European Project Track Paper

VeriDevOps: Automated Protection and Prevention to Meet Security Requirements in DevOps

Andrey Sadovykh*‖, Gunnar Widforss†, Dragos Truscan‡, Eduard Paul Enoiu†,
Wissam Mallouli§, Rosa Iglesias¶, Alessandra Bagnto* and Olga Hendel†

*Softeam, France, {andrey.sadovykh, alessandra.bagnato}@softeam.fr
†Mälardalen University, Sweden, {gunnar.widforss, eduard.paul.enoiu, olga.hendel}@mdh.se
‡Åbo Akademi University, Finland, dragos.truscan@abo.fi
§Montimage EURL, France, wissam.mallouli@montimage.com
¶Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Spain, riglesias@ikerlan.es
‖Innopolis University, Russia, a.sadovykh@innopolis.ru

***Abstract*—Current software development practices are increasingly based on using both COTS and legacy components which make such systems prone to security vulnerabilities. The modern practice addressing ever changing conditions, DevOps, promotes frequent software deliveries, however, verification methods artifacts should be updated in a timely fashion to cope with the pace of the process. VeriDevOps, Horizon 2020 project, aims at providing a faster feedback loop for verifying the security requirements and other quality attributes of large scale cyber-physical systems. VeriDevOps focuses on optimizing the security verification activities, by automatically creating verifiable models directly from security requirements formulated in natural language, using these models to check security properties on design models and then generating artefacts such as, tests or monitors that can be used later in the DevOps process. The main drivers for these advances are: Natural Language Processing, a combined formal verification and model-based testing approach, and machine-learning-based security monitors. VeriDevOps is in its initial stage - the project started on 1.10.2020 and it will run for three years. In this paper we will present the major conceptual ideas behind the project approach as well as the organizational settings.**

***Index Terms*—Model-Driven Engineering, Cybersecurity, Security-by-design, Prevention and Reaction, Requirement, Design checking, Testing and validation, Runtime Analysis, Root cause analysis, Natural Language Processing, Machine Learning**

I. Introduction

The European Union Agency for Cybersecurity (ENISA) has outlined that the total number of software vulnerabilities has grown dramatically since 2002 [1]. In the past five year number of vulnerabilities has doubled and gone beyond 20 000. The security vulnerability reports are omnipresent in many application domains [2]. The average time to close or patch a vulnerability may reach 67 days [3], which leads to a high threat for organizations and may even threaten human life. Elaborated security mechanisms must be properly implemented prior to deployment in order to provide an effective level of protection against threats. The number of security scenarios to be ensured explodes. Moreover, in the embedded software domain, the number of system interactions with the environment that are subject to security attacks is increasing. This may result in security vulnerabilities that can cause losses for end-users as well as drastic increase in costs for both production and maintenance [4]. In such cases, traditional security verification approaches do not support continuous feedback loops. Furthermore, security quality attributes are often treated after code delivery or at the infrastructure level with specific patches [5] while it is generally agreed that those attributes must be addressed holistically at the design level (Security-by-design) [6].

Automation is an important technique [7] in today's agile software development, Continuous Delivery (CD) [8] pipelines and DevOps [9] practices. Current system development practices are increasingly based on using both commercial off-the-shelf and legacy components [10] which render such systems prone to security vulnerabilities.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957212. We acknowledge all the experts who contributed to numerous discussions and to writing the VeriDevOps proposal.

II. Objectives and Concepts

VeriDevOps aims at bringing together fast and cost-effective security verification through formal modelling and verification, as well as test generation, selection, execution and analysis capabilities to enable companies to deliver quality systems with confidence in a fast-paced DevOps environment.

Fig. 1 depicts the overall concept of the VeriDevOps project. Given an existing system under continuous integration/delivery, security requirements come in different forms. These can be standard requirements, such as those from ISA/IEC 62443 standard for control systems or description of vulnerabilities from common repositories, as well as reports from security experts. In all cases, these requirements should be immediately taken into account according to their severity and at all relevant levels. In this way, protection mechanisms such as firewalls and intelligent traffic monitors may be the first to be re-configured in order to avoid an immediate danger and secure the system perimeter. Next, the design of the system should be examined in order to locate the root-cause of the potential security breach and identify the remediation methods on code level as a patch or upgrade, at the design level, as a major redesign. The use of security requirements for protection and prevention suffers

