# >>> Tea or Coffee ?

## >>> A small story about hardware security...

Name: Cyril MARIN

Date: January 28, 2019

# >>> DISCLAIMER


The information is furthermore not intended as AN advice in any way.

The consequences of decisions, solely based on the information on this presentation, remain at **one's own risk**.

# >>> Who am I ?


## MARIN Cyril

*cyril.marin@open-groupe.com*



* Developper
  * Mainly Java and occasionally Python
  * Knowlege in system/low level programming
* Familiar in DevOps concepts
  * currently working to implement DevOps best practices and tools on a customer large project
* Not security expert, but interested by the security aspects and willing to experiment on that topic

[1] https://www.open.global/

## >>> Why this presentation ?

### Starting from a simple discussion

* More than expected security issues discovered

### Exploitation is globally quite simple

* Can be done by almost anyone with:
  * time
  * some researches
  * minimum knowledge
  * bit of motivation

### Lessons can be learned for other areas

* Software development
* And IT in general

# And now, I will tale you my little coffee machine story

# >>> My little coffee machine story

## A typical work day:

* Say hello to colleagues
* Plug laptop in
* Check emails
* Realize there is too much...

# >>> My little coffee machine story

**A typical work day:**

* Say hello to colleagues
* Plug laptop in
* Check emails
* Realize there is too much...
* And then, ... Caffein !

# >>> My little coffee machine story

And it's in front of the coffee machine that someone with an evil mind asked:

*Is coffee machine secured ?*

*Is it possible to have some coffee for free ?*

>>> How easy could it be to have free coffee ?

Is my smarphone compatible ?

* Yes, it reacts when tag is near.

## >>> How easy could it be to have free coffee ?

### Is my smarphone compatible ?

* Yes, it reacts when tag is near.

### First naive try

* Try to dump with NFC smartphone app
* Buy a coffee
* Dump again
* Compare data

## >>> How easy could it be to have free coffee ?

### Is my smarphone compatible ?

* Yes, it reacts when tag is near.

### First naive try

* Try to dump with NFC smartphone app
* Buy a coffee
* Dump again
* Compare data
* No change ...

## >>> How easy could it be to have free coffee ?

### Is my smarphone compatible ?

* Yes, it reacts when tag is near.

### First naive try

* Try to dump with NFC smartphone app
* Buy a coffee
* Dump again
* Compare data
* No change ...

### What happen ?

Tags seem to be protected. Dumping is not as simple as I thought.
Only public fields extracted.

# >>> Further analysis

## What type is the RFID chip of coffee machine tag ?

* Scanned tag with smartphone app (NFC Tools)

[1] https://searx.me/

# >>> Further analysis

## What type is the RFID chip of coffee machine tag ?

* Scanned tag with smartphone app (NFC Tools)
    * NXP Mifare Classic 1k

[1] https://searx.me/

# >>> Further analysis

## What type is the RFID chip of coffee machine tag ?

* Scanned tag with smartphone app (NFC Tools)
    * NXP Mifare Classic 1k

## Let's do some Searxing...


[1] https://searx.me/

# >>> Mifare Classic 1/2

## In short

* Introduced in 1994
* 2 versions:
  * 1k : 1024 bytes of storage, split into 16 sectors
  * 4k : 4096 bytes of storage, split into 32 + 8 sectors
* Each sector is secured by 2 keys (A and B, 48bits length)
* Uses CRYPTO-1 proprietary hardware encryption algorithm
* Not all smartphone compatible

[1] https://en.wikipedia.org/wiki/MIFARE

## >>> Mifare Classic 2/2

**And about security strength ?**

* Full reverse engeneering published in March 2008. CRYPTO-1 encryption can be broken in about 200s with a laptop.
* In 2009, with specific ASIC Proxmark 3, only 10s needed.
* Hardened version of Mifare Classic have been released in 2011, but, in 2015, an attack able to recover "secured" data was published.

Since the discovery of last attack in 2015, NXP is officially recommending to migrate from MIFARE Classic product-based systems to higher security products.[1]

[1] *https://www.mifare.net/en/products/chip-card-ics/mifare-classic/security-statement-on-crypto1-implementations/*

# >>> Move from theory to practice

## Github hosts some open source cracking tools

* Nested attack implementation
  * Need at least one key already known
* Dark side attack implementation
  * Find one key from tag

