Threat modelling of industrial controllers: A firmware security perspective

Anitha Varghese, Arijit Kumar Bose
ABB Corporate Research,
Bangalore, India
E-Mail: anitha.varghese@in.abb.com; arijit.bose@in.abb.com

***Abstract*—Industrial devices are becoming more and more networked, as M2M communications gain acceptance. Industrial controllers are usually resource constrained real time devices, and the firmware of these controllers play a vital role enabling the controllers to function accurately and reliably. Binary image of the firmware in these embedded controllers are usually key intellectual property of any OEM making the device. Developing the firmware for a control function is a time consuming process, and being able to get the binary image of firmware can accelarate the product building for counterfieting. We provide a survey of the different threats, different attack methods to materialize the threats, attacker capability required to mount an attack based on an existing threat, and the impact of these attacks on the system. In this paper, we discuss the traditional industrial firmware management process like its distribution, updation and installation in industrial controllers, and describe the security threats in the firmware management methods. We also provide mitigation methods to the threats identified.**

Keywords-Embedded Controller; Firmware; Threats; Industrial; Anticounterfeiting; Confidentiality

I. INTRODUCTION

As industrial controllers become more and more networked, security features in industrial controllers become a key differentiator for market acceptance. Although physical security at key points in a factory or plant precludes tampering of embedded controllers used for industrial automation, a strong security solution should ensure security even when the controllers and the communication ports to the controller are accessible to an attacker. This paper aims to identify and examine generic threats, to the industrial controllers.

In this paper we describe different kinds of threats to industrial controllers such as unauthorized changes to the configuration parameters of the embedded controllers, unauthorized changes to the firmware, unauthorized read of the status messages, configuration parameters etc., and the protection of intellectual property in the form of firmware in the embedded controllers.

One of the factors that ensure security of the industrial controllers is that the physical access to ports used for configuration, monitoring and upgrading firmware is limited. This assumption is common to the different threats we consider later in this section, and hence the topic is described before we get into the threat analysis. But physical security may be inadequate in the event of an attacker gaining access to the controller.

II. EMBEDDED CONTROLLERS : PRODUCTION AND OPERATIONAL SCENARIOS

This section describes the typical system of an industrial controller, in place from supplier through to shipment to the customer, along with the typical installed scenario and serves as a base system to identify the vulnerabilities in the system with respect to the security objectives. The system is shown pictorially in Fig.1.

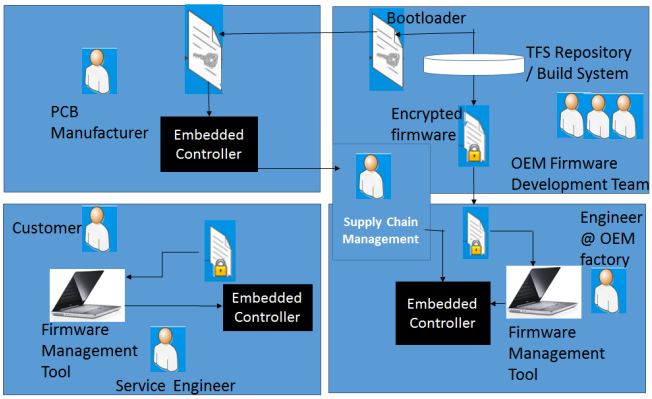

Figure1. Production and Operation of an Industrial Controller

A. Firmware

Typically the firmware for the industrial controller is developed and maintained by the OEM that makes the product. A custom boot-loader (developed by the OEM) is used to load the firmware first time or for subsequent upgrades in the field. The boot-loader typically contains routines (or modules) to read the input interface (a serial interface, such as USB interface) and receive the firmware apart from chip specific FLASH memory routines for read, write and erase functions. The firmware (binary image) is protected (made confidential) by encryption using a strong encryption algorithm to prevent any copying of design or code. This is very important to ensure that firmware is not copied during distribution.

B. Hardware

The embedded hardware is typically manufactured by a third-party supplier outside OEM control. The third-party

supplier is provided with the boot-loader application by OEM. The manufacturer burns the boot-loader firmware image into the embedded controller and ships back device units with the custom boot-loader ready for use. Encryption "key" is embedded in the boot-loader code and is used to decrypt the firmware during the first firmware load in production or firmware upgrades during the lifecycle of the product.

C. Production Setup

An encrypted firmware is downloaded into the device at OEM factory by engineer or technician. Post firmware download, each device undergoes acceptance tests and is ready for shipment after successfully passing tests and quality checks.

D. Operational Scenario

The industrial controller is usually kept inside a cabinet, and access to the cabinet is limited to authorized personnel. Embedded controllers usually have different communication interfaces such as serial connection to the keypad, which is usually mounted such that it is accessible from outside the cabinet, USB interfaces accessible from outside the cabinet, which can be used to connect computers or other USB enabled devices, to configure the controller, collect event logs etc.

