Avviso pubblico per la presentazione di Proposte di intervento per la realizzazione di attività di ricerca fondamentale relative al Partenariato Esteso SERICS (PE00000014), nell'ambito dello Spoke 6 Software and Platform Security (UNIVERSITA' CA' FOSCARI, VENEZIA) ammesso a finanziamento con l'Avviso Pubblico nr 341 del 15-02-2022 "Partenariati estesi alle università, ai centri di ricerca, alle aziende per il finanziamento di progetti di ricerca di base" – nell'ambito del Piano Nazionale di Ripresa e Resilienza, Missione 4 "Istruzione e ricerca" – Componente 2 "Dalla ricerca all'impresa" – Investimento 1.3, finanziato dall'Unione europea – NextGenerationEU

Allegato Tecnico

Critical services and all modern infrastructures heavily rely on software running on varieties of platforms. A vulnerability either in the software or in the underlying platform might enable Denial of Service (DoS) attacks, compromise data integrity and confidentiality and, in the worst case, allow remote code execution. Software and platform vulnerabilities are exploited by malware and are the root cause of many security incidents in ICT in a multitude of settings, from mobile devices to web and cloud-based applications. In critical ecosystems, these security incidents can have disastrous consequences, possibly involving financial loss and physical threat to people.

Spoke 6 is coordinated by UNIVE and brings together complementary initiatives to address the thematic line in its overall complexity. It relies on the implementation of the following project scopes (i.e., Ambiti Progettuali):

- SCAI - Supply Chain Attack Avoidance
- SWOPS - Securing softWare frOm first PrincipleS

SCAI explores innovative solutions to protect the software management and development process; SWOPS addresses the formal foundations of secure programming, to enable the implementation of secure-by-construction software systems.

Spoke 6 launches Open Calls / Bando a Cascata to address certain tasks foreseen in each of the aforementioned project scopes. For each project scope, this document introduces the corresponding set of tasks that have to be managed by the participants with their proposals.

The document is organized as follows. It first outlines the overall plan of the activities and milestones of Spoke 6. Then, it provides useful details about the projects included in Spoke 6. Finally, the section "Open Calls / Bando a Cascata" details the objectives of the tasks of each project that are the subject of this notice.

Plan of activities and milestones

Extended Partnership started activities on 1st Jan 2023 (M1). The duration of the project is 36 months.

The overall plan of the activities and milestones of the Spoke is summarized in the diagram shown in Figure 1. The figure details each type of activity, separated by horizontal bars, to which the partner exposes the costs of the project . The figure also displays the checkpoints at which the Spoke leader and partners must summarize the findings obtained in the corresponding period as vertical red lines. Analogously, after selecting the most -suited proposal for the Open Call corresponding to each project, the winning candidate should provide technical reports, one at each checkpoint in the diagram (red lines). The technical report will describe the findings obtained in the corresponding period and a software implementation of the best-performing techniques and best-suited method.

In addition, the winning candidate should monthly provide an update on the activities carried out on the ongoing Open Call.

Milestone	Description	2023				2024				2025			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
M1.0	Personnel recruitment												
M7.0	Research Phase 1 - Spoke 6												
M17.0	Research Phase 2 - Spoke 6												

Figure 1. Gantt diagram.

Milestones are set at the end of each period, when the results obtained in the last considered ones will be revised, and the work to be done in the subsequent period will be planned in light of the results obtained so far.

SCAI

Supply Chain Attack Avoidance

Partners

The project includes five SERICS research partners (IMT, UNIBA, UNIROMA1, UNISA, UNIVE), one industrial partner (DELOITTE), and will include other partners recruited through Open Calls (OC).

Abstract

Supply chain attacks are an increasingly popular approach to implement a variety of malicious goals. The project will explore innovative solutions to protect the software management and development process, to perform security tests through continuous dyna mic analyzes, and to protect software, detecting harmful activities and preventing or limiting their impact, along a self-defense paradigm.

