PEUT ON AVOIR CONFIANCE DANS "L'INTERNET DES OBJETS" SANS TRANSPARENCE?

Aurélien Francillon

Atelier Francophone sur la transparence et l'opacité des systèmes d'information, Lyon, Avril 2018
Introduction
Economic aspects
Why make secure products?
Trust in embedded devices
Verifying trust
Conclusion
Introduction

- **This talk: a “consequence” of about 10 years of working on the security of embedded systems software**

- **Practical approach**
  - Attacking systems
  - Analyzing systems (real products)
  - Developing new security mechanisms to make software more secure

- **Unfortunately**
  - A lot of this “systems security” knowledge is not public
  - Why is it so often so bad?
Problems found in a large scale analysis

- Analysed ~30000 Firmware images
- Hard-coded passwords, SSL keys...
  - SSL private keys which are used by 40,000 IP on the internet...
- Same vulnerabilities across different products
  - Code sharing, Vulnerability sharing
- Several hundreds of vulnerable firmware images... tens of CVEs
- Web analysis: Many basic problems

Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces
A. Costin, A. Zarras, A. Francillon AsiaCCS 2016
A Large Scale Analysis of the Security of Embedded Firmwares
A. Costin, J. Zaddach, A. Francillon, D. Balzarotti, Usenix Security 2014
Security for the 99%

- There are some very secure devices
  - Smartcards, HSMs, ...
  - Not flawless but with a reasonable level of security
  - This is “1%” of the devices

Data source: “guts feeling”

Aurélien Francillon
Security for the 99%

- **There are some very secure devices**
  - Smartcards, HSMs, ...
  - Not flawless but with a reasonable level of security
  - This is “1%” of the devices

- **The remaining 99% is not**
  - Soho equipment
  - Computers peripherals
  - (Some) Industrial systems, etc.

- **Security for the 99% ?**
An economic problem

- Intuitively security requires an extra effort
  - Costs money
  - Customers may not want to pay for it

A bit more complicated...

- Anderson / Schneier “economics of security”

- Security an externality:
  - Manufacturer often not responsible for operating the device
  - No direct loss in case of breach
  - So “why bother with security”
Market for Lemons or silver bullets?

- Markets with asymmetric information
  - Market for Lemons: Used car market (Akerlof)
    - When selling a product seller knows more, buyer less
    - This drives down the average price of an used car
  - Security products: Both seller and buyer lack information (Grigg)
    - Spafford: how to test a unicorn detection device?
    - Market for silver bullets
- Security products v.s. Product security
  - Product security is a lemons' market
Motivations for trust/security on Manufacturers' side

Security considered when:

- **There are active attacks on asset to protect**
  - Conditional access for Pay TV
  - Actual goal is to resist to the attacks

- **Must not fail**
  - E.g., critical military system
  - No need to be profitable

- **Regulations, standards, certifications to pass**
  - ID documents, payment processing
  - Actual goal is to get the certification

- **But what’s for the 99% ?**
Economically speaking: Security or not?

- **In the short term, probably no...**
  - Time to market, Cost
  - Users wants features

  Schneier:
  “Any smart software vendor will talk big about security, but do as little as possible, because that's what makes the most economic sense.”

- **In the long term**
  - A big problem
  - Maintenance, legacy, users defiance
  - Costs can be higher than the initial development
  - Life Cycle (How long will the manufacturer support it?)
Transparency v.s. security

- **Kerckhoffs 2nd design principle:**
  - “... It should not require secrecy, and it should not be a problem if it falls into enemy hands”

- **Often interpreted as:**
  - “if the system is not open and does not receive public scrutiny then it is not secure”
  - Or “Security by obscurity is bad”
  - A wrong interpretation

- **Hiding the system details is actually making attacks much harder**
  - Many more factors

- **However, this has other bad effects...**
Small digression...

- One day I was given old scope for free to play at home...
- It worked 5 minutes and then the Magic Smoke escaped...
Small digression...

Tektronix 2445
Service manual
330 pages
8 horizontal dot positions within each horizontal display area determined by bits 0-2 of character ROM.

Small characters are displayed in the top half of a given display area.

16 vertical dot positions above line reference determined by bits 3-6 of character ROM.

Subframe - 8 characters.

Large characters require 2 adjacent display areas.

Only small characters may be displayed on the bottom line.

*Note: Reference level is -001V with D03, D04, D05, and D06 all LO.
ALL COMPONENTS MOUNTED ON A5-SCALE ILLUMINATION CIRCUIT BOARD ARE SHOWN ON SCHEMATIC DIAGRAM.

