Analyzing Thousands of Firmware Images and a Few Physical Devices. What's Next?

Aurélien Francillon
Eurecom, France
aurelien.francillon@eurecom.fr

ABSTRACT

In this talk we will make an overview of security problems that have been found with a large scale automated static analysis (within the firmware.re project [1, 2]) and with a more focused and more manual dynamic analysis (using the Avatar project [4, 5]). We will then discuss what we think we can do about this and how. We argue that to be more trustworthy devices need to be made more transparent (so that users can inspect them), controllable (so that users can do something about it) and resistant to attacks (so that devices will resist to basic attacks).

Keywords

Embedded systems security; Trust

Studying Embedded (In)security

Most of the connected or smart devices we analyzed have a disappointing security level. What is especially disappointing is that they are massively produced devices and that the problems found are often problems that we know how to address. One of the most surprising case is that of firmware images which include public and private key pairs. Those keys are used for serving web pages over HTTPS. Not only those keys are used as is in thousands of similar devices that are online, they are not even changed across different brands (likely because of white labels products).

This should not be problems that we find in real devices, in particular, a lot of research efforts have been put into constructing secure systems and those are very basic problems.

In fact, many products have a good level of security (e.g., secure smart cards), but others are really insecure. Some are security devices: security is at the core of their purpose; while other are not. Unfortunately, such systems with a poor level of security seem to outnumber the secure systems. We nevertheless often rely on their security in our daily life and their failure can have serious consequences.

So What's Next?

A part of the problem is an economic problem: developing secure products is difficult and expensive, and may be overlooked if incentives are unclear.

Second, trust is something that is not blindly granted but that is earned, e.g., by verifying it [3]. Currently, trusted computing mechanisms often rely on unconditional trust on the systems manufacturer. However, users have too few ways to verify that the systems are trustworthy other than blindly trusting the manufacturer, i.e., one can only trust a system fully if he can inspect it. Unfortunately, the first security measures that are implemented in embedded systems often prevent such an independent analysis (e.g., deactivation of a debug port, secure boot, encrypted file system, obfuscation). There is a conflict situation: such measures are often useful in securing a system (slowing down an attacker) but should not jeopardize our ability to analyze them (independently discover software vulnerabilities). Ideally, this problem should be solved during design, we call this Design For Security Testing.

We should design systems where the users, i.e., the devices owners, can decide whom and what to trust. We call this Design For User Trust, where users are in control of the system. Without this, a user may not be able to solve the problem he is able to detect. Finally, devices needs to be designed in a more resilient way, for this we still need to create new methodologies for design, test and building blocks which are easy to integrate in future systems.

We conclude that more research is needed to make it easier to build secure systems, in particular, in the areas of concrete architectures for Design For User Trust and Design For Security Testing.

1. REFERENCES

[1] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti. A Large Scale Analysis of the Security of Embedded Firmwares. USENIX Security, 2014.

[2] A. Costin, A. Zarras, and A. Francillon. Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces. ASIACCS, 2016.

[3] A. Francillon. Trust, But Verify: Why and how to establish trust in embedded devices (invited paper). DATE, 2016.

[4] J. Zaddach, L. Bruno, A. Francillon, and D. Balzarotti. Avatar: A Framework to Support Dynamic Security Analysis of Embedded Systems' Firmwares. NDSS, 2014.

[5] J. Zaddach, A. Kurmus, D. Balzarotti, E. O. Blass, A. Francillon, T. Goodspeed, M. Gupta, and I. Koltsidas. Implementation and implications of a stealth hard-drive backdoor. ACSAC, 2013.

TrustED'16 October 28-28 2016, Vienna, Austria

ACM ISBN 978-1-4503-4567-5/16/10.

DOI: http://dx.doi.org/10.1145/2995289.2995296