# >>> Move from theory to practice

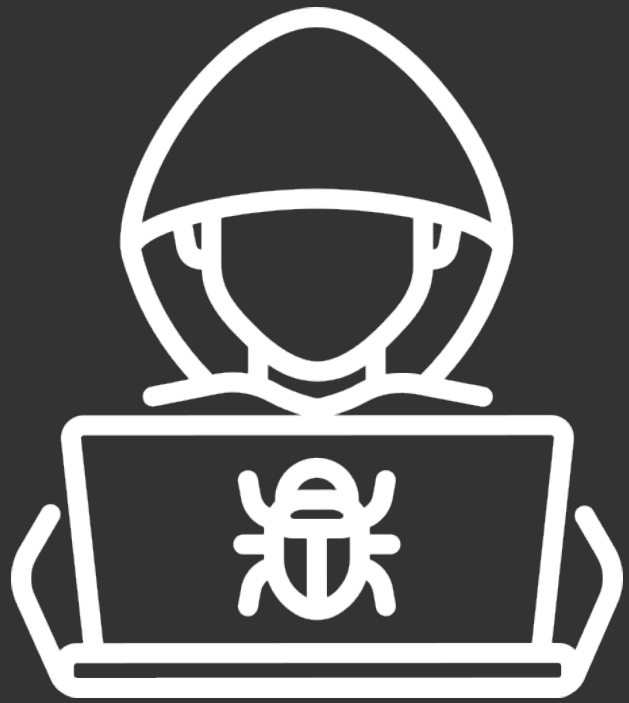
## A small investment from Aliexpress

| | |
|---|---|
| USB NFC reader (ACR122U) | ~18€ |
| NFC tags with writable block 0 (x5) | ~2€ |
| Your favorite Linux distro... | 0€ |
| Total | ~20€ |

# >>> Move from theory to practice

## The hacking process

* Install tools
* Recover A & B security keys
* Dump tag memory
* Buy a coffee
* Dump tag memory again
* Compare dumps
* Identify changes in memory
* Update money amount on tag



Enjoy free coffee !

# >>> Move from theory to practice

## A bit of programming

* Custom Android application
    * First draft working app in 2 hours
    * Only one view
        * 2 java classes
        * 1 xml layout
    * Very few functionnalities
        * display money
        * reset money

## Advantages

Allows to update tag easily and quickly

# >>> Let see a little demo

# >>> A quick recap

* A and B keys recovery
  * Allows to dumping/cloning tag
  * Every needed tools available on Github
  * Took ~1h of computing
  * Cost ~20€ of hardware

# >>> A quick recap

* A and B keys recovery
  * Allows to dumping/cloning tag
  * Every needed tools available on Github
  * Took ~1h of computing
  * Cost ~20€ of hardware
* Dump analysis
  * Identify memory sector, block and bytes were money balance is stored
  * Took a few minutes

# >>> A quick recap

* A and B keys recovery
  * Allows to dumping/cloning tag
  * Every needed tools available on Github
  * Took ~1h of computing
  * Cost ~20€ of hardware
* Dump analysis
  * Identify memory sector, block and bytes were money balance is stored
  * Took a few minutes
* Custom Android app
  * Can display money and refill tag
  * Took 2 hours for first usable version

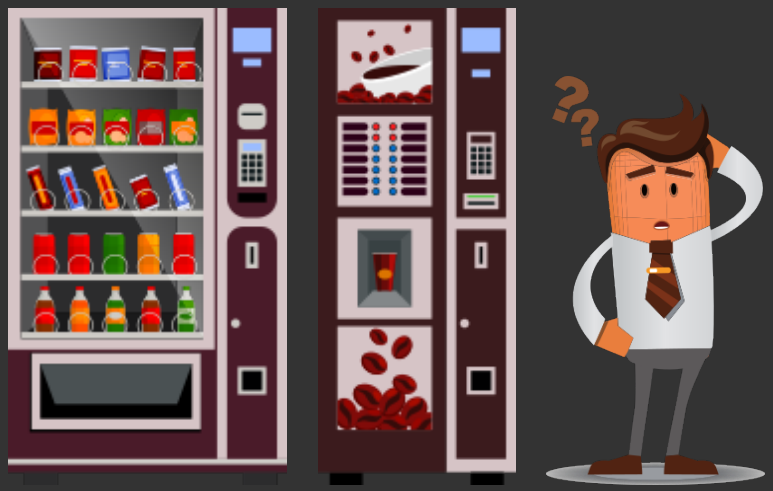
# >>> A quick recap

* A and B keys recovery
  * Allows to dumping/cloning tag
  * Every needed tools available on Github
  * Took ~1h of computing
  * Cost ~20€ of hardware
* Dump analysis
  * Identify memory sector, block and bytes were money balance is stored
  * Took a few minutes
* Custom Android app
  * Can display money and refill tag
  * Took 2 hours for first usable version

**Total**

Less than a day of work load and cost ~20€

# And now ?



# >>> Commercial use example of Mifare Classic tags

1. Micropayment (coffee machine and more...)

[1] https://en.wikipedia.org/wiki/MIFARE

## >>> Commercial use example of Mifare Classic tags

1. Micropayment (coffee machine and more...)
2. Secured building access
3. Health cards
4. Blood donor cards
5. Password storage
6. Public transportation
7. Loyalty cards
8. Hotel room keys
9. ...

