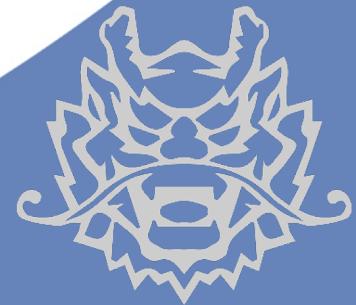


Qiling Framework: Introduction

November, 2020



About xwings



JD.COM

Beijing, Stays in the lab 24/7 by hoping making the world a better place

- > IoT Research
- > Blockchain Research
- > Fun Security Research



Qiling Framework

Cross platform and multi architecture advanced binary emulation framework

- > <https://qiling.io>
- > Lead Developer
- > Founder



HACKERSBADGE.COM

Badge Maker

Electronic fan boy, making toys from hacker to hacker

- > Reversing Binary
- > Reversing IoT Devices
- > Part Time CtF player

Badge Designer for Hacking Conferences



Some Recent Talk (Partial)

- > 2016, Qcon, Beijing, Speaker, nRF24L01 Hijacking
- > 2016, Kcon, Beijing, Speaker, Capstone Unicorn Keystone
- > 2017, Kcon, Beijing, IoT Hacking Trainer
- > 2018, Kcon, Beijing, IoT Hacking Trainer
- > 2018, Brucon, Brussel, Speaker, IoT Virtualization
- > 2018, H2HC, San Paolo, Speaker, IoT Virtualization
- > 2018, HITB, Beijing/Dubai, Speaker, IoT Virtualization
- > 2018, beVX, Hong Kong, Speaker, HackCUBE - Hardware Hacking

- > 2019, DEFCON USA, Qiling Framework Preview
- > 2019, Zeronights, Qiling Framework to Public
- > 2020, Nullcon GOA, Building Reversing Tools with Qiling
- > 2020, HITB AMS, Building Reversing Tools with Qiling
- > 2020, HITB Singapore, Training, How to Hack IoT with Qiling
- > 2020, HITB UAE, Training, Lightweight Binary Analyzer
- > 2020, Blackhat USA, Building IoT Fuzzer with Qiing
- > 2020, Blackhat Singapore, Lightweight Binary Analyzer
- > 2020, Blackhat Europe, Deep Dive Into Obfuscated Binary

Qiling Framework

- > Cross platform and cross architecture binary instrumentation framework
- > Emulate and instrument ARM, ARM64, MIPS, X86 and X86_64
- > Emulate and instrument Linux, MacOS, iOS, Windows and FreeBSD
- > High-level Python API access to register, CPU and memory
- > 1,700+ Github star, more than 9,000+ pypi download, 60+ contributors worldwide
- > Contributor from Dell, Intel, SentinelOne and etc

About lazymio && kabeor

~ \$ whoami
Lazymio



~ \$ file **Lazymio**
The sheperd lab, JD security, Security Engineer.
CTF player, member of Lancet.
GeekPwn 2019 Hall of Fame.

~ \$ ls -l **Lazymio**
Reverse engineering.
Binary analysis.
Writing code for fun.

~ \$ which **Lazymio**
Github: <https://github.com/wtdcode>
Blog: <https://blog.lazym.io/>
Twitter: <https://twitter.com/pwnedmio>

Name: kabeor



Security Engineer at The Shepherd Lab, JD Security.

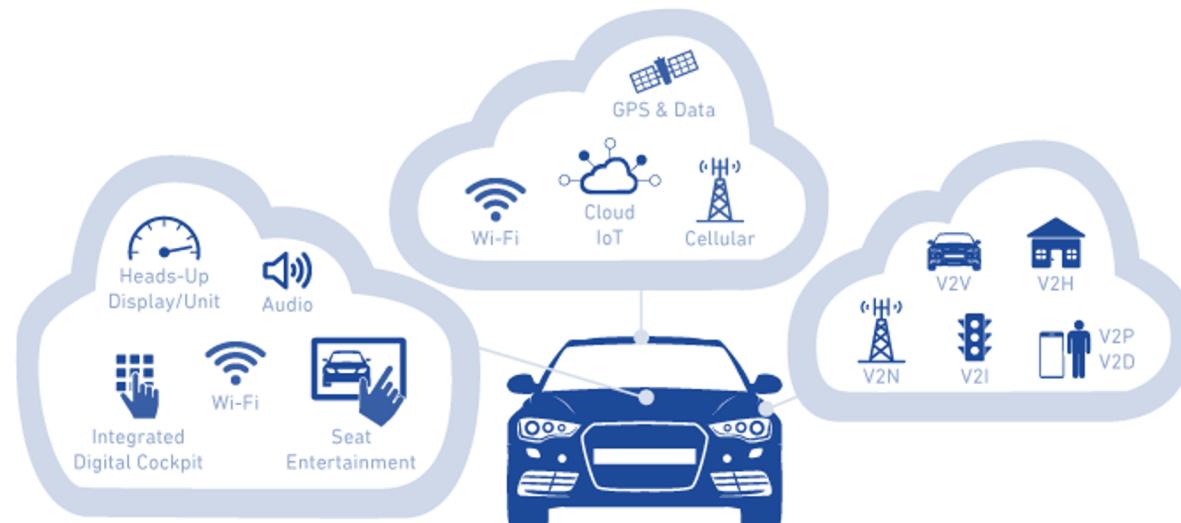
Core developer of Qiling.

BlackHat Asia & Europe 2020 - Speaker
China kanxue SDC 2020 - Speaker
HTIB Training 2020 - Speaker

Github: <https://github.com/kabeor>
Blog: <https://kabeor.cn>
Twitter: https://twitter.com/Angrz3_K

