Vulnerability-Tolerant Architectures for Resource-Constrained Devices

Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation - Executive Summary
Doctoral Program in Computer and Systems Engineering (35^{th}cycle)

Vulnerability-Tolerant Architectures for Resource-Constrained Devices

By

Gianluca Roascio

******

Supervisor(s):
Prof. Paolo Prinetto, Supervisor

Doctoral Examination Committee:
Prof. Alessandro Armando, Referee, Università degli Studi di Genova
Prof. Giorgio Di Natale, Referee, Université Grenoble Alpes
Prof. Alessandro Cilardo, Università degli Studi di Napoli Federico II
Prof. Gabriele Costa, Scuola IMT Alti Studi Lucca
Prof. Matteo Sonza Reorda, Politecnico di Torino

Politecnico di Torino
2023

Summary

The research work described in this dissertation is composed of elements from a path that began almost four years ago, investigating cybersecurity from a *foundation perspective*: device security is no longer seen as an ancillary feature that can be added, but as a foundational element of systems, which must already be designed to respond to an external threat, that is always assumed to exist. Based on my research group's and my department's extensive experience and tradition on the design of hardware and computing architectures, the work has delved into the broad (but recent) strand combining *security* with *hardware*. This intermingling generates two different facets, which nevertheless tend toward a single end: the design of hardware that is *defended* against attacks that directly target its physical and electronic nature, and the design of hardware that is *defending*, i.e., that by virtue of its security also enables the protection of the upper abstraction layers of computing systems, such as software and communications.

Hardware, like software, data, or communication infrastructure, must be designed, built, tested, used, and maintained while considering potential cyber attacks and their consequences. Indeed, hardware runs software and constitutes *the last line of defense*: if the hardware is compromised, any mechanism introduced to secure may be rendered ineffective. Unprotected hardware can be a weak point in the chain, providing at the same time an easy and powerful access point to the system functions and data.

The present research path, borrowing a well-established term in the culture of hardware design, investigates principles and methods of the design of *vulnerability-tolerant architectures*, i.e., platforms designed to ensure system security even in the presence of a vulnerability found, for example, in the software being executed. The concept traces the already well-known concept of *fault tolerance*, whereby the presence of a hardware fault does not cause the system to fail. In this dissertation,

the concept takes shape from a major software security problem for its prevalence and danger, namely the presence of memory corruption vulnerabilities, which enable the attacker to force malicious executions on the victim system, sometimes without even the need to inject code.

The target domain is *embedded systems*, i.e., the small electronic controllers that are hidden in the objects that surround our daily lives. Here, the effects of the dangerousness of such attacks are manifested to an even greater extent, as the hardware is often designed in such a way as to exclusively respond to functional needs, and with policies of maximum resource saving, leaving apart those features that could constitute a safeguard against attacks. Likewise, the code executed by such systems is largely written in languages, such as C or Assembly, which allow a deep optimization of resources, but at the same time open the way for the corruption vulnerabilities mentioned above. The goal is then to create sensitivity and knowledge about secure hardware design, particularly to address control flow corruption vulnerabilities: hardware architectures must assume such weaknesses as inescapably present in the software they will eventually run in their life cycle, and be able to block penetration attempts by construction.

The scientific contribution of this research work can be split in serveral parts. First, an important point of view on low-level computer security is introduced, with definitions and taxonomies that can be of special help in the hardware security field. It is defined what is meant by *threat*, *danger*, *weakness*, and *attack* in computer security, and what are the pillar properties on which security is based. The concept of *vulnerability* is detailed, and what are the categories within which to classify one of them. The *zero-trust* principle is discussed, and how a design that wants to be secure from the ground up and provide security to other domains must assume the presence of vulnerabilities and threats in the surrounding environment as certain. In the context of computer security, the role of hardware has been discussed, as the physical substrate of the systems from which they cannot disregard, and on which the design of a vulnerability-tolerant architecture is necessarily based.

Among all the security issues, the study focused on issues originating from the software these devices run, and opening the possibility for the attacker to corrupt the memory of these devices in such a way as to force them to execute sequences of unintended code. Thus, the so-called *binary attacks* have been discussed, their history, the classical defenses that the community has devised to defend itself, and

also how attackers have engineered the attack to escape them. The concept of *control-flow integrity* (CFI) was introduced as a security property, ensuring that a program only follows execution paths as originally designed for the mission of that system. The original philosophy of this concept, together its advancement and conjugation throughout the recent history of the scientific literature, have been discussed discussed. A taxonomy was given to the various CFI proposals, which distinguishes the various techniques based on the domain in which they are applied, the static or dynamic nature of their policies, the type of software instrumentation applied to the programs, and most importantly, the *granularity* of the checks that are performed to verify compliance with the flow integrity.

