ercim-news.ercim.eu

Number 139 October 2024

ERCIM NEWS

01001100011100000000100

11011011110101010000101

1001SPECIAL011100101000

001001100001101THEME011

11010010101001100010000

00010001000000000011100

1011SOFTWARE01001111011

01100011011101101110111

00010000SECURITY1110010

10111100001010000001001

10111010111001010111010

Also in this issue’s Research and Innovation section:
Sustainable Agriculture through Industry 5.0 Technologies

Editorial Information

ERCIM News is the magazine of ERCIM. Published quarterly, it reports on joint actions of the ERCIM partners, and aims to reflect the contribution made by ERCIM to the European Community in Information Technology and Applied Mathematics. Through short articles and news items, it provides a forum for the exchange of information between the institutes and also with the wider scientific community. This issue has a circulation of about 2,000 printed copies and is also available online ,at https://ercim-news@ercim.eu.

ERCIM News is published by ERCIM EEIG
BP 93, F-06902 Sophia Antipolis Cedex, France
+33 4 9238 5010, contact@ercim.eu
Director: Dominique Hazaël-Massieux, ISSN 0926-4981

Contributions
Contributions should be submitted to the local editor of your country

ERCIM News online edition: *https://ercim-news.ercim.eu/*

Next issue:
January 2025: Large-Scale Data Analytics

Subscription
Subscribe to ERCIM News by sending an email to en-subscriptions@ercim.eu

JOINT ERCIM ACTIONS

SPECIAL THEME

Introduction to the Special Theme

Artificial Intelligence for Software Protection

Architectural Security and Risk Analysis

Code Analysis

Hardware-Assisted Protections

RESARCH AND INNOVATION

ANNOUNCEMENTS / IN BRIEF

29th International Conference on Formal Methods for Industrial Critical Systems (FMICS'24)

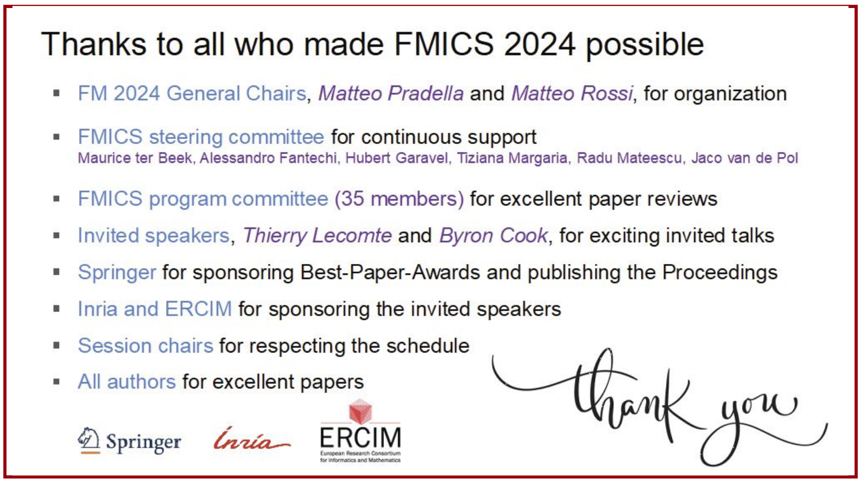
by Maurice ter Beek (CNR-ISTI)

The yearly conference of the ERCIM Working Group on Formal Methods for Industrial Critical Systems, FMICS [L1], the key conference at the intersection of industrial applications and formal methods, reached its 29th edition. This year the participants met in Milan, Italy, during 9-11 September 2024.

The aim of the FMICS conference series is to provide a forum for researchers and practitioners interested in the development and application of formal methods in industry. It strives to promote research and development for improving formal methods and tools for industrial applications.

FMICS 2024 [L2] was chaired by Anne Haxthausen (Technical University of Denmark, Lyngby) and Wendelin Serwe (Inria Grenoble, France) and organised by the general chairs Matteo Pradella and Matteo Rossi (Politecnico di Milano, Italy) and their team as a co-located event of FM 2024, the 26th International Symposium on Formal Methods. As part of this co-location, FM and FMICS organised a joint Industry Day on 11 September. This is one of the reasons for which FMICS 2024 overall attracted about 100 participants from many countries worldwide, from academia as well as industry, making this a very well attended edition.

