VESSEDIA: Verification Engineering of Safety and Security Critical Industrial Applications Context of the project

Virgile Prévosto

VESSEDIA: Verification Engineering of Safety and Security Critical Industrial Applications

Context of the project

As shown by the growing number of vulnerabilities discovered in the code of many connected devices, as well as the emergence of high-level attacks exploiting those vulnerabilities, it becomes crucial to provide effective means to detect and fix as many such security holes as possible. While security concerns must be addressed at every single step of a system lifecycle, from its early design to its everyday use and maintenance, the VESSEDIA project (https://www.vessedia.eu) proposes to focus on two extremely sensitive phases, namely system modeling and software development. VESSEDIA started at the beginning of 2017, for a planned duration of 3 years, and is funded by the European Union's Horizon 2020 Programme, under grant number 731453. It is coordinated by Austria's Technikon, with France's CEA Tech List acting as technical coordinator. All in all, the project gathers 10 partners from 7 countries ranging from Finland to Spain:

- Technikon (Austria)
- CEA Tech List (France)
- Dassault Aviation (France)
- Search Labs (Hungary)
- Fraunhofer FOKUS (Germany)
- Inria Lille (France)
- Turku University of Applied Sciences (Finland)
- KU Leuven (Belgium)
- Fundacion Deusto (Spain)
- Amossys (France)

The main aim of the project is to propose a wide set of software and system analysis tools for helping security assessment of connected applications, specifically in the IoT domain. This builds upon existing source code analysis tools, such as Frama-C (https://frama-c.com) and in particular its plug-ins EVA and WP, or Verifast (https://github.com/verifast/verifast), and system modelling tools such as the Eclipse UML modeller Papyrus (https://www.eclipse.org/papyrus/). The project is mainly based on formal methods in order to give very strong guarantees on the absence of entire classes of weaknesses in the code under analysis.

Objectives

While formal methods have been successfully deployed for verifying safety properties of critical embedded software, extending their use to security properties in non-highly critical domains (that don't require stringent certification criteria, as is the case for aeronautics or nuclear power plants for instance) poses significant challenges.

First, the code under analysis routinely uses much more complex patterns than what is typically allowed by coding rules in critical domains and is meant to be used in a less well-controlled environment. Furthermore, some security properties, such as non-interference or integrity, demand a quite different setting than the verification of functional properties or the absence of run-time errors that is at the heart of Frama-C and Verifast.

Another important challenge is to ensure that formal verification tools can easily integrate into existing development toolchains and verification processes. This includes both providing a high-level description to specify the intended properties of the code as input to the tools, and the development of smart user interfaces that can integrate the results of various basic analyzers in a flexible way, in order to adapt to many operational settings.

Finally, a closely related concern consists in establishing strong links with existing standards, including Common Criteria (https://www.commoncriteriaportal.org/), CERT Secure coding rules (https://wiki.sei.cmu.edu/confluence/display/seccode), or the CWE list (https://cwe.mitre.org/data/index.html).

Technical Approach

In order to tackle these challenges, VESSEDIA is divided into 7 work-packages. First, WP1 will propose methodologies for verifying various kinds of safety and security properties. Its results will be used as input throughout the other work-packages for driving the development of the tool in appropriate directions.

Namely, the core of the project consists in extending the verification and validation tools involved in the project to new kinds of properties and code patterns for C and C++ languages. These development are addressed in WP2, whose final aim consists in providing an integrated toolbox dedicated to formal verification.

Similarly, WP3 will provide the toolbox users with high-level models for easily expressing their requirements at design level and tracing them during all phases of software development and verification.

In order to ensure a smooth integration within existing evaluation and certification processes, WP4 will propose metrics allowing a quantitative assessment of the security assurance requirements of the target of evaluation. This will also serve as an input for WP2 and WP3 to extend the tools so that they can provide appropriate information.

All tools and methods developed in the previous work-packages will be put to use into WP5's use cases, that target the Contiki OS (http://www.contiki-os.org/) dedicated to small connected devices, a firmware update protocol based on Contiki, and an aircraft maintenance system.

Dissemination of VESSEDIA's results will be covered by WP6, which notably include a standardization effort in order to propose an ISO standard for classifying code analysis tools.

Finally, WP7 is a classical management work-package, aimed at ensuring a smooth communication among all partners and with the European Commission.

Results

At the end of the project, we aim at proposing a powerful set of formal source code analysis tools capable of providing strong safety and security guarantees. By demonstrating the benefits of formal methods on a representative set of IoT applications, VESSEDIA will promote the usage of such techniques for security assessments, offering a key component in the protection of critical infrastructures. In addition, VESSEDIA toolset will include runtime verification techniques, allowing to take countermeasures in case formal proofs are unattainable for some parts of the code. These tools will also support quantitative measurements of the progress of the analyses in order to facilitate the evaluation of an application with respect to its intended security properties. Finally, the toolbox will bridge the gap between high-level security requirements, expressed at model level for the entire system, and verification activities done at implementation level (i.e. source code) for each component.

In order to achieve this goals by the end of 2019, a certain number of results have already materialized during the first year of the project, notably in the scope of WP1. First, the set of security requirements that are in the scope of the project has been precisely characterized in deliverable D1.1, together with some directions on how VESSEDIA tools could help assessing them. Similarly, a UML profile dedicated to modelling expected security properties of a system is being developed for the Papyrus UML editor. In addition, an early draft for an ISO standard categorizing software verification tools is being developed together with the relevant ISO working group (JTC1/SC7/WG4).

VESSEDIA tools have also seen several extensions, including notably more automated verification for the Verifast analyzer, an early prototype for the generation of formal code specifications verifiable by Frama-C from UML models using the Diversity symbolic execution engine of the Papyrus platform, new Frama-C plugins aimed at verifying well-established secure coding rules (CERT or CWE nomenclature), and a new tool for monitoring control-flow and data-flow integrity within the Linux kernel.

Finally, analysis of the use-cases has begun, with in particular some promising experiments with Frama-C/WP and Frama-C/EVA on Contiki, combination of static and dynamic analyses on the code for the aircraft maintenance system, and UML modelling as well as the use of Verifast on the firmware update use case. Feedback from WP5 also led to further proposals for improving the tools.

All in all, the first year of the project has seen very interesting developments, with promising research directions that bode well for the future of the project.