Advanced Embedded Security and Vulnerability Research Training Syllabus

OVERVIEW

The purpose of this course is to introduce students with an embedded security and vulnerability research principles. Students that having this course will learn on how to learn to approach an embedded system in a security research manner, how to find communication ports, how to debug and emulate non x86 CPU architectures, RISC CPU architecture internals, how to find vulnerabilities using a code review and fuzzing, how to exploit common software vulnerabilities (buffer overflow, use after free, type confusion etc.), exploit mitigations such as ASLR, DEP, stack cookies etc. walkthrough through exploitation of such vulnerabilities.

STUDENT REQUIREMENTS

- C/C++ coding experience
- Python experience
- Basic experience with a debugger

WHAT STUDENTS SHOULD BRING

- Laptop
- VMWare with Ubuntu
- IDA Pro - Windows/OSX
- Administrator/root access – mandatory

SYLLABUS

Module 1 – Embedded 101

In this module you will learn about what's an embedded device, what are the difference between embedded devices and non-embedded devices, communication interfaces, development for embedded device, debugging, MIPS introduction, the difference between embedded security research and regular security research, security research tools.

This module focuses on the following artifacts:

- Differencing between embedded and non-embedded devices
- Regular security research and embedded security research
- Development and debugging for embedded systems (cross-compiling, gdb server)
- Communication interfaces (I2C, SPI, UART, JTAG etc.)
    - Lab:
        In this lab the students will communicate through a serial port after detection and identification of a black box hardware.

- Tools
    - QEMU
    - Unicorn

- MIPS Introduction
    - Lab:
        In this lab the students will exercise MIPS instruction set by solving programming puzzles.

Module 2 – CPU Architecture

In this module you will learn about Virtual Memory, Caching, MIPS pipeline stages, stalls (data hazards, delay slots), branch predictor and more about MIPS.

This module focuses on the following artifacts:

- RISC vs CISC
- MIPS CPU Pipeline stages
    - Data hazards
    - Delay slots
    - Branch prediction
- Virtual Memory
- Caching (and cache coherency)

- Rowhammer Vulnerability Explanation
- Meltdown and Spectre Hardware Vulnerabilities Explanation
- MIPS
- Lab:
  In this lab the students will exercise MIPS instruction set by solving programming puzzles.

Module 3 – Reverse Engineering

In this module you will learn about reverse engineering in an embedded systems, dynamic and static reverse engineering, tools for ease the reverse engineering process, reverse engineering automations

This module focuses on the following artifacts:

- Disassemblers and Debuggers
- CPU Emulators
- IDA Walkthrough
- GDB and GDB Server
- Unicorn
- IDAPython
- Automated reverse engineering – python
- Lab:
  In this lab the students will solve reverse engineering challenges and write automated reverse engineering tools.

Module 4 – Vulnerability Research

In this module you will learn about the process of vulnerability research and exploit development, common data structures such as stack, heap, fuzzing etc.

This module focuses on the following artifacts:

- Process of vulnerability research and exploit development
- Hands on labs on Stack overflow, Heap overflow, Use-After-Free, Double-Free real vulnerabilities
- Pwntools
- Shellcoding 101
  - Problems and how to solve them
  - Lab:
    - In this lab the students will write reliable shellcodes in MIPS architecture
- Fuzzing

- Type of fuzzers
- Symbolic execution
- AFL
- Lab:
  In this lab the students will fuzz real world programs to find security vulnerabilities.

Module 5 – The Pwn CTF Challenge

The Pwn challenge is a CTF (capture-the-flag) competition which involved in vulnerability research and exploit development.