The international program committee, with 35 members from 15 different countries, received 22 submissions and decided to accept 14 papers after a rigorous reviewing process. The program moreover included two excellent invited keynote presentations, namely “The Business of Proof” by Byron Cook (Amazon Web Services, UK) and “B+ or how to model system properties in a formal software model” by Thierry Lecomte (Clearsy, France), both of which attracted many participants also from FM and from the other co-located conferences FACS, LOPSTR, PPDP and TAP. ERCIM and Inria generously sponsored the invited speakers (Figure 1).



Thanks to all who made FMICS 2024 possible

- FM 2024 General Chairs, *Matteo Pradella* and *Matteo Rossi*, for organization
- FMICS steering committee for continuous support
  Maurice ter Beek, Alessandro Fantechi, Hubert Garavel, Tiziana Margaria, Radu Mateescu, Jaco van de Pol
- FMICS program committee (35 members) for excellent paper reviews
- Invited speakers, *Thierry Lecomte* and *Byron Cook*, for exciting invited talks
- Springer for sponsoring Best-Paper-Awards and publishing the Proceedings
- Inria and ERCIM for sponsoring the invited speakers
- Session chairs for respecting the schedule
- All authors for excellent papers

thank you

Figure 1: FMICS 2024 PC chairs Anne Haxthausen and Wendelin Serwe acknowledged ERCIM sponsoring of the invited speakers during the conference opening presentation.

Figure 2: Jan Steffen Becker received the FMICS 2024 Best Paper Award from the PC chairs Anne Haxthausen and Wendelin Serwe.

Figure 3: John Hatcliff received the FMICS 2024 Best Tool Paper Award from the PC chairs Anne Haxthausen and Wendelin Serwe.

Following a tradition established over the years, Springer sponsored the FMICS best paper awards. This year, the program committee selected the contributions “Safe Linear Encoding of Vehicle Dynamics for the Instantiation of Abstract Scenarios” by Jan Steffen Becker (German Aerospace Center, Oldenburg) as the FMICS 2024 Best Paper (Figure 2) and “Logika: The Sireum Verification Framework” by Robby, John Hatcliff and Jason Belt (Kansas State University, USA) as the FMICS 2024 Best Tool Paper (Figure 3).

FMICS 2025 will take place in Aarhus, Denmark, under the CONFEST 2025 umbrella, alongside CONCUR, FORMATS and QEST, and organised by the general chairs Jaco van de Pol and Andreas Pavlogiannis (University of Aarhus, Denmark) and their team during 25-30 August 25-30.

Links:
[L1] https://fmics.inria.fr/
[L2] https://fmics.inria.fr/2024/

Reference:

[1] A.E. Haxthausen and W. Serwe (eds.), Formal Methods for Industrial Critical Systems: Proceedings of the 29th International Conference on Formal Methods for Industrial Critical Systems (FMICS'24), Milan, Italy, 9-11 September 2024. Lecture Notes in Computer Science, volume 14952, Springer, Cham, 2024. DOI: https://link.springer.com/book/10.1007/978-3-031-68150-9

Please contact:
Maurice ter Beek, CNR-ISTI, maurice.terbeek@isti.cnr.it

12th International Workshop on Computational Intelligence for Multimedia Understanding