[1] https://en.wikipedia.org/wiki/MIFARE

## >>> Commercial use example of Mifare Classic tags

1. Micropayment (coffee machine and more...)
2. Secured building access
3. Health cards
4. Blood donor cards
5. Password storage
6. Public transportation
7. Loyalty cards
8. Hotel room keys
9. ...

Only frivolous things without importance...

[1] https://en.wikipedia.org/wiki/MIFARE

## >>> Let's try something

**What if I check a client's professionnal badge ?**

It's used for building access.

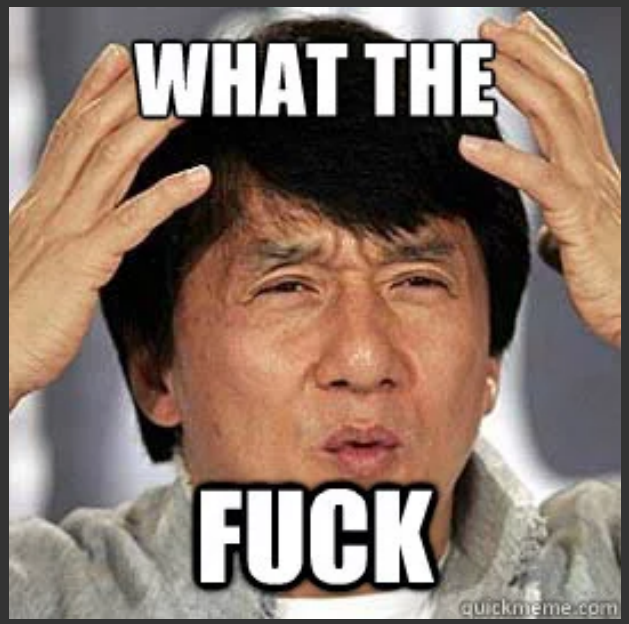
Let's test it with my smartphone and...

>>> Let's try something

What if I check a client's professionnal badge ?

It's used for building access.

Let's test it with my smartphone and...




# Oh wait !

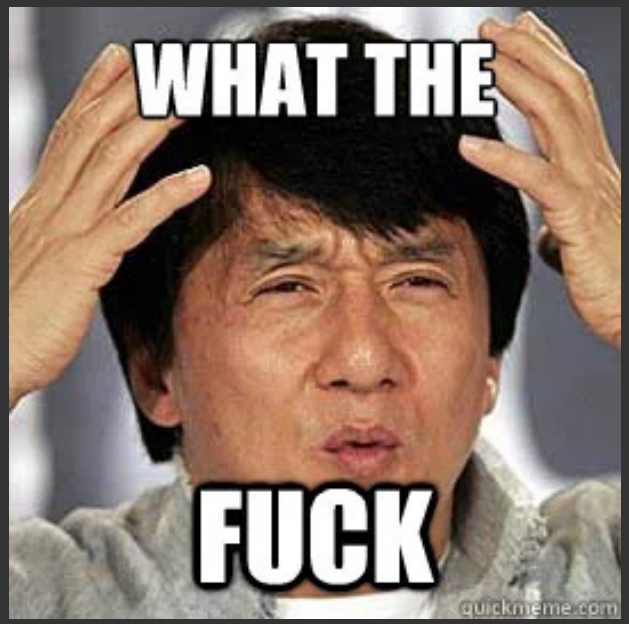
Mifare Classic 1k tag...

>>> Let's try something

## What if I check a client's professionnal badge ?

It's used for building access.

Let's test it with my smartphone and...




# Oh wait !

Mifare Classic 1k tag...

## What does it means ?

It's easy to clone.

So to usurp identity too.

# >>> Some lessons

1. Obscurity is not security
    * Closed source / Without auditability
    * Still crackable !

# >>> Some lessons

1. Obscurity is not security
   * Closed source / Without auditability
   * Still crackable !
2. Cryptography is hard
   * Follow standards !

# >>> Some lessons

1. Obscurity is not security
   * Closed source / Without auditability
   * Still crackable !
2. Cryptography is hard
   * Follow standards !
3. Even if NXP is officially recommending to not use MFC tags since 2015, we can still find many nowaday.
   * Fleet upgrade issue ($$)
   * Information issue ?
     * from manufacturer to subcontractor
     * from subcontractor to final client

# Thank you