1) COMPONENTS WITHIN PARENTHESES MAY NOT BE LOCATED PRECISELY AS SHOWN BUT ARE NEAR THEIR INDICATED POSITION.

2) INDICATES COMPONENTS THAT WERE MANUALLY ADDED TO THE BOARDS AS A RESULT OF MODIFICATION.

3) LABELLED ON SOME BOARDS AS "P" VICE "U."

4) COMPONENT ON BACK OF BOARD.

5) USED FOR FUTURE TV OPTION.
In the “good old days”...

- Before there was documentation for:
  - Mini computers
    - Steve Wozniak learned computers on DEC schematics
  - Apple II
    - Andrew "bunnie" Huang learned from Apple II schematics

- Today datasheets often not available, even for:
  - Raspberry Pi
  - Intel Edison
  - Any secure device
Lack of transparency

- Basic security measures often make them less transparent
- Makes third party audit very hard
  - But does not mean the device is secure...
- Secrecy leads to suspicion
  - What is the device doing with my data?
  - Trying to hide a poor level of security?
  - Something nasty to hide?
- To trust something we need to
  - Blindly trust?
  - Verify it, inspect it?

Aurélien Francillon
From an actual smartphone chip

Aurélien Francillon
Trust and transparency

- **Asymmetric market**
  - Manufacturer knows
  - Customer cannot evaluate security

- **Lack of transparency damages market of secure devices, users cannot:**
  - Educate themselves
  - Learn about security
  - Evaluate security
  - Compare devices

- **How could they want to pay more for security?**

**Increase: Transparency**
Security or lock out

- **Who is in control of the device**
  - Your Manufacturer?
  - Your government, another one?
  - Trusted Computing as “Treacherous Computing” (R. Stallman)

- **Users should eventually be in control**

**Increase: User Control**
Design problem

We need systems to be designed for:

- **User Trust**
  - Letting the choice to the user, owner of the device to which software is running on the device
  - Let the user know which software it is running

- **Security Analysis**
  - We need to be able to independently inspect those systems
Design for User Trust

- **To trust the systems, users needs to:**
  - Know what is running
  - Chose what can be running
  - Be in control
  - Be able to verify

- **Currently there are devices which**
  - We can control, but have zero security (e.g., unlocking android)
  - Are secure but under the control of someone else (iPhone)
Design for User Trust: examples

- Joanna Rutkowska proposal of a state-less laptop
  - Without R/W memories
  - All firmware loaded from an external, trusted, device

- My laptop
  - Has an UEFI Firmware
  - Loaded with my own keys
  - Secure boot, only code I signed

https://wiki.gentoo.org/wiki/Sakaki%27s_EFI_Install_Guide/Configuring_Secure_Boot
BIOS Password Backdoors in Laptops

Synopsis: The mechanics of BIOS password locks present in current generation laptops are briefly outlined. Trivial mechanisms have been put in place by most vendors to bypass such passwords, rendering the protection void. A set of master password generators and hands-on instructions are given to disable BIOS passwords.

When a laptop is locked with password, a checksum of that password is stored to a so-called FlashROM - this is a chip on the mainboard of the device which also contains the BIOS code and other settings, e.g. memory timings.

For most brands, this checksum is displayed after entering an invalid password for the third time:
## Bios Backdoor scripts

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Hash Encoding</th>
<th>Example of Hash Code/Serial</th>
<th>Scripts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asus</td>
<td>Machine Date</td>
<td>01-01-2011</td>
<td>pwgen-asus.py</td>
</tr>
<tr>
<td>Compaq</td>
<td>5 decimal digits</td>
<td>12345</td>
<td>pwgen-5dec.py Windows binary</td>
</tr>
<tr>
<td>Dell</td>
<td>serial number</td>
<td>1234567-595B 1234567-D35B 1234567-2A7B</td>
<td>Windows binary&amp;source</td>
</tr>
<tr>
<td>Fujitsu-Siemens</td>
<td>5 decimal digits</td>
<td>12345</td>
<td>pwgen-5dec.py Windows binary</td>
</tr>
<tr>
<td>Fujitsu-Siemens</td>
<td>8 hexadecimal digits</td>
<td>DEADBEEF</td>
<td>pwgen-fsi-hex.py Windows binary</td>
</tr>
<tr>
<td>Fujitsu-Siemens</td>
<td>5x4 hexadecimal digits</td>
<td>AAAA-BBBB-CCCC-DEADBEEF</td>
<td>pwgen-fsi-hex.py Windows binary</td>
</tr>
<tr>
<td>Fujitsu-Siemens</td>
<td>5x4 decimal digits</td>
<td>1234-4321-1234-4321-1234</td>
<td>pwgen-fsi-5x4dec.py Windows binary</td>
</tr>
</tbody>
</table>
Design for Security Testing