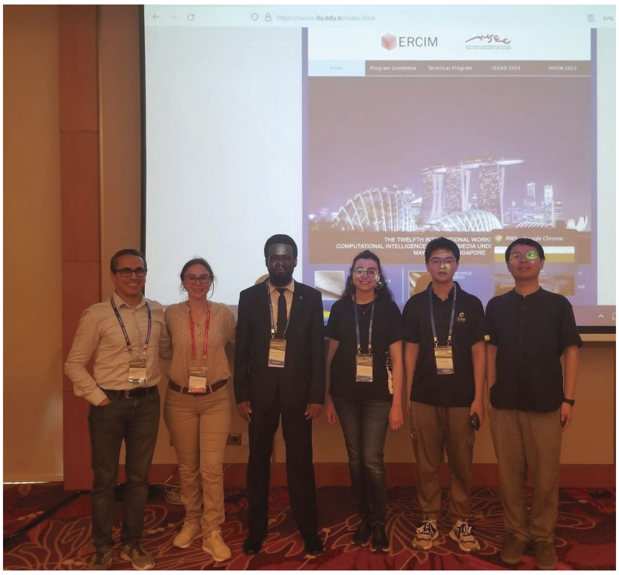
by Behçet Uğur Töreyin (İTÜ), Maria Trocan (ISEP) and Davide Moroni (CNR-ISTI)

Approximately forty researchers attended the International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), organized annually by the ERCIM Working Group Multimedia Understanding through Semantics, Computation and Learning (MUSCLE). The workshop took place as a satellite event of IEEE ISCAS 2024 held in Singapore on 21 May 2024.

Multimedia understanding is an essential part of many intelligent applications in our daily lives, whether in our households or commercial, industrial, service, and scientific environments. Analyzing data acquired from a multitude of multimodal sensors to provide them with semantics is essential to exploit their full potential. Multimodal and cross-modal analyses are essential for maximizing their utility.

Through IWCIM, the MUSCLE Working Group [L2] aims to address these emergent topics by growing a community of scientists and practitioners from the academy and industry. The mission continues with the planning of the upcoming thirteenth IWCIM, details of which will be announced shortly.

The IWCIM 2024 website is hosted by Istanbul Technical University [L1], and full-text papers may be accessed via IEEE Xplore.



Highlights from the venue.

We are excited to announce that the next edition will be featured as a special session during IEEE ISCAS 2025, which will take place in London from May 25th to 28th, 2025.

Links:
[L1] https://iwcim.itu.edu.tr/technical.html
[L2] https://wiki.ercim.eu/wg/MUSCLE/

Please contact:
Davide Moroni, CNR-ISTI, davide.moroni@isti.cnr.it

ERCIM “Alain Bensoussan” Fellowship Programme

The ERCIM postdoctoral Fellowship Programme has been established as one of the premier activities of ERCIM. The programme is open to young researchers from all over the world. It focuses on a broad range of fields in Computer Science and Mathematics.

> “ERCIM postdoctoral fellowship is an unmatched opportunity to sharpen the skill set and knowledge while being in the research conducive environment of Europe. This fellowship has immense potential to provide accelerated career growth, increase research cooperation and knowledge sharing by fostering collaborations.
>
> **Amandeep CHEEMA**
> Former ERCIM Fellow

The fellowship scheme also helps young scientists to improve their knowledge of European research structures and networks and to gain more insight into the working conditions of leading European research institutions. The fellowships are of 12 months duration (with a possible extension), spent in one of the ERCIM member institutes.

Where are the fellows hosted?

Only ERCIM members can host fellows. When an ERCIM member is a consortium the hosting institute might be any of the consortium’s members. When an ERCIM Member is a funding organisation, the hosting institute might be any of their affiliates. Fellowships are proposed according to the needs of the member institutes and the available funding.

The fellows are appointed either by a stipend (an agreement for a research training programme) or a working contract. The type of contract and the monthly allowance/salary depends on the hosting institute.

ERCIM encourages both researchers from academic institutions and scientists working in industry to apply.

Why to apply for an ERCIM Fellowship?

The Fellowship Programme enables bright young scientists from all over the world to work on a challenging problem as fellows of leading European research centers. In addition, an ERCIM fellowship helps widen and intensify the network of personal relations and understanding among scientists. The programme offers the opportunity to ERCIM fellows:

- to work with internationally recognized experts,
- to improve their knowledge about European research structures and networks,
- to become familiarized with working conditions in leading European research centres,
- to promote cross-fertilization and cooperation, through the fellowships, between research groups working in similar areas in different laboratories.

Equal Opportunities

ERCIM is committed to ensuring equal opportunities and promoting diversity. People seeking fellowship within the ERCIM consortium are not discriminated against because race, color, religion, gender, national origin, age, marital status or disability.

