iSEC
information security inc.

Triton

Information Security Inc.

iSEC
information security inc.

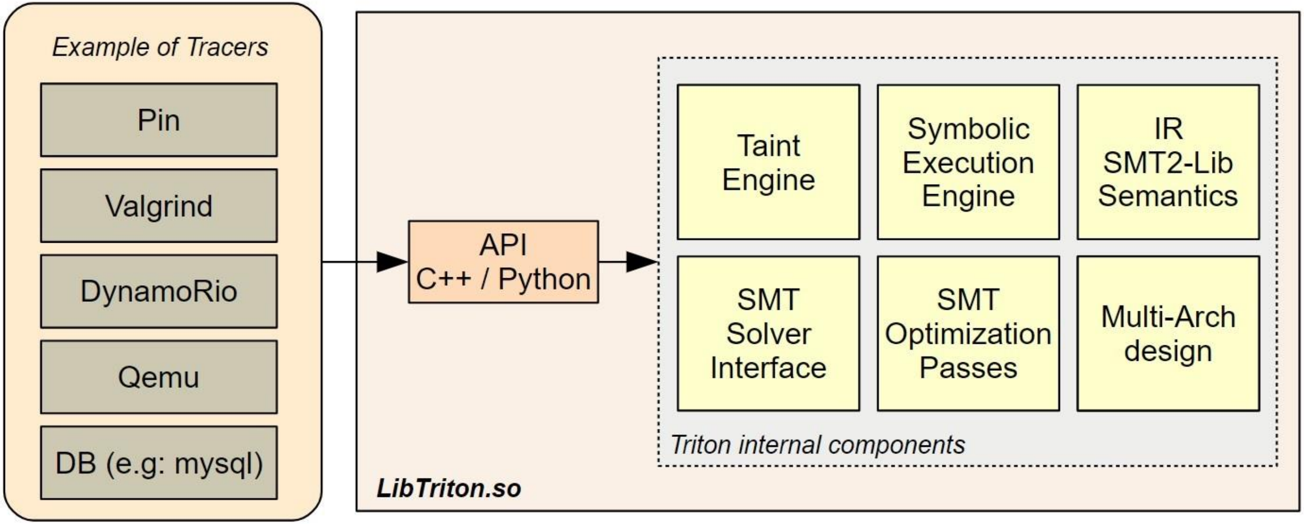
Contents

- About Triton
- From Source to Binary Code
- Triton’s design
- Demo Setup
- Required libraries
- Installing Triton
- Using Triton
- References

About Triton

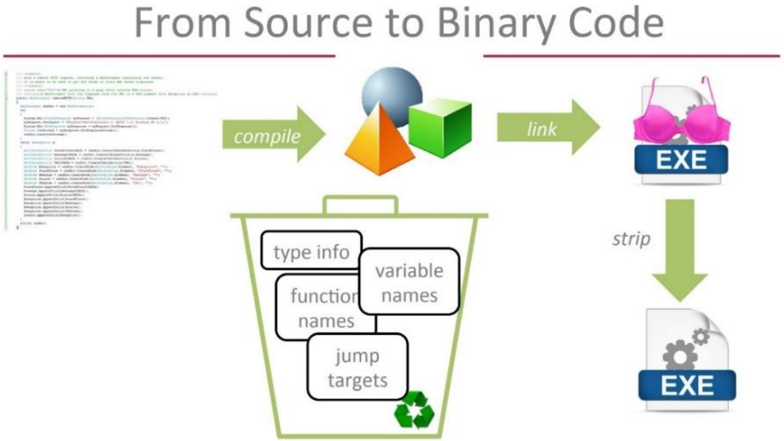
- **Triton** is a dynamic binary analysis (DBA) framework
- It provides Dynamic Symbolic Execution (DSE) engine, a Taint Engine, AST representations of the x86 and the x86-64 instructions set semantics, SMT simplification passes, an SMT Solver Interface and Python bindings

