FI for Software Developers

Target audience:
Developers of Embedded Systems

Type of course:
Secure Boot, Authentications, Security Checks, etc

Duration:
4 hours

Fault Injection (FI) attacks are routinely used in security testing to alter the intended behavior of a device by changing either the data in the memory or the control flow of a program. The result of a successful fault injection attack is often the ability to entirely bypass complex security and protection mechanisms.

Aims and objectives

In this course you will learn how to assess the impact of instruction skipping and data corruption attacks on security critical source code, and propose a cost effective remediation plan.

Features

Prerequisites	Experience with C/C++ programming
Content	Video lectures: ~1h10m Interactive exercises: ~50 knowledge assessment questions and code review exercises

Evaluating fault injection resilience

Fault injection attacks are routinely performed in high-end hardware certification projects. These are the most powerful and versatile attacks against an embedded system. At Riscure, we utilize advanced equipment (which we also manufacture) in order to test the resilience against faults of any device.

With the right guidance, the learning curve for a specialist who performs fault injection attacks can be relatively short, especially if one has a background in electrical engineering.

If not properly hardened, any target is susceptible to Fault Injection attacks. As a developer, you benefit most from learning what makes Fault Injection attacks difficult. Learn how to spot the 'interesting' areas of code and add countermeasures which are effective without adding a big performance or cost penalty.

AGENDA

1. INTRODUCTION TO FI

Real world examples

What is a fault, and why is it relevant?

Where are FI attacks used?

LEARNING OBJECTIVES

- Explain what FI is and why it is relevant
- Explain the methodology of FI attacks
- Explain the impact of FI on a provided use case

2. CHARACTERIZATION OF FAULTS

How does a glitch happen?

Types of glitches: how do different glitches affect a given target?

Effects of glitches

- Instruction skipping
- Data corruption

LEARNING OBJECTIVES:

- Enumerate different types of glitches and their effect
- Estimate the effect of instruction skipping and data corruption on a given code
- Build awareness to the sensibility of assembly code to FI: the same code compiled using different compilers may lead to different vulnerable points

3. EVALUATING THE COMPLEXITY OF FI ATTACKS

The parameter search problem

Creation of faults

- Typical devices for FI
- Single vs multiple faults

Reporting FI attack performance: the success rate and attack reproducibility metrics

LEARNING OBJECTIVES:

- List the practical challenges of FI
- Plan the best approach for a given scenario
- Explain the success rate and attack reproducibility metrics, and why they are relevant

4. OVERVIEW OF COUNTERMEASURES

Software countermeasures:

- redundancy
- control flow check
- values check

Hardware countermeasures

- glitch detectors
- shields
- redundancy

Tradeoff: cost vs effect (from both the developer's and the attacker's perspective)

LEARNING OBJECTIVES:

- Implement basic software countermeasures against FI on a given example
- For each hardware countermeasure, explain what it is designed to accomplish
- Plan the software and hardware countermeasures based on given budget