Conditions

Candidates must:

- have obtained a PhD degree during the last eight years (prior to the application year deadline) or be in the last year of the thesis work with an outstanding academic record. Before starting the grant, a proof of the PhD degree will be requested;
- be fluent in English.

Application deadlines

Deadlines for applications are currently 30 April and 30 September each year.

Since its inception in 1991, more than 790 fellows have participated in the program. In 2023, 19 young scientists began an ERCIM PhD fellowship, and throughout the year, 63 fellows were hosted. The Fellowship Program is named in honor of Alain Bensoussan, the former president of Inria, one of the three founding institutes of ERCIM.

http://fellowship.ercim.eu

> “ERCIM allowed me to be part of a world-leading research institution while maintaining the independence needed to develop a personal and distinct research profile. The fellowship gave me the chance to significantly expand my personal network by participating to various scientific events all over the world.
>
> **Giacomo SORELLI**
> Former ERCIM Fellow

Nicola Messina Winner of the 2024 Cor Baayen Award

Nicola's research is distinguished by its high quality, interdisciplinary approach, and significant impact. His work spans across multiple domains, including Artificial Intelligence, Computer Vision, Deep Learning, and Multimedia Information Retrieval. He has made substantial contributions to both the theoretical and applied aspects of these fields, with measurable scientific and practical outcomes.

During his PhD, Nicola began exploring the ability of neural networks to understand and process relationships between objects in computer vision. He also made important contributions in multimodal artificial intelligence by developing innovative and efficient methods for aligning representations of complex neural networks in both visual and language modalities.

His work in multimedia information retrieval has demonstrated immediate real-world applicability. This is exemplified by VISIONE, a large-scale video search system that won the 2024 international Video Browser Showdown competition and secured second place in 2023. VISIONE has been employed by the Italian national public broadcaster RAI, as part of the AI4Media European project, to facilitate efficient browsing of audiovisual archives.

Nicola Messina (left) receives the Cor Baayen Award from ERCIM President Björn Levin at the ERCIM Fall Meetings in Budapest on 16 October 2024.

More recently, he has expanded his research focus to include the application of attentive deep learning techniques for structural health monitoring and the preservation of cultural heritage. This new direction highlights his ability to engage in highly interdisciplinary research and to address complex challenges across different fields.

Nicola's professional experience also includes international collaborations with esteemed European universities and participation in various Italian and European research projects, such as AI4Media, AI4EU, INAROS, AI4CHSites, ADA, and Smart News. In addition to his research, he is involved in teaching and dissemination activities, offering instruction and guidance to students on topics related to deep learning, computer vision, and multimodal processing.

In 2021 and 2022, Dr. Messina received the ISTI Young Researcher Award, recognizing him as one of the top young researchers (under 32 years old) at CNR-ISTI, Italy.

Overall, Dr. Nicola Messina is an outstanding early-career researcher who has achieved remarkable success during his PhD and postdoctoral work, combining scientific innovation with societal and commercial impact.

Nicola Messina is currently employed at the Information Science and Technologies Institute, National Research Council (CNR-ISTI), Italy, where he is associated with the Artificial Intelligence for Multimedia and Humanities Laboratory (AIMH). He received his PhD in 2022 from the University of Pisa, Italy, with a thesis titled "Relational Learning in Computer Vision," supervised by Fabrizio Falchi, Giuseppe Amato, and Marco Avvenuti.

More information about the Cor Baayen Award:
https://kwz.me/hDG

Cor Baayen Early Career Researcher Award 2024

Winner:
Nicola Messina, nominated by Fabrizio Falchi, Information Science and Technologies Institute, National Research Council (CNR-ISTI), Italy

Honorary mentions:
- Jana Wagemaker, nominated by CWI, The Netherlands
- David Saulpic, nominated by Inria, France
- Błażj Osiński, nominated by University of Warsaw, Poland

Finalists:
- Lesly-Ann Daniel, nominated by INRIA, France
- Diogo Barradas, nominated by INESC, Portugal
- António Correia, nominated by INESC, Portugal
- Agustin Zuniga Corrales, nominated by VTT, Finland
- Shadi Shafighi, nominated by University of Warsaw, Poland

Selection Committee:
The Selection Committee was composed of Gabriel David (INESC-TEC), Monica Divitini (NTNU – chair of the ERCIM Human Capital Task Group), Anna Gambin (University of Warsaw), Georgia Kapitsaki (University of Cyprus), Bruno Levy (Inria), Kostas Magoutis (FORTH), and Fabrizio Sebastiani (CNR-ISTI).