- When do we really need to be able to analyse embedded devices?
  - Each firmware version: Detect vulnerabilities
  - Each independent device: Shipped with bad FW
  - Regularly: Check for compromise
  - Exceptionally: Forensics

- Need for independent analysis
- Requires some access to the device (DFUT)
  - But not reducing the security of the device... Authenticate users?
Design for Security Testing

- Currently first security measures in an embedded system makes it harder to test:
  - Locking JTAG
  - Encrypt/Sign code

- Testing embedded systems is difficult We developed a tool for security testing
  - Avatar http://s3.eurecom.fr/tools/avatar/

Increase: Attack Resistance
In Summary

- **We need more transparency**
  - Datasheets!
  - Access to debug ports!

- **Not because it makes devices more secure but it makes:**
  - Auditable
  - Trustworthy
  - Forensics possible

- **We need mechanisms that**
  - Put users in control
  - Do not introduce new vulnerabilities
  - Are easy to integrate in products
Questions?

Trust, But Verify: Why and how to establish trust in embedded devices (invited paper)
Aurélien Francillon
Proceedings of Design, Automation and Test in Europe (DATE), 2016
http://www.s3.eurecom.fr/docs/date16_francillon.pdf
Backup slides
Liability

- Schneier argues for liability
  - Did not happen... will it one day?

- Probably in some regulated / life threatening markets?
  - Toyota sudden unintended acceleration
    - 9 Million cars recalled
    - 37 deaths alleged

- Will this occur for the 99%?
  - I guess not
Hard disk drive security

- A disk Drive runs a firmware
  - with its own OS
  - Can be updated

- Could be compromised
  - what would be the consequences?
  - The required effort

- To discover it we did it
  - Took a disk and reverse engineered it
  - designed a backdoor

- So yes, feasible but difficult, but a few days later...

Implementation and Implications of a Stealth Hard-Drive Backdoor

J. Zaddach, A. Kurmus, D. Balzarotti, E. Blass, A. Francillon, T. Goodspeed, M. Gupta, I. Koltsidas, best student paper award, ACSAC 2013,
(TS//SI//REL) IRATEMONK provides software application persistence on desktop and laptop computers by implanting the hard drive firmware to gain execution through Master Boot Record (MBR) substitution.

(TS//SI//REL) IRATEMONK Extended Concept of Operations
(TS//SI//REL) This technique supports systems without RAID hardware that boot from a variety of Western Digital, Seagate, Maxtor, and Samsung hard drives. The supported file systems are: FAT, NTFS, EXT3 and UFS.

(TS//SI//REL) Through remote access or interdiction, UNITEDRAKE, or STRAITBAZZARE are used in conjunction with SLICKERVICAR to upload the hard drive firmware onto the target machine to implant IRATEMONK and its payload (the implant installer). Once implanted, IRATEMONK’s frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

**Status:** Released / Deployed. Ready for Immediate Delivery

**Unit Cost:** $0

**POC:** [Contact Information]

---

TOP SECRET//COMINT//REL TO USA, FVEY
(TS/SCI/REL) IRATEMONK provides software application persistence on desktop and laptop computers by implanting the hard drive firmware to gain execution through Master Boot Record (MBR) substitution.

(TS/SCI/REL) This technique supports systems without RAID hardware that boot from a variety of Western Digital, Seagate, Maxtor, and Samsung hard drives. The supported file systems are: FAT, NTFS, EXT3 and UFS.

(TS/SCI/REL) Through remote access or interdiction, UNITEDRAKE, or STRAITBAZZARE are used in conjunction with SLICKERVICAR to upload the hard drive firmware onto the target machine to implant IRATEMONK and its payload (the implant installer). Once implanted, IRATEMONK’s frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0
10. What is the most sophisticated thing about the EQUATION group?

Although the implementation of their malware systems is incredibly complex, surpassing even Regin in sophistication, there is one aspect of the EQUATION group’s attack technologies that exceeds anything we have ever seen before: this is the ability to infect the hard drive firmware.

We were able to recover two HDD firmware reprogramming modules from the EQUATIONDRUG and GRAYFISH platforms. The EQUATIONDRUG HDD firmware reprogramming module has version 3.0.1 while the GRAYFISH reprogramming module has version 4.2.0. These were compiled in 2010 and 2013, respectively. If we are to trust the PE timestamps.
(TS//SI//NF) Left: Intercepted packages are opened carefully; Right: A “load station” implants a beacon

(TS//SI//NF) In one recent case, after several months a beacon implanted through supply chain interdiction called back to the NSA covert infrastructure. This call back provided us access to further exploit the device and survey the network.