978-3-9819263-5-4/DATE21/©2021 EDAA

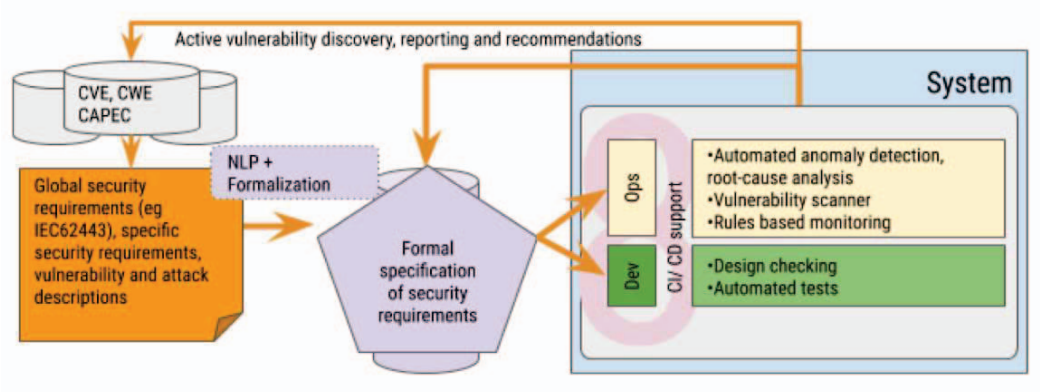

Fig. 1. Conceptual architecture of the VeriDevOps project

from limited automation support which is mostly limited to vulnerability scanners. There is still a tremendous amount of manual work to configure protection means at operations level and locate and prevent the vulnerabilities at design level, beyond the use of tools for scanning the libraries and tool chains used during the implementation. Despite the large volume of academic research on software testing and verification, there are relatively few commercial and industry-strength tools for security testing that require formal specifications of the system. In addition, the formalization of requirements is still a very human-intensive activity; much information is informally exchanged among the engineers and due to this, most verification activities cannot be automated and need human intervention. We argue that this formalization of security requirements and the creation of environment and system models could increase the product quality, and make the development and operation more efficient and less costly.

Thus, the key challenge of the project is to automatically express and manage security requirements in an effective and unambiguous way, such that both engineers and stakeholders have a common understanding of their content. Once these security requirements are unambiguously specified and decomposed, one needs to verify the compliance of the realizations to required security behavior by formal verification and testing for both protection and prevention means.

In order to save time and lower the effort for adjusting the prevention and protection mechanisms, VeriDevOps automates the specification and analysis of requirements with security relevance, testing of system realizations and the integration of these techniques and tools with current VeriDevops practices in industry:

1) *Analysis and formalization of a textual description of security requirements* from several sources. NLP and Patterns/boilerplates will be used for preventing inconsistencies and ambiguities from being introduced into the specification as well as the use of methods for automatic translation of patterns into temporal logic.
2) *Automated configuration of trace monitors based on formal specification of security requirements.* We automatically configure these traces over time, we monitor them continuously using formal specifications of security requirements and use runtime monitoring to detect various conflicts and vulnerabilities.
3) *Automated test generation for attacks based on formal specification of security requirements.* We will identify invalid states and conflicting security requirements in natural language requirement specifications and use this type of information to create negative tests, which attempt to force the system to enter invalid states, and can uncover vulnerabilities that may not be detected by positive test cases. One improvement would be to define guidelines and a format for testers to specify test scenarios that evaluate both security and energy properties, which are frequently neglected areas of testing.
4) *Automated design and code checks based on formal specification of security requirements.* This checking and analysis activities can be done by simulating the resulting model, or by formally verifying a description of the system bearing a formal semantics.
5) *Methods for threat detection and analysis*, as well as automated monitoring, vulnerability detection, root-cause analysis, and resilience mechanisms at operations level based on the identified security requirements.

VeriDevOps will also investigate the human and organizational factors that affect how systems are documented and, particularly, how the requirements are written and used during security testing. We will discuss these key ideas further in detail in the following sections.

III. TARGETED RESULTS OF VERIDEVOPS

The Veridevops project will provide several exploitable results, as follows:

A **methodology** for formal validation of security properties in DevOps context providing comprehensive guidelines for the adopters of VeriDevOps methods and tools. The methodology will also serve as a teaching material for IT-oriented universities. The methodology will be supported by several security catalogues, a Data Set and by the VeriDevOps Integrated Framework. The **Security Catalogues** will include:

- a Catalogue of common patterns for formal specification of security requirements;
- a Catalogue of resilience enablers for operations level including descriptions of attacks and responses with applicable instruments to detect and protect a system;
- a Catalogue of security recommendations that suggests countermeasures based on detected vulnerabilities and security flaws at development time.

