r2 Debugger Improvements

Zi Fan Tan (abcSup)

About me

- Undergraduate student from Malaysia
- Computer Science major
- Enjoy reverse engineering software and games
- Started contributing to radare2 this year

GSoC 2020

- r2 Debugger Improvements
  - Reverse Debugging
  - WinDbg KDNet Support
- Mentors - yossizap, Florian Märkl

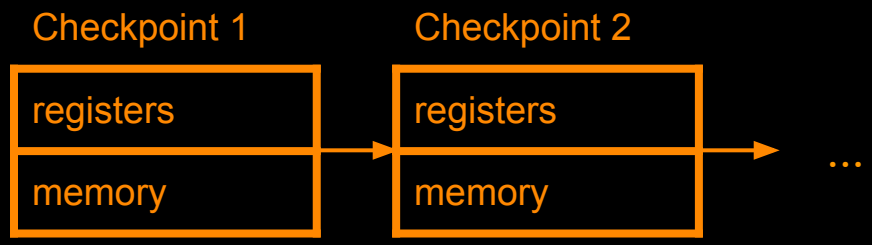
Reverse Debugging

Reverse Debugging

- The ability to “reverse” program executions
- Impossible on hardware
- Use software to keep track of previous states
- View program states and control flow in history

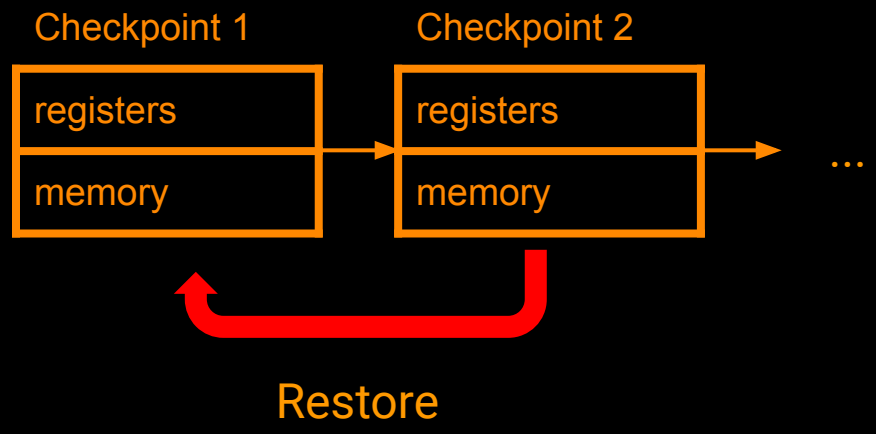
Reverse Debugging - Implementation

Capture and restore registers and memory snapshots aka checkpoints



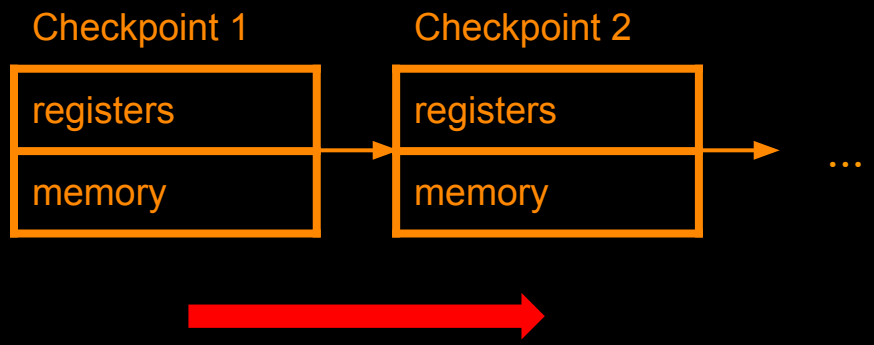
Reverse Debugging - Implementation

Capture and restore registers and memory snapshots aka checkpoints