Make IoT Reverse Engineering Great

Today's IoT Analysis



To under a car's firmware

First, you need THAT car & not destroy THAT car

How We Fix It

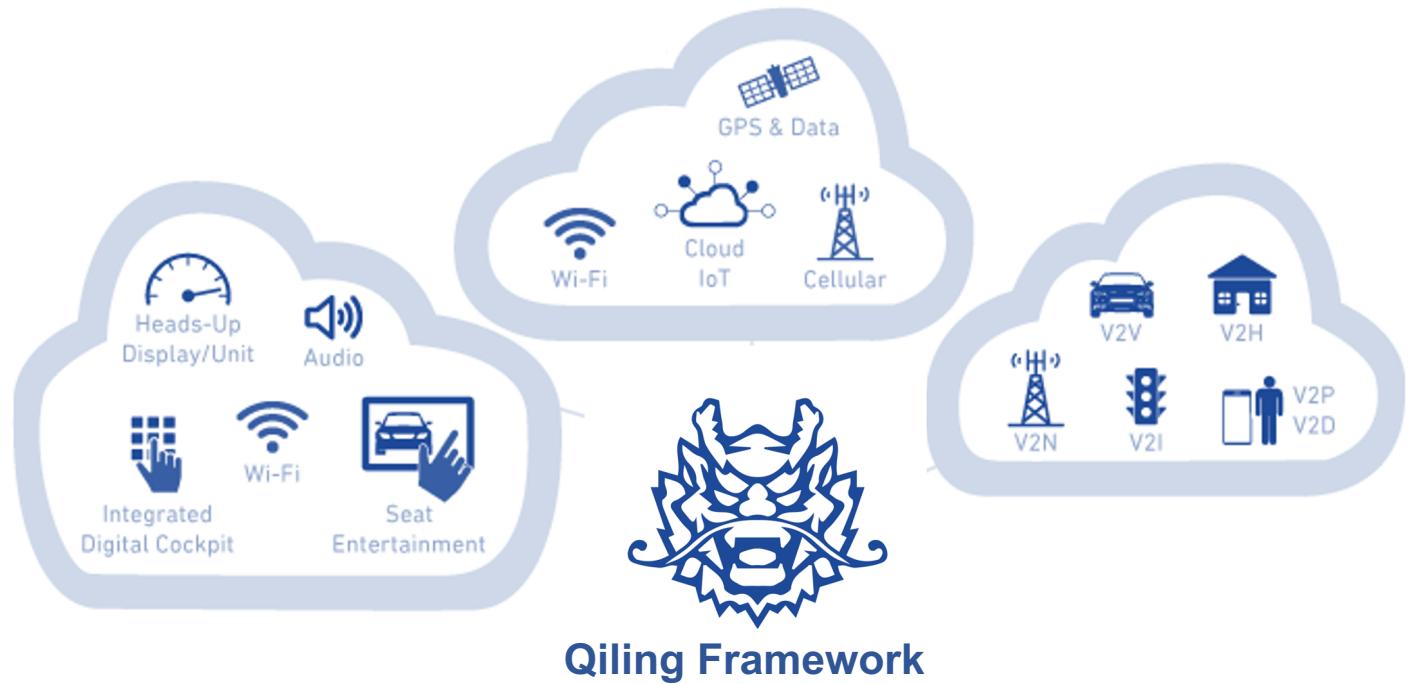
GPS

WiFi

Bluetooth

Audio

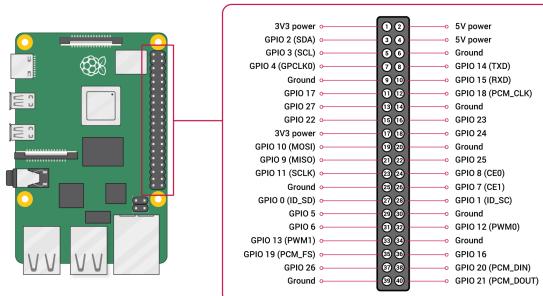
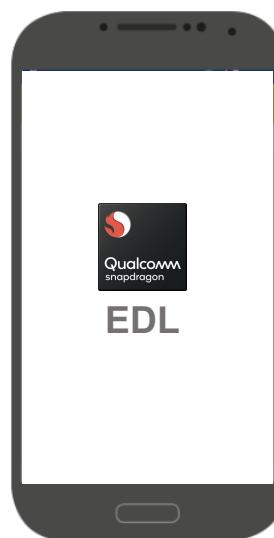
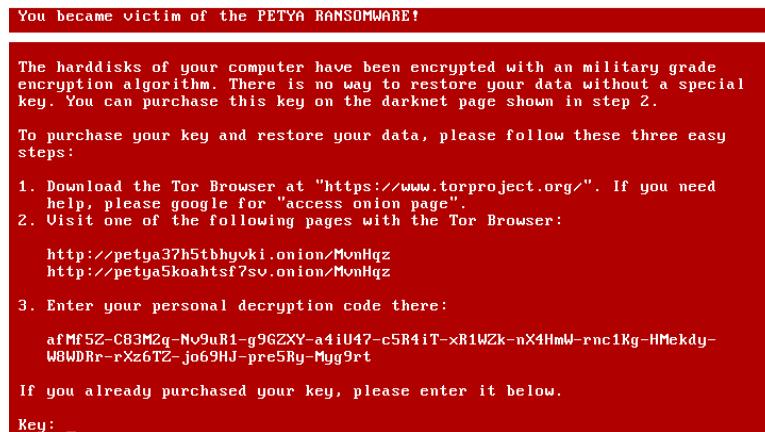
Screen



Bring the entire car firmware's binary into emulation
with virtual devices support

Wait, There are Virtual Machines

Current Virtual Machine Limitation



MBR

UEFI

Smart Contract

GPIO

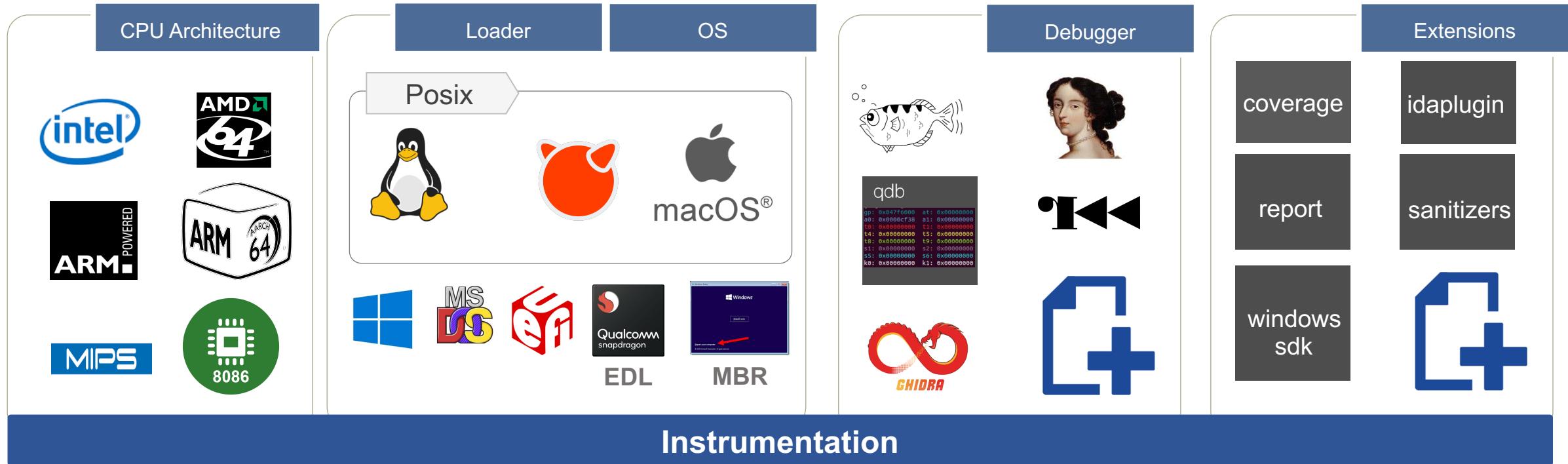
Anti-Anti Debug

Qualcomm EDL

Most modern platform are either limited or NONE emulation or proper analysis tools

Qiling Framework

Overview



External Hardware Emulation

Qiling Framework

Features

- Cross platform: Windows, MacOS, Linux, BSD, UEFI, MBR
- Cross architecture: X86, X86_64, Arm, Arm64, MIPS, 8086
- Multiple file formats: PE, UEFI(PE), MachO, ELF, EDL (ELF), COM
- Emulate & sandbox machine code in a isolated environment
- Provide high level API to setup & configure the sandbox
- Fine-grain instrumentation: allow hooks at various levels (instruction/basic-block/memory-access/exception/syscall/IO/etc)
- Allow dynamic hotpatch on-the-fly running code, including the loaded library
- True Python framework, making it easy to build customized analysis tools on top
- GDBServer support - GDB/IDA/r2
- IDA Plugin
- OS profiling support

	8086	x86	x86-64	ARM	ARM64	MIPS
Windows (PE)	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	<input type="checkbox"/>	-
Linux (ELF)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MacOS (MachO)	-	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-	<input type="checkbox"/>	-
BSD (ELF)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UEFI	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-	-
DOS (COM)	<input checked="" type="checkbox"/>	-	-	-	-	-
MBR	<input checked="" type="checkbox"/>	-	-	-	-	-

Similarity

User Mode Emulation



gemu-usermode

- The TOOL
- Limited OS Support, Very Limited
- No Multi OS Support
- No Instrumentation
- **Syscall Forwarding**



usercorn

- Very good project !
- It's a Framework !
- Mostly *nix based only
- Limited OS Support (No Windows)
- Go and Lua is not hacker's friendly
- **Syscall Forwarding**



Binee

- Very good project too
- Only X86 (32 and 64)
- Limited OS Support
- Only PE Files
- Just a tool, we don't need a tool
- Again, is GO



WINE

- Limited ARCH Support
- Limited OS Support, only Windows
- Not Sandbox Designed
- No Instrumentation



Speakeasy

- Very good project too
- X86 32 and 64
- PE files and Driver
- Limited OS Support
- Only Windows



Zelos

- Very good project !
- It's a Framework !
- Linux based only (No Windows)
- Incomplete support for Linux multi arch

Framework

Framework, NOT Tools

EFI Fuzzer

A screenshot of the GitHub repository for Sentinel-One/efi_fuzz. The repository has 15 commits, 2 branches, and 0 tags. The most recent commit by liba2k is "Adding docker support. (#5)" from 24 days ago. Other commits include "NotMyUefiFault" and "Initial public commit". The repository includes files like README.md, efi_fuzz.py, requirements.txt, and sanitizer.py.

Decoder

A screenshot of the GitHub repository for nmantani/FileInsight-plugins. The repository has 214 commits, 1 branch, and 16 tags. The most recent commit by nmantani is "Use "py.exe --list" instead of hard-coded paths to check Python 3 ins..." from 2 days ago. A screenshot of the McAfee FileInsight hex editor interface is shown, displaying a file named "malicious.pdf" with various hex and ASCII data.

VAC3 Emulator

A screenshot of the GitHub repository for ioncodes/vacation3-emu. The repository has 3 commits, 1 branch, and 0 tags. The most recent commit by ioncodes is "added link" from Sep 29. The repository includes files like README.md, images, .gitignore, README.md, emu.py, and typedefs.h.

