CHEAPSCATE:
ATTACKING IOT WITH LESS THAN $60

Rafael Boix Carpi
Principal Trainer & Security Specialist
Riscure
A BRIEF STORY ABOUT
A HARDWARE CTF AND IOT
A brief story about a hardware CTF

2017: 8.4 BILLION “THINGS” CONNECTED TO INTERNET
A brief story about a hardware CTF

The typical IoT device is quite cheap but does a lot of stuff

- Heart of the device: general purpose microcontroller (MCU)
- TON of features for extremely low $$$
  - WiFi / Bluetooth / memory / USB / ...
  - Lots of interfaces & sensors
  - Devkit packed with peripherals typically < 20$
A brief story about a hardware CTF

The typical IoT device is quite cheap but does a lot of stuff:

- Heart of the device: general purpose microcontroller (MCU)
- TON of features:
  - WiFi / Bluetooth / memory / USB / …
  - Lots of interfaces & sensors
- Devkit packed with peripherals typically < 20$

HMM...

WHAT'S THE CATCH?
A brief story about a hardware CTF

GENERAL PURPOSE MCUs ARE NOT SECURE AGAINST SCA AND FI ATTACKS
A brief story about a hardware CTF

GENERAL PURPOSE MCUs ARE NOT SECURE AGAINST SCA AND FI ATTACKS
A brief story about a hardware CTF

Lots of people solved the software challenges
But very few attempted the SCA and FI challenges 😞

Typical reasons given:

- “The SCA equipment is very expensive”
- “SCA and FI are too difficult”
- “These attacks are only for evaluation labs”
- “I’m allergic to mathematics”
- “I will destroy my device”
A brief story about a hardware CTF

My goals today

- You will learn that SCA is affordable for everyone
- You will learn that FI is affordable for everyone
- If you are an IoT developer, and:
  - you handle sensitive info
  - you believe these attacks don’t apply to you
  - you don’t try these attacks
  - you don’t learn how to defend

THEN you WILL get hacked: please do something!!
Rules of the game

In order to prove the point, I will use the following rules:

- Cheapest tooling I could find
- Generic for (almost any) target
- Using Open Source Software only
- Result reproducible by a script-kiddie profile
SIDE-CHANNEL ANALYSIS: THEORY AND APPLICATION

Let’s break an AES implementation!
SCA: theory and application

Challenge: Piece of SCAke (available on riscure.com/Github)

Goal
- Get the AES key from the device

Info
- The device performs AES encryption of a message
- Then replies the encrypted message
SCA: theory and application

The attack works with any crypto implementation
You just need to have the device at reach
SCA in a nutshell

1 – Talk or listen to a device doing crypto (e.g. AES)
2 – Measure power consumption of device doing crypto
3 – SCA program computes math with collected data
4 – You get the crypto key
SCA steps in practice (attacker)

1. Trace acquisition
2. Signal processing
3. Leakage detection (optional)
4. Attack key
SCA steps in practice (attacker)

- Trace acquisition
- Signal processing
- Leakage detection (optional)
- Attack key: CPA

How do I get traces?
Measuring power from an embedded device

Power trace:
- Measured power
- I/O data

Computer

Resistor or Current Probe

Power measurement
Measuring power in an embedded device

- This approach has some issues
- There are other “more efficient” ways to measure power (current probe, EM, ...) but we will not see them today...

Because a resistor is CHEAP!! ($<0.01)$
Cost of the SCA setup

- Resistor
  - US$ 0.01
- Scavenged USB cable
  - Free
- USB soldering iron
  - US$ 4.64, free shipping!
  - Comes with tin!
  - It actually works! AMAZING!
- Hantek 6022BE “oscilloscope”
  - US$ 53.32, free shipping!
  - Comes with probes and cables!
- USB to serial cable
  - US$ 1.10, free shipping!

**Total new hardware cost: US$ 59.07**
Demo time
General steps for SCA

1. Trace acquisition
2. Signal processing
3. Leakage detection (optional)
4. Attack key

Improve signal?
Signal processing: alignment

In order to do math, we need to compare “apples to apples”

IoT things typically have LEDs that blink when busy

The RHMe board has a blinky LED that is ON when busy

- Let’s align measurements with the LED activity
Demo time
General steps for SCA

1. Trace acquisition
2. Signal processing
3. Leakage detection (optional)
4. Attack key: CPA

Does it leak?
Does it leak?

You can use any tool (e.g. R, SciPy, Octave, ...) to compute statistics.

Useful statistic to find info leakage: Welch T-test

- This test shows if I can tell apart random messages from a fixed one.
General steps for SCA

1. Trace acquisition
2. Signal processing
3. Leakage detection (optional)
4. Attack key

Profit?
Warning: math ahead!

I will explain the attack a bit in depth

If you don’t get it now, **don’t worry:**

- SCA programs have everything implemented
- run the scripts and check results
- check later the presentation until it becomes clear
CPA attack: divide and conquer!