Reverse Debugging - Implementation

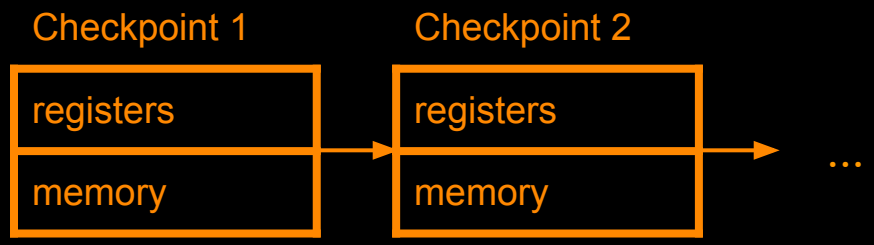
Capture and restore registers and memory snapshots aka checkpoints



Continue until the previous instruction or a breakpoint

Reverse Debugging - Implementation

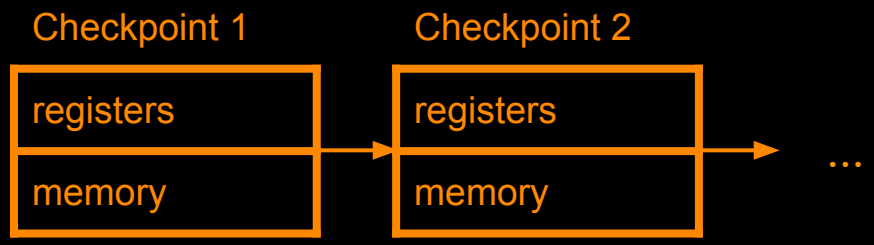
Capture and restore registers and memory snapshots aka checkpoints



- Hard to track the previous PC value after JMP

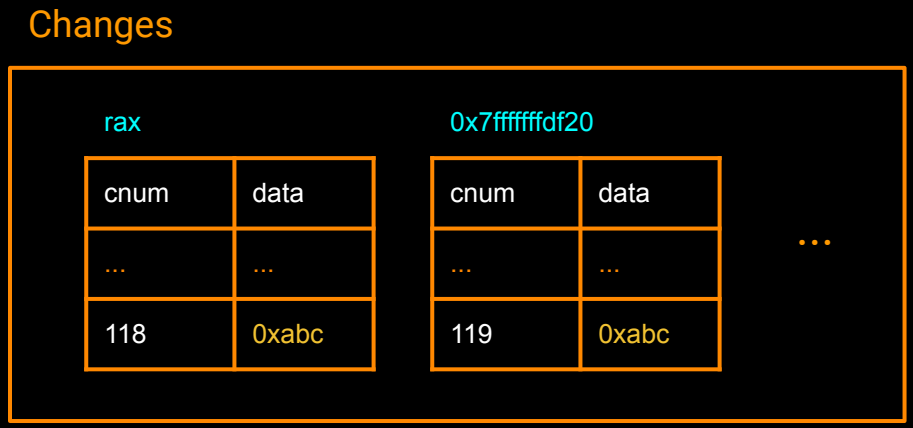
Reverse Debugging - Implementation

Capture and restore registers and memory snapshots aka checkpoints



+

Trace registers and memory changes with index and apply them accordingly



Reverse Debugging - Implementation

During Execution

- Record registers and memory changes at every executed instruction

```
118 mov rax, 0xabc
119 mov dword [ rbp - 4 ], rax
```