The **Data Set** will contain labelled training data for the extraction and classification of security requirements.

Last but not least, the **VeriDevOps Integrated Framework** will incorporate tools for

1) security requirements extraction, classification and formalization,
2) anomaly detection supported by Artificial Intelligence for root cause analysis for events and problems at operations,
3) active prevention via advanced automated security testing from formal security specifications,
4) vulnerability localization and classification via root cause analysis at code and design level and
5) a THreat Oracle Engine (THOE) tool that gathers vulnerabilities, exploits and threat detection for industrial components. Finally, the project will also provide evaluation results of applying the VeriDevOps technologies to two complementary case studies, each targeting a set of typical scenarios from two different industrial domains.

IV. AMBITION

VeriDevOps *disrupts the traditional approaches on several levels by considering security holistically*, at all levels of the DevOps approach. First, it proposes the time gains by automatically translating security requirements to formal specifications. Second, the formal specifications contribute both at operations, as a means to trace the anomalies, and at development, as an enabler for automated verification.

To this extent, we are going to advance the state of the art by *tailoring formal verification of security requirements to DevOps and real-world CD pipelines*. We will use extended timed models for the formal specification of security requirements based on the semantic framework of stochastic timed automata. The availability of such models will allow us to take advantage of the UPPAAL tooling ecosystem for verification of the security model using either exhaustive model checking or scalable statistical model-checking. In addition, we will aim at generating security tests and runtime monitors from the verified specifications to be used at development and operations phases, respectively. Moreover, within VeriDevOps, we will identify patterns in verifying security properties and convert them into reusable template classes.

Furthermore, our ambition is focused on studying *knowledge extraction methods for formalization of the textual description into a security requirement by applying the modern Natural Language Processing models*, such as BERT for word embeddings. This question is not sufficiently addressed in the state-of-the-art research towards formalization of security requirements. VeriDevOps will use the weak (or programmatic) supervision approach which has not been studied before in complex domains such as security requirements verification.

In *vulnerability scanning*, we target a system represented by a collection of micro services that will process in real time a set of events in the environment and outside for detecting an incident or threat. These events will be related to detecting abnormal and suspicious activities, and to external published databases (i.e. for discovered vulnerabilities and exploits). For example, monitoring of application and operating system software, network traffic and human misbehaviour will be analysed, but also external sources of published vulnerabilities and exploits that help correlate with a reliable and updated asset database.

With respect to *security incidents detection and reaction*, VeriDevOps addresses updating the risk assessment at run-time based on data that is continuously collected through monitoring. VeriDevOps will present a model-based approach that makes use of measurable indicators at the system and network levels in order to obtain a risk picture that is continuously or periodically updated.

VeriDevOps targets to develop specific *test generation and selection methods based on formal specifications that can be used for model-based security* testing. These models that can be built and reused in different contexts; and contribute to empirically understand the return on investment. Building on standards and practices such as IEC 62443, VeriDevOps will develop a consolidated cybersecurity methodology incorporating the roles played in cybersecurity by people, processes and software technologies. In addition, we will develop new approaches for model-based mutation testing and fuzz testing by defining new model-level mutation operators suitable for testing security aspects.

V. CASE STUDIES

Two case studies will ensure VeriDevOps has a direct impact on the European Industry, as follows: 1) ABB's automation industry and operation of cranes and 2) FAGOR ARRASATE's manufacturing machines.

ABB Marine Ports design implement and commission automation solutions for container cranes in ports. These cranes can be fully autonomous and co-work in systems to load boats, self-driving trucks and trains in a highly intensive logistic system. The development process is lean and cost efficient. VeriDevOps is to further shorten the continuous integration and continuous delivery (CI/CD) cycles in DevOps by modelling and testing in simulation of a crane system in order to be able to perform tests at the developer's desk environment before commissioning to reduce time on site but also being able to test functionality, including the safety and security aspects, before the actual crane is manufactured. By using VeriDevops technologies, requirements for new or enhanced functionality of the standard platform have to be written in such a way that they are maintainable for future extensions but also fit test cases for software module testing and overall regression test. FAGOR ARRASATE's case study requires high-quality reliable software for manufacturing machines to serve customers coming from highly demanding sectors, such as

leading automotive manufacturers, stampers, home appliances or metallic furniture producers. Security concerns have been further accentuated due to digitalization of their machines via Edge and Cloud infrastructures. Besides, the new challenge is to combine cybersecurity with CI/CD practices.