Before, digging inside the threat modelling of industrial firmware, we first define certain terms[1].

III. Definitions

- **Threat:** A potential for violation of security, which exists when there is a circumstance, capability, action, or an event that could breach security and may cause harm. The potential for occurrence of a harmful event, such as an attack.
- **Attack :** Attack is the materialization of a threat which may be successful or unsuccessful
- **Attack Surface:** Exposure, reachable and exploitable vulnerabilities that can be used to commit a security breach or launch an attack
- **Attack Methods:** Ways in which the threat can be materialized attack and exploit any vulnerability. There might be multiple ways, in which case, those will be detailed in the section
- **Attacker capability:** The resources in terms of access, tools etc.an attacker needs to realize an attack corresponding to a given threat
- **Probability of an attack:** Probability of an attack indicates the relative probability of an attack based on the particular threat.
- **Impact:** Qualitative description of the probability of a particular attack happening, and the consequences. The impact of an attack on operation or possible harm

IV. Threat Model

In this section, we provide threat analysis of the existing firmware management approach as described in Fig. 1. The list of threats has been identified based on the use cases associated with the firmware distribution, installation and upgradation process as illustrated in Fig. 1. Before, going in to the detailed description of threats, we first provide a generic introduction to threat modelling in the next subsection.

A. Threat model background

Threat modeling is a technique to identify threats to any product and helps in proposing the suitable countermeasures for mitigating the threats that could be relevant to the product.

Threat modelling helps the designers and architects to identify security problems at an early stage of a product life cycle. Many type of security vulnerabilities can emerge during a product's design phase. These vulnerabilities can reside in the product software or in its hardware design, also on how the product is planned to be deployed and used and in many other aspects. Such vulnerabilities can lead to different kind of threats to the product, where an attacker can exploit these vulnerabilities to launch an attack.

Therefore, while designing a product, it is very important to identify the security issues and thus address these security problems with appropriate countermeasures. Threat modelling helps perform this activity. Performing a threat modelling during the product's design phase helps to design a product, which is secure by design.

There are several approaches available to develop threat models like OWASP[2], STRIDE[3]etc. These are widely used approaches for performing a threat modelling for any product or system mainly because these are generic in nature. However, prior art shows that these approaches are mostly used for threat modelling of a software application. For threat modelling of the firmware management process Fig.1, we have used the STRIDE based threat modelling approach. STRIDE is a system developed by Microsoft for classifying computer security threats. It provides a mnemonic for security threats in six categories as described below[4][5]:

Threat Category: Spoofing

- Property: Authentication.
- Definition: Attempting to gain access to a system by using a false identity. This can be accomplished using stolen credentials of a legitimate user.
- Example: Impersonating or masquerading as an authentic and legitimate user trying to access the firmware.

Threat Category: Tampering

- Property: Integrity.
- Definition: Unauthorized modification of data, for example as it flows over a network between two computers.
- Example: Corrupting the firmware of a controller when it is being transferred over a network or unauthorizely modifying the firmware that is stored in a controller.

Threat Category: Repudiation

- Property: Non-repudiation.
- Definition: Ability of users (legitimate or otherwise) to deny that they have performed specific actions or transactions. Without adequate auditing, repudiation attacks are difficult to prove.
- Example: Illegitimately gaining access to the firmware without leaving any trace of attacker's identity.

Threat Category: Information Disclosure

- Property: Confidentiality.
- Definition: Unwanted exposure of private data.
- Example: Unauthorized disclosure and leakage of firmware contents.

Threat Category: Denial of Service

- Property: Availability.
- Definition: Process of making a system or application unavailable.
- Example: Firmware crashing leading to controller not being able to boot, Unsuccessful upgradation of firmware leading to booting failure in controller.

Threat Category: Elevation of Privilege

- Property: Authorization.
- Definition: A user with limited privileges gaining the controls of a privileged user to gain privileged access to an application.
- Example: An unprivileged user gaining access to the firmware of a controller by illegitimately escalating its access rights.

In the next sub-section, we provide a threat modelling of the firmware management technique as mentioned in Fig.1. The threats have been grouped and classified as per the STRIDE six categories of threat.

Category: Spoofing

I. Threat: Unauthorized access to firmware.

Definition: If there is a lack of user authentication and access control guards on the firmware, a rogue user can impersonate as a legitimate user and can access the firmware that is present in the controller.

Illegitimate access to the firmware can create a surface for an attacker to change the configuration settings of the firmware to malformed settings, which could eventually lead to malfunctioning of the embedded controller.

II. Threat: Downloading rogue firmware package.

Definition: Attacker can download and install rogue firmware into the embedded controller. Since, the controller does not verify the authenticity of the firmware package; the devastating firmware package would look like a legitimate firmware in front of the embedded controller. The controller can continue installing it without performing an authentication check of the firmware.

