OPPOSING FORCE

Stealing PLC Intellectual Property
[ a Red Teaming Story ]

hackinthebox
Keeping Knowledge Free for Over a Decade
Matteo Beccaro | @_bughardy_

Founder & CTO at Opposing Force

- The first Italian firm specialize in offensive physical security

Member of since May 2016

[ did you enjoy yesterday’s roundtable? ]
Agenda

- Introduction
- What is Red Teaming
- Tales from the field
- PLC Hacking: the case study
- Q&A
Agenda

- Introduction
- What is Red Teaming
- Tales from the field
- PLC Hacking: the case study
- Q&A
We got hired for a Red Teaming mission from an industrial manufacturer in Italy.
A very important preliminary note:

several technical details for the described engagement are under NDA

private Q&A session could be an option for specific questions but...

it will cost you [lots of] drinks
Engagement details

1. No addresses provided for the target’s physical facility
2. Seven [7] days long engagement
3. The only provided information was the company’s name
4. Full control of attack scenarios to be implemented
Engagement rules

1. Allowed targeting any physical, cyber and human assets
2. No disruptive attacks were allowed
3. No harm against personnel
One deadly simple goal for the Red Teaming mission:

stealing the PLC “recipe” of a specific machinery model
- Introduction
- What is Red Teaming
- Tales from the field
- PLC Hacking: the case study
- Q&A
What is Red Teaming

- Simulation of advanced attack scenarios, goal(s)-driven
- Not limited to the cyber security domain
- Could combine cyber, physical and human hacks
- Design to specifically testing the maturity of a Blue Team
- Often sadly used as synonymous of PT since.. It is cool!
- Often intended as a ”I need some extra budget, please hack me hard so I can “motivate” the top management..”
What is Red Teaming | 

We opted for a physical intrusion approach..
Agenda

- Introduction
- What is Red Teaming
- Tales from the field
- PLC Hacking: the case study
- Q&A
Tales from the field
Multiple possible scenarios were identified:

1. Entering and stealing the PLC
2. Entering and download the PLC data directly from the machine
3. Entering and placing a backdoor in the Company network for later access
Our equipment

- Camera with telescopic lens
- SDR
- Drone for aerial surveillance
- Time and patience
After 36 hours..
Tales from the field |
We found that the factory has an alarm system with a control panel at the entrance.
..unluckily we could not get the pin code from the surveillance..
We mapped the IR sensors placed inside the factory

We then bought the same alarm model in order to find possible vulnerabilities that we could exploit in order to bypass it.
We found that the anti-jamming feature has a fifteen [15] seconds timeout after which the alarm starts.

Moreover we found that IR sensors were not capable of covering the entire facility, and so leaving unprotected multiple interesting blind spots.
Tales from the field |
Agenda

- Introduction
- What is Red Teaming
- Tales from the field
- PLC Hacking: the case study
- Q&A
We bought two devices of the same model and brand in order to find vulnerabilities that would allow us to download the receipt.
The "recipe" download/upload features can be protected by a user-defined passcode.
The passcode is a 4 digits code.
After 5 failed attempts there is a 30 minutes timeout/lock.

5 tries / 30 minutes = 0.167 try/minute
Analysing the MODBUS traffic during the download of a recipe with a known passcode we found that:

- There seem to be a magic packet that reset the failed attempts counter
- The passcode verification is done client-side

This means there must be a way to read the passcode from the device memory!
Further investigations demonstrated that the PLC accepts commands to read/write arbitrary addresses in the memory.

We could then:

- Reset the failed attempt counter setting its memory value to 0
- Read the address from the memory in which the passcode is stored
The commands are a slightly modified MODBUS ASCII commands. A sample structure is the following:

\[ PKT = :0103000028C040D4/r/n \]

: = Starting Char
01 = Device Address
03 = MODBUS Command
000028C040 = Payload
D4 = LRC Checksum
\r\n = Ending Chars
The command 03 means Read Holding Register. Its payload is slightly modified and it’s made by

$PKT = 0103000028C040D4r\n$

000028C0 = Memory Address
40 = Bytes to read
KEEP CALM
IT'S NOT A BUG
IT'S A FEATURE
We also found out that the passcode is stored in clear-text in the device memory.
Moreover we can read/write, without any sort of authentication, that part of the memory using MODBUS commands.
The software use this functionality in order to check if the submitted passcode is correct or not.
The passcode is encoded using a custom scheme that we retrieved easily after some attempts:

- 0 --> 0x30 (ASCII: 0)
- 1 --> 0xb1
- 2 --> 0xb2
- 3 --> 0x33 (ASCII: 3)
- 4 --> 0xb4
- 5 --> 0x35 (ASCII: 5)
- 6 --> 0x36 (ASCII: 6)
- 7 --> 0xb7
- 8 --> 0xb8
- 9 --> 0x39 (ASCII: 9)
- A --> 0x41 (ASCII: A)
- B --> 0x42 (ASCII: B)
- C --> 0xc3
- D --> 0x44 (ASCII: D)
- E --> 0xc5
- F --> 0xc6
We started to write an exploit to be used against the stolen device..
PLC hacking

High-level exploitation strategy

1. Verify connection
2. Retrive the passcode
3. Success?
   - Yes: Decode the passcode
   - No: Retrive the passcode
4. Dump receipt
Did we succeed?
Agenda

- Introduction
- What is Red Teaming
- Tales from the field
- PLC Hacking: the case study
- Q&A
Any question?
Don’t be shy..
Thank you