VeriDevOps adopts a case study-driven approach that involves end-users from the beginning, so that the resulting tools are suitable for the needs of the end users. In order to foster project cooperation among partners and stimulate the incubation of new ideas and technologies, we are going to use regular project level hackathons. We have applied a similar approach in previous large European projects with very satisfactory results [11].

The results of the project will be disseminated via academic publications submitted to conferences and journals, via the project web site (http://www.veridevops.eu), by keeping active social media channels, by attending relevant industrial events and fairs, and by producing high-quality and comprehensive marketing material such as posters, videos or brochures.

Standardisation efforts will be mostly focused at raising awareness in standardisation circles and possibly, providing some information needed to implement the project correctly from the standards' point of view. These activities will rely on the experience of the project participants which are active in numerous standardization bodies and activities regarding security, membership of European Cyber Security Organisation (ECSO) and ISO/IEC-62443, and software development methodologies, such as Object Management Group (OMG).

VI. CONSORTIUM

The VeriDevOps consortium combines comprehensive RD as well as technical expertise by Softeam (France), Åbo Akademi University (Finland), Mälardalen University (Sweden), Montimage (France) and Ikerlan (Spain) in Requirements Engineering, NLP, Formal specifications, Model checking, Model-based testing, Traffic and traces analysis, Security and Safety of industrial systems, as well as in other topics that are of strategic importance for the successful execution of the project. Most partners are either dedicated leaders of major scientific and open-source software communities or are active contributors to the industrial-value software stacks. The consortium is thus well-balanced from the expertise coverage perspective and is well-adapted to the changes in the IT environment. Moreover, the industrial partners (in the project, ABB AB (Sweden) and Fagor Arrasate S. Coop. (Spain) are expected to ensure a proper reflection of the industrial requirements in the project and will work towards a fast take-up of the project's outcomes by their respective user communities.

VII. CONCLUSIONS

VeriDevOps is a European collaborative research project that tackles problems of securing modern industrial systems both on design and runtime sides. We have presented our initial concepts on workflow starting from natural language processing for automated creation of formal security specifications, which serve to fulfill multiple prevention and protection activities, such as automated model-based testing or threat and root cause analysis. As a result, the VeriDevOps framework will enable more secure industrial systems. The framework will be validated in two complementary industrial case studies. The framework is planned to contribute to the open-source community, while its adoption and exploitation will be reinforced by the endorsement of the case study partners. VeriDevOps is driven by a multidisciplinary consortium with proven leadership in software development, security engineering, as well as in the domains of industrial control systems and smart manufacturing.

REFERENCES

[1] The European Union Agency for Cybersecurity (ENISA). Is software more vulnerable today? https://www.enisa.europa.eu/publications/info-notes/is-software-more-vulnerable-today, March 2018. Accessed: 2020-10-29.

[2] Accenture, 2020 Cyber Threatscape Report. https://www.accenture.com/_acnmedia/PDF136/Accenture2020-CyberThreatscapeFullReport.pdf. Accessed: 2020-10-29.

[3] EDGESCAN, 2018 VULNERABILITY STATISTICS REPORT. https://www.edgescan.com/wp-content/uploads/2018/05/edgescan-stats-report-2018.pdf. Accessed: 2020-10-29.

[4] D. Papp, Z. Ma, and L. Buttyan. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pages 145–152, 2015.

[5] M. Zhivich and R. K. Cunningham. The real cost of software errors. *IEEE Security Privacy*, 7(2):87–90, 2009.

[6] J. L. Bayuk. Systems security engineering. *IEEE Security Privacy*, 9(02):72–74, mar 2011.

[7] Matthias Tichy, Michael Goedicke, Jan Bosch, and Brian Fitzgerald. Rapid continuous software engineering. *J. Syst. Softw.*, 133:159, 2017.

[8] Jez Humble and David Farley. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Adobe Reader)*. Pearson Education, July 2010.

[9] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, May 2015.

[10] T. Sedano, P. Ralph, and C. Péraire. Software development waste. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 130–140, 2017.

[11] Andrey Sadovykh, Dragos Truscan, Pierluigi Pierini, Gunnar Widforss, Adnan Ashraf, Hugo Bruneliere, Pavel Smrz, Alessandra Bagnato, Wasif Afzal, and Alexandra Espinosa Hortelano. On the use of hackathons to enhance collaboration in large collaborative projects : - a preliminary case study of the MegaM@Rt2 EU project -, 2019.