WP Breakdown Structure

WP1- Securing the Software Process

WP Description: The goal of this work package is to address three aspects of the software process which may strengthen the security of the entire supply chain. The first one is the management of the development process which should be usable and at the same time secure (Task 1.1). The second one is a new quantitative evaluation of the ICT infrastructure security that includes the risk information (Task 1.2). The third one is the study of a methodology for facilitating the development of trustworthy systems composed of heterogeneous software components (Task 1.3).

Task 1.1- Usable security (UNIBA, UNIVE)

Task 1.2- Innovative techniques for evaluating cyber risk (see the section **Open Calls**)

Task 1.3- Innovative techniques for developing trustworthy systems (UNIBA, UNIVE, IMT, UNIROMA1, DELOITTE)

WP2 - Security Testing for Programs

WP Description: This work package will address software vulnerabilities in supply chain products, by investigating security testing strategies tailored for these products. The main directions will be to develop new API-level analyses (Task 2.1), new static and dynamic analyses (Task 2.2), and new fuzzing strategies (Task 2.3). We aim to both foundational and applied research on fuzz testing techniques that seamlessly integrate a wide spectrum of new methods. This will lay the groundwork for innovative platforms that will enable companies to continuously run fuzz tests right at their production sites.

Task 2.1 - API-level analysis (UNIVE, IMT, UNISA)

Task 2.2 - Innovative static and dynamic analyses for locating vulnerabilities (see the section **Open Calls**)

Task 2.3 - Innovative fuzz testing techniques (see the section **Open Calls**)

WP3 - Software Security and Protection

WP Description: In this work package we will investigate new techniques to protect software and detect malicious behaviors. We will investigate analysis and protection techniques (Task 3.1), runtime monitoring techniques (Task 3.2), static and dynamic analysis of mobile apps and devices (Task 3.3). The detection strategies will span from static analysis to runtime monitoring of software across different systems: mobile apps, IoT devices, cloud services, and desktop applications. The protection strategies will range from software hardening and watermarking techniques to runtime policies to correct or mitigate existing vulnerabilities which could be exploited to cause harmful and faulty behavior.

Task 3.1 - Software analysis and protection (UNIBA, UNIVE)

Task 3.2 - Runtime monitoring for malicious behavior detection (see the section **Open Calls**)

Task 3.3 -Security and Privacy of mobile apps and devices (see the section **Open Calls**)

SWOPS

Securing softWare frOm first PrincipleS

Partners

IMT, UNICA, UNIFI, UNIVE

Abstract

The complexity of software systems requires a major shift in perspective, in which software security is considered in the very early stages of the software life cycle. The project will address the formal foundations for revolutionizing secure programming to simplify the implementation of secure-by-construction software systems. More precisely, the project has three aims: the first is to develop semantic models and high -level programming abstractions that empower developers with the capabilities of writing robust and secure code from the first principle; the second one is to develop methods and techniques to analyze a piece of software both at a static time and at run -time in order to assess its security properties continuously, and assure that it is genuine, exposes an acceptable behavior, and is free from vulnerabilities; the third one is to contribute to the design, development, and implementation of infrastructure -level mechanisms that allow the secure execution and composition of software.

WP Breakdown Structure

WP1- Core Programming Languages

WP Description: This WP aims to define core languages with novel and high-level constructs for programming secure distributed and decentralized software systems. These core languages will allow software developers to deal with various programming paradigms and express security policies and properties for distributed systems, so to serve as the foundation for the methodologies developed in the project.

Task 1.1- Innovative techniques for secure programming (IMT, UNICA, UNIFI, UNIVE)

Task 1.2- Innovative abstractions for secure programming(see the section **Open Calls**)

WP2 - Language-Based Security Techniques

WP Description: This WP will develop formal methodologies for secure programming and techniques to reason about the security of programs implemented in the language defined in WP1. In particular, we will study static techniques for analyzing security prope rties; for the security properties of interest that cannot be ensured statically, we will provide techniques for the run-time instrumentation and dynamic verification.