rax

cnum	data
...	...
118	0xabc

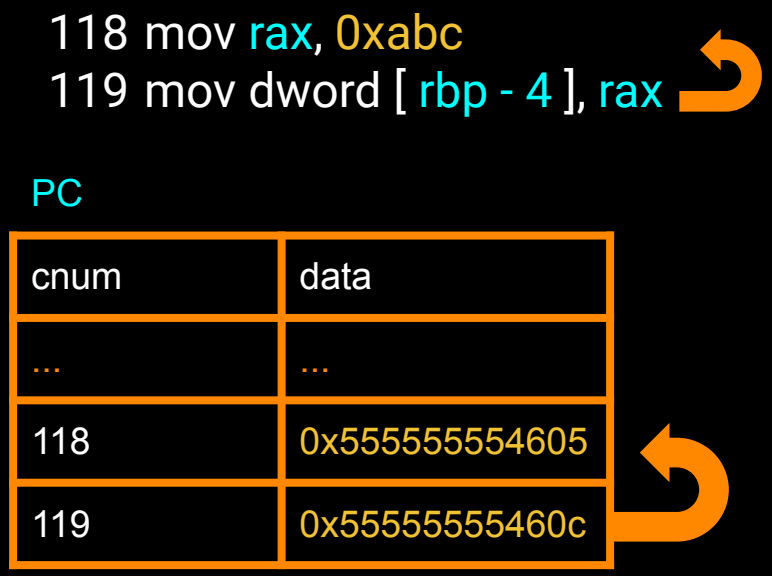
0x7fffffffdf20

cnum	data
...	...
119	0xabc

Reverse Debugging - Implementation

Step/Continue Back

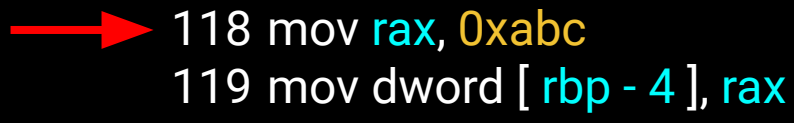
- Look up the previous PC value or the latest PC value that hits a breakpoint.



Reverse Debugging - Implementation

Step/Continue Back

- Apply latest registers and memory changes before that index (e.g. cnum = 118)



rax

cnum	data
...	...
114	0x1234
118	0xabc

0x7fffffffdf20

cnum	data
...	...
115	0x0
119	0xabc

other registers and memory...

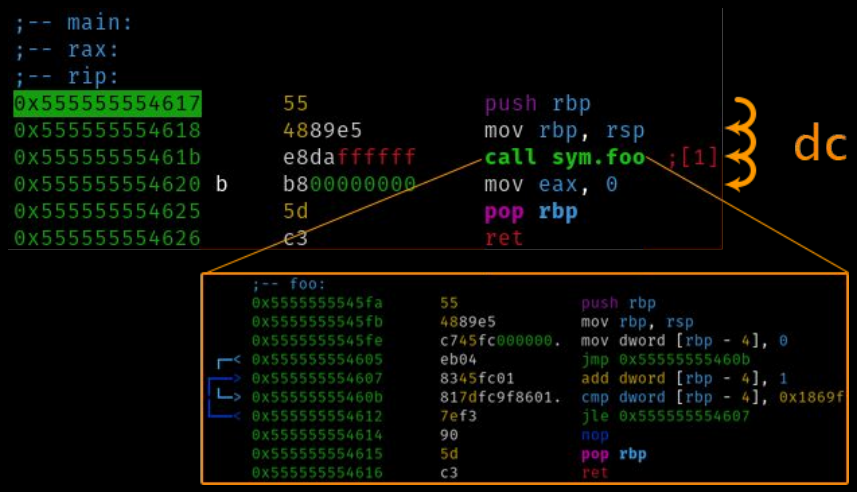
Reverse Debugging - Usage

- Support x86 and ESIL (WIP)
- Commands
  - Start and stop session - dts+, dts-
  - Step and continue back - dsb, dcb
  - Save and load session - dtst, dtsf
- Config
  - dbg.trace_continue - Trace every instruction when continuing

Reverse Debugging - Usage

dbg.trace_continue

-e dbg.trace_continue=true (default)



- Trace every instruction along the way

-e dbg.trace_continue=false

- Take a registers and memory snapshot at the next debugger stop (e.g. breakpoint)

Demo

Reverse Debugging - Future improvements

- Dynamic taint analysis
- Integration with r2 Project
- System calls emulation (read, write, mmap, ...)