The decision was unanimous.

Introduction to the special theme

Software Security

by Sebastian Schrittwieser (University of Vienna) and, Michele Ianni (University of Calabria)

Software is gaining unprecedented importance in many industries. The automotive sector is a prime example of this massive change: once primarily confined to embedded systems such as engine control units, software now serves as the central interface for almost all vehicle components. Features such as advanced driver assistance systems, infotainment and connectivity services all rely heavily on software. In addition, cost considerations are driving the replacement of hardware components with software equivalents - from analogue switches and buttons being replaced by a central touchscreen with software-based controls, to dedicated hardware sensors such as LIDAR being replaced by vision-based artificial intelligence (AI). This shift not only reduces manufacturing costs, but also enables entirely new business models. Concepts such as over-the-air updates, paid activation of modular features and subscription models are only possible through software-centric approaches.

However, this increased reliance on software brings new security challenges. In so-called Man-At-The-End (MATE) attacks, reverse engineers have full control over the systems on which they execute, analyze, and modify the targeted software. Such attacks pose significant threats to new business models, including unauthorized activation of functionality, software piracy, and intellectual property infringement. Software integrity is also critical. Modified software can lead to unintended side effects, potentially compromising human safety in critical situations. Fundamentally, ensuring robust software security to protect against analysis and modification is not only a business imperative, but also a safety requirement.

Similar to many other domains, AI is already having a significant impact on software security. Large Language Models (LLMs), such as GPT-4, are improving code analysis and semantic understanding of software, leading to remarkable improvements in both speed and quality. These models can automatically generate code documentation, identify vulnerabilities, and suggest optimisations. While research is currently focused on code analysis, the field of software protection is poised to benefit significantly from AI advances in the coming years as well. AI can be used to develop more sophisticated protection strategies, including code diversification - syntactic variations of software copies to prevent so-called class breaks, where attacking one instance allows adversaries to compromise all other instances as well.

However, AI-driven advances in software security do not come without challenges. Obfuscation, for example, is a widely used technique that AI could enhance in the future. Obfuscation aims to make code harder to understand by transforming its structure, adding irrelevant code, increasing the complexity of the control flow, and encrypting data. The primary goal is to prevent reverse engineering and protect intellectual property. Despite its usefulness in securing software, obfuscation also presents significant challenges. While it provides protection by hiding the inner workings of code, attackers, including malware developers, can weaponise obfuscation techniques. Bad actors can use the same strategies to hide malware, making it harder for security systems to detect and for analysts to understand the intent of the code. Obfuscated malware can evade signature-based detection mechanisms and slow down reverse engineering efforts, creating a dangerous gap in software security.

Similarly, watermarking is another key technique for protecting software, particularly in the fight against piracy and unauthorized copying of programs. By embedding invisible marks in software, developers can trace the origin of pirated copies or unauthorized modifications. This can be crucial in legal disputes or intellectual property enforcement. However, like obfuscation, watermarking is not immune to abuse. In malicious hands, watermarking can be exploited to make compromised software appear to belong to a legitimate source, complicating the attribution process and potentially implicating innocent parties. This illustrates the double-edged nature of many software protection strategies: while they are essential for securing legitimate software, they also present new challenges when adopted by adversaries.

The risks associated with software exploitation are not limited to intellectual property theft or software piracy. Exploitation can lead to serious vulnerabilities that threaten data security, financial stability of companies, and even human safety. A critical problem is the exploitation of zero-day vulnerabilities - previously unknown flaws in software that can be exploited before developers can patch them. These vulnerabilities can be sold or shared among attackers, making them extremely difficult to defend against in real time. Software exploitation, particularly in critical sectors such as automotive, medical or finance, can have catastrophic consequences if attackers are able to manipulate the software to perform unintended actions.