This has led to the presentation of the formal theory behind PROLEPSIS, that is the main scientific output of this dissertation. PROLEPSIS is a theoretical and practical framework for devising vulnerability-tolerant architectures for embedded systems, that guarantee control-flow integrity which is simultaneously *fine-grained*, *real-time* and *interrupt-aware*. With respect to what has been elaborated in the control-flow integrity research field, PROLEPSIS advances the knowledge importing a model which is completely abstract from practical details, such as the used computer architecture or monitor technology, while offering a maximum coverage against any code redirection, and granting minimal overhead on runtime, memory size and hardware resource occupancy. Using a formal language derived from some distinct precedents in the literature, the behavior of an embedded program is described, and how it can be hijacked to *illegal executions* through the intervention of an attacker. The concept of *tainted branch* is then introduced and defined, as a branch from which execution can escape from predefined paths. An algorithm for finding a *minimal* number of such branches within a program has been devised, as well as the behavior of a monitor that performs detection and inhibition of hijacking attempts at these points. Such a monitor is based on the classical *Policy Decision Point* (PDP) architecture, whereby *Policy Information Points* (PIP) and *Policy Enforcement Points* (PEP) are inserted within the program to be supervised. *Theorems* on which the operation of the monitor is based are formally demonstrated. A protected execution environment according to the rules of such a monitor corresponds to a *vulnerability-tolerant architecture*, at least for vulnerabilities originating control-flow attacks.

To give empirical strength to the formal findings, the monitor has been implemented targeting two examples of vulnerability-tolerant architecture. In the former one, the monitor is on an external hardware with respect to an ARM-based processor,

and receives control information necessary for verification; in the latter, the monitor represents an extension of a RISC-V processor, and is implemented internally. The technical details of the two architectures are presented, and based on scientifically established practices, performance parameters have been evaluated, in terms of binary size increase, additional time required for execution, and cost in terms of hardware resources. In the case of the external monitor, a 17.88% additional code base size has been measured, and a 1.10% additional execution time over a standard set of benchmarks, with a cost on the external reconfigurable hardware hosting the monitor logic below 3%. In the case of the internal monitor, we have got even better results, i.e., around 0.4-0.6% additional cost on all evaluation parameters. These results significantly outperform previous solutions, and provide evidence that the framework implementation can lead to solutions falling within the real-time requirements of the target domain. In both cases, such results are better than a group of reference solution identified for each type of monitor, where usually fine-grained solutions have double-digit percentage costs in all parameters, while coarse-grained solutions are more efficient but highly ineffective. Regarding practical security evaluation, the framework includes a new runtime evaluation benchmark based on established practices in the literature, but specifically designed for embedded architectures. The experimental results have shown 100% coverage on all considered attack cases, empirically demonstrating the theoretical conclusions reached by the elaboration of the formal theory.

The results of this dissertation can lead to architectural solutions with a clear degree of novelty compared to what has been scientifically elaborated so far. In addition to the abovementioned results, it is important to note that the framework abstracts from the analysis of software vulnerabilities that the architecture is then charged with executing. Many CFI solutions presented in the literature require some degree of *branch regulation*, i.e., to change the nature of control-flow transfers, or even the high-level code style itself, to reduce vulnerabilities. On the contrary, the advantage of PROLEPSIS is not to require any security analysis, but offering solutions that can be resilient to a precise attack pattern in all its variants. Among other advantages, it is also worth mentioning how it does not require any *pre-existing architectural support*. Everything required to implement protection is imported, without requiring anything from the architecture. On the contrary, many proposals require cryptographic extensions or modules such as the Last Branch Record (LBR) from which to read program activity. Above all, PROLEPSIS is not based on any

secret: even when the attacker gets the control-flow graph information, he cannot exploit it in any way to his advantage. The communication with the monitor has a precise protocol, and excluding the possibility for the attacker to inject or modify the code, there are no combinations of using instrumentation instructions to escape the security checks.

In the end, a final corollary part of the dissertation describes the research work done on a topic which is discussed centrally in the thesis, i.e., on *Education and Training* of Hardware Security. The importance of the topic stems from our experience with the main topics covered in this thesis, and the particular relevance of secure hardware design practices on which to train the next generation of experts. The Appendix details and shows the results of a study carried out in order to face the challenge of integrating this topic within the practical training paths specialized on cybersecurity, also in the context of relevant projects in the Italian and European contexts.