Research in Firmware Security

Stoyanov

October 20, 2020

Abstract

In recent years embedded devices have become very popular in many countries worldwide, they are often used in households, factories, and even in infrastructure objects. Their usage is predicted to increase steadily in the next decades. Our focus is that all embedded devices run on Firmware, which is basically computer software that is meant to work with specific hardware. The Firmware is not as secure as we would like it to be, there are many ways of hacking (reverse engineering) it so that one (the hacker) can find how exactly it operates and if there are any holes that can be exploited for benefit or used to harm either the users or the creators of the firmware.

0.1 Introduction

The research we are conducting is about firmware security mainly but we will touch on the topic of embedded devices also because the two work together. In general, most devices with firmware are relatively easy to reverse engineer by hackers who want to find vulnerabilities that they can exploit to attack different parts of the eco-system. Therefore the goal of this research is to find a way to increase firmware security.

0.2 Research Design

1. Reverse Engineering, how it is done questions: for the needs of this paper we are going to conduct an applied research.

1. The first thing is to understand how is firmware being reverse engineered?
    - What tools an techniques are used?
    - How does those tools work, what algorithms they execute?
    - Also the result is very important, what holes are found?
    - How dangerous are those they?
    - Can those holes be used to illegally access other parts of an software eco-system?

2. Simple ways to enhance firmware security (if any): Are there any software solutions to patch known vulnerabilities? Are there any hardware solutions to patch known vulnerabilities? How to implement those solutions? Any other ways for improving security?

3. Applied solutions result check: 1. Do the applied software solution work? 1.1. Did they create any other issues? 2. Did the applied hardware solutions work? 2.1. Did they create any other issues? 3. Overall solution check. 3.1 How much harder is now to penetrate firmware's code? 3.2 Did the number of reverse engineered/penetrated devices decrease? 3.3 Do the proposed solutions worth the effort for the achieved result?

0.3 Data collection

0.4 Analysis