



Qiling Framework

Qiling : Advanced Binary Emulation Framework

- Qiling (Python)
- Unicorn engine
- QEMU



About the project

- First public presentation november 2019:
https://www.qiling.io/docs/qiling_beta2019.pdf
- Cross platform: Windows, MacOS, Linux, BSD
- Cross architecture: X86, X86_64, Arm, Arm64, Mips
- Multiple file formats: PE, MachO, ELF
- isolated environment + GDB remote debug
- Fine-grain instrumentation: allow hooks at various levels (instruction/basic-block/memory-access/exception/syscall/IO/etc)

```
(00:19:53):xwings@kamino:~/qiling>
(169)$ cat examples/shellcodes/lin32_execve.asm
xor eax,eax
push eax
push 0x68732f2f
push 0x6e69622f
xchg ebx,esp
mov al,0xb
int 0x80
```

```
(00:19:56):xwings@kamino:~/qiling>
(170)$ python3 qltool.py shellcode --arch x86 --os linux --asm --output debug -f examples/shellcodes/lin32_execve.
asm
>>> Load ASM from FILE
>>> SET_THREAD_AREA selector : 0x83
>>> SET_THREAD_AREA selector : 0x8b
>>> SET_THREAD_AREA selector : 0x90
>>> Tracing basic block at 0x1000000
>>> 0x1000000 31 c0 xor eax, eax
|--->>> REG0= 0x0 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
>>> 0x1000002 50 push eax
|--->>> REG0= 0x0 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
>>> 0x1000003 68 2f 2f 73 68 push 0x68732f2f
|--->>> REG0= 0x0 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
>>> 0x1000008 68 2f 62 69 6e push 0x6e69622f
|--->>> REG0= 0x0 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
>>> 0x100000d 87 e3 xchg ebx, esp
|--->>> REG0= 0x0 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
>>> 0x100000f b0 0b mov al, 0xb
|--->>> REG0= 0x10ffff4 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
>>> 0x1000011 cd 80 int 0x80
|--->>> REG0= 0x10ffff4 REG1= 0x0 REG2= 0x0 REG3= 0x0 REG4= 0x0 REG5= 0x0
execve(b'/bin//sh', [b''])
(00:20:07):xwings@kamino:~/qiling>
(171)$ |
```

- How about executing wannacry on a Linux host on Qiling Framework and debugging it using IDApro :

The image displays the IDA Pro interface with several panels:

- Disassembly View:** Shows assembly instructions such as `sub esp, 300`, `push esi`, `mov ecx, 0ch`, `lea ecx, [ecx+eax]`, and `rep movsd`. A yellow highlight is visible on the `rep movsd` instruction.
- General registers:** Lists registers like EAX, ECX, EDI, etc., with their current values. For example, EAX is 00400000 and ECX is 00000000.
- Debugger Console:** Shows a list of breakpoints and a `gdb>` session. The session includes commands like `resume at: 0x408145`, `breakpoint added: 0x408145`, and `breakpoint remove: 0x408145`.
- Hex View:** Displays the raw hex dump of the code, with a yellow highlight on the instruction `00408170 51 50 58 50 6A 01 50 08`.
- Stack View:** Shows the stack memory layout with addresses and hex values.

<https://github.com/qilingframework/qiling>

qilingframework / qiling

Watch 33

★ Star 554

🍴 Fork 88

↔ Code

🔔 Issues 10

🔗 Pull requests 0

▶ Actions

📁 Projects 1

🛡 Security

📊 Insights

Qiling Advanced Binary Emulation framework <https://qiling.io>

binary

emulator

framework

unicorn-emulator

malware

analysis

qiling

reverse-engineering

📄 980 commits

🌿 2 branches

📦 0 packages

📦 0 releases

👤 19 contributors

📄 GPL-2.0

● Python 99.9%


● Dockerfile 0.1%

Branch: master ▾

New pull request

Find file

Clone or download ▾

 xwings Merge branch 'master' of github.com:qilingframework/qiling

✓ Latest commit f787bc8 5 hours ago

📁 docs gdb: update docs 14 hours ago

📁 examples gdb: command ! 4 days ago

📁 qiling f9 bug and fix f9 issue 5 hours ago