Installation of a rogue firmware on the embedded controller can lead to a deviation of the controller behavior than what it is supposed to be. This could lead to harmful operations by the controller.

Category: Tampering

III. Threat: Unauthorized modification of firmware.

Definition: If there is an absence of firmware access control mechanism, an unauthorized person can modify the firmware content of the controller.

It could also happen that the firmware management tool (Refer Fig. 1) may be reading a tampered firmware package. This means the disc or a memory stick that contains the firmware package may be tampered. By installing such corrupted firmware in the controller can lead to malfunction of the controller, which could eventually lead to devastating effects.

Category: Repudiation

IV. Threat: Denying from an act of illegitimate firmware access on the controller

Definition: Due to the absence of any tracking mechanisms that tracks who and when the firmware of the controller is being accessed, any rogue operations and illegitimate access on the firmware can go unnoticed without revealing the intruders identity. The attacker can be successful in hiding its presence and trace.

Category: Information Disclosure

V. Threat: Unauthorized disclosure of controller's firmware

Definition: Firmware may contain very sensitive information, which if gets leaked can result into financial and even competitive losses. If the firmware in the controller is not stored with confidentiality protections, then an attacker can scan the controller's memory and obtain the firmware binary image. The binary image can be reverse engineered to deduce the source file of the firmware. The memory scanning and reverse engineering can be performed by using sophisticated memory analysis and reverse engineering tools.

In addition, if the firmware is not being confidentially transferred to the firmware management tool (Refer Fig.1), there are chances of an attacker obtaining the binary image of firmware. One example such unauthorized disclosure could be, if the firmware package is transferred to the firmware management tool over insecure network channel.

Category: Elevation of Privilege

VI. Threat: Escalation of privileges of a normal user to gain unauthorized access and operations on the embedded firmware.

Definition: If the firmware of the controller does not incorporates fine granularity of access controls, a normal legitimate user of the controller can unauthorizely gain the possession to access the firmware. In a set of legitimate users of the controller, all of them will be able to access the firmware. Such escalation of privilege rights can lead to a normal user gaining the ability to perform more than its intended operation. This could lead to undesirable events like incorrect operations on the controller's firmware.

In the next subsection, we provide some brief recommendations for a secure firmware management of industrial controllers.

B. Recommendations for firmware management in industrial embedded space

As described in section B, that the threats to the controller's firmware fundamentally lies with its authentication, integrity, auditability, confidentiality and authorization. Threats related to "Spoofing" can be mitigated by introducing authentication mechanism in the controller's firmware. Such authentication can be done by username/password or using digital certificate based tokens like X.509 based tokens[6].

"Tampering" threats to the controller's firmware can be counteracted by digitally signing the firmware packages. This signature can be performed by a keyed hash algorithm like HMAC[7] or by asymmetric signature operations like RSA[8]. Signing the firmware packages would also ensure the authenticity of the firmware i.e. whether or not the firmware is coming from a legitimate firmware manufacturer.

"Repudiation" based threats can be mitigated by accounting and auditability. Each access to the controller's firmware should be recorded. Digitally signing the firmware package can also help to determine the manufacturing source of the firmware.

To preserve the confidentiality of the firmware and thereby handling the information disclosure type of threats can be performed by encrypting the firmware package. This encryption should be done both when the firmware is stored inside the controller and also when it is transferred through any means to the firmware management tool (Refer,).

Privilege escalation based threats can be mitigated by introducing granularity in access controls to the firmware. Different applications and users should have different levels of access and operational rights on the controller. The "Principle of least privilege"[9], could be a good reference principle for implementing granular access controls on the firmware. "Role based access controls" can also be introduced in the controller to restrict normal users from accessing the controller's firmware.

V. CONCLUSION

In this paper,we describe generic threats to industrial controllers under some assumptions of the production and operation of the industrial controllers. Threat analysis of specific controllers and their application scenarios, and implementing mitigation techniques are important to ensure safe and reliable operation of industrial controllers in the event of an attack.

REFERENCES

[1] Bruce Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", John Wiley & Sons

[2] Open Web Application Security Project (OWASP) https://www.owasp.org/

[3] Microsoft STRIDE threat modelling, The STRIDE threat model, commerce server 2002. http://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

[4] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan , "Improving Web Application Security: Threats and Countermeasures" Microsoft Corporation

[5] Shawn Hernan, Scott Lambert, Tomasz Ostwald and Adam Shostack , "Uncover security design flaws using the STRIDE approach"

[6] X.509 format digital certificates, Wikipedia

[7] Keyed-Hashing for Message Authentication https://www.ietf.org/rfc/rfc2104.txt

[8] PKCS #1: RSA cryptography specifications,Version 2.0, RFC 2437 https://www.ietf.org/rfc/rfc2437.txt

[9] Principle of least privilege, Wikipedia, en.wikipedia.org/wiki/Principle_of_least_privilege