Mitigating these risks requires a multi-layered approach in software security. Patching software and regularly addressing known vulnerabilities is a basic defense mechanism. In addition, adopting secure coding practices, where software is designed with security in mind from the outset, can help reduce the attack surface. AI-driven solutions also play an important role in identifying vulnerabilities earlier, allowing for faster patching and response. For example, AI models can sift through massive code bases and pinpoint areas that are likely to contain vulnerabilities based on past patterns. This automation allows developers to focus their efforts on the most critical parts of the software.

One area that remains particularly vulnerable is the Internet of Things (IoT). IoT devices, from smart home appliances to industrial sensors, are notoriously difficult to secure. Many IoT devices are low-cost and designed with limited computing power, which means they lack the robust security features of more sophisticated systems. Moreover, these devices are often not regularly updated, leaving them vulnerable to known vulnerabilities long after they have been discovered. IoT devices are a prime target for attackers because, once compromised, they can be used as an entry point into wider networks or reused in large-scale attacks such as distributed denial of service (DDoS) campaigns. The security challenges associated with IoT are exacerbated by the decentralized nature of these devices and their widespread deployment. Securing the IoT requires cross-industry collaboration, with standards, regulations and lightweight security measures that can be effectively implemented on resource-constrained devices.

Analyzing software for vulnerabilities relies on a variety of techniques. Static analysis, which examines code without executing it, can help identify potential vulnerabilities by analyzing code structure, syntax and control flow. Dynamic analysis, on the other hand, observes software in action and detects runtime vulnerabilities by observing how the software behaves under different conditions. Combining these approaches with techniques such as fuzzing - where random inputs are thrown at the software to discover unexpected behavior - can help uncover vulnerabilities that would otherwise remain hidden. AI and machine learning models are increasingly being used to enhance both static and dynamic analysis, speeding up the process and improving accuracy by detecting subtle signs of potential threats that might elude human analysts.

From a European perspective, the approach to software security and intellectual property protection presents its own unique set of challenges and opportunities. Europe has always placed a strong emphasis on privacy and data protection, as evidenced by regulations such as the General Data Protection Regulation (GDPR). However, while the region leads the way on privacy, it lags behind the US, Japan and China in most areas of software innovation. Europe needs to increase its focus on defending against MATE attacks and protecting intellectual property, especially as software becomes central to new business models. The example of Skype - a European-developed VoIP software that became a market leader primarily through its innovative technology, but also because of its superior software protection which prevented clones for many years - illustrates how robust security practices can provide a competitive advantage. However, to maintain this edge, Europe must continue to evolve its software security strategies, especially as AI-driven threats become more sophisticated.

Three key EU legislative frameworks— the Digital Operational Resilience Act (DORA), the Network and Information Security Directive 2 (NIS2), and the Cyber Resilience Act (CRA)—have been established to enhance cybersecurity and operational resilience across various sectors. Each of these initiatives emphasizes stringent security requirements for software, including secure development practices, lifecycle management, and risk assessments. Collectively, these frameworks push organizations to adopt more secure software development practices, invest in advanced security technologies, and ensure continuous compliance with evolving regulations. By embedding security into every stage of the development process, organizations not only comply with these regulations but also improve their overall resilience against cyber threats, positioning themselves to navigate the increasingly complex cybersecurity environment effectively. While these regulations represent a crucial step toward enhancing software security, it is more important than ever for organizations to prioritize it. Doing so is essential for safeguarding digital assets and maintaining resilience in an increasingly interconnected world.

The articles in this special theme section offer a comprehensive panorama of the current European research activities in software security and protection. By showcasing a diverse range of studies and innovative approaches, they highlight the ongoing advancements and key developments shaping the future of the field.

