HERE BE BACKDOORS: *A Journey Into The Secrets Of Industrial Firmware*

Ruben Santamarta.
Security Researcher. IOActive Labs.

Abstract

PLCs, Smart Meters, SCADA, Industrial Control Systems…nowadays all those terms are well known for the security industry. When critical Infrastructures come into play, the security of all those systems and devices that control refineries, Water treatment or nuclear plants pose a significant attack vector.

For years, the isolation of that world provided the best 'defense' but things are changing and that scenario is no longer valid. It is feasible to attack devices used in critical infrastructures without even having them physically..

This paper covers the approach followed to discover backdoors, confidential documentation or software, vulnerabilities. The main intention is providing a clear explanation of the methods used to face this kind of challenges.

Introduction

Industrial Automation has played a significant role in the advance of humanity; it has improved the security, accuracy and overall capabilities of our modern society. A large number of sectors rely on the security and resilience of the Industrial Control Systems to provide critical services to people.

Access to some devices used by these ICS is usually restricted because of several reasons: commercial issues, embargo, security concerns, etc. Therefore, from a security researchers' perspective this fact poses an important challenge since we face the task of analyzing a device without phisically having one of them.

However, this lack of resources can be compensated by developing an alternative approach that would allow researchers research into these devices through the analysis and emulation of the firmware they run.

Thus, we will be able to discover vulnerabilities or to conduct pentests against critical systems that cannot stopped or duplicated. The objective of the approach here presented is emulating as much as possible the target device/ICS in a controlled environment.

Methodology

The first step of almost every research is the information gathering stage. Usually you have to spend hours or even days collecting as much information as possible about your target, since the most important requirement to analyze something is understanding how it works.

There are 3 main items we should be able to get

- Documentation [3]
- Software
- Firmware

It is recommended to use open source intelligence in order to do so. We should focus our search on 2 different sources

- Vendor website
- 3rd party/Integrators websites

Documentation

We have to identify not only vendor documentation, but all those documents detailing ICS projects where the device has been deployed are potentially valuable for us.

The information collected can provide the researcher with a lot of helpful information, i.e:

- Device Internals
    - Architecture
    - Standars supported
    - Operating System

    - Protocols
    - Security
- Default configurations/accounts
- Programming API / SDK

Software

By reverse engineering th software vendors provide to client or end-users to control or configure the device we can extract a lot of useful information such as binary protocol internals or backdoor accounts. Sometimes the firmware is deployed by a custom user-client windows application which drops and uploads the new image to the device.

Firmware

This is the most important part. If we manage to find a valid firmware image we will be able to analyze how the device is working, we could even perform dynamic analysis of the binaries embeded into the firmware by using cross-platform emulators.

The task is also the most time-consuming and complicated part of the research. In general terms, to analyze the firmware we have follow the following steps.

- Identify
    - Compression algorithms
    - Encryption algorithms
    - Architecture
    - Embedded blobs
    - Bootloader
    - Static libraries
    - Strings
    - File Systems [1]
    - Arbitrary files/blobs
    - Operating System
    - Kernel Images
    - Binaries

- Recontruct Firmware [4]
    - Symbols
    - Entry points
    - Base Address
    - Functions

- Analyze file system / binaries
    - Dump file system / binaries
    - Mount a file system
    - Emulate binaries through closs-platform emulators [2]

Tools

Despite the complexity we only need just a few tools

- gdb
- Radare
- IDA Pro
- QEMU
- binwalk
- other common tools such as hex editors or compilers.

In order to illustrate this approach we will cover below a real case.

Schneider ION Smart Meters Backdoor

Introduction

Usually smart meters contain certain security measures at both software and hardware level mainly implemented in order to prevent fraud. Smart meters are complex devices that not only can be used to revenue metering but also as SCADA devices to control and monitorize power quality, substations, etc...

Discovering a backdoor in a Smart Meter

The first step was collecting as much documentation as possible, since I didn't have access to the actual device any information would be very valuable.

Basically, these Smart Meter allow remote (via telnet) access by using a user limited account which lets the customer modify some basic settings. However, one of the things that took my attention when reading the documents was a section that described a Factory Login account as 'reserved', with no other information.

In order to try to figure out what that Factory Login account was and its capabilities I reverse engineered the firmware, which was available at the Schneider website. This firmware contains a customized Real Time Operating System. It took little time to discover what was behind the Factory Login account.

The Factory Login 'Backdoor' account

This account grants full control over the smart meter to anyone who is able to login with valid credentials. For example, those smart meters that are intended to be used as revenue meter come locked from factory to prevent fraud. By using this factory account anyone could unlock these meters to modify its billing, among other things.

By reverse engineering the firmware I was able to figure out how to generate the password for all these smart meters. Basically, the password is a 32 bit number computed using a hash algorithm seeded with a hardcoded 'secret' string and the serial number of the smart meter.
The only variable element, the serial number, can be known by the attacker without any problem since it is prompted when connecting to the smart meter via telnet.

At this point I was curious to know why and how Schneider was using that backdoor so I reverse engineered the official software Schneider provides to its customers to control these smartmeters: IONSetup.
I discovered a hidden functionality which allowed to login into the smart meter by using this Factory Account. So apparently, there were two different versions of IONSetup, one released for the customers which

had some features disabled and one internal release for Schneider's support staff containing additional features enabled, although both versions shared the same code base.

This hidden functionality used a very specific string as the username for the Factory Login so I did a google search on it. The first result I got was an open ftp server containing confidential documents from Schneider, some of them were detailing the backdoor (literally, schneider documents refer to this account as the 'Backdoor account') account functionality. According to these documents this account let Schneider to have an advantage over its competitors.

Impact

To sum up the scenario, an attacker could generate at anytime valid credentials for the Factory Login account in order to hack into the smart meter, even remotely, therefore fully controlling it.
The consequences derived from controlling these smart meters can vary depending on the scenario where it is deployed. However, technically the following attacks would be possible:

Power disruption

Indirectly
By using false data injection, triggering false alarms etc… we can force a disruption in the supply if we are able to trick the supplier into thinking that something wrong is happening at our facility/home.

Directly
These devices have complex capabilities, including the ability to interact with other devices in order to adjust their consumption or turn them on/off. Basically, these meters can talk common SCADA protocols, such as Modbus, which means there is room for using on of these smart meters to attack other industrial devices.

Pivoting

From the attacker’s perspective, controlling one of these devices is pretty much the same that having access to a regular computer. It’s a device with networking capabilities that can be used in a lateral attack to access other part of the company’s network.

Fraud

An attacker can tamper the meter in order to modify the collected data used to calculate its billing.

References

[1] http://blog.ioactive.com/2012/02/solving-little-mystery.html
[2] http://reversemode.com/index.php?option=com_content&task=view&id=77&Itemid=1
[3] http://reversemode.com/index.php?option=com_content&task=view&id=78&Itemid=1
[4] http://reversemode.com/index.php?option=com_content&task=view&id=80&Itemid=1

About IOActive

Established in 1998, IOActive is an industry leader that offers comprehensive computer security services with specializations in smart grid security, software assurance, network penetration testing, and compliance. Boasting a well-rounded and diverse clientele, IOActive works with a majority of Global 500 companies including power and utility, hardware, retail, financial, media, aerospace, high-tech, and software development organizations. As a home for highly skilled and experienced professionals, IOActive attracts talented consultants who contribute to the growing body of security knowledge by speaking at such elite conferences as Black Hat, Ruxcon, Defcon, BlueHat, CanSec, and WhatTheHack. For more information, visit www.ioactive.com.

IOActive™
Using Our Past to Secure Your Future.