Electronic Access Control Security

Matteo Beccaro || HackInTheBox
Amsterdam, May 27th, 2016
- Matteo Beccaro
- Founder & Chief Technology Officer at Opposing Force
  - The first Italian company specialize in offensive physical security
- Twitter: @_bughardy_ | @_opposingforce
What do you need?

Extract the zip

What you will find in the archive:

- VM with all tools and libraries for the hands-on parts
- VirtualBox installer
- VirtualBox guest-addition

username: opposingforce
password: opfor2016
Module 1 – Introduction
- Historical introduction on access control attacks

Module 2 – Attacking NFC
- NFC: what are we talking about?
- Weapons for NFC-based solutions
- Penetration test methodology
- Hands-on
- Case studies
Module 3 – Attacking RF communications
- Radio Frequency and EAC Systems
- Exploring Radio Frequency communications in practice
- Hands-on: receiving your first transmission
- SIGINT with GNU Radio
- Understanding RF communications security

Module 4 – The challenge
- Introducing the challenge
- The awards 😊
Module 1 || introduction
Access Control system?

A system composed by several elements which aim is to limit the access to certain resources only to authorized people.

The system is composed by two type of elements:

- Human
- Technological
What was an Access Control system?

The technological elements
Introduction

- What was an Access Control system?

  The human elements...
What was an Access Control system?

...often fail
First access control hackers? Magicians..
First access control hackers?

Social Engineers
What is an Access Control system?
What is an Electronic Access Control system?

- It may employ different technologies
  - NFC
  - RF
  - Biometrics
  - Mag-stripe
  - Mobile phones
  - etc.
Module 2 || attacking NFC
Module 2 – Attacking NFC
- NFC: what are we talking about?
- Weapons for NFC-based solutions
- Penetration test methodology
- Hands-on
- Case studies
What is NFC?

- NFC stands for Near Field Communication
- Frequency at 13.56 MHz
- 3-5 cm of range
- Widely used for
  - Access control systems
  - Electronic ticketing systems
  - Mobile phone applications
Notorious NFC families:

- MIFARE
  - MIFARE Classic
  - MIFARE Ultralight
  - MIFARE DesFire
- HID iClass
- Calypso
- FeliCa
- 1-4 KB memory storage device
- Strong access control mechanisms
  - A key is required to access data sectors
  - Use of Crypto1 Crpto1 algorithm
  - Sadly broken..
  - ..but still so widely used (!) – RFID door tokens, transport tickets, etc.
MIFARE Ultralight

- 64 byte memory storage device
- Basic security mechanisms
  - OTP (One-Time-Programmable) sector
  - Lock bytes sector
  - Mostly used for disposable tickets
  - It has some more secure children:
    - ULTRALIGHT C
    - ULTRALIGHT EV
- 2 KB, 4KB or 8 KB memory size
- Advanced security mechanisms (3DES, AES, etc.)
- File system structure is supported
- Several variants are available
  - DESFIRE
  - DESFIRE EV1
  - DESFIRE EV2
- Same encryption and authentication keys are shared across every HID iClass Standard Security installations (!)
- Keys have already been extracted (!!!)
- Two variants
  - iClass Standard (very common)
  - iClass High Secure (not that common)
- Both variants are BROKEN
NFC-based Electronic Access Control systems

- We need to create a common **methodology**
- We need **tools** to effectively assess these systems
- We need **secure architectures** as references and best practices
The token

- Usually a NFC card
  - MIFARE Ultralight
  - MIFARE Classic
  - HID

- The card can store
  - Timestamp of the last stamping
  - Details on the location where we used the token
  - Credentials, access level, etc.
- What about MIFACE Classic?
  - It is just BROKEN
- What about MIFARE Ultralight?
  - Well, it’s bleeding..
    - Lock attack
    - Time attack
    - Reply attack..
- HID
  - BROKEN, again
- Can operate offline or online
- Wire or wireless connected to the controller
  - RS232, Ethernet, etc.
- Usually supports multiple standards
- Can store secrets and keys used for authentication
- Usually it can
  - Read token(s) data
  - Send token data to the controller
  - Give a feedback to users on operation’s success
- Connected both to readers and backend
  - Wiegand, Ethernet, rs232
- Receives data from the reader(s)
  - Support multiple readers technologies
- Sends the data to the backend
  - Open the door
  - Deny the access
The backend can be cloud-based or not. It is usually wired connected through RS232, Ethernet, etc. It performs multiple operations, including providing token validation logic, statistics, and logging.
Module 2 – attacking NFC
- NFC: what are we talking about?
- Weapons for NFC-based solutions
- Penetration test methodology
- Hands-on
- Case studies
Tools of the trade |

