Ph.D. in Information Technology: Thesis Defenses

March 29th, 2018

DEIB Conference Room (building 20) – 9.30 am

Andrea CONTINELLA – XXX Cycle

"Defending from Financially-Motivated Software Abuses"

Advisor: Prof. **Stefano Zanero**

Abstract:

Software is involved in every aspect of our world, from our homes to large enterprises, and, in particular, it manages our data. As a consequence, software abuses can drastically impact our lives, for instance causing substantial financial losses or affecting people's privacy. This raised the attention of cybercriminals, who found in this scenario a lucrative business. In fact, in the past twenty years the motivation behind the cybercriminals' modus operandi has changed. No longer searching only for notoriety and fame, they have turned their attention to financial gain. Indeed malicious software, "malware," is one of the most dangerous Internet threat nowadays.

This dissertation details our research on the analysis and detection of the current software abuses, with the aim of protecting users from such threats. Specifically, we focus on three main threats, which have been the cause of billion dollars losses in the past years. First, we concentrate on information-stealing malware, also known as "banking Trojans." The purpose of these Trojans is to steal banking credentials and any other kind of private information by loading code in memory and hooking the network-related operating-system APIs used by web browsers. Second, we focus on a major class of malware, known as ransomware, which encrypts files, preventing legitimate access until a ransom is paid. Finally, we analyze the privacy issues in the mobile world by studying the problem of privacy leaks. Mobile apps are notorious for collecting a wealth of private information from users. Such information is particularly attractive. For instance, cybercriminals are known to sell users' private information on the underground markets, and advertisement libraries massively collect users' data to illicitly increase their profits.

Our contributions regarding banking Trojans focus on extracting robust, behavioral signatures of their malicious behavior, by combining web-page differential analysis and memory forensics techniques. The produced signatures can then be used, on the client side, to detect such Trojans in a more generic way, independently from their specific implementation, and protect victims' machines.

Our contributions regarding ransomware focus on designing behavioral detection models and proposing a novel defense mechanism to mitigate its effectiveness by equipping modern operating systems with practical

self-healing capabilities. We designed our detection models after an analysis of billions of low-level, I/O filesystem requests generated by thousands of benign applications, which we collected from clean machines in use by real users for about one month.

Our contributions regarding mobile privacy leaks focus on proposing a novel, obfuscation-resilient approach to detect privacy leaks by applying network differential analysis. To make differential analysis practical, our approach leverages a novel technique that performs root cause analysis of non-determinism in the network behavior of Android apps.

Alessandro DI FEDERICO – XXX Cycle

"Compiler Techniques for Binary Analysis and Hardening"

Advisor: Prof. **Giovanni Agosta**

Abstract:

Despite the growing popularity of interpreted or byte-compiled languages, C/C++ and other languages targeting native code are still dominantly used for system programming. Programs compiled to native code present a set of challenges compared to alternatives. In particular, in this work we focus on how they can be efficiently analyzed, how existing security measures (known as "binary hardening techniques") perform, and how new ones can be introduced to secure features that have received little attention. We propose rev.ng a binary analysis framework based on QEMU, a popular dynamic binary translator and emulator, and LLVM, a mature and flexible compiler framework. rev.ng can easily handle a large number of architectures and features a set of analyses to recover basic blocks locations, function boundaries and prototypes in an architecture- and ABI-independent way. rev.ng can be used for instrumentation, debugging, decompilation, retrofitting of security features and many more purposes. Our prototype encompasses about 17 kSLOC of C++ code and has been publicly released under a Free Software license. The core component of rev.ng is revamb: a static binary translator which can accurately identify all the basic blocks, and, in particular, the targets of indirect jumps for switch statements. Along this work, we will make heavy use of analysis techniques popular in the compiler literature, such as Monotone Frameworks, to recover an accurate control-flow graph, identify function boundaries and the number and location of function arguments and return values. We will also discuss how rev.ng can handle native dynamic libraries, how it can be easily employed for instrumentation purposes, how it can be extended to handle even more architectures and how its performance compares to tools with analogous purposes such as QEMU, Valgrind, Pin and angr. We also study two often overlooked features of C/C++ programs: variadic functions and the RELRO link-time

protection mechanism. We propose HexVASAN, a sanitizer for variadic functions to ensure that the number and type of arguments used by the variadic function match those passed by the caller, and leakless, an exploitation technique to bypass the RELRO protection in its several forms.

