

Intro Video: https://youtu.be/NyL26OQf_dl

2 Classes

Assembly
Reverse Engineering

Price:

\$150 for each class, or \$250 for both classes, paypal

<https://paypal.me/BDSPodcast/150usd>

(use "Notes" to denote which class)

If you just want the videos for both classes, it's also \$150, but specify you just want the videos in the "Notes" section when you pay

<https://www.paypal.me/BDSPodcast/250usd>

(if you want both classes)

****Links have been FIXED****

Email bds.podcast@gmail.com if there is an issue with paying

One student will be gifted with a license of Binary Ninja <https://binary.ninja/> (A \$150 value)

Time:

Mondays (8:30pm Central)

Assembly Class

Starts 2 April 2018

Ends 23 April 2018

Reverse Engineering Class

Starts 30 April 2018

Ends 21 May 2018

Topics we discussed:

developing self modifying code
encrypting code and executing it
parsing PE files
c++ code and data structures

Homework will occur as needed

Raspberry pi using QEMU

<https://www.amazon.com/Raspberry-Pi-RASPBERRYPI3-MODB-1GB-Model-Motherboard/dp/B01CD5VC92>

Linux →

<https://azeria-labs.com/emulate-raspberry-pi-with-qemu/>

Windows → <https://qemu.weilnetz.de/w64/>

<https://godbolt.org/>

<http://www.scs.stanford.edu/05au-cs240c/lab/pcasm-book.pdf>

<https://www.amazon.com/Raspberry-Pi-RASPBERRYPI3-MODB-1GB-Model-Motherboard/dp/B01CD5VC92>

<http://blog.canonical.com/2012/05/25/ubuntu-arm-server-ami-on-amazon-ec2/>

<https://hub.docker.com/r/armhf/alpine/>

<https://binary.ninja/demo/>

Requirements

- Linux server setup (brian?)
 - Debian 9.3 VM
 - Gdb
 - GCC (32bit libs)
 - objdump
 - Readelf (should be default)
 - Nasm
 - Make
 - Build-essentials
 - Binary ninja
 -
- ssh client
 - Putty
 - OS X Terminal
 - Linux Terminal

Assembly Training

Session 1

- Assembly Basics
 - Registers
 - Stack
 - Mnemonics
 - Operations
 - Add, Sub, Mul, Div, mod
 - AND, OR, XOR, NOT
 - 2's Complement
 - Define global variables
- RE some basic C programs

Session 2

- Memory layout
 - Code
 - Data
 - Stack
 - Heap
- Compare and test
 - Unconditional jumps
 - Conditional jumps
- If and if/else basic blocks
- While loops
- Shifting

Session 3

- Using GDB
 - Start debugging session
 - Breakpoints
 - Examine
 - Memory
 - Stack
 - Registers
 - Dumped core
 - Backtrace
 - Allow dump
- Stack
 - Push
 - Pop
 - Add vs pop

- Calling/writing functions
 - Prolog
 - Epilogue
 - Save registers
 - Arguments
 - Local Variables
- RE some basic C functions

Session 4

- Arm assembly
 - Fixed width
 - Low power usage
- New registers
- Loading registers
- Branching, Link Register
- Barrel shifter
- Conditional Execution

- AES Homework

Reverse Engineering Training

Session 1

- Reverse Engineering Process
 - Disassembly
 - Dead Listing
 - Recursive Descent
 - Errors
- https://tsyrklevich.net/tbb_payload.txt
- File Formats
 - PE
 - ELF
- Parsing example
 - Parsing an ELF in linux
- C structures and usage
- Fread/fwrite/fseek
- Homework
 - Write a parser for PE file

Session 2

- Common design patterns
 - if/else
 - Loops
 - Functions
- API Calls
 - Thunk Tables
 - GOT
- Types of calling conventions
 - Std call
 - C Decl
 - Fast call

Session 3

- C++ Code
 - Objects vs Struct
 - Vtables
 - This
 - Inheritance variables
- Compiler differences
- AES Sample code

Session 4

- Mixing Code and Data
- Windows
 - VirtualAlloc
 - VirtualProtect
 - Process Execution Block
 - Iterating locating functions
- Locate functions and call them dynamically
- Create a DLL function
 - DLL hooking
- Speculative Execution
 - Why do we need it
 - What went wrong