- HydraNFC
- ProxMark3
- ChameleonMini
- NFCulT
- HydraNFC (~90 €)
- Users Texas Instrument TRF7970A NFC chipset (13.56MHz only)
- MIFARE 1k and 14443A UID emulation
- ISO 14443A sniffing (also autonomous mode)
- 2 different raw modes
- ProxMark3 (~200 €)
- HF and LF capabilities
- Very large community
- Supports almost every known RFID tag
- Support sniffing and emulation
- ChameleonMini (~100 €)
  - [http://kasper-oswald.de/gb/chameleonmini/](http://kasper-oswald.de/gb/chameleonmini/)
- HF (13.56MHz) only
- Almost same capabilities as HydraN
- Different chipset
- The firmware is only available for old revision
- NFCuIT (~0 €)
- Originally designed for ticketing systems, it can be also used for generic EAC system security assessment
- Mobile app for NFC-enabled Android smartphones
  - Implements Lock, Time and Reply attacks
  - A “custom edit mode” is available for bit by bit data editing
- The app currently supports the MIFARE Ultralight format only
  - MIFARE Classic support will be released on summer 2016
The custom editing feature is useful to better understand the structure of data stored onto the token.

- Quick encoding from hex to bin and back
- The app allows token bit by bit data editing
Module 2 – Attacking NFC
- NFC: what are we talking about?
- Weapons for NFC-based solutions
- Penetration test methodology
- Hands-on
- Case studies
Access Control system attack surface
Access Control system attack surface

Token

Reader
Reader
Reader
Reader

Channel

Controller / Middleware

Channel

Channel

Backend System

Client(s)
<table>
<thead>
<tr>
<th>Attack Surface</th>
<th>Attacks to Perform</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFC Interface</td>
<td>Analyze the authentication mechanisms</td>
<td>Secrets extraction, MiTM attacks</td>
</tr>
<tr>
<td>Hardware board</td>
<td>Side channel attacks</td>
<td>Secrets dumping or guessing</td>
</tr>
<tr>
<td>Memory</td>
<td>Assess logic vulnerabilities in the implementation</td>
<td>Bypass security mechanisms</td>
</tr>
</tbody>
</table>
Access Control system attack surface
<table>
<thead>
<tr>
<th>Attack Surface</th>
<th>Attacks to Perform</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFC Interface</td>
<td>Analyze the authentication mechanisms</td>
<td>Secrets extraction, MiTM attacks</td>
</tr>
<tr>
<td>Hardware board</td>
<td>Analyze the exposed interface (JTAG, UART, etc.)</td>
<td>Firmware or secrets dumping</td>
</tr>
<tr>
<td>Ethernet, wiegand, etc.</td>
<td>Is MITM possible? Intercepting the exchanged data</td>
<td>Intercepting secrets or sensitive data</td>
</tr>
</tbody>
</table>
Access Control system attack surface

Diagram showing the components of an access control system, including:
- Token
- Reader
- Reader
- Reader
- Reader
- Channel
- Controller / Middleware
- Channel
- Backend System
- Channel
- Client(s)