Triton's design



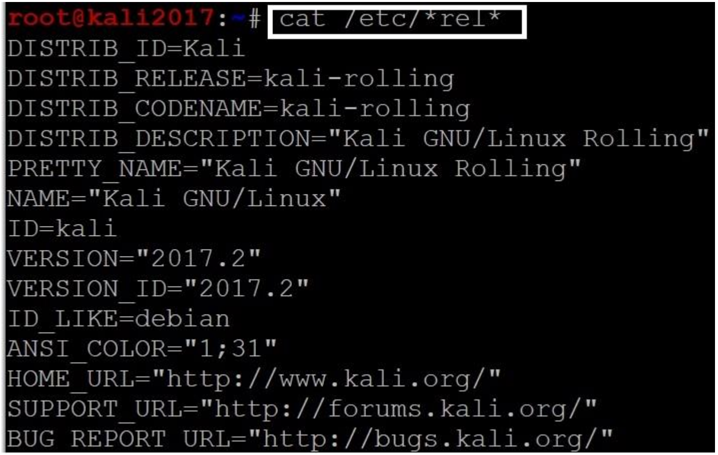
From Source to Binary Code

- Binaries lack significant information present in source



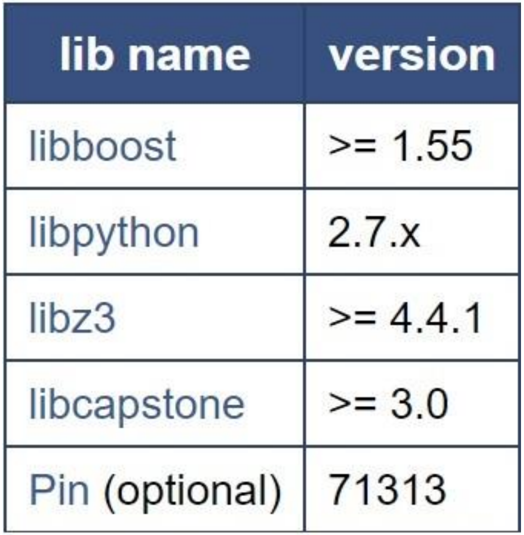
Demo Setup

- Setup
- Kali Linux 2017



```
root@kali2017:~# cat /etc/*rel*
DISTRIB_ID=Kali
DISTRIB_RELEASE=kali-rolling
DISTRIB_CODENAME=kali-rolling
DISTRIB_DESCRIPTION="Kali GNU/Linux Rolling"
PRETTY_NAME="Kali GNU/Linux Rolling"
NAME="Kali GNU/Linux"
ID=kali
VERSION="2017.2"
VERSION_ID="2017.2"
ID_LIKE=debian
ANSI_COLOR="1;31"
HOME_URL="http://www.kali.org/"
SUPPORT_URL="http://forums.kali.org/"
BUG_REPORT_URL="http://bugs.kali.org/"
```

