VSS 501: Hardware Hacking Bootcamp

Description

This course is designed to provide students with an in-depth understanding of embedded protocol reverse engineering, instrumentation, and analysis. The course covers a wide range of protocols, including UART, I2C, SPI, JTAG, and SWD. Throughout the course, students will perform attacks on multiple real-world targets, including flash extraction, debug interface instrumentation, and firmware analysis.

Objectives

- Understand the principles of embedded protocol reverse engineering and instrumentation
- Learn how to reverse engineer and analyze UART, I2C, SPI, JTAG, and SWD protocols
- Develop the ability to perform attacks on real-world targets, including flash extraction, debug interface instrumentation, and firmware analysis
- Gain practical experience in the use of hardware hacking techniques through hands-on lab exercises

Outline

- Module 1: Fundamentals / Tool Review
  - Objective 1: Review EE fundamentals and measurement tools
  - Objective 2: Hardware hacking overview and history
  - Objective 3: Define a process for analyzing a new embedded device
- Module 2: Universal Asynchronous Receiver Transmitter
  - Objective 1: Understand what UART is and how it works
  - Objective 2: Identify and detect an active UART on a PCB
  - Objective 3: Calculate an unknown baud rate
  - Objective 4: Create and utilize UART analyzers in Pulseview
  - Objective 5: Use the Raspberry PI as a UART interface
  - Objective 6: Discover and interface with a UART on the router
- Module 3: Bootloaders and UBoot
  - Objective 1: Learn what UBoot is and how it is used
  - Objective 2: Review the Linux boot process
  - Objective 3: Understand UBoot environment and console commands
  - Objective 4: Manipulate environment variables to gain access to systems
  - Objective 5: Script UBoot console interactions with Depthcharge and Python
- Module 4: Serial Peripheral Interface
  - Objective 1: Learn how the SPI interface works and what components utilize it

    - Objective 2: Identify SPI transactions at the signal level
    - Objective 3: Identify SPI commands at a protocol level
    - Objective 4: Set up a SPI decoder with a logic analyzer
    - Objective 5: Extract SPI flash with flashrom
    - Objective 6: Interface with a SPI device using Python
- Module 5: Firmware Analysis and Dissection
    - Objective 1: Learn how to perform an initial analysis of a firmware blob
    - Objective 2: Learn to extract segments of interest from firmware images via binwalk and dd
    - Objective 3: Extract components of interest from the router firmware image
    - Objective 4: Patch and modify router firmware image to gain advanced access
    - Objective 5: Learn how to repackage and reflash firmware images
- Module 6: Inter-Integrated Circuit
    - Objective 1: Learn and understand how the I2C protocol works
    - Objective 2: Understand how to approach I2C as a reverse engineer
    - Objective 3: Analyze and review I2C traffic and addressing
    - Objective 4: Practice sniffing I2C traffic for data of interest
    - Objective 5: Extract an I2C-based memory device
    - Objective 6: Reflash and modify an I2C-based memory device
- Module 7: Joint Test Action Group
    - Objective 1: Review and understand the JTAG specification
    - Objective 2: Learn to navigate the JTAG state machine
    - Objective 3: Use the Pi to identify a JTAG interface
    - Objective 4: Interface with a JTAG TAP using urJTAG and OpenOCD
    - Objective 5: Extract memory, set breakpoints and control target execution via JTAG, both manually and via pre-existing tools
    -
- Module 8: Serial Wire Debug
    - Objective 1: Learn what Serial Wire Debug is
    - Objective 2: How to identify Serial Wire Debugging
    - Objective 3: Learn how to interface with SWD via OpenOCD
    - Objective 4: How to identify an unknown ARM SoC via SWD
    - Objective 5: Extract firmware via SWD
    - Objective 6: Modify and upload new firmware via SWD

Outcomes

After taking this course, students will understand how to approach embedded devices and the protocols that comprise them from a black box perspective. Students will have experience extracting multiple types of flash chips via various interfaces and understand how hardware level debuggers work at the protocol level. Upon completion of the course, students will receive the following:

- A certificate of completion
- All slides for the course materials
- Video recorded lectures from the course (if remote)
- An SD card containing the software and tooling used for the Raspberry Pi
- All hardware used throughout the course

Prerequisites

This course is targeted towards security researchers who want to learn more about the process of firmware extraction and embedded systems analysis. Students should be familiar with the Linux command line and be comfortable with a scripting language such as python.

Interfacing with the Raspberry Pi requires an available USB type A port. A virtual machine is provided to automate the configuration of the Raspberry Pi. Students should be able to load and run virtual machines if they are not comfortable installing Pulseview and configuring an ethernet interface on their host machine.

Schedule / Pricing

Duration: 5 Days
Cost Per Student: $3,500