The diagram highlights the attack surface components within the access control system.
<table>
<thead>
<tr>
<th>Attack Surface</th>
<th>Attacks to Perform</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware board</td>
<td>Analyze the exposed interface (JTAG, UART, etc.)</td>
<td>Firmware or secrets dumping</td>
</tr>
<tr>
<td>Eth, serial Interfaces, etc.</td>
<td>Is MITM possible? Intercepting the data</td>
<td>Intercepting secrets or sensitive data</td>
</tr>
<tr>
<td>Computer Application</td>
<td>Analyzing exposed network services</td>
<td>Complete control of the machine (e.g., add new users)</td>
</tr>
</tbody>
</table>
Access Control system attack surface
<table>
<thead>
<tr>
<th>Attack Surface</th>
<th>Attacks to Perform</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web application(s)</td>
<td>Classic web app-related attacks</td>
<td>Data exfiltration, service interruption, etc.</td>
</tr>
<tr>
<td>Network service(s)</td>
<td>Classic network services-related attacks</td>
<td>Data exfiltration, service interruption, etc.</td>
</tr>
<tr>
<td>Physical location</td>
<td>Try to get physical access to the servers</td>
<td>Basically, heavily PWNED</td>
</tr>
</tbody>
</table>
Access Control system attack surface
<table>
<thead>
<tr>
<th>Attack Surface</th>
<th>Attacks to Perform</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware board</td>
<td>Identify forgotten or backdoor pins</td>
<td>Data exfiltration, firmware dumping</td>
</tr>
<tr>
<td>External wires</td>
<td>Try to intercept data passing through those wires</td>
<td>Intercepting sensitive information</td>
</tr>
<tr>
<td>Wireless connection</td>
<td>Intercept and inject data</td>
<td>Intercepting sensitive information, send spoofed information</td>
</tr>
</tbody>
</table>
Module 2 – Attacking NFC
- NFC: what are we talking about?
- Weapons for NFC-based solutions
- Penetration test methodology
- Hands-on
- Case studies
Fire up your
Module 2 – attacking NFC
- NFC: what are we talking about?
- Weapons for NFC-based solutions
- Penetration test methodology
- Hands-on
- Case studies
MIFARE Ultralight ticketing system
MIFARE Ultralight ticketing system
Absence of a UID blacklist in the backend

Lock bit for the OTP sector is not checked by the stamping machine

Timestamps are not encrypted nor signed
MIFARE Classic hotel door lock

Diagram:
- Token
- Reader
- Reader
- Reader
- Channel
- Controller / Middleware
- Channel
- Backend System
- Channel
- Client

Opposing Force
MIFARE Classic hotel door lock
<table>
<thead>
<tr>
<th>0C8BCF38 70881200</th>
<th>ãœ8pâ</th>
<th>Card’s UID</th>
</tr>
</thead>
<tbody>
<tr>
<td>17EA0000 00000000</td>
<td>Í</td>
<td>Room number: (\text{int}(0x17ea, 16) = 6122)</td>
</tr>
</tbody>
</table>
Module 3 || attacking RF communication
Module 3 – Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security
Radio Frequency identification is widely used to control physical accesses

- Advantages
  - Automatic identification
  - High reliability
  - High security
Different technologies based on operating frequency band

- **Low Frequency (LF)** – 125 KHz
- **High Frequency (HF)** – 13.56 MHz
- **Ultra High Frequency (UHF)** – 433 MHz, 860-960 MHz and 2.4 GHz
Low Frequency band

- Tags
- Access control token
Radio Frequency and EAC Systems

**High Frequency band**
- Door locks
- Ticketing systems
Radio Frequency and EAC Systems

Ultra High Frequency band

- Automated Gates
- Keyless Entry Systems
- Alarms
- Smart Locks
Radio Frequency and EAC Systems

- Common technologies and protocols
  - Fixed and rolling code
  - NFC
  - Bluetooth
  - ZigBee
  - Z-Wave
Module 3 – Attacking RF communications
- Radio Frequency and EAC Systems
- Exploring Radio Frequency communications in practice
- Hands-on: receiving your first transmission
- SIGINT with GNU Radio
- Understanding RF communications security
How to explore wireless communications?
- Software Defined Radio (SDR) devices with GNU Radio
- Software implementation of most parts of a radio system
  - Cheap hardware
  - High flexible
Exploring Radio Frequency communication

Three SDR-compatible devices

<table>
<thead>
<tr>
<th>Device</th>
<th>Frequency Range</th>
<th>Bandwidth</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTL-SDR Dongle</td>
<td>24 MHz – 1.76 GHz</td>
<td>2.4 MHz</td>
<td>~ 20 €</td>
</tr>
<tr>
<td>HackRF</td>
<td>1 MHz – 6 GHz</td>
<td>20 MHz</td>
<td>~ 300 €</td>
</tr>
<tr>
<td>USRP B200</td>
<td>70 MHz – 6 GHz</td>
<td>56 MHz</td>
<td>~ 700 €</td>
</tr>
</tbody>
</table>
Exploring Radio Frequency communication