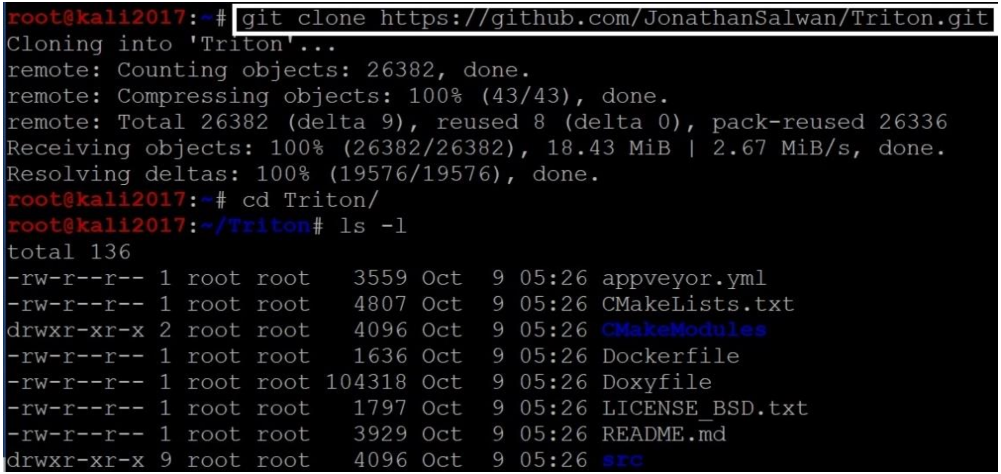
Required libraries



lib name	version
libboost	>= 1.55
libpython	2.7.x
libz3	>= 4.4.1
libcapstone	>= 3.0
Pin (optional)	71313

Installing Triton

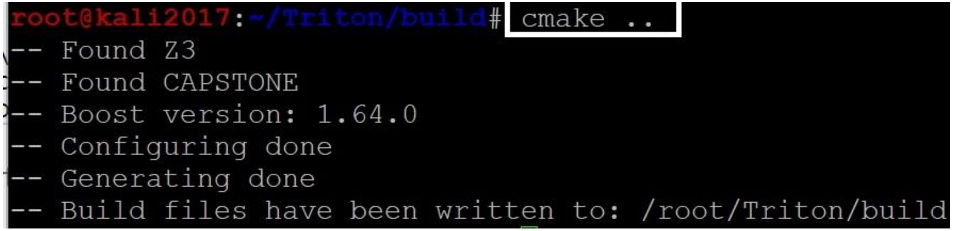
- Clone the GitHub repository



```
root@kali2017:~# git clone https://github.com/JonathanSalwan/Triton.git
Cloning into 'Triton'...
remote: Counting objects: 26382, done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 26382 (delta 9), reused 8 (delta 0), pack-reused 26336
Receiving objects: 100% (26382/26382), 18.43 MiB | 2.67 MiB/s, done.
Resolving deltas: 100% (19576/19576), done.
root@kali2017:~# cd Triton/
root@kali2017:~/Triton# ls -l
total 136
-rw-r--r-- 1 root root   3559 Oct  9 05:26 appveyor.yml
-rw-r--r-- 1 root root   4807 Oct  9 05:26 CMakeLists.txt
drwxr-xr-x 2 root root   4096 Oct  9 05:26 CMakeModules
-rw-r--r-- 1 root root   1636 Oct  9 05:26 Dockerfile
-rw-r--r-- 1 root root 104318 Oct  9 05:26 Doxyfile
-rw-r--r-- 1 root root   1797 Oct  9 05:26 LICENSE_BSD.txt
-rw-r--r-- 1 root root   3929 Oct  9 05:26 README.md
drwxr-xr-x 9 root root   4096 Oct  9 05:26 src
```

Installing Triton

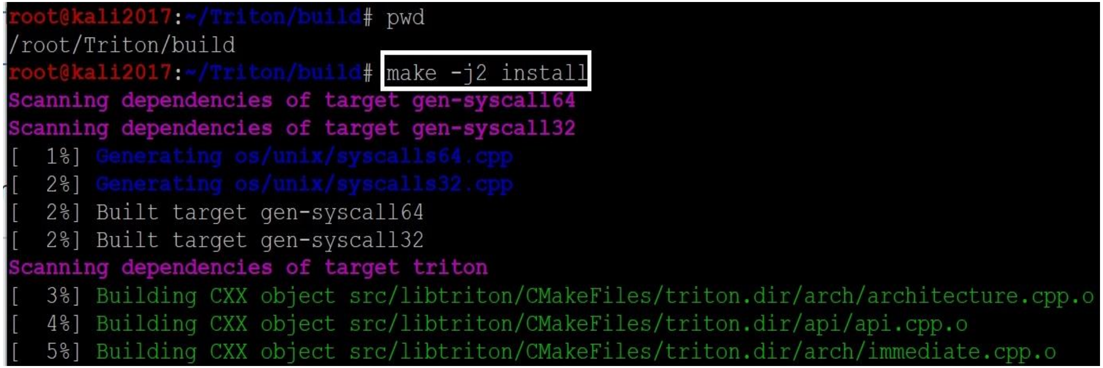
- Build libTriton



```
root@kali2017:~/Triton/build# cmake ..
-- Found Z3
-- Found CAPSTONE
-- Boost version: 1.64.0
-- Configuring done
-- Generating done
-- Build files have been written to: /root/Triton/build
```