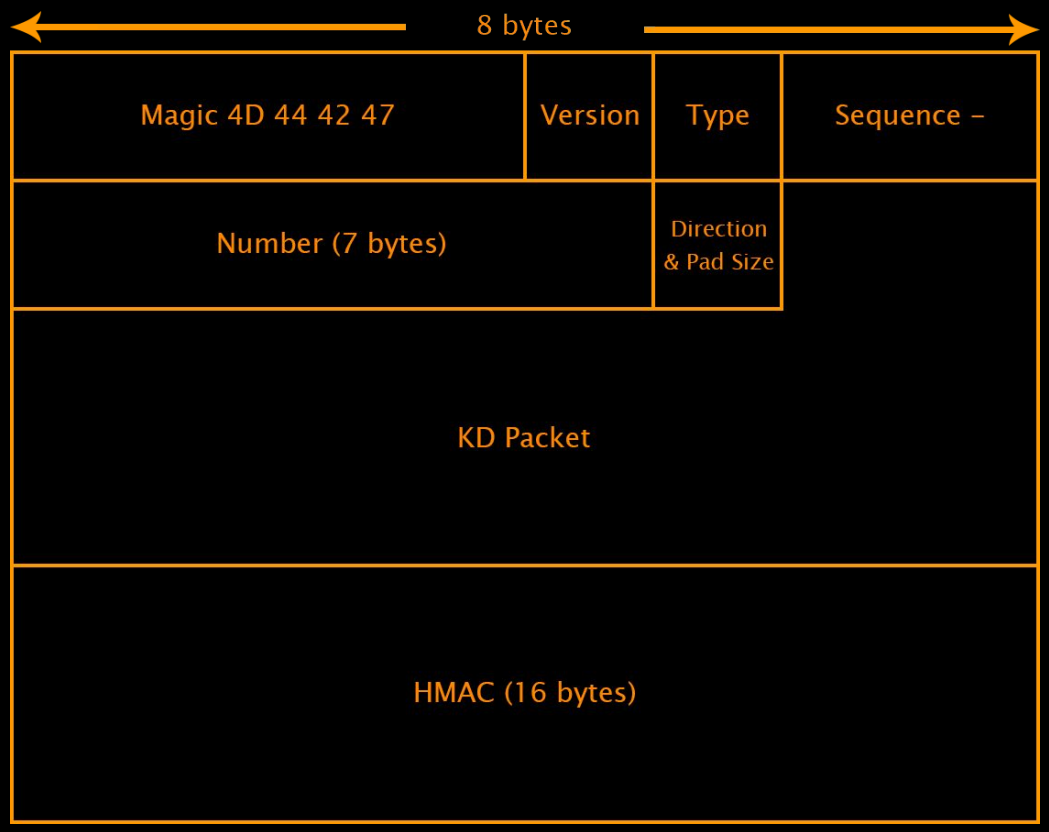
WinDbg KDNet Support

WinDbg KDNet Support

- Debug Windows kernel over network
- Replace serial connections, similarly to VirtualKD
- KD packets wrapped in UDP packets
- Implementation can be found by analyzing DbgEng.dll

