SYSCALL7

Software Reverse Engineering and Security Analysis

Description: This 3-day, hands-on course offers an examination of the fundamental techniques of software reverse engineering used by attackers and security researchers alike. The lectures and exercises provide a practical foundation for all areas of software security research, including forensics, penetration testing, vulnerability research, exploit development, and malware analysis. The course covers general concepts and techniques that apply to the full spectrum of computing targets, from bare-metal and RTOS- powered embedded systems to mobile and desktop systems running Android, Windows, Linux and MacOS.

Attendees will gain hands-on experience with state-of-the-art techniques and tools of the hacking trade through a series of lab exercises that reinforce the content of the lecture material. The exercises will incorporate several popular instruction sets, including x86, MIPS, PowerPC, and ARM. Students will learn how to use Binary Ninja to reverse binaries, and each student will take home their own licensed copy of Binary Ninja! Each student will also take home a hardware device that will be used in several student exercises throughout the course. The reversing skills learned in the course are transferable to IDA Pro and other disassemblers.

Topics covered during this course include:

- Static Analysis
- Dynamic Analysis
- Firmware Specific Techniques
- Vulnerabilities and Exploits
- Final Project

Prerequisites: Attendees should be experienced with the C and/or C++ programming languages and should have some prior exposure to assembly language and Python.

Requirements: Each student will need to provide their own laptop and must have administrator access. The laptop must have WiFi support and at least one USB Type A port (i.e., not Type C). Some of the lab exercises will be performed in a VM, and the students should install VMware on their system before arriving for the first day of class. (The free trial version of VMware is sufficient if you do not already own a license.)

Course Syllabus

The following summary covers the major course topics and may be modified at the instructor's discretion based on the needs and pace of the course.

Introduction

- Course Overview
- What is reverse engineering?
- Motivations and ethical considerations
- Approaches to reverse engineering

Static Analysis

- Computer architecture refresher
  - Review of instruction sets (x86, ARM, PowerPC, MIPS)
  - Addressing modes
  - Control flow
  - The stack
  - The heap
  - Object file section types

  - Working with executables and object files
- Role of Application Binary Interfaces
  - Function prologues and epilogues
  - Calling conventions
  - Variadic arguments
  - Position-independent code
- How to use disassemblers
  - Overview of disassembler tools
  - Introduction and basic usage of ODA
  - Introduction and basic usage of Binary Ninja
- Analysis of data structures
  - Refresher on C structs
  - Identifying and modelling structs in Binary Ninja
- Reversing C++
  - C++ classes
  - The this pointer
  - Virtual function tables
  - Inheritance
  - Name mangling
- How to use decompilers
  - Introduction and basic usage
- Scripting tools
  - Scripting with the Binary Ninja API
  - Writing Binary Templates in the 010 Editor

Dynamic Analysis

- Getting the most from Debuggers
  - Software and hardware breakpoints
  - Conditional breakpoints
- Introspection techniques
  - Patching
  - Hooking

    - Instrumentation and probing
- Analysis of network communications
    - Scanning with nmap
    - Sniffing with Wireshark
    - Dissecting protocols with Wireshark plugins
    - Scripting network protocols with Python/Scapy
- Analysis of USB communications
    - Overview of USB protocol
    - Common tools for monitoring USB
- Windows tools and techniques
    - System monitoring tools
    - Hooking the Windows API and custom DLLs
    - Debugging on Windows (x64dbg, ollydbg, WinDbg)
- Anti-reversing techniques

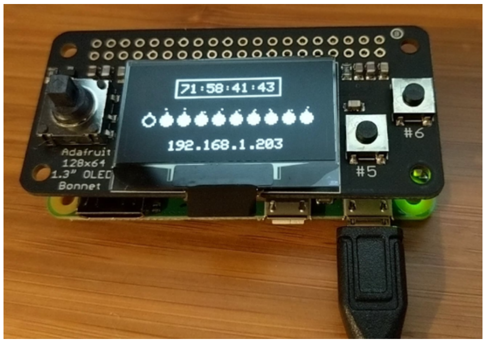
Firmware-Specific Techniques

- Extracting Firmware Images
- Image deconstruction
- Recovering symbol tables
- Suppressing watchdog timers
- Working with Embedded Linux

Vulnerabilities and Exploits

- Common vulnerability categories
- Introduction to fuzzing
- Introduction to exploits
- Introduction to Return Oriented Programming

Lab Exercises



Throughout the course students will be reversing the firmware for a "bomb" device, which is implemented with Raspberry Pi-based hardware. The lab exercises throughout the course will guide and challenge students to defuse the various stages of the "bomb", which get progressively more difficult as the class goes on. Students get to keep the hardware device and will be able to complete any remaining lab exercises on their own if they are not able to complete them in class.

Instructor Bio

Anthony DeRosa is an infosec expert specializing in the vulnerability research of networking and embedded computing devices. He has 15 years of experience in reverse engineering using static reverse engineering tools, and he is experienced with the runtime debugging environments of gdb, WinDbg, and JTAG-based debuggers like Lauterbach, Green Hills, American Arium, and BDI. Through fuzzing and a combination of static and dynamic reverse engineering, he has uncovered serious vulnerabilities and recommended techniques for securing these systems against the weaknesses uncovered. Mr. DeRosa is the founder of ODA (onlinedisassembler.com), a collaborative reverse engineering platform. He has a Master of Science in Electrical and Computer Engineering from Johns Hopkins University. He has consulted for several 3-letter agencies within the US intelligence community over his 15 year career.