Installing Triton

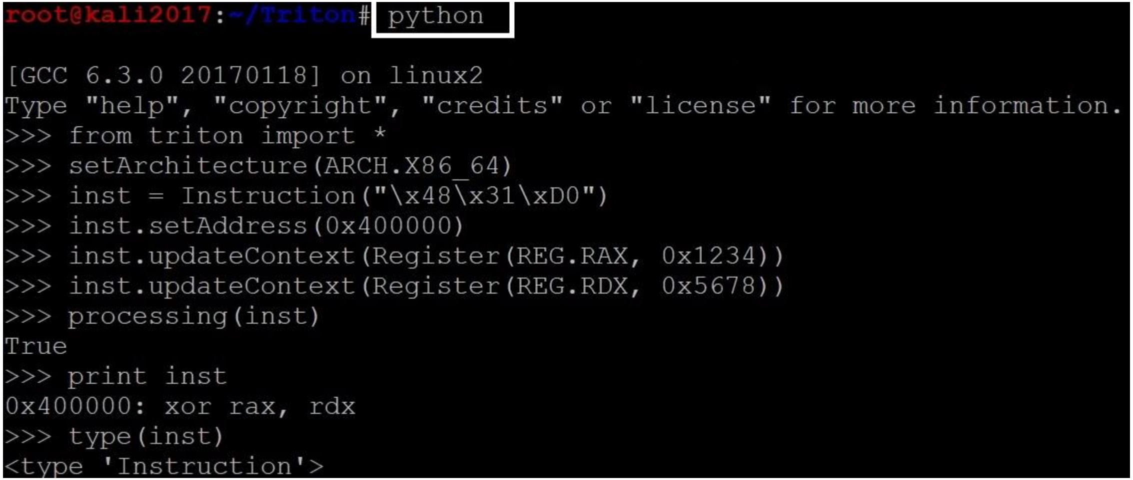
- Build libTriton



```
root@kali2017:~/Triton/build# pwd
/root/Triton/build
root@kali2017:~/Triton/build# make -j2 install
Scanning dependencies of target gen-syscall64
Scanning dependencies of target gen-syscall32
[  1%] Generating os/unix/syscalls64.cpp
[  2%] Generating os/unix/syscalls32.cpp
[  2%] Built target gen-syscall64
[  2%] Built target gen-syscall32
Scanning dependencies of target triton
[  3%] Building CXX object src/libtriton/CMakeFiles/triton.dir/arch/architecture.cpp.o
[  4%] Building CXX object src/libtriton/CMakeFiles/triton.dir/api/api.cpp.o
[  5%] Building CXX object src/libtriton/CMakeFiles/triton.dir/arch/immediate.cpp.o
```

Using Triton

- Define an opcode and context (# xor rax,rdx)



```
root@kali2017:~/Triton# python

[GCC 6.3.0 20170118] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from triton import *
>>> setArchitecture(ARCH.X86_64)
>>> inst = Instruction("\x48\x31\xD0")
>>> inst.setAddress(0x400000)
>>> inst.updateContext(Register(REG.RAX, 0x1234))
>>> inst.updateContext(Register(REG.RDX, 0x5678))
>>> processing(inst)
True
>>> print inst
0x400000: xor rax, rdx
>>> type(inst)
<type 'Instruction'>
```

References

- Quarkslab
https://triton.quarkslab.com/

- Kali Linux
https://www.kali.org/

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

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

- Taint analysis
http://shell-storm.org/blog/Taint-analysis-and-pattern-matching-with-Pin/