Please contact:
Sebastian Schrittwieser
sebastian.schrittwieser@univie.ac.at
University of Vienna, Austria

Michele Ianni
University of Calabria, Italy
michele.ianni@unical.it

Next Generation Vulnerability Detection with LLMs

by Mila Dalla Preda, Niccolò Marastoni, Federica Paci (University of Verona)

Ensuring software security starts with detecting vulnerabilities in the early stages of development: while traditional rule-based and machine-learning methods require expert input, Large Language Models (LLMs) are emerging as powerful, autonomous alternatives that could transform the approach to vulnerability detection.

The detection of vulnerabilities in source code during the early stages of development is crucial for ensuring robust software systems that can resist cybersecurity threats. This is especially true as the number of reported vulnerabilities is dramatically rising every year (see Figure 1). In recent years, several approaches have been proposed to identify security vulnerabilities, but each has its own limitations. Rule-based methods require expert intervention to define rules or patterns indicative of known vulnerabilities. Moreover, the rules must be adjusted to detect new or unknown vulnerabilities. Recently, deep learning-based approaches have emerged as a promising alternative, capable of automatically learning vulnerability patterns without direct expert involvement [1].

Large Language Models (LLMs) have demonstrated significant potential in interpreting and generating code, suggesting the possibility of using them to detect vulnerabilities in code [2]. In this paper, we present the project VULCAN, which aims to investigate whether LLMs, despite not being specifically trained to detect vulnerabilities in code, can effectively identify vulnerabilities in source code.

While most LLM-based vulnerability-detection approaches rely on fine-tuning, VULCAN leverages LLMs' in-context learning capability, allowing them to tackle new tasks without specific training. To harness this capability, it is essential to adopt prompt engineering strategies that design effective natural language instructions to guide the models in detecting vulnerabilities in source code.

There are different prompting strategies: zero-shot, few-shot, and chain-of-thought. With zero-shot prompting, the model only receives the description of the new task to be performed. Few-shot prompting includes examples of how the task should be executed. Chain-of-thought prompting gives the model a sequence of steps to execute the task.

VULCAN's main goal is to develop a framework (see Figure 1) to create effective prompts for different prompt-engineering strategies and to evaluate and compare their impact on the accuracy of LLMs in detecting vulnerabilities. However, designing effective prompts for each strategy and evaluating the accuracy of LLM in detecting vulnerabilities requires addressing several challenges.

The first challenge is the limit on the number of input tokens for LLMs, which means that programs with many lines of code cannot be directly included in the prompts. It is therefore necessary to design a concise representation of the code that preserves the semantic relationship between the program's instructions, which is crucial for detecting vulnerabilities.

The second challenge is related to designing prompts for few-shot prompting, which requires adopting strategies to select examples that are relevant for identifying vulnerabilities. Inspired by the recent success of using retrieval modules to augment large-scale neural network models, VULCAN will adopt Retrieval-Augmented Generation (RAG) to select se-

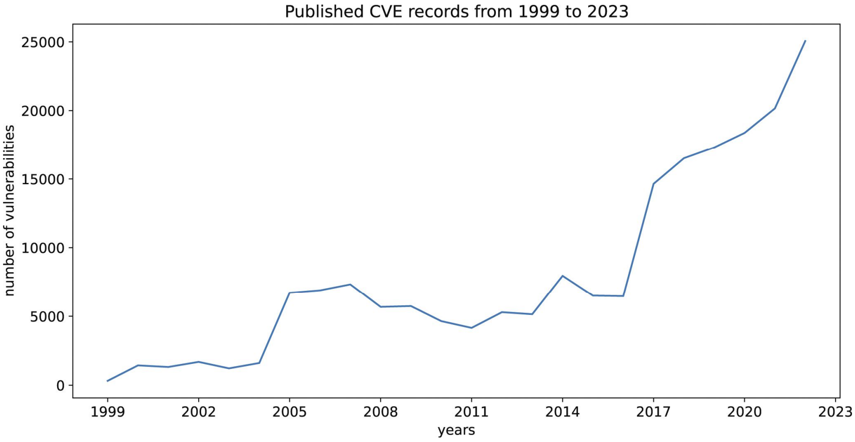


Figure 1: Published CVE records from 1999 to 2023. Source: https://www.cve.org/About/Metrics

