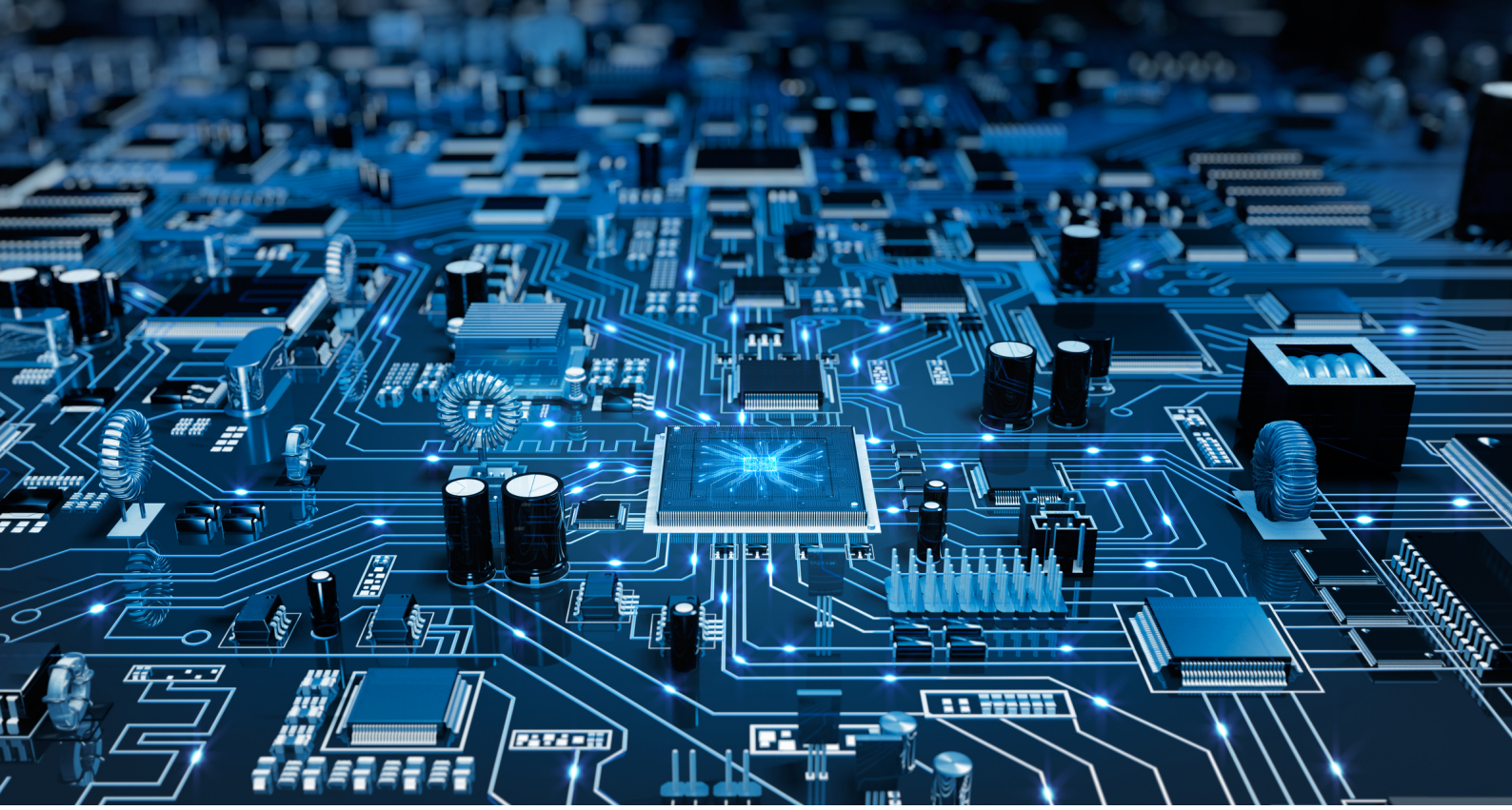

linklayer

Linklayer Training

Syllabus

Table of Contents

Training Overview

Linklayer designs and runs training on information security topics. Our training have been provided publicly through the Black Hat and hardwear.io. We also offer private training upon request.

For more information on training, send us an email: hello@linklayer.com.

Courses

Reverse Engineering Firmware With Ghidra

This hands-on course teaches the concepts, tools, and techniques required to reverse engineer firmware and assess embedded devices. To ensure the tools taught are available to all, we will make use of Ghidra, a powerful open-source reverse engineering tool developed by the National Security Agency. This free, capable tool eliminates the high cost of entry of expensive commercial tools that are currently used for these tasks.

