Objective Embedded systems are omnipresent in our everyday life. For example, they are the core of various Common-Off-The-Shelf (COTS) devices such as printers, mobile phones, home appliances, and computer components and peripherals. They are also present in many devices that are less consumer oriented such as video surveillance systems, medical implants, automotive elements, military systems, SCADA and PLC devices, and basically anything we usually call “electronics”. Moreover, the emerging phenomenon of the Internet-of-Things (IoT) will make them even more widespread and interconnected. The security and reliability of this broad range of devices is paramount to ensure both the proper functioning of our society and our physical safety. Unfortunately, embedded devices did not reach yet the same level of security we obtained for software of typical personal computers. For example, because of the very heterogeneous set of platforms and architectures, embedded systems still lack a solid set of vulnerability discovery and analysis techniques. The goal of this thesis is to bridge this gap by improving the state of the art of vulnerability discovery in software binaries. In particular, the student will focus on the development of novel static and dynamic analysis techniques that can be applied to the study of real-world, complex firmware images. The proposed approach will need to cope with a number of challenges, ranging from scalability issues, heterogeneity of targets, need for low false positive rates, and the intrinsic difficulty of running dynamic analysis on real embedded devices. To ensure the deployability of the developed techniques, real examples and test cases for the Ph.D. research will be provided by a close industrial support and collaboration with Siemens. Background and Previous Work The work performed in this thesis builds upon two lines of research which are ongoing in our group at Eurecom. The first is related to the use and application of advanced dynamic analysis techniques on the firmware of embedded devices. Zaddach et al. designed and implemented an open source system named Avatar [1], whose goal is to execute a firmware inside an instrumented emulator. Emulating firmwares of embedded devices requires accurate models of all hardware components used by the system under analysis. Unfortunately, the lack of documentation and the large variety of hardware on the market make this approach infeasible in practice. Avatar fills this gap and overcomes the limitation of pure firmware emulation by acting as an orchestration engine between the physical device and an external emulator [7]. By injecting a special software proxy in the embedded device, Avatar can execute the firmware instructions inside the emulator while channeling the I/O operations to the physical hardware. Since it is infeasible to perfectly emulate an entire embedded system and it is currently impossible to perform advanced dynamic analysis by running code on the device itself, Avatar takes a hybrid approach. It leverages the real hardware to handle I/O operations, but extracts the firmware code from the embedded device and emulates it on an external machine. A similar architecture, but supported by a FPGA bridge to increase the throughput, was used by Koscher et al. [5] in their Surrogates system.
The Avatar design was partially motivated by a previous experiment, in which the same author tried to measure the complexity of injecting malicious functionalities in the firmware of a common hard drive [8] – just few months before this threat was first announced as part of the NSA scandal. The paper showed that it is possible to compromise the firmware of a commercial off-the-shelf hard drive, by resorting only to public information and reverse engineering. This requires approximately the same amount of effort and expertise as the development of many existing forms of professional malware (e.g., large scale botnets). Moreover, we also claimed that this attack was well within the capabilities of current cyber-espionage tools (again, this was few months before the world discovered that indeed government were using these exact approach to backdoor several brands of hard disks).

Davidson et al. [4] presented a tool to perform symbolic execution of embedded firmware for MSP430-based devices. Like Avatar, this tool is based on the KLEE symbolic execution engine. However, it relies on firmwares source code as well as on documented SoCs, peripherals mapping, or on a simple emulation layer for them, all of those are rarely available for commercial devices. The second, parallel, line of work, consist instead in the use of static analysis techniques to enable a scalable and large scale analysis of the security of firmware images. The first study in this direction was published by Costin et al. [3] at the Usenix Security conference in 2014. The idea was to collect firmware images for as many devices and vendors as possible and then use a specially designed distributed architecture to unpack and run simple static analysis tasks on the collected firmware images. This paper showed the advantages of an horizontal, largescale exploration to quickly “propagate” vulnerabilities from known vulnerable devices to other systems that were previously not known to be affected by the same vulnerability [2]. Firmalice [6], by Shoshtaishvili et al., presented instead a more sophisticated static analysis technique to analyze the binary code of a firmware and detect the presence of authentication bypass vulnerabilities. The system was tested on three devices, finding evidences of backdoors in all of them.