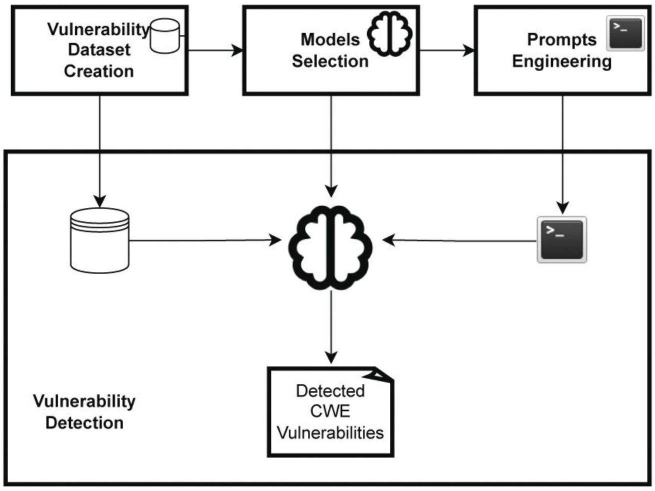


Figure 2: VULCAN framework components.

mantically similar examples to a test program to be included in the few-shot prompt. VULCAN will employ different code similarity measures to select the nearest neighbours of a given test program from a dataset of vulnerable programs.

Then, to verify the effectiveness of this method, we will investigate the impact of different code similarity measures, the order of examples, and the number of examples included in the prompt on the accuracy of LLMs in detecting vulnerabilities. On the other hand, the chain-of-thought strategy requires generating prompt models with specific steps to identify vulnerabilities, which differ for each vulnerability type. Therefore, we will explore how the steps for identifying a specific vulnerability type can be automatically generated.

Finally, a main challenge is related to the datasets of vulnerable code to use to assess LLMs' accuracy in detecting source code vulnerabilities. Indeed, existing vulnerability datasets suffer from many issues that impact their usability to assess the accuracy of LLMs in detecting vulnerabilities, including: 1. an unbalanced number of vulnerable and non-vulnerable programs, 2. inaccurate vulnerability labelling, and 3. data duplication [3]. SVEN [L1] is a notable exception; security experts manually labelled the vulnerabilities present in the code samples, but the number of samples is relatively small, which makes it inadequate to properly assess the accuracy of LLMs in vulnerability detection. Therefore, VULCAN will create a new, larger, balanced dataset of vulnerable programs with an accurate vulnerability labelling process.

The VULCAN's framework aims to address the above challenges to allow security analysts and software developers to harness the in-context learning capability of LLMs in detecting software vulnerabilities. As part of the framework, various experiments will be conducted to compare the effectiveness of the prompting strategies with general-use and code LLMs on the created dataset containing vulnerable programs. The VULCAN project will also develop a tool that leverages the LLM and prompting strategy that proves to be the most effective based on experimental results and will integrate it as a plug-in into the Eclipse development environment.

Links:
[L1] https://github.com/eth-sri/sven

References:

[1] Z. Li, et al., "Vuldeepecker: a deep learning-based system for vulnerability detection," in: 25th Annual Network and Distributed System Security Symposium (NDSS), 2018.
[2] Y. Guo, et al., "Outside the comfort zone: analysing LLM capabilities in software vulnerability detection," in Proc. of ESORICS, 2024.
[3] Y. Ding, et al., "Vulnerability detection with code language models: how far are we?," arXiv preprint arXiv:2403.18624, 2024. https://arxiv.org/abs/2403.18624.

Please contact:

Mila Dalla Preda, University of Verona, Italy
mila.dallapreda@univr.it

Federica Paci, University of Verona, Italy
federica.paci@univr.it

Increased Software Security with Large Language Models

by Zoltán Ságodi (University of Szeged), Péter Hegedűs (University of Szeged), and Rudolf Ferenc (University of Szeged)

As AI-driven language models increasingly demonstrate their ability to generate and repair source code, the role of human developers faces a profound transformation. This paper explores both the potential and challenges of leveraging these models for programming tasks and vulnerability mitigation, highlighting where human expertise remains essential.