During this training you will:

- Learn general techniques for binary reverse engineering
- Identify, unpack, load, and analyze various types of firmware into Ghidra
- Use reverse engineering techniques to find exploitable vulnerabilities in an embedded Linux device
- Map device vector tables, peripheral memory, and system calls to find exploitable vulnerabilities in a bare-metal device
- Identify remotely exploitable vulnerabilities in a Bluetooth Low Energy device

Labs attacking an embedded Linux system and a bare-metal Bluetooth Low Energy device will be used to deliver a hands-on experience. You can expect to leave this course with the skills to reverse firmware for a variety of embedded targets. You'll also take home a target board to continue building your skills after the course.

The global embedded system market is predicted to be worth over $200 billion by 2020. An embedded system is a combination of software (called firmware) and hardware which together facilitate the accurate functioning of a target device. These increasingly popular devices are not only found in the home, but automotive, telecommunications, healthcare, industrial, and military & aerospace.

Working with firmware requires skills beyond ordinary binary reversing. This course begins with an introduction to reverse engineering ARM binaries, then moves into skills for various types of firmware. We will use Ghidra, the NSA's open-source reverse engineering tool, throughout the course. This highly extensible tool supports many different processor architectures, making it well suited for firmware reversing. Ghidra's feature-set is comparable to costly tools such as IDA Pro.

Two targets will be explored in the course: an embedded Linux device and a bare-metal ARM

device with Bluetooth Low Energy. These types of devices represent what's inside many products in the wild.

This course is divided into four half-day blocks. Each course block adopts a Mission Essential Task List (METL) approach where students are taught a list of tasks required in order to successfully implement the skills in the hands on section. We will follow this agenda:

- Block 1 - Introduction to Embedded Reverse Engineering & Hello Ghidra
- Block 2 - Embedded Linux Device
- Block 3 - Bare-Metal Device 1: Device Peripherals and Interrupts
- Block 4 - Bare-Metal Device 2: RTOS, System Calls, Bluetooth Low Energy, Debugging

Course Outline

Block 1 - Introduction to Embedded Reverse Engineering & Hello Ghidra

Summary: We will begin with a basic overview of methodology, and some processes that can be used for binary reversing. We will talk about basic tools for looking at binaries, and solve our first simple "crackme" together.

Tasks:

1. Introduction to reverse engineering techniques
2. Learn types of binaries and what's inside
3. Simple RE using "strings"
4. Create a Project and import a binary
5. Ghidra basics: the UI, functions, tools and tips and tricks
6. Open binary and auto-analyze in CodeExplorer
7. Navigate the Listing view to locate various labels
8. Use the Functions view to navigate by functions
9. Use the Symbols view to navigate by symbols
10. Label symbols and functions
11. Use the Back and Forward navigation buttons
12. Manage UI panes (split, dock, stack, detach)
13. Navigate cross-references via the Listing view and Find References To
14. Read and understand a simple function in assembly and disassembled code
15. Demo: reverse engineering a simple crackme in Ghidra

Lab: Crackmes

Students will work on a set of simple "crackme" binaries to get experience with loading and analyzing binaries. This will familiarize students with the process of working with binaries and expose them to increasingly difficult reverse engineering challenges. This lab will provide students

with an opportunity to gain experience with Ghidra, the main tool that will be used for the remainder of the course.

Block 2 - Embedded Linux Device

Summary: This block will consist of looking at our first real hardware target: an embedded Linux device. A lecture component will discuss working with these targets. Students will get a device and firmware image to work on for the rest of this block. We will solve a challenge together before letting students work on their own. Periodically, we will discuss the solutions to the various challenges.

Tasks:

1. Embedded Linux systems introduction. What they are,how they are used and nuances often found in common implementations.
2. Embedded Linux system security. Common vulnerabilities you are likely to see.
3. Grab real update from support page, unpack firmware image with binwalk
4. Navigate embedded Linux root filesystem
5. Identify relevant configuration and binaries
6. Load binaries into Ghidra project
7. Navigate binaries to identify security critical functions
8. Identify vulnerabilities in binaries
9. Demo: identifying a real command injection vulnerability (CVE-2018-5383)

Lab: Embedded Linux Device Students will attack an embedded Linux device designed for this course. The device contains a number of vulnerabilities which are modelled after findings in real devices. Students will have to unpack the firmware and load binaries into Ghidra, locate potentially vulnerable code paths, then identify and exploit vulnerabilities. This lab will give students the opportunity to practice a process for reverse engineering firmware in devices such as routers, IP cameras, NASes, etc…

Block 3 - Bare Metal Device 1

Summary: In this block, we'll focus on a bare-metal target. The block will consist of a lecture on bare-metal targets, what makes them different, and specific techniques for analyzing their firmware. Students will receive a device and firmware image which will be used for the remainder of the day.

