initial concrete concrete data for the input symbolic
                         buffer
  --dumpafter DUMPAFTER
                         Dump state after every N instructions; 0 to disable
  --env ENV              Specify symbolic environment variable VARNAME=++++++
  --errorfile ERRORFILE
                         where to write the memory error locations
  --file FILES           Specify symbolic input file, '+' marks symbolic bytes
  --maxstorage MAXSTORAGE
                         Storage use cap in megabytes.
  --maxstates MAXSTATES
                         Maximun number of states to mantain at the same time
  --maxsymb MAXSYMB      Maximun number of symbolic bytes to inject
  --names NAMES          File with function addresses to replace with known
                         models
  --offset OFFSET        Buffer header size to leave concrete
  --policy POLICY        Search policy.
                         random|adhoc|uncovered|dicount|icount|syscount|depth.
                         (use + (max) or - (min) to specify order. e.g.
                         +random)
  --profile              Enable profiling mode.
  --procs PROCS          Number of parallel processes to spawn
  --replay REPLAY        The trace filename to replay
  --size SIZE            Specify buffer full size
  --timeout TIMEOUT      Timeout. Abort exploration aftr TIMEOUT seconds
  -v                     Specify verbosity level from -v to -vvvv
  --workspace WORKSPACE
                         A folder name for temporaries and results.(default
                         mcore_?????)
```

Using Manticore

- Usage -> manticore ./path/to/binary
- Runs, and creates a mcore_* directory with analysis results

```
(manticore) root@admin1-virtual-machine:~# manticore Simple
2017-10-06 00:49:15,894: [56514] MANTICORE:INFO: Loading program Simple (platform: Linux)
2017-10-06 00:49:17,290: [56514] MANTICORE:INFO: Starting 1 processes.
2017-10-06 00:51:04,802: [56552] MANTICORE:INFO: Generated testcase No. 0 - Program finished with exit status: 0L
2017-10-06 00:51:04,829: [56514] MANTICORE:INFO: Results in /root/mcore_5viL0U
2017-10-06 00:51:04,830: [56514] MANTICORE:INFO: Instructions executed: 137922
2017-10-06 00:51:04,830: [56514] MANTICORE:INFO: Coverage: 8280 different instructions executed
2017-10-06 00:51:04,830: [56514] MANTICORE:INFO: Total time: 107.538961887
2017-10-06 00:51:04,831: [56514] MANTICORE:INFO: IPS: 1282
```

Using Manticore

- Usage -> manticore ./path/to/binary
- Runs, and creates a mcore_* directory with analysis results

```
(manticore) root@admin1-virtual-machine:~# cd mcore_5viL0U/
(manticore) root@admin1-virtual-machine:~/mcore_5viL0U# ls -alh
total 3.9M
drwx------  2 root root 4.0K Oct  6 00:51 .
drwx------ 48 root root 4.0K Oct  6 00:49 ..
-rw-r--r--  1 root root   36 Oct  6 00:51 command.sh
-rw-r--r--  1 root root 1.9K Oct  6 00:51 test_00000000.messages
-rw-r--r--  1 root root 1.8M Oct  6 00:51 test_00000000.pkl
-rw-r--r--  1 root root    0 Oct  6 00:51 test_00000000.smt
-rw-r--r--  1 root root    0 Oct  6 00:51 test_00000000.stdin
-rw-r--r--  1 root root   11 Oct  6 00:51 test_00000000.stdout
-rw-r--r--  1 root root 3.0K Oct  6 00:51 test_00000000.syscalls
-rw-r--r--  1 root root 2.0M Oct  6 00:51 test_00000000.trace
-rw-r--r--  1 root root 1.1K Oct  6 00:51 test_00000000.txt
-rw-r--r--  1 root root 121K Oct  6 00:51 visited.txt
```

Using Manticore

- Reading test_00000000.messages can see program exit code; Memory map and registry info

```
(manticore) root@admin1-virtual-machine:~/mcore 5viL0U# more test_00000000.messages
Command line:
  '/root/manticore/bin/manticore Simple'
Status:
  Program finished with exit status: 0L

================ PROC: 00 ================
Memory:
0000000000400000-0000000000401000   r x 00000000 Simple
  0000000000600000-0000000000601000   r   00000000
  0000000000601000-0000000000602000   rw  00000000
  0000000000602000-0000000000623000   rw  00000000
  00007fffff9ed000-00007fffffbad000   r x 00000000
  00007fffffbad000-00007fffffdad000       00000000
  00007fffffdad000-00007fffffdb1000   r   00000000
  00007fffffdb1000-00007fffffdb3000   rw  00000000
  00007fffffdb3000-00007fffffdb7000   rw  00000000
  00007fffffdb7000-00007fffffdd1000   r x 00000000 /lib64/ld-linux-x86-64.so.2
  00007ffffffba000-00007ffffffbb000   rw  00000000
  00007ffffffbb000-00007ffffffbc000   rw  00000000
  00007ffffffbc000-00007ffffffbd000   rw  00000000
  00007ffffffda000-00007ffffffdc000   rw  00000000
  00007ffffffdc000-00007ffffffdd000   r   00000000
  00007ffffffdd000-00007ffffffdf000   rw  00000000
  00007ffffffdf000-0000800000000000   rwx 00000000 CPU:
Instruction: 0x00007fffffab9746:        syscall
RAX: 0x00000000000000e7
RCX: 0x00007fffffab9748
RDX: 0x0000000000000000
RBX: 0x0000000000000000
RSP: 0x00007fffffffda8
RBP: 0x00007fffffdad8e0
RSI: 0x000000000000003c
RDI: 0x0000000000000000
 R8: 0x00000000000000e7
 R9: 0xffffffffffffff98
R10: 0x00007fffffffd00
R11: 0x0000000000000044
R12: 0x00007fffffdad8e0
R13: 0x00007fffffdb2c40
R14: 0x0000000000000000
R15: 0x0000000000000000
RIP: 0x00007fffffab9748
EFLAGS: 0x0000000000000044
CF: 0
SF: 0
```

References

- HackingReviews
https://www.hacking.reviews/2017/04/manticore-dynamic-binary-analysis-tool.html?m=0

- Ubuntu Linux
https://www.ubuntu.com/download

- Wikipedia
https://en.wikipedia.org/wiki/Static_program_analysis
https://en.wikipedia.org/wiki/Symbolic_execution

- PIC (Position Independent Code)
https://en.wikipedia.org/wiki/Position-independent_code

- Executable stack protection
https://en.wikipedia.org/wiki/Executable_space_protection