Pietro FEZZARDI – XXX Cycle

"Discrepancy Analysis: a Methodology for Automated Bug Detection in Hardware Designs Generated with High-Level Synthesis"

Advisor: Prof. **Fabrizio Ferrandi**

Abstract:

This thesis describes the definition, implementation, and evaluation of a methodology for automated bug detection, called Discrepancy Analysis, targeted at hardware designs generated with High-Level Synthesis. Discrepancy Analysis is based on a notion of equivalence between the execution of the hardware generated with High-Level Synthesis and the execution of the software obtained from the original high-level source code used to generate that hardware. Using this notion of equivalence, the thesis describes how to compare automatically the two executions, and how to detect and isolate the first mismatch if present. All these operations are executed without human interaction, relieving users from the timeconsuming and error-prone tasks to select the necessary signals for debugging, analyzing the signal traces to identify the malfunction, and backtracking it to the original high-level source code. The methodology is tightly integrated with the High-Level Synthesis process. As a consequence, it supports all compiler optimizations available during High- Level Synthesis. This coupling with the High-Level Synthesis tool also allows to automatically select in the generated designs the signals necessary for automated bug detection. Despite the tight coupling with the High-Level Synthesis tool, the discussion is kept as general as possible and only relies on common features that are present in all the known commercial and academic tools. The thesis also describes two extensions of Discrepancy Analysis: one to support automated bug detection in hardware generated with High-Level Synthesis of multithreaded code; one to support automated bug detection on pointers and memory accesses. Two bug detection flows based on Discrepancy Analysis are presented. The first is based on simulation of the hardware at the Register Transfer Level and performs the automated bug detection process offline after execution. The second flow is for on-chip bug detection. The generated hardware is instrumented with dedicated checker components, that analyze the execution on the fly, halting the circuit if a mismatch occurs and notifying it to users. Both the debug flows have been implemented and tested with BAMBU, an open source research framework for High-Level Synthesis developed at Politecnico di Milano.

The results have been evaluated in terms of performance, coverage, and other advantages brought to the overall debugging experience, like the considerable reduction of the size of the waveforms files that can be achieved with a heuristic for automated signal selection. This evaluation showed Discrepancy Analysis to be fast, accurate, and effective in identifying several different classes of bugs, coming from the original high-level code, from external libraries of components, and even subtle bugs injected by the High-Level Synthesis tool itself. A thorough and extensive analysis of these classes of bugs has been carried on, both on the baseline version and on the presented extensions for multithreaded code, for pointers, and for on-chip debugging. The technique used to compress the execution traces for On-Chip Discrepancy Analysis, based on Efficient Path Profiling, also showed reductions of the memory consumption necessary for on-chip debugging up to 95% compared to previous state-of-the-art.

Simone LIBUTTI – XXX Cycle

"Multicore Resource Management: a Horizontal Perspective"

Advisor: Prof. **William Fornaciari**

Abstract:

Modern computing systems strive to provide ever increasing performance levels despite increasingly strict system-wide optimization objectives. This is a vast problem that spans over a wide variety of architectures: due to the wild technological development caused by the spread of devices such as Smartphones, high-end embedded systems are quickly closing the gap with desktop computers; similarly, high performance and cloud-based systems are scaling up towards exascale to serve increasingly demanding workloads. Indeed, this technological trend poses several, nontrivial problems: embedded systems are usually subject to thermal and energy constraints in order to maximize battery life, to minimize faults and, at least in the case of hand-held devices, to provide a comfortable user experience, while bigger systems are typically subject to thermal and power constraints in order to minimize supplying and cooling costs and, again, to prevent faults. On the other hand, users do not care about system optimization objectives: they just want their applications to comply with some Quality of Service requirement.

This dissertation explores the problem of resource management from a horizontal perspective, by analyzing the problem of CPU resource management spanning from high-end embedded to High Performance Computing systems. For each of those architectures, we try to understand what is yet missing to obtain an optimal resource management and how we can fill some of those gaps.

Davide QUARTA – XXX Cycle

"Embedded System Security: Attacks, Impacts & Defenses"

Advisor: Prof. **Stefano Zanero**

Abstract:

Embedded systems and Internet-enabled devices are nowadays part of our everyday life, and play a huge role in raising the quality of our life: smartphones, home appliances, ICS (Industrial Control Systems), medical devices, and can be found even in cars.

The security of embedded systems have been studied and tested extensively giving birth to best practices, and considering the automotive environment for example, to standards that need to be applied in order to guarantee a baseline for the cyber-security of IT and OT (Operative Technology) systems that are part of a car. Yet, there is one class of embedded devices that is still lagging behind: Industrial Control Systems, in particular Industrial Robots. Up to now, there has been no in-depth, hands-on research that demonstrates to what extent robots can actually be compromised, and we hope our work will pave the way for more scrutiny and attention to their security. The powerful features that enriches currently available Industrial Robots, unlocks a broad attack surface, and creates novel venues for attacks. Depending on the actual setup and security posture of these devices, an attacker could trigger attacks with consequences ranging from massive financial damage, to affecting critical goods production, and up to jeopardizing the safety of human workers.

In this dissertation, we thoroughly analyze the attack surface of Industrial Robots, also considering the presence of typical IIoT (Industrial Internet of Things) devices associated with the robot, and user interaction. To guarantee confidentiality and integrity of the code running on the robot controller, we propose Tarnhelm, a system that allows for transparent execution of arbitrary parts of an unmodified application, while being flexible and providing confidentiality for the code. Our approach provides a portable and lightweight r^x primitive on ARM Trustzone, and it seamlessly applies also to consumer devices.

PhD Committee:

Prof. **Stefano Zanero**, DEIB – Politecnico di Milano

Prof. **Jeronimo Castrillon**, TU Dresden

Prof. **Alessandro Cilardo**, Università Federico II, Napoli