The block will focus on mapping peripheral registers and interrupt tables. This will be applied to several different device peripherals (GPIO, UART, and I2C). We will demo one of these peripherals together, then allow students to work on their own with instructor support. We will discuss the results before breaking for lunch.

Tasks:

1. Learn what's different about bare metal targets

2. Bare metal target security and common vulnerabilities for these targets
3. Identification of attack surfaces
4. Microcontroller peripheral concepts
5. Microcontroller vector table concepts
6. Loading Intel Hex formatted firmware into Ghidra
7. Using device datasheet to assign memory map in Ghidra
8. Using the datasheet to identify, understand, and label the vector table
9. Using the datasheet to identify, understand, and map peripheral memories and registers
10. Demo: Locating and reverse engineering GPIO peripheral functionality

Lab: Bare Metal Device Peripherals

Students will practice locating peripheral memory and reverse engineering device functionality. This will familiarize students with the process of reading datasheets, understanding peripheral functionality, and reverse engineering peripheral usages. This lab will give students an opportunity to work with simpler device peripherals (UART and I2C) on real hardware.

Block 4 - Bare Metal Device 2

Summary: During this block, we will cover Real Time Operating Systems (RTOS), system ROMs, and embedded Software Development Kits (SDK) in a lecture. We will discuss how these systems are used by developers and how they can greatly accelerate reverse engineering of a complex target.

This block will focus on Bluetooth Low Energy firmware. First, a demo of mapping syscalls will be used to show how to extract encryption keys from the device. Students will then be able to work on reverse engineering various Bluetooth Low Energy functions using the skills gained throughout the course.

The course will conclude with a discussion and Q&A.

Tasks:

1. Learn about RTOS and SDK packages
2. Research the nRF51 “SoftDevice”
3. Using RTOS/SDK to your advantage
4. Identify SVC instructions and calling convention
5. Label SoftDevice syscalls
6. Identify syscall usages (encryption, BLE, etc…)
7. Locate security critical functions
8. Determine hardcoded AES key by syscalls
9. Connect JLink debugger to target

Lab: Reverse Engineering BLE

Students will use the skills learned throughout the course to identify the Bluetooth Low Energy handler functions on the device. With these functions located, students will reverse engineer and attack the device's BLE interface. This lab will give students an opportunity to work with more complex device features.

Why Take This Course?

Embedded is on the rise and firmware surrounds us today. From toasters to aircraft, you'll find firmware responsible for controlling devices. As a result, embedded device security is a growing field within information security. Having the skills to analyze these types of targets is an asset for anyone in the information security industry as well as those developing devices. The skills taught in this course will aid students in attacking and defending embedded devices.

Ghidra is a highly capable tool which has been released to the public for free. It provides comparable functionality to IDA Pro for analyzing firmware binaries. This course is one of the first to provide a hands-on introduction to the tool, which can be used for any type of binary reversing. It is also one of very few courses that focuses on reverse engineering of firmware.

Top Takeaways

- Hands-on skills in binary reverse engineering using Ghidra
- Experience with unpacking, loading, and reversing embedded Linux targets
- Bare-metal firmware reverse engineering techniques using a real-world target

Lab / Lecture Breakdown

This course aims to be 30% Lecture and 70% Labs

Each block contains a lab component to reinforce the lecture. In total, this course will contain 4 multi-part labs. Each lab is organized as self-paced exercises, allowing all students to work at their own pace.

Bespoke Training

Linklayer has developed and delivered bespoke training based on client request. If you have a specific need, we may be able to provide support or refer you to a subject matter expert.

Team

Eric Evenchick

Eric Evenchick has worked in security, design, and development roles for hardware and software companies. He now specializes in embedded device security, automotive security, and bespoke tool development. Eric's work with embedded systems began with development of research vehicles at the University of Waterloo, in partnership with General Motors and the US Environmental Protection Agency. This experience lead to roles in developing automotive firmware and reverse engineering vehicle systems at companies including Tesla Motors and Faraday Future.

Eric has previously held the roles of Technical Director at NCC Group and Principal Research Consultant at Atredis. In these roles, he performed security assessments on a wide variety of hardware and software targets. Eric holds a Bachelor of Applied Science in Electrical Engineering from the University of Waterloo. He has presented at numerous software and security conferences including Black Hat, escar, SecTor, ToorCon, NorthSec, and PyCon USA. His work has been featured by several prominent publications, including Wired and Forbes.