Hacking Hardware Debuggers: Syllabus

VoidStar Security LLC

Description

This three-day course teaches students how to identify, instrument, and utilize hardware-level debuggers. Focusing on JTAG - Joint Test Action Group and SWD - Serial Wire Debug, students will first learn to use these interfaces on an exemplar target and then test their abilities on a commercial-off-the-shelf (COTS) target.

Students will learn how to interact with these interfaces at the signal level and develop high-level tools through labs and exercises. After writing tools to instrument the target, students will utilize tools such as UrJTAG and OpenOCD to gain access to the target systems.

Course Objectives

After participating in this course, students will:

- Identify and interface with a JTAG test access port
- Manually instrument and navigate JTAG scan chains
- Develop OpenOCD config files for high-level JTAG access
- Utilize JTAG to extract RAM from a target device and modify its behavior
- Understand how the SWD protocol is implemented
- Use SWD to read and write memory manually
- Extract and modify flash memory via SWD using OpenOCD

Students will utilize open source tooling and develop tools to interface with the targets included in the kit. All exercises and laboratories are performed using open source tooling on a Linux-based SBC. The targets in the kit include a development board (for learning purposes) and then a COTS target for each interface being targeted.

Upon completion of the course, students will receive:

- 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 Linux based SBC
- Access to a Discord channel with past students of VSS courses

Course Structure

This course is primarily made up of 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.

1. Hardware Debugging Overview and History
    a. Purpose of Hardware Debugging
    b. Past and Present Debug Standards
    c. ARM ADI Specification Review

    d. Hardware Kit Setup and Review
2. JTAG Protocol Review
    a. History of JTAG Specification
    b. Practical Applications of JTAG
    c. State Machine Navigation
    d. Register Usage and Access
    e. JTAG and ARM ADI Integration
3. JTAG for Reverse Engineers:
    a. How to Identify JTAG Test Points
    b. Detecting Pinouts via ID CODE
    c. Detecting Pinouts via BYPASS
    d. How to Determine Register Length
    e. Manually Navigating the State Machine
4. JTAG Exercises: Examplar Development Board
    a. Identify JTAG Pinout
    b. Determine Register Length
    c. Manually Extract ID CODE
    d. Manually Read/Write Memory
    e. Develop OpenOCD Configuration
5. JTAG Exercises: Target - Transcend SSD
    a. 3.a - 3.e
    b. Debug SSD Using GDB
    c. Modify SSD Read Operation
    d. Modify Firmware on SSD via JTAG
6. SWD Protocol Review
    a. History of SWD Specification
    b. Practical Applications of SWD
    c. SWD Packet Structure and Format
    d. SWD and ARM ADI
    e. SWD ADI Examples
7. SWD Exercises: Exemplar Development Board
    a. Identify SWD Pinout and Extract ID
    b. Manually Read/Write Memory Regions
    c. Read/Write Memory via OpenOCD
    d. Extract Flash via OpenOCD
    e. Modify Firmware and Reflash via OpenOCD
8. SWD Exercises: Target - Xbox One Controller
    a. Identify SWD Pinout and Extract ID
    b. Manually Read/Write Memory Regions
    c. Read/Write Memory via OpenOCD
    d. Extract Flash via OpenOCD
    e. Modify Firmware and Reflash via OpenOCD

Requirements

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

Interfacing with the Linux-based SBC requires an available USB type A port. A virtual machine is provided to automate the configuration of the Linux-based SBC. Students will be provided with a virtual machine and setup instructions before the course begins.

Pricing and Private Offerings

Public and private versions of this course are available. The class size for remote courses is limited due to exercise complexity.

For private remote offerings, alternate timings can be arranged; for example, a course can be performed over the course of multiple weeks once a week, etc.

Course Type	Location	Minimum Number of Students	Cost Per Student
Public	Remote	5	$1250
Private	Remote	5-8	$1500
Private	Onsite	5	$2000 + Travel fees
Private	Onsite	5+ (max of 20)	$1750 + Travel fees

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