Sk3wlDbg:

Emulating all (well many) of the things with Ida

Chris Eagle
Sk3wl 0f r00t
Disclaimer

- Everything I say today is my own opinion and not necessarily the opinion of my employer.
Who am I?

- Senior lecturer of computer science
- Computer security researcher
- Reverse engineer
- Inveterate Capture the Flag player
- Performer of stupid IDA tricks
Introduction

- CPU emulators are useful in a variety of cases
  - System design before hardware is available
  - Running code from obsolete platforms
  - Studying code without need to stand up full hardware system

- Some emulators go well beyond CPU to emulate full system including hardware
Goals

– Make lightweight CPU emulator available in a static reverse engineering context

– Temporarily step away from reading a disassembly to confirm behavior

– Incorporate results of a computation back into a static analysis
End result - Sk3wlDbg

- Lightweight emulator integrated into a disassembler
  - Disassembler - IDA Pro
  - Emulator - Unicorn Engine
IDA Pro

- Commercial disassembler
- Supports many processor families
- Integrated debugger supports x86 and ARM targets
- Decompiler
Unicorn Engine
- Announced at BlackHAT USA 2015
- Same people that did Capstone
- Emulator framework based on QEMU
- Supports x86, x86-64, ARM, ARM64, Sparc, MIPS, M68k
- Related projects
Some other emulators

- **Bochs**
  - “Bochs is a highly portable open source IA-32 (x86) PC emulator written in C++”

- **QEMU**
  - “QEMU is a generic and open source machine emulator and virtualizer.”
  - [http://www.qemu.org](http://www.qemu.org)
Emulators and IDA Pro

- 2003 ida-x86emu
  - For deobfuscating x86 binaries

- 2009 Hex-Rays adds Bochs “debugger” module

- 2014 msp430 for use with microcorruption
  - https://microcorruption.com

- 2016 Unicorn integration
  - Because why not
Rationale

- Looked at QEMU and Bochs briefly when writing ida-x86emu
  - Much too heavy weight for what I wanted
  - Too lazy to dig into the code to learn them and strip down
- The Unicorn people did all the heavy lifting
- Brings more architecture to the table
Implementation – two choices

- Emulate over the IDA database itself using the database as the backing memory
  - ida-x86emu does this
  - Forces changes on the database – NO UNDO
- Leverage the IDA plugin architecture to build a debugger module
  - IDA’s Bochs debugger module does this
Result
– Many unhappy dev hours, unhappy wife
– Mostly undocumented IDA plugin interface
  VS
– Mostly untested emulator framework
– BUT...
It’s Alive!

- Sub-classed IDA debugger_t for all supported Unicorn CPU types
- Simple ELF and PE loaders map file into Unicorn
- Fallback loader just copies IDA sections into Unicorn
- Integration issues
  - IDA remains a 32-bit executable
  - Can only interface w/ 32-bit libraries
  - Unicorn doesn’t have great support for 32-bit builds
  - Unicorn’s underlying QEMU code depends on glib
    - Complicates use on Windows
Demo

- Probably not a good idea very alpha code

- Bugs could be Unicorn’s or they could be mine
The way forward

- Better user interface when launching emulator
- Extensible hooking for library functions and system calls
- Option to load shared libraries into emulation along with executable loaded in IDA
• Where to get it
  - https://github.com/cseagle/sk3wldbg
Questions ???

- Contact info
  - Email: cseagle @ gmail . com
  - Twitter: @sk3wl