Hardware Hacking Bootcamp: Syllabus and Overview

VoidStar Security LLC

2023-11-01

Hardware Hacking Bootcamp: Syllabus and Overview

VoidStar Security LLC

Course Description

- Course Length: 5 Days / 40 Hours
- Course Availability: Onsite

Have you ever wondered how to go from opening up an embedded device such as your home router or an old cell phone to extracting valuable data? If so, then this course is for you!

This five-day course reviews the fundamentals of hardware reverse engineering and analysis. With a focus on reverse engineering protocols that comprise embedded systems, students learn how interfaces such as UART, SPI, I2C, JTAG, and SWD work while developing and using tools to interface with these protocols. Through interfacing with these protocols, students learn how to extract flash memory, interact with hardware-level debuggers, and modify the boot process of the target systems included in the kit.

This course includes a hardware hacking kit that comes with all of the tools needed to get started with hardware hacking, such as:

- Multimeter
- Logic Analyzer
- Raspberry Pi 4, preloaded with all of the software required for the course
- Breadboard, jumper wires and header pins
- Four different target devices

Course Objectives

After participating in this course, students will have experience with the following:

- Non-invasive hardware analysis (component identification/documentation)
- Tracing and identifying points of interest on PCBs
- Analyzing, instrumenting, and decoding standard embedded protocols such as SPI, UART, I2C, SWD
- Extracting firmware over multiple interfaces
- Unpacking/analyzing binary images
- Reverse engineering and instrumenting hardware debug interfaces
- Modifying, repacking, and re-flashing firmware

Labs include extracting SPI/I2C-based flash chips, discovering and gaining access to consoles using UART, and identifying, enumerating, and actuating hardware-level debuggers such as JTAG and SWD. JTAG and SWD labs include leveraging these interfaces to modify a running process via physical memory access and escalating the privileges of a running process.

All exercises and laboratories utilize open-source tooling on a Raspberry Pi. Students will learn to use the Raspberry Pi to reverse engineer and exploit all the targets in the kit.

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

Course Outline

This course is built on practical exercises. Students will first instrument the target devices manually using Python-based tooling and pre-existing tools. An outline and task list for each module can be found below.

This course includes multiple modules, one for each protocol of interest. For each module, we will perform the following:

- Protocol Overview and Analysis
- Understanding and Reviewing Captured Protocol Traffic
- Protocol Analysis from a Reverse Engineering Perspective
- Tools for Reverse Engineering Specific Protocols
- Practical Attacks and Applications on Provided Targets

After each protocol module, students will attack the included targets to reinforce what was learned in the analysis segment. Using this knowledge, students will perform hardware attacks on the targets included in their kits.

Module Listing

- Module 1: Fundamentals / Tool Review
  - Review tools and equipment needed for hardware hacking
  - **Lab:** Calculating Voltage, Resistance and Continuity

  - Printed Circuit Board Construction and Reverse Engineering
  - **Lab:** Reverse Engineering PCBs in Kit
  - Component Identification and Documentation

- Module 2: Universal Asynchronous Receiver Transmitter
  - UART protocol overview
  - Identify UART transactions at the signal/protocol level
  - How to identify and detect an active UART on a PCB
  - **Lab:** Calculate an unknown baud rate
  - **Lab:** Create and utilize UART analyzers in Pulseview
  - **Lab:** Use the Raspberry PI as a UART interface
  - **Target Exercise**: Discover and interface with a UART on the router

- Module 3: Bootloaders and UBoot
  - UBoot history and overview
  - Linux boot process review and deep dive
  - Understand the U-Boot environment and console commands
  - **Lab(s):** Manipulate environment variables to gain access to systems
  - **Target Exercise:** Interface with target UBoot Console and perform initial analysis
  - **Target Exercise:** Script UBoot console interactions to extract flash data

- Module 4: Serial Peripheral Interface
  - SPI protocol overview
  - Identify SPI transactions at the signal/protocol level
  - **Lab:** Set up an SPI decoder with a logic analyzer
  - **Lab:** Manually extract SPI flash with Python
  - **Lab:** Extract SPI flash with flashrom
  - **Target Exercise:** Extract target SPI flash with flashrom

- Module 5: Firmware Analysis and Dissection
  - Overview of common firmware image structure
  - Approaching a new firmware image as a reverse engineer
  - **Lab:** Extract segments of interest from firmware images via `binwalk` and `dd`
  - **Lab:** Extract components of interest from the router firmware image
  - **Lab:** Patch and modify router firmware image to gain advanced access

- Module 6: Inter-Integrated Circuit
  - I2C protocol overview

  - Identify I2C transactions at the signal/protocol level
  - Understand how to approach I2C as a reverse engineer
  - **Lab:** Analyze and review I2C traffic and addressing
  - **Lab:** Capturing and analyzing I2C traffic for data of interest
  - **Lab:** Extract data from an I2C-based memory device using Python
  - **Target Exercise:** Re-flash and modify an I2C-based memory device

- Module 7: Joint Test Action Group
  - JTAG specification and state machine review
  - JTAG for reverse engineers
  - **Target Exercise:** Enumerate and identify pins on an undocumented JTAG pinout via BYPASS and IDCODE scans
  - **Target Exercise:** Interface with a JTAG TAP using urJTAG and OpenOCD
  - **Target Exercise:** Write a custom OpenOCD config file for the target device
  - **Target Exercise:** Extract memory via JTAG, manually and via pre-existing tools
  - **Target Exercise:** Utilize JTAG to escalate privilege on a Linux-based target
  - **Target Exercise:** Utilize JTAG to list processes on a Linux-based target
- Module 8: Serial Wire Debug
  - SWD specification and protocol overview
  - SWD for reverse engineers
  - **Target Exercise:** Identify a Serial Wire Debug interface
  - **Target Exercise:** Write a custom OpenOCD config file for the target device
  - **Target Exercise:** Interface with SWD via OpenOCD
  - **Target Exercise:** Identify an unknown ARM SoC via SWD
  - **Target Exercise:** Extract firmware via SWD
  - **Target Exercise:** Modify and upload new firmware via SWD

Each module corresponds with a real hardware target included in the kit; targets include:

- MIPS Travel Router
- ARCompact 64GB SSD
- ARM-based USB controller
- Arcade Cabinet
- ARM-based single-board computer

Target Audience / Pre-requisites

This course targets IT professionals and security researchers who want to learn more about how hardware-level debuggers work and how to approach them as a reverse engineers. Students should be familiar with the Linux command line and comfortable with a scripting language like Python. C experience is also helpful but not required. While some hardware experience will be beneficial, it is not necessary for this course.

Pricing

Public and private versions of this course are available. The class size for remote courses is limited due to exercise complexity. For some courses, self-paced online variants are provided. These versions cover the core concepts but do not include bonus exercises and additional background information that arises from classroom discussion.

Course Type	Location	Minimum Number of Students	Cost Per Student (USD)
Public	Onsite	10	$4000
Private	Onsite	5	$3500
Private	Onsite	5+	$3000

If you are interested in taking this course or organizing a private offering, please reach out to contact@voidstarsec.com