IoT Emulator **MacOS Emulator**
IOS Emulator **Binary Decrypt**

Qiling Framework

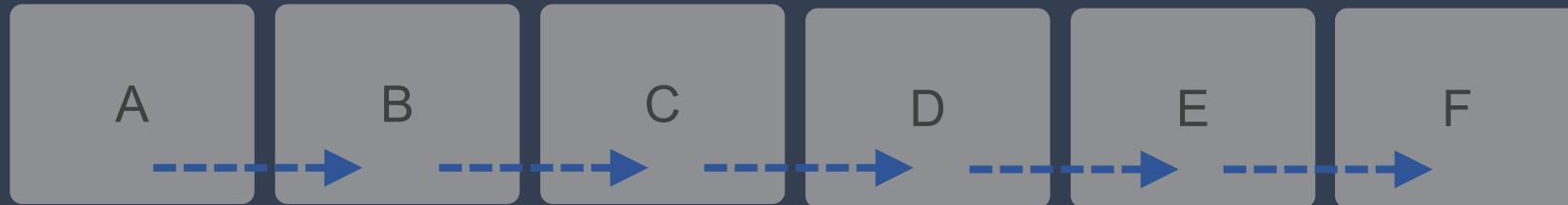


Instrumentation (Qiling's API)

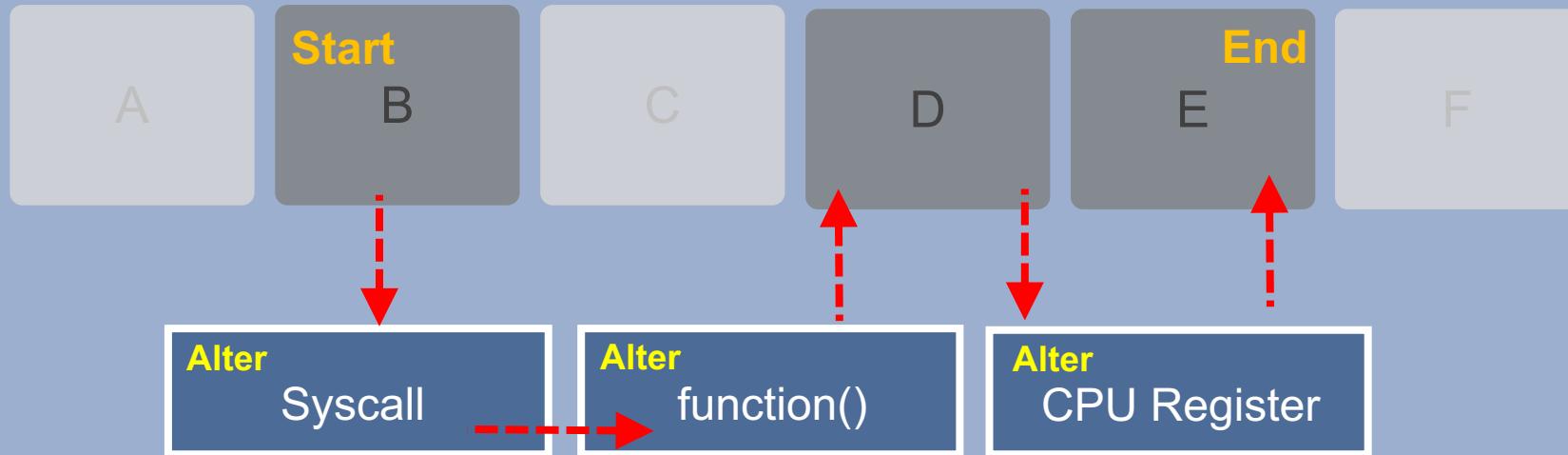
Instrumentation

What Is Instrumentation

Binary Execution Flow

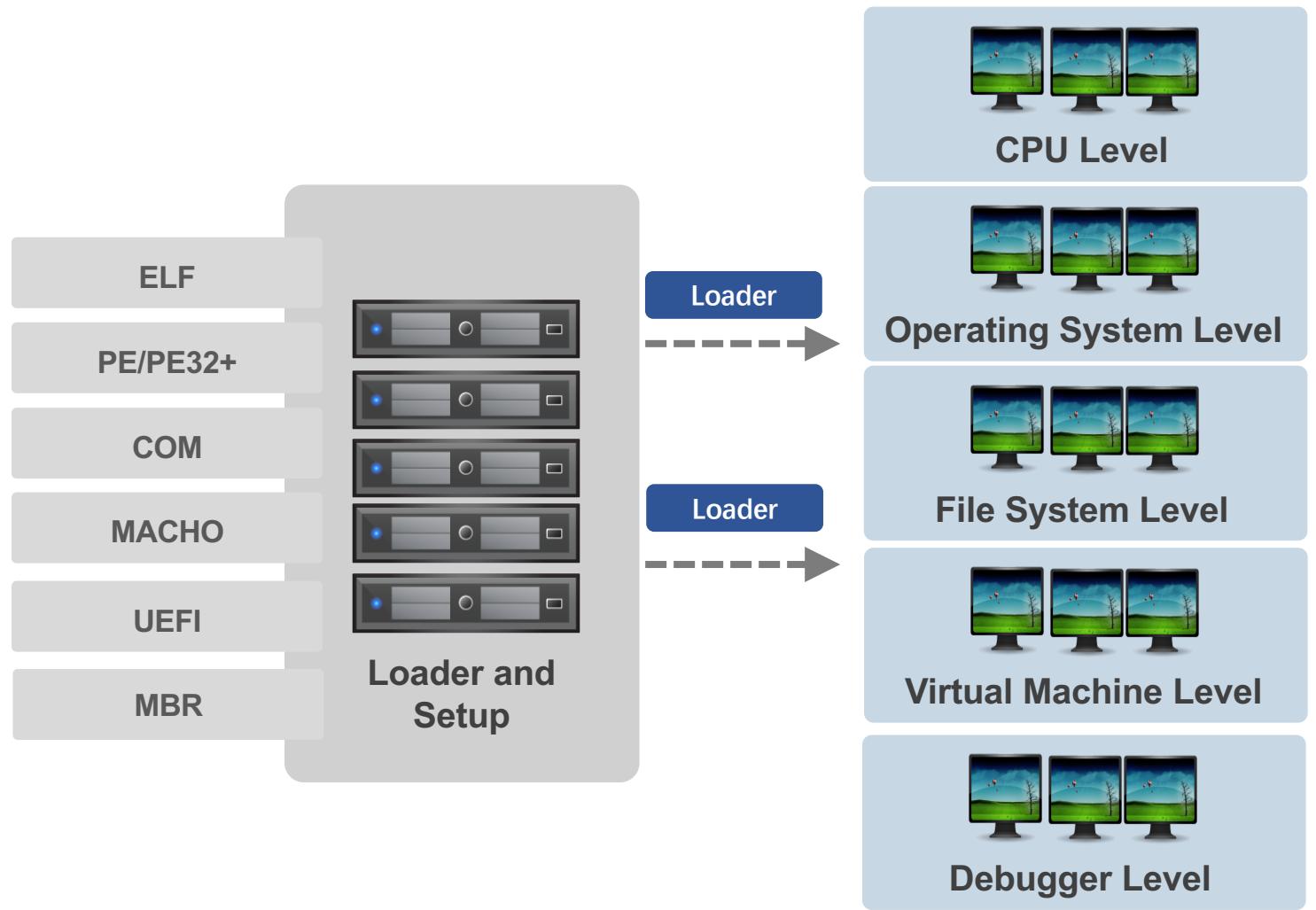


Qiling's Instrumentation