Task 2.1 - Formal techniques for the run-time instrumentation and dynamic verification of high-level security properties (IMT, UNIVE, UNIFI, UNICA)

Task 2.2 - Program analysis techniques for verifying security properties (see the section **Open Calls**)

WP3 - Infrastructure and Run-time models

WP Description: This WP deals with the secure deployment and execution of software and services. In particular, it focuses on the secure compilation of software as a fundamental building block for the deployment of provably correct programs. This includes formal methods that support the review of the security properties of programs even when they are compiled and executed in different environments. Furthermore, this WP will consider the problem of secure service composition by investigating new methodologie s for ensuring that software units, each having peculiar security requirements and offering specific security guarantees, can be composed in a secure way.

Task 3.1 - Formal techniques for explainable and verified secure compilation (IMT, UNIVE)

Task 3.2 - Secure composition of micro-services (see the section **Open Calls**)

Open Calls / Bandi a Cascata

For each project scope (i.e., Ambito Progettuale) included in Spoke 6, the sections below provide a detailed breakdown of the tasks and their corresponding main objectives. The selected proposal must successfully complete the tasks outlined below, ensuring that the requirements and objectives of the project milestones are met.
Each proposal should focus on addressing a specific project scope. In the case of more than one partner participating in the same proposal for addressing the same project scope, each of them must clearly state their role, expected outcomes, and corresponding budget.
Additionally, the proposal has to plan the activities over time by producing a GANTT chart including milestones in accordance with the overall project GANTT reported in Figure 1 and respect the deadlines for documentation and software deliveries.

Project Scope: SCAI

WP1 Task 1.2- Innovative techniques for evaluating cyber risk

The main objective of this task is to provide new methods for assessing the cyber risk associated with the ICT infrastructures of the various suppliers of the modules used to build a given ICT system. The primary goal is to map risk information into a quantitative assessment of infrastructure security.

WP2 Task 2.2 - Innovative static and dynamic analyses for locating vulnerabilities

The main objective of this task is to exploit innovative combinations of static and dynamic analysis to help fuzzers locate untested regions of programs, and thus crucially counteract the undesirable early-saturation effect of current fuzz test methods.

WP2 Task 2.3 - Innovative fuzz testing techniques

The main objective of this task is to integrate fuzz testing methods with the ability of reusing knowledge about known attacks and field executions, to trigger program executions that depend on sophisticated inputs that the fuzzers are unlikely to automatically calculate.

WP3 Task 3.2 - Runtime monitoring for malicious behavior detection

The primary goal of this task is to develop techniques to extract malicious, harmful or unwanted behavior of a program from an instrumented sandbox. This will be accomplished by mapping

specific program behaviors to program execution patterns and checking them through an appropriate engine.

WP3 Task 3.3 - Security and Privacy of mobile apps and devices

The goal of this task is to design a novel approach that can identify the entry points of software that can be maliciously exploited by an attacker to inject code or to steal data, with particular focus on Android devices. This will be achieved by leveragi ng techniques, such as static and dynamic analysis, type systems and CFG analysis.

Project Scope: SWOPS

WP1 Task 1.2- Innovative abstractions for secure programming

This task will consider emerging programming abstractions that have appeared in the recent years. This may include programming paradigms for mobile code, distributed systems, serverless computing, and more. These abstractions will be studied to determine n ovel approaches for formally defining their behavior and security properties.

WP2 Task 2.2 - Program analysis techniques for verifying security properties

Languages provided with formal semantics are the cornerstone for applying rigorous analysis techniques. These techniques are mathematically grounded and they provide the highest guarantees since a program cannot violate the properties that were formally verified on it. This task will be in charge of studying the formal verification methodologies that can be applied to formal languages, such as those investigated in the context of WP1. Moreover, the proposed analysis techniques will be used for supporting the dynamic verification of Task 2.1..

WP3 Task 3.2 - Secure composition of micro -services

Micro-services are emerging as the leading paradigm for the orchestration and composition of software modules. Although formal verification techniques, such as those studied in this project, can guarantee the security properties of each component in isolat ion, their composition may invalidate part of the security guarantees. This Task will thus focus on investigating techniques that allow composing the security properties of several micro-services participating in a single, global orchestration.