GNU Radio

- Platform to develop radio applications, called **flowgraphs**
  - Series of connected signal processing blocks
- GNU Radio libraries include blocks to perform signal processing

![Diagram of flowgraph with nodes Source, Blk_2, Blk_3, and Sink connected]

[Image: flowgraph.png]
Exploring Radio Frequency communication

- GNU Radio
  - Supports the programming of custom C++ blocks
- GNU Radio Companion (GRC)
  - Graphical UI to program GNU Radio applications
  - Supports the creation of UI for applications
Exploring Radio Frequency communication

- **GRC Interface**

![GRC Interface Diagram](image-url)
Exploring Radio Frequency communication

- **GRC Interface**

![diagram of GRC interface with variables and nodes]

```
/home/opfor/grc_gnuradio
```

Loading: `/home/opfor/Downloads/AM_PB_RX.grc`  >>> DoIn

Showing: `/home/opfor/Downloads/AM_PB_RX.grc`

OPPOSING FORCE
Exploring Radio Frequency communication

- GRC Interface
Exploring Radio Frequency communication

- GRC Interface

```
/home/opfor/grc_gnuradio

Loading: `/home/opfor/Downloads/AM_PB_RX.grc`
>>> Done

Showing: `/home/opfor/Downloads/AM_PB_RX.grc`
```
Exploring Radio Frequency communication

- **GRC Interface**

![GRC Interface Diagram]

The diagram shows a block diagram of a GRC (GNU Radio Companion) flowgraph. The flowgraph includes nodes such as Signal Source, Multiply, Throttle, Low Pass Filter, and WX GUI Notebook. The nodes are connected with arrows indicating the flow of data. The flowgraph is open, with options and settings visible, such as the file name `/home/opfor/grc_gnuradio` and commands being run to load and show a flowgraph file.

- **BLOCK LIBRARY**
  - `[Audio]`
  - `[Bluetooth]`
  - `[Boolean Operators]`
  - `[Byte Operators]`
  - `[Channel Models]`
  - `[Coding]`
  - `[Control Port]`
  - `[Debug Tools]`
  - `[Depreciated]`
  - `[Digital Television]`
  - `[Equalizers]`
  - `[Error Coding]`
  - `[FCO]`
  - `[File Operators]`
  - `[Filters]`
  - `[Fourier Analysis]`
  - `[GSM]`
  - `[GUI Widgets]`
  - `[Impairment Models]`
  - `[Instrumentation]`
  - `[IQ Balance]`
  - `[Level Controllers]`
  - `[Math Operators]`
  - `[Measurement Tools]`
  - `[Message Controllers]`
  - `[Misc]`
  - `[Modulators]`
  - `[Networking Tools]`
  - `[NOAA]`
Exploring Radio Frequency communication

- "Hello World" in GNU Radio
Exploring Radio Frequency communication

- “Hello World” in GNU Radio
Exploring Radio Frequency communication

- RTL-SDR Source Block
Exploring Radio Frequency communication

- WX GUI FFT Sink Block
Module 3 – Attacking RF communications
- Radio Frequency and EAC Systems
- Exploring Radio Frequency communications in practice
- Hands-on: receiving your first transmission
- SIGINT with GNU Radio
- Understanding RF communications security
Build a FM receiver
Fire up your
Module 3 – Attacking RF communications

- Radio Frequency and EAC Systems
- Exploring Radio Frequency communications in practice
- Hands-on: receiving your first transmission
- SIGINT with GNU Radio
- Understanding RF communications security
Define a methodology to study real world signals

Three main steps

- Intercept and record signal
- Study characteristics
- Reverse transmitted data
Define a methodology to study real world signals

Three main steps

1. Intercept and record signal
2. Study characteristics
3. Reverse transmitted data
SIGINT with GNU Radio

- **GQRX**
  - SDR receiver and spectrum analyzer based on GNU Radio and QT Graphical toolkit
  - User-friendly interface
  - Supports RTL-SDR, HackRF, USRP and other SDR devices
  - Records signal to WAV file
SIGINT with GNU Radio