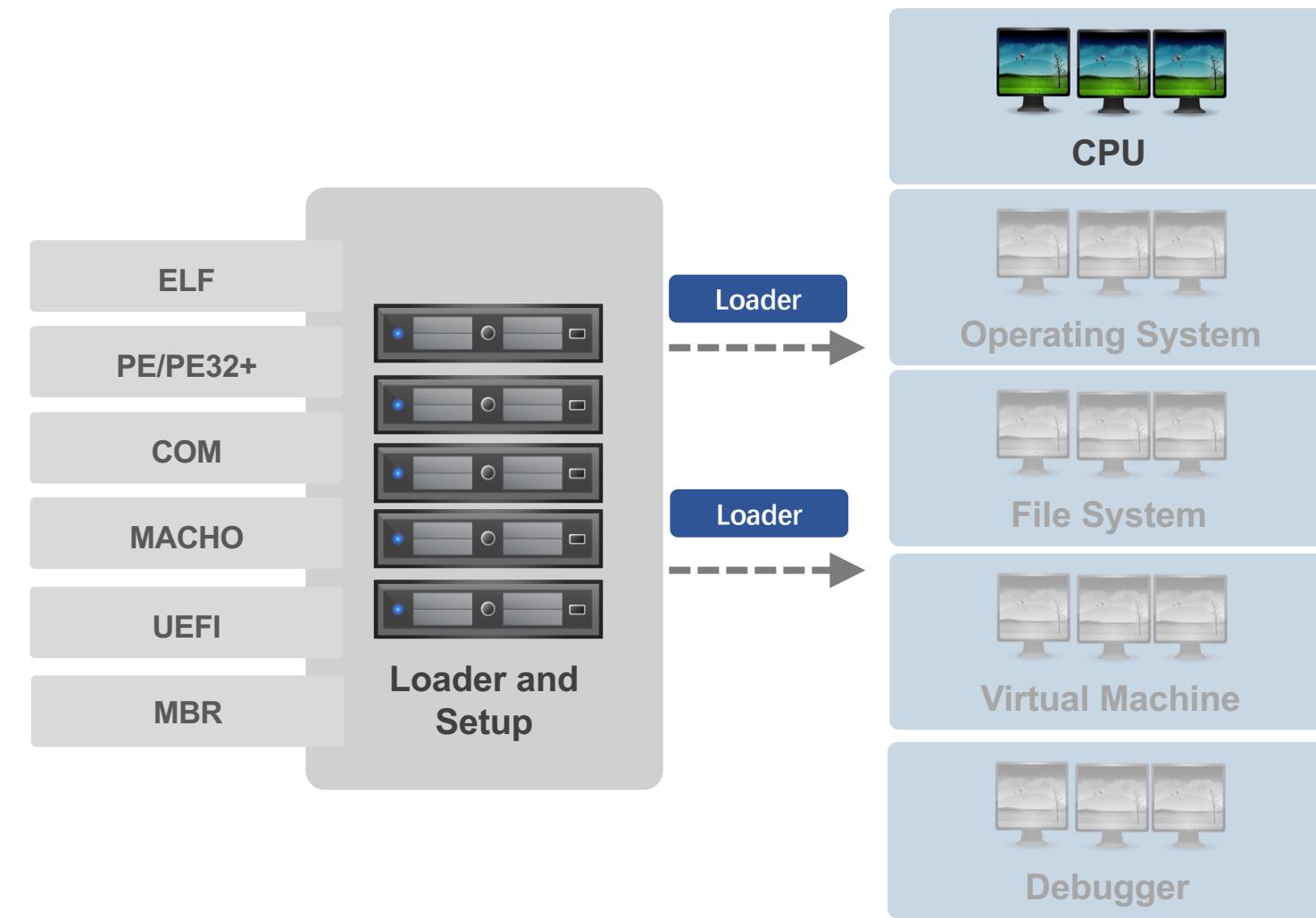


Qiling and APIs

Qiling Framework and Its Interface

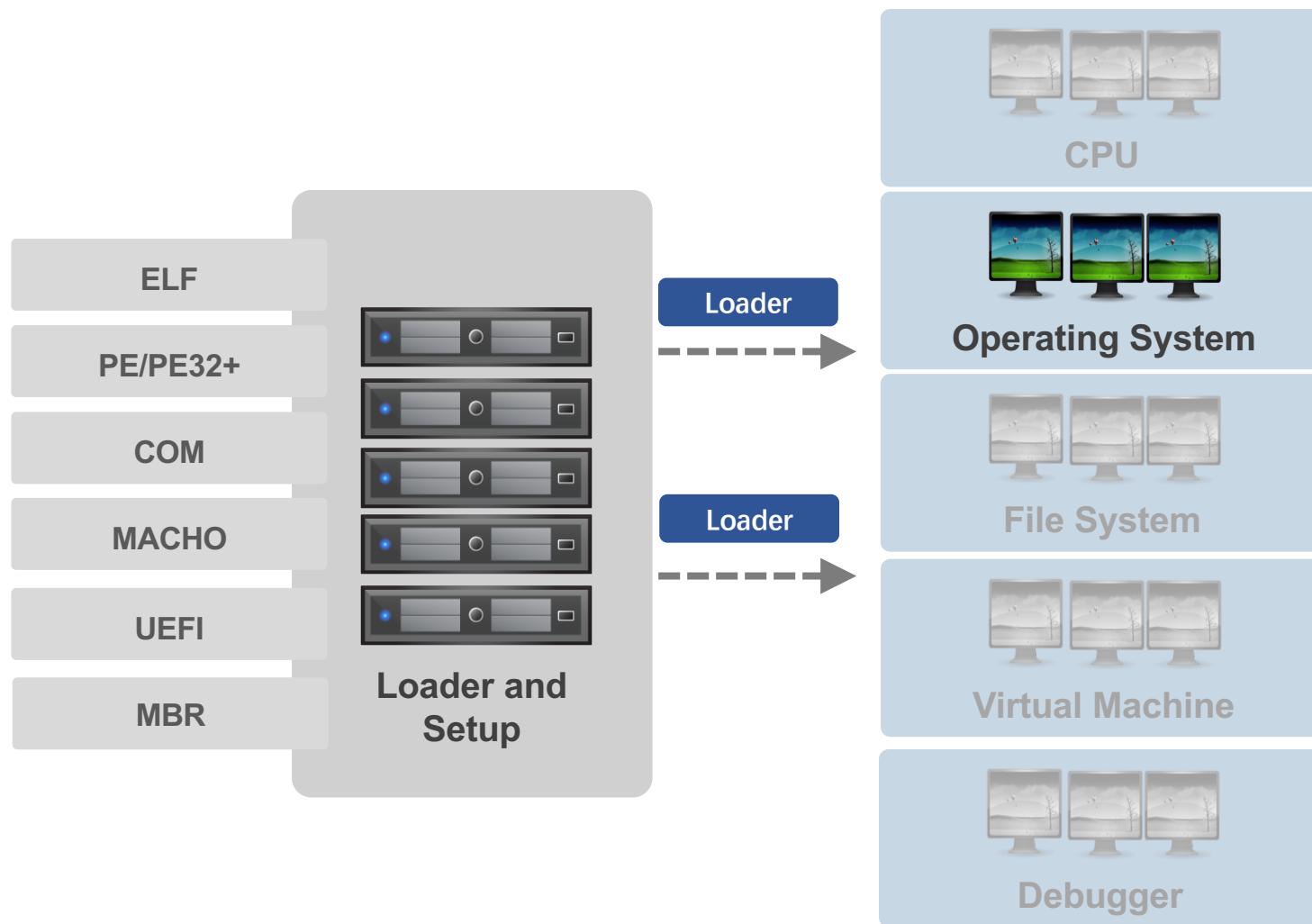


Qiling Framework: CPU



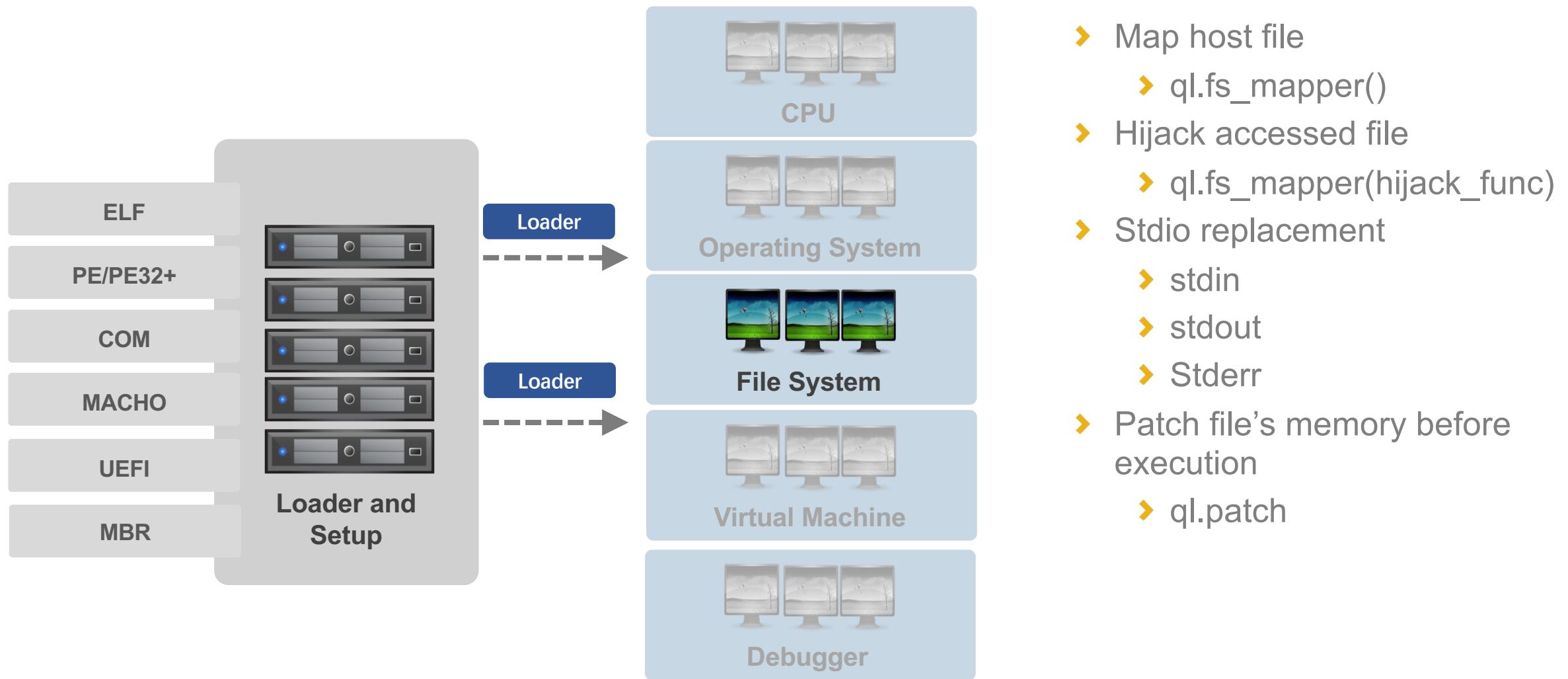
- Access to Register
 - Reading register
 - ql.reg.eax
 - Writing to register
 - ql.reg.eax = 0x41
 - Different Hooks
 - ql.hook_code
 - ql.hook_address
 - and more