Sub-keys in AES
- are used *independently*
- can be attacked *separately*

Bruteforce key: $2^{128}$ tries

Divide and conquer
$16 \times 2^8 = 2^{12} = 4096$ tries

We reduce from $2^{128}$ to 4096!!!
CPA attack: divide and conquer!

- Modern ciphers are fully public
- If we guess part of the key, we can rebuild part of the internals
- Can we check if the values we rebuild match device behavior?
The CPA attack

Data: FF, C7, 09, 8B

Power traces for different data values.

Correlation trace
Side-Channel Analysis takeaways

**SCA Takeaway 1: homebrew SCA is dirt-cheap and allows scalable attacks**

- **Total cost for presented setup:** <US$60
- **Q: What about time? 2 full days for first attack** *
  - 1.8 days fighting the “scope” & drivers & OSS
  - 0.1 days to build the physical setup (removing caps)
  - 0.1 days to build an OSS SCA setup & measure
  - *Note: if you spend US$150 in a decent scope, time is ~2 hours. With professional tools: <2 minutes*
- **Q: What about repeating the attack with same setup? 2 minutes!!**
- **Result: full key retrieval & scalable attack**

**SCA Takeaway 2:**

*Open-Source Software SCA setup complexity for IoT-like devices is minimal*
FAULT INJECTION: THEORY AND APPLICATION

Let’s break a security check implementation!
Challenge: Fiesta

Goal
- Unlock the device

Info
- The device boots
- And then keeps saying “Lock” forever
Hypothesis

Guess is that the code in the board is similar to this:

```java
boolean unlocked=false;
boot_CPU_and_IO();
while(1){
    unlocked=do_some_check();
    if(unlocked){
        print(secret);
    } else{
        print("Lock");
    }
}
```
Hypothesis

What would you glitch here?

```java
boolean unlocked=false;
boot_CPU_and_IO();
while(1){
    unlocked=do_some_check();
    if(unlocked){
        print(secret);
    }else{
        print("Lock");
    }
}
```
Attack plan

What did I glitch:

boolean unlocked=false;
boot_CPU_and_IO();
while(1){
    unlocked=do_some_check();
    if(unlocked){
        print(secret);
    }
    else{
        print(“Lock”);
    }
}
How to generate a glitch 101

Simplest FI attack I can think of: VCC glitching

Clock of the MCU in the RHMe2 board is 16MHz

- 1clk = 62.5ns
- Hopefully timing won’t be an issue

Brainstorm for FI setup

- MCU, a transistor and two PSUs
- MCU with DAC attached a buffer
- MCU with multiplexer chip and a buffer
- ....
CheapoGlitcher

Free development board I got with ARM MCU@ 180MHz

- If you want to buy a similar board, it is ~US$ 15

GPIO pins can be driven @ 90MHz max ➔ 11ns glitch ➔ fast enough

GPIO supplies enough current to power up the IoT board

I’m just going to toggle the GPIO at different instants

Will it even work?
CheapoGlitcher: will it work?

5 hours of high-caffeine drinks in the SHA2017 Amsterdam conference:

100% success rate
CheapoGlitcher unexpected result
Fault Injection (FI) takeaways

**FI Takeaway 1:** you can glitch general purpose MCUs with almost anything
- RHMe2: 100% reproducibility, uber-cheap stuff and 5 hours of effort

**FI Takeaway 2:** countermeasures for FI is a must
- Attack was possible due to the infinite loop: free unlimited retries
- Think about your implementation: do you have such a structure?

```c
/* glitch here */
if(mbedtls_pk_verify(..., hash, signature, ...)) {
  /* do not boot up the image */
  while(1);
} else {
  /* boot up the image */
  boot();
}
```
CONCLUSION & TAKEAWAYS
Conclusion

My goals today

- SCA is affordable for everyone ✓
- FI is affordable for everyone ✓
Conclusion

My goals today

- Developer: should you do something?

Do you know the answer for these questions?

- Is my security going to be somehow bypassed with FI?
- Are any of my secrets going to leak with SCA?
1. **Understand threat**
   - Learn how SCA/FI attacks work (read papers or take a training)
   - Try the attacks yourself!

2. **Develop a solution**
   - Remember: current attacks can be way more advanced than what was presented here
   - Effective countermeasures include multiple levels
     - Software
     - Hardware
     - Protocols
   - Try design patterns as described in documents here: [https://www.riscure.com/gocheap/](https://www.riscure.com/gocheap/)

3. **Verify**
   - Design != Implementation → vulnerabilities are more persistent than you think
   - Independent testing avoids blind spots → security evaluation labs can help
Questions?

Rafael Boix Carpi
Principal Trainer & Security Specialist
boixcarpi@riscure.com
http://www.riscure.com