Frequency Selector

106.000 000 MHz
SIGINT with GNU Radio

REAL-TIME SPECTRUM
SIGINT with GNU Radio

INPUT

CONTROLS

[Image of a signal analysis software interface showing a graph and input controls]
SIGINT with GNU Radio

RECEIVER OPTIONS
SIGINT with GNU Radio

DEMODULATE D SPECTRUM
SIGINT with GNU Radio
- Black-box interception of a RF signal
  - If the frequency is unknown, search power peaks in the spectrum
Define a methodology to study real world signals

Three main steps:

1. Intercept and record signal
2. Study characteristics
3. Reverse transmitted data
Modulation

- Impresses a waveform, called carrier, with another signal that contains data to be transmitted
### Signal Identification Guide

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Description</th>
<th>Frequency</th>
<th>Mode</th>
<th>Bandwidth</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Buzzer (MDZI) (UZEN)</strong></td>
<td>Traditionally, the signal is used for public announcements.</td>
<td>4.625 MHz</td>
<td>AM</td>
<td>2.8 kHz</td>
<td>Russia</td>
</tr>
<tr>
<td><strong>Tire Pressure Monitoring System (TPMS)</strong></td>
<td>Signal is from a Chrysler TPMS (Tire-Pressure Monitoring System) sensor.</td>
<td>315 MHz</td>
<td>AM</td>
<td>433 kHz</td>
<td>Worldwide</td>
</tr>
<tr>
<td><strong>Toyota Car Key</strong></td>
<td>Wireless entry rolling code car key.</td>
<td>315 MHz</td>
<td>AM</td>
<td>40 kHz</td>
<td>Worldwide</td>
</tr>
</tbody>
</table>
SIGINT with GNU Radio

- **Audacity**
  - Useful to study recorded signals
  - Support RAW data files used with USRP and HackRF utilities
Case Study: remote control at 433 MHz
Case Study: remote control at 433 MHz
Case Study: remote control at 433 MHz
Let’s study the signal

- Amplitude Modulation (AM)
- Only two amplitude levels
  - Binary transmission using **On-Off Keying (OOK)** modulation
- Repeated trains of pulses
  - Different lengths to encode the ‘0’ and ‘1’ bit
Define a methodology to study real world signals

Three main steps:

1. Intercept and record signal
2. Study characteristics
3. Reverse transmitted data
Focus on a single train

- The first pulse indicates the beginning of the "message"
- **Short** pulses represent binary ‘0’ while **long** pulses represent binary ‘1’

- Transmitted message is **001010010001**
Module 3 – Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security
Case study’s solution security

- The remote control always sends same fixed code (!)
- Malicious people can record and replay signals thus obtaining an unauthorized access

Solution

- Rolling code
- Rolling Code
  - Remote control always sends **different** codes
  - Sender and receiver are synchronized with an internal counter
  - An hardware algorithm calculates the ‘next’ code on the basis of the internal counter’s value
  - A widely used algorithm is **KeeLoq**
  - Rolling code is NOT a unbreakable mechanism.
Module 4 || the challenge
Module 4 – The challenge
  - Introducing the challenge
  - The awards 😊
You are now part of a Red Team, which has been engaged to breach the physical security of a high security facility controlled by a super secret, and “probably” evil, organization known as h4k3rZ T34mZ.

Your task is to open the external facility’s electric gate, thus allow your team to enter the facility and proceed with the intrusion.
You find one employee’s remote controller. It seems to be broken and you can’t use it to open the gate but you decide to open it to see inside....
Agenda

- Module 4 – The challenge
  - Introducing the challenge
  - The awards 😊
Awards

The first two to complete the challenge will win a:

**RTL-SDR Dongle** from [http://www.rtl-sdr.com](http://www.rtl-sdr.com)
Feedback and questions please..
Don’t be shy.. ;-D
Thank you
Start hack Access Control systems NOW!

We have **10 tool box** at the special price of **150 €**

- Each toolkit contains
  - Plastic box with Opposing Force sticker 😊
  - 1 HydraBus with its case
  - 1 HydraNFC
  - 1 Mini USB cable
  - 1 SDR-RTL dongle with its antenna
  - 1 Breaboard with some jumpers
  - 1 NFC MIFARE ULTRALIGHT card