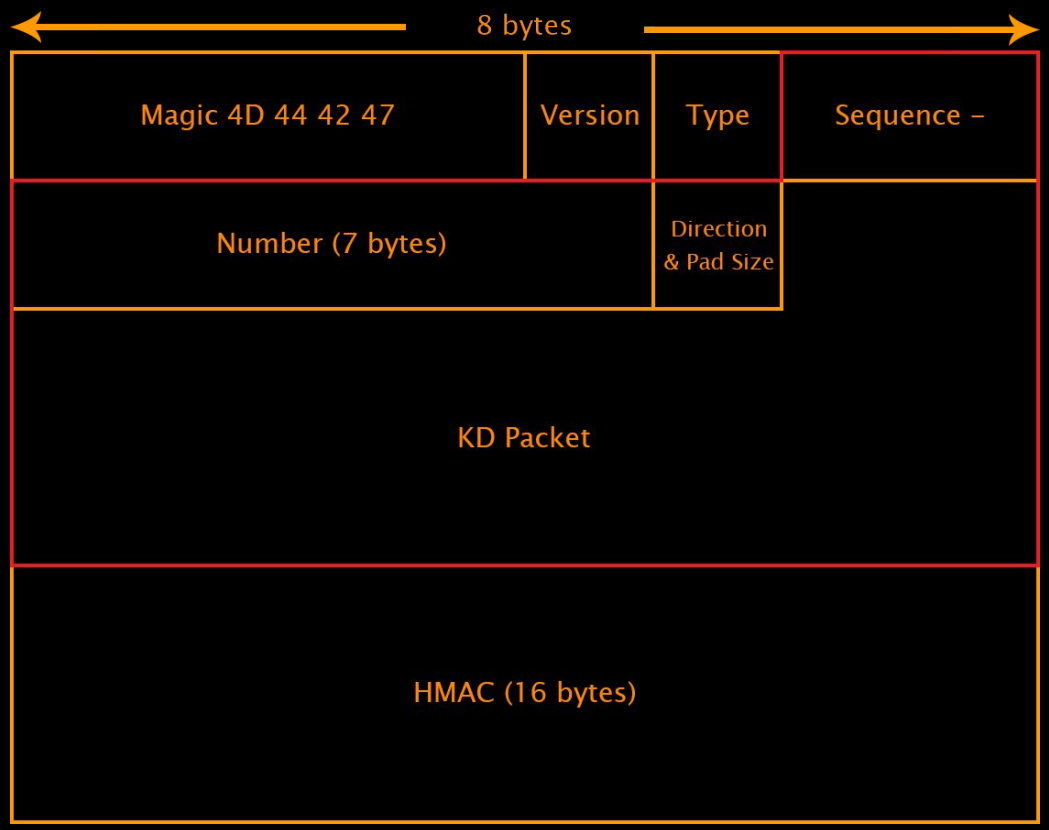
WinDbg KDNet Support - Packet structure

- KDNet Header
  - Magic, Version, Type
- KDNet Data
  - Sequence Number
  - Direction (first bit)
  - Pad size (last 7 bits)
  - KD Packet
- HMAC



WinDbg KDNet Support - Packet structure

- KD Packet
  - 16-byte aligned
- KDNet Data
  - encrypted with AES256
  - used to generate SHA256HMAC (first 16 bytes are used)



WinDbg KDNet Support - Usage

- Easier to setup compared to serial ports on VM

Setting up on Windows

C:\ bcdedit /debug on

C:\ bcdedit /dbgsettings net <hostip>:w.x.y.z port:<hostport>

Connecting to KDNet on r2

$ r2 -a x86 -b 64 -d winkd://<targetip>:<hostport>:w.x.y.z

DbgEng - WinDbg Backend

Credits to @GustavoLCR

DbgEng - WinDbg Backend

- Provide API to execute the same functions as WinDbg commands
- New windbg:// IO and Debug handler using DbgEng.dll*
- Debug programs, crash dump files (.dmp), remote target and KD
- Same command-line parameters when using WinDbg.exe

```
C:\ radare2 -d “windbg://notepad.exe”
C:\ radare2 -d “windbg://-k net:port=<hostport>,key=<w.x.y.z>”
C:\ radare2 -d “windbg://-z file.dmp”
```

* Previous KD interface has been renamed to winkd://

Credits to @GustavoLCR

DbgEng - WinDbg Backend

- Direct access to WinDbg backend shell (=!)

```
[0x7ffa3e8ece30]> =!k
Child-SP          RetAddr           Call Site
0000001f`564bfc78 00000000`00000000 0x00007ffa`3e8ece30
[0x7ffa3e8ece30]> =!lm
start             end               module name
00007ff6`36f50000 00007ff6`36f82000   notepad    (deferred)
```

Credits to @GustavoLCR

Thank you!

Q&A