Qiling Framework: Operating System



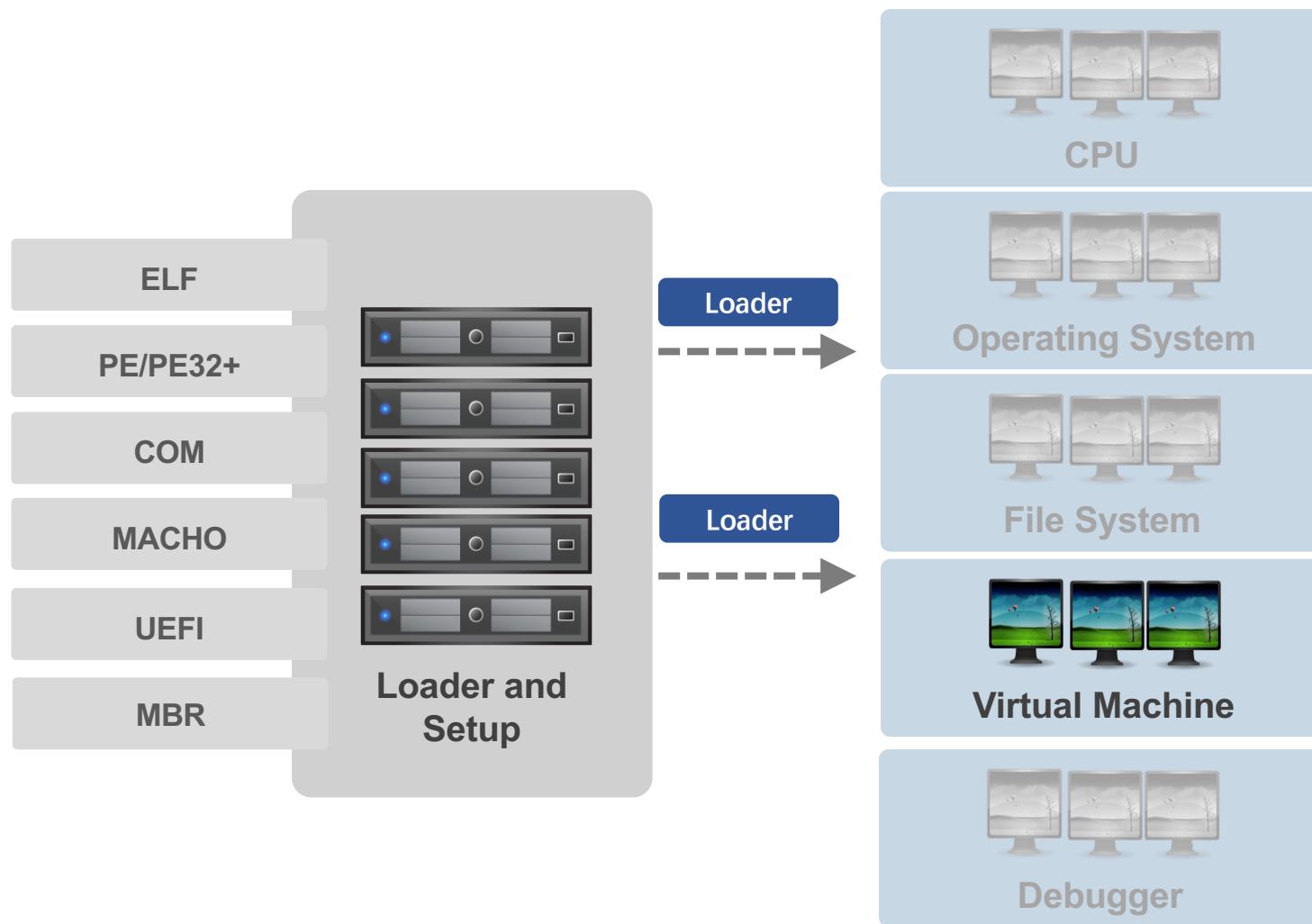
- Access to memory
 - `ql.mem.read()`
 - `ql.mem.write()`
- Search pattern from memory
 - `ql.mem.search()`
- Stack related operation
 - `ql.stack_pop`
 - `ql.stack_push`
- Syscall replacement
 - `ql.set_syscall()`
 - `ql.set_api()`
- Replace library call with
 - `ql.set_api()`

Qiling Framework: File System



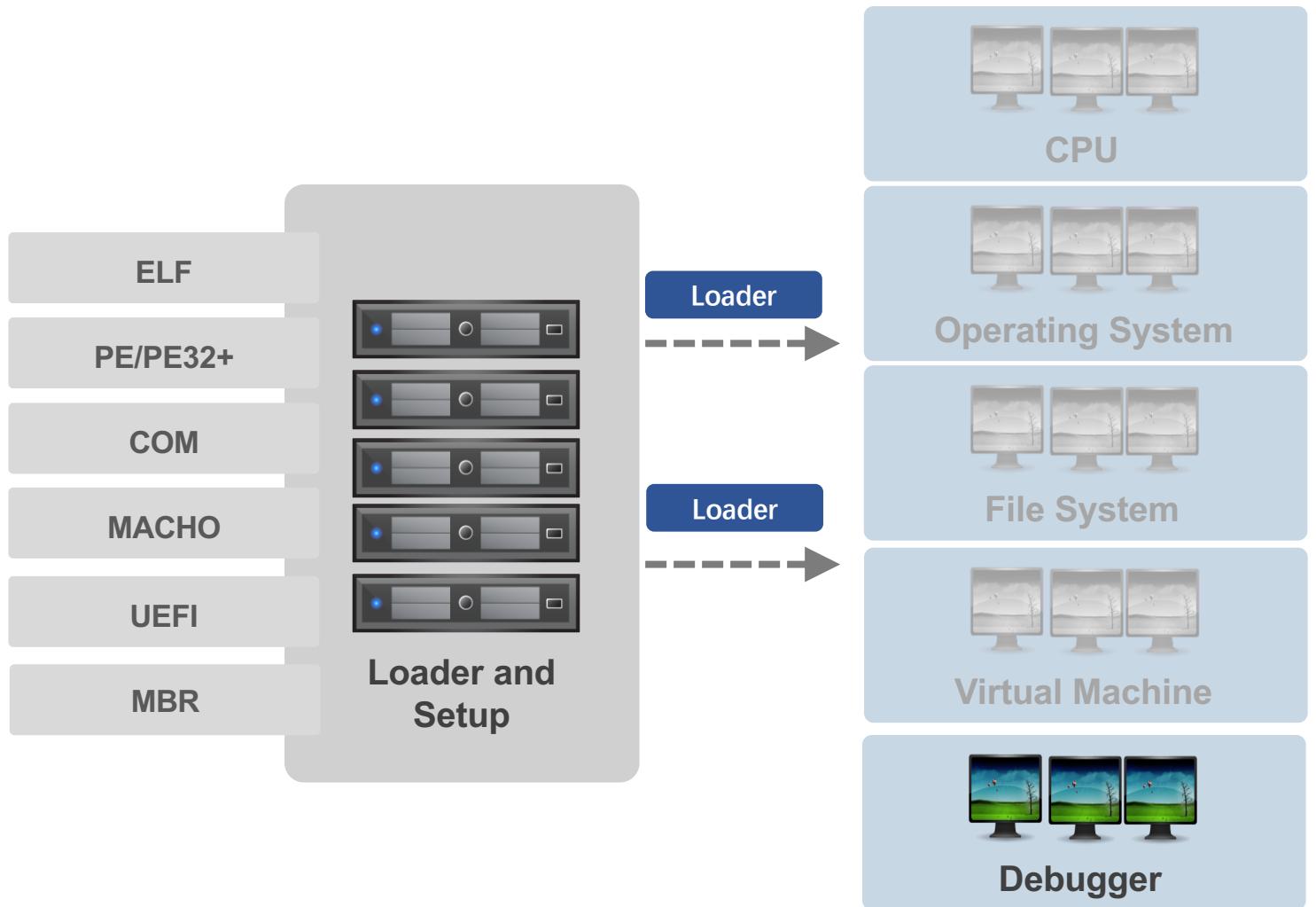
More APIs: <https://docs.qiling.io>

Qiling Framework: Virtual Machine



- Save current state
 - `ql.save()`
- Restore current state
 - `ql.restore()`
- Save/restore memory only
 - `ql.mem.save()`
- Save/restore register only
 - `ql.reg.save()`

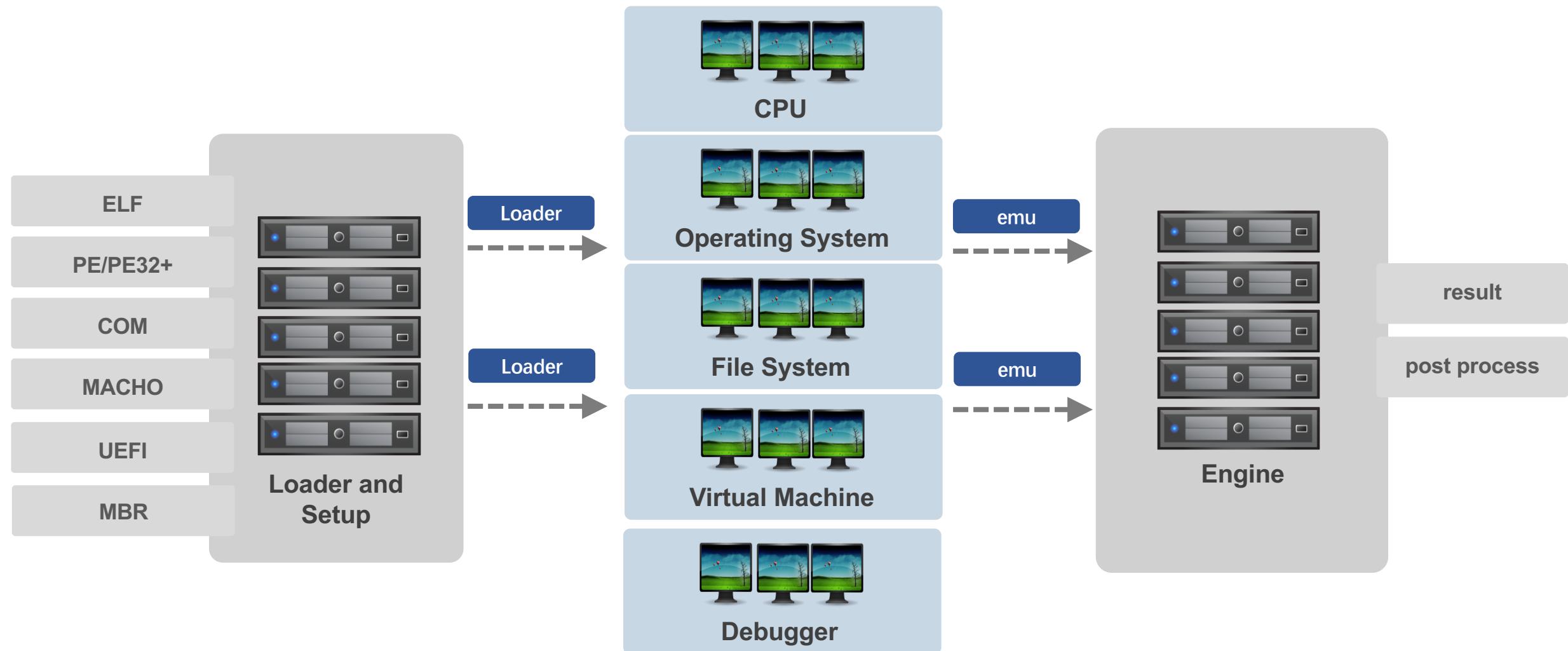
Qiling Framework: Debugger



- Open API for RSP compatible Debugger
- Build In debugger – Qdbg
 - Able to reverse debug

More APIs: <https://docs.qiling.io>

Qiling Framework: In a Nutshell



Base OS can be Windows/Linux/BSD or OSX
And not limited to ARCH

Demo

Demo Setup

➤ **ONLY If you wish to try yourself**

➤ Required OS

- Ubuntu 18.04 / 20.04
- WSL2

➤ Install Qiling Framework

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt install python3-pip git cmake build-essential libtool-bin python3-dev automake flex bison libglib2.0-dev libpixman-1-dev clang python3-setuptools llvm
- git clone <https://github.com/qilingframework/qiling.git>
- pip3 install --user <https://github.com/qilingframework/qiling/archive/dev.zip>

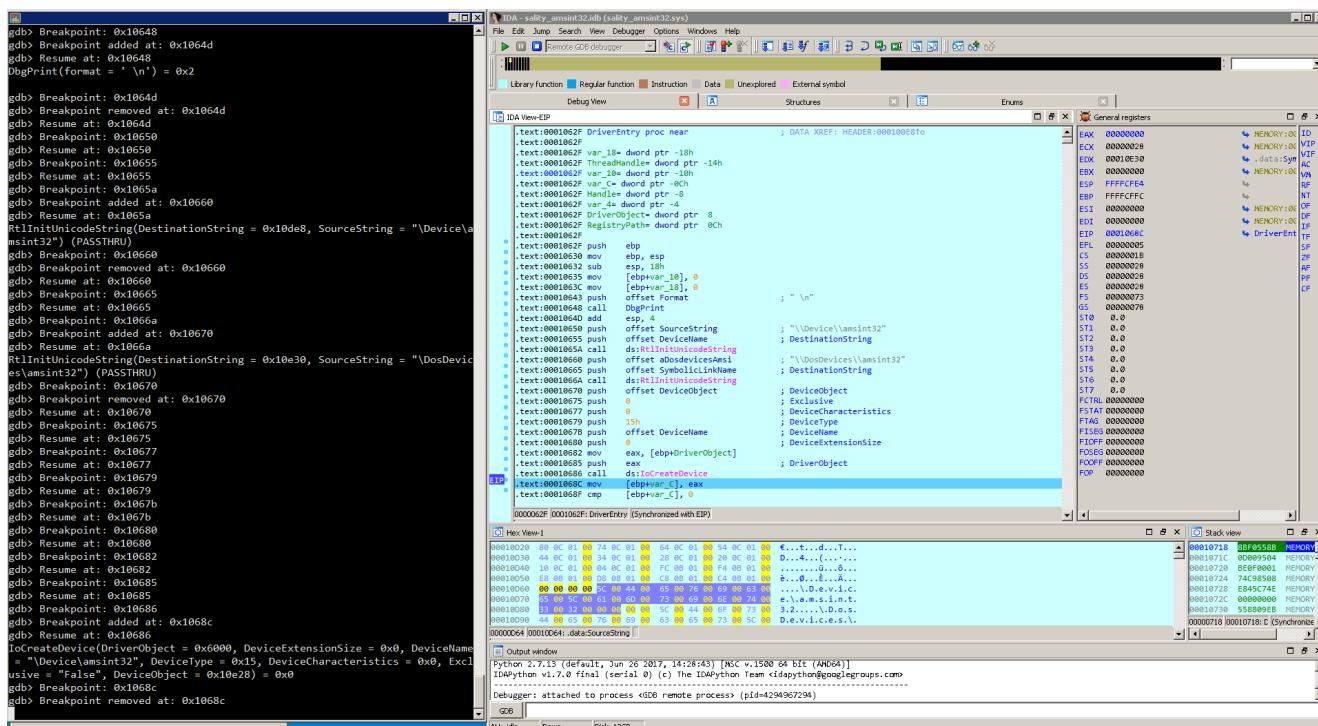
➤ Install AFL++

- git clone <https://github.com/AFLplusplus/AFLplusplus.git>
- cd AFLplusplus
- make
- cd unicorn_mode
- ./build_unicorn_support.sh

Microsoft ❤️ Linux

Malware & Rootkit Analysis

- Support Both Win32/64
- Support PE and System Driver
- Anti-Anti Debug
- Script able
- Cross platform support

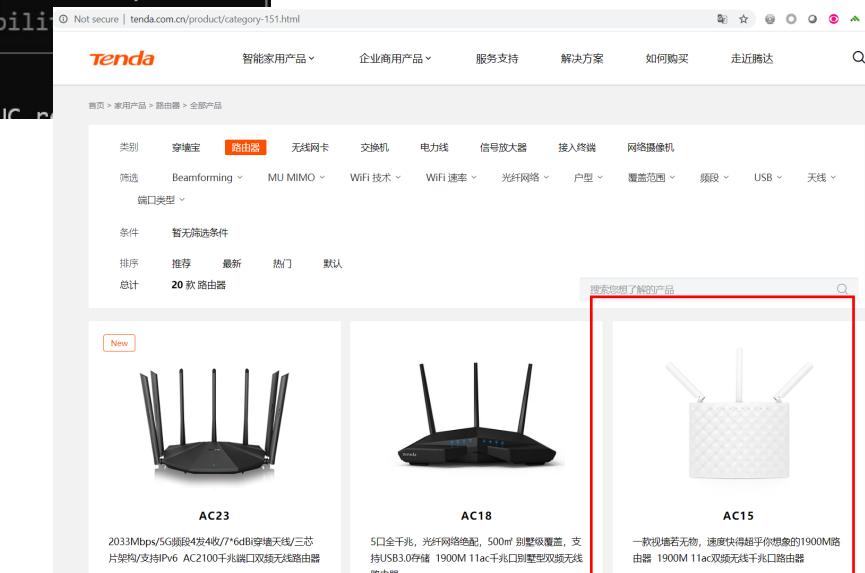


```
(22:43:04):xwings@bespin:<~/projects/qiling>
(15)$ python3 qltool run -f examples/rootfs/x86_windows/bin/al-khaser.bin --rootfs j
xamples/rootfs/x86_windows
[+] Loading examples/rootfs/x86_windows/bin/al-khaser.bin to 0x400000
[+] PE entry point at 0x403d6a
[+] Initiate stack address at 0xffffdd000
[+] TEB addr is 0x6000
[+] PEB addr is 0x6044
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/kernel32.dll to 0x10000000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/kernel32.dll
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/user32.dll to 0x100d4000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/user32.dll
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/advapi32.dll to 0x1019d000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/advapi32.dll
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/ole32.dll to 0x102c0000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/ole32.dll
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/oleaut32.dll to 0x1039a000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/oleaut32.dll
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/shlwapi.dll to 0x1042c000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/shlwapi.dll
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/setupapi.dll to 0x10470000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/setupapi.dll
[+] Done with loading examples/rootfs/x86_windows/bin/al-khaser.bin
GetCurrentThreadId() = 0x0
GetCurrentProcessId() = 0x2005
QueryPerformanceCounter(lpPerformanceCount = 0xfffffcfe4) = 0x0
IsProcessorFeaturePresent(ProcessorFeature = 0xa) = 0x1
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/api-ms-win-core-synch-1-2-0.dll to 0x108b9000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/api-ms-win-core-synch-1-2-0.dll
LoadLibraryExW(lpLibFileName = "api-ms-win-core-synch-1-2-0", hFile = 0x0, dwFlags = 0x800) = 0x108b9000
GetProcAddress(hModule = 0x108b9000, lpProcName = "InitializeCriticalSectionEx") = 0x0
InitializeCriticalSectionAndSpinCount(lpCriticalSection = 0x424d64, dwSpinCount = 0x0)
a0 = 0x1
[+] Loading jexamples/rootfs/x86_windows/Windows/SysWow64/api-ms-win-core-fibers-1-1-1.dll to 0x108bc000
[+] Done with loading jexamples/rootfs/x86_windows/Windows/SysWow64/api-ms-win-core-fibers-1-1-1.dll
LoadLibraryExW(lpLibFileName = "api-ms-win-core-fibers-1-1-1", hFile = 0x0, dwFlags = 0x800) = 0x108bc000
GetProcAddress(hModule = 0x108bc000, lpProcName = "FlsAlloc") = 0x0
TlsAlloc() = 0x0
GetProcAddress(hModule = 0x108bc000, lpProcName = "FlsSetValue") = 0x0
TlsSetValue(dwTlsIndex = 0x0, lpTlsValue = 0x424d3c) = 0x1
LoadLibraryExW(lpLibFileName = "api-ms-win-core-synch-1-2-0", hFile = 0x0, dwFlags = 0x800) = 0x108b9000
GetProcAddress(hModule = 0x108b9000, lpProcName = "InitializeCriticalSectionEx") = 0x0
InitializeCriticalSectionAndSpinCount(lpCriticalSection = 0x425380, dwSpinCount = 0x0)
a0 = 0x1
InitializeCriticalSectionAndSpinCount(lpCriticalSection = 0x425398, dwSpinCount = 0x0)
a0 = 0x1
InitializeCriticalSectionAndSpinCount(lpCriticalSection = 0x4253b0, dwSpinCount = 0x0)
a0 = 0x1
InitializeCriticalSectionAndSpinCount(lpCriticalSection = 0x4253c8, dwSpinCount = 0x0)
```

Fuzzer

- Required Firmware
 - AC15
- Run Tenda AC15
 - start_tendaac15_httpd.py
 - Test with browser
- Check crash point
 - addressNat_overflow.sh
- How to find and save snapshot
 - saver_tendaac15_httpd.py
- How to build and run fuzzer
 - fuzz_tendaac15_httpd.py

```
american fuzzy lop ++2.65d (python3) [explore] {0}
process timing
    run time : 0 days, 0 hrs, 12 min, 52 sec
    last new path : 0 days, 0 hrs, 0 min, 7 sec
last uniq crash : 0 days, 0 hrs, 0 min, 15 sec
last uniq hang : none seen yet
overall results
    cycles done : 2
    total paths : 36
    uniq crashes : 1
    uniq hangs : 0
cycle progress
    now processing : 21*0 (58.3%)
    paths timed out : 0 (0.00%)
map coverage
    map density : 1.55% / 1.60%
    count coverage : 1.36 bits/tuple
stage progress
    now trying : havoc
    stage execs : 2742/32.8k (8.37%)
    total execs : 122k
    exec speed : 161.7/sec
findings in depth
    favored paths : 4 (11.11%)
    new edges on : 8 (22.22%)
    total crashes : 9 (1 unique)
    total tmouts : 0 (0 unique)
fuzzing strategy yields
    bit flips : 0/3480, 0/3468, 0/3444
    byte flips : 0/435, 0/401, 0/385
    arithmetics : 2/23.0k, 0/4022, 0/1454
    known ints : 1/2313, 0/10.5k, 0/16.6k
    dictionary : 0/0, 0/0, 0/0
    havoc/rad : 17/48.6k, 1/1312, 0/0
    py/custom : 0/0, 0/0
    trim : 0.00%/154, 65.57%
path geometry
    levels : 4
    pending : 25
    pend fav : 0
    own finds : 35
    imported : n/a
stabi
```



The screenshot shows a product listing for Tenda routers. At the top, there's a navigation bar with links like '首页', '家用产品', '路由器', '无线网卡', etc. Below it is a search bar. The main content area displays a grid of routers. The 'AC15' model is visible on the right side of the grid, with its image highlighted by a red rectangular box. To the left of the AC15, there are other routers like the 'AC23' and 'AC18'. Each router has a small image, a model name, and a brief description below it.

MBR Analysis

- Sample:
 - Flare-On 5 (2018) Challenge 8 - doogie
 - MBR file
 - Quick look by qltool.
 - `python3 qltool run -f examples/rootfs/8086/doogie/doogie.bin --rootfs examples/rootfs/8086/ --console False`
 - Try some inputs, but only get gibberish.
 - Tips: Feburary 06, 1990.

```
python3 /Users/mio/qiling
```

February 06, 1990... Despite being a 16-year-old reverse engineering genius, I seem to have forgotten the password to my PC. Can you help me???

Password:

```
~/q/e/r/8/doogie (doogie|...) $ file doogie.bin  
doogie.bin: DOS/MBR boot sector; partition 1 : ID=0x7, active,  
start-CHS (0x0,32,33), end-CHS (0x3ff,254,63), startsector  
2048, 41938944 sectors  
~/q/e/r/8/doogie (doogie|...) $ █
```

```
fish /Users/mio/qiling
Y ff 0A }0~Vdr\ c0^?mK sJ cE a@ tX aU ukL iV gwS xm jD ^?? 1Z~Gtf3 ^OT nH hD i0
l0 ^FA ↵
~/qiling (doogie_fix_crlf...) $
```

IDA Plugin

Setup
Reload User Scripts

Execute Till
Execute Selection
Continue
Set PC
Step F9
Edit Register

Restart

View Register
View Stack
View Memory

Save Snapshot
Load Snapshot

Auto Analysis For Deflat
Mark as Real Block
Mark as Fake Block
Mark as Return Block
Deflat

Remove Junk Code by Patterns
Nop Items without Color

IDA View-A Hex View-1 Structures Enums Imports Exports

Function name Seg

_init_proc .ini
sub_8048330 .plt
__gmon_start_ .plt
__libc_start_main .plt
_write .plt
__isoc99_scanf .plt
start .text
sub_8048380 .text
sub_8048410 .text
sub_8048434 .text
sub_8048451 .text
sub_80484F7 .text
main .text
init .text
fini .text
sub_80485F2 .text
sub_8048600 .text
_term_proc .text
__libc_start_main .text
write .text
__isoc99_scanf .text
__gmon_start_ .text

; Attributes: bp-based frame fuzzy-sp
; int __cdecl main(int, char **, char **)
main proc near
; _unwind {
push ebp
mov esp, ebp
and esp, 0FFFFFFF0h
sub esp, 10h
mov dword ptr [esp+8], 17h ; n
mov dword ptr [esp+4], offset aReversingKrEas ; "Reversing.Kr Easy ELF\n"
mov dword ptr [esp], 1 ; fd
call _write
call sub_8048434
call sub_8048451
cmp eax, 1
jnz short loc_804855B

call sub_80484F7
mov eax, 0
jmp short locret_804857C

loc_804855B:
; n
mov dword ptr [esp+8], 6
mov dword ptr [esp+4], offset aWrong ; "Wrong\n"
mov dword ptr [esp], 1 ; fd
call _write
mov eax, 0

locret_804857C:
leave

QL Register View Reg value at { IDA Address:0x8048524 | QL Address:0x8048524 } Stack at 0x7FF3CE60

eax: 0x7769ADD8	ecx: 0x59FE442C	edx: 0x7FF3CEA4	7FF3CDE8: 774BF2F0
ebx: 0x00000000	esp: 0x7FF3CE60	ebp: 0x7FF3CE78	7FF3CDEC: 00000001
esi: 0x77699000	edi: 0x00000000	eip: 0x08048524	7FF3CDF0: 00000000
ef: 0x00000006	cs: 0x0000001B	ss: 0x00000028	7FF3CDF4: 00000001
ds: 0x00000028	es: 0x00000000	fs: 0x00000000	7FF3CDF8: 047E1940
gs: 0x00000063	st0: 0x00000000	st1: 0x00000000	7FF3CDFA: 047D2121
st2: 0x00000000	st3: 0x00000000	st4: 0x00000000	7FF3CE00: 08048034
st5: 0x00000000	st6: 0x00000000	st7: 0x00000000	7FF3CE04: 00000009

7FF3CE08: 00000000	7FF3CE0C: 047E1000	7FF3CE10: 00000000	7FF3CE14: 00000000	7FF3CE18: 00000000	7FF3CE1C: 08048380
--------------------	--------------------	--------------------	--------------------	--------------------	--------------------

Output window

```
esi : 0000000076990000 [INFO][custom_script:13] edi : 0000000000000000 eip : 0000000008048521
[INFO][custom_script:13] cr0 : 0000000000000000 cr2 : 0000000000000000
[INFO][custom_script:13] cr1 : 0000000000000000 cr4 : 0000000000000000 cr5 : 0000000000000000
[INFO][custom_script:13] cr3 : 0000000000000000 cr6 : 0000000000000000 cr7 : 0000000000000000 cr8 : 0000000000000000
[INFO][custom_script:13] cr9 : 0000000000000000 cr10: 0000000000000000 cr11: 0000000000000000 cr12: 0000000000000000 cr13: 0000000000000000 cr14: 0000000000000000
[INFO][custom_script:13] cr1 : 0000000000000000 cr10: 0000000000000000 cr11: 0000000000000000 cr12: 0000000000000000 cr13: 0000000000000000 cr14: 0000000000000000
[INFO][custom_script:13] st0 : 0000000000000000 st1 : 0000000000000000 st2 : 0000000000000000 st3 : 0000000000000000 st4 : 0000000000000000 st5 : 0000000000000000 st6 : 0000000000000000 st7 : 0000000000000000
[INFO][custom_script:13] cs : 000000000000001b ss : 0000000000000028
[INFO][custom_script:13] ds : 0000000000000028 fs : 0000000000000028
[INFO][custom_script:13] es : 0000000000000028 gs : 0000000000000063
```

Python

Execution path drawing

```
.text:0804851B
.text:0804851B
.text:0804851B ; Attributes: bp-based frame fuzzy-sp
.text:0804851B
.text:0804851B ; int __cdecl main(int, char **, char **)
.text:0804851B main proc near
.text:0804851B ; _ unwind {
.text:0804851B push    ebp
.text:0804851C mov     ebp, esp
.text:0804851E and    esp, 0FFFFFFF0h
.text:08048521 sub    esp, 10h
.text:08048524 mov    dword ptr [esp+8], 17h ; n
.text:0804852C mov    dword ptr [esp+4], offset aReversingKrEas ; "Reversing.Kr Easy ELF\n\n"
.text:08048534 mov    dword ptr [esp], 1 ; fd
.text:0804853B call   _write
.text:08048540 call   sub_8048434
.text:08048545 call   sub_8048451
.text:0804854A cmp    eax, 1
.text:0804854D jnz    short loc_804855B
```

```
.text:0804854F call   sub_80484F7
.text:08048554 mov    eax, 0
.text:08048559 jmp    short locret_804857C
```

```
.text:0804855B loc_804855B:          ; n
.text:0804855B mov    dword ptr [esp+8], 6
.text:08048563 mov    dword ptr [esp+4], offset aWrong ; "Wrong\n"
.text:0804856B mov    dword ptr [esp], 1 ; fd
.text:08048572 call   _write
.text:08048577 mov    eax, 0
```

```
.text:0804857C locret_804857C:
.text:0804857C leave
.text:0804857D retn
.text:0804857D ; } // starts at 804851B
.text:0804857D main endp
.text:0804857D
```

Future

- Sync with Qemu5
 - More architectures, instructions
- Android Java bytecode layer instrumentation
- IOS emulation support.
- More robust Windows emulation.
 - Introduce wine&&cygwin
- Smart Contract emulation.
- Single-chip microcomputer emulation.

Everything Else

>About Qiling Framework

- <https://qiling.io>
- <https://github.com/qilingframework/qiling>
- <https://docs.qiling.io>
- @qiling_io



Star us

qilingframework / qiling

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 2 branches 10 tags Go to file Add file Code

xwings Merge pull request #532 from qilingframework/dev ... ✓ 7f27ec3 on Sep 30 3,445 commits

.github	adding gitee sync actions	2 months ago
docs	clean up docs and plan for filter	6 months ago
examples	refine tcp and udp sockets	2 months ago
qiling	getting ready for 1.1.3	last month
tests	refine tcp and udp sockets	2 months ago
.gitignore	clean up 8086 folder	2 months ago
.travis.yml	Fixing travis docker build error	3 months ago
AUTHORS.TXT	core.py: move exit_code to os	6 months ago
COPYING	import	15 months ago
CREDITS.TXT	fixed some typo errors and updated donation details	2 months ago
ChangeLog	update changelog	last month

About

Qiling Advanced Binary Emulation Framework

qiling.io

binary emulator framework
unicorn-emulator malware analysis
qiling reverse-engineering
cross-architecture uefi unicorn-engine

Readme

GPL-2.0 License

Releases 10

Version 1.1.3 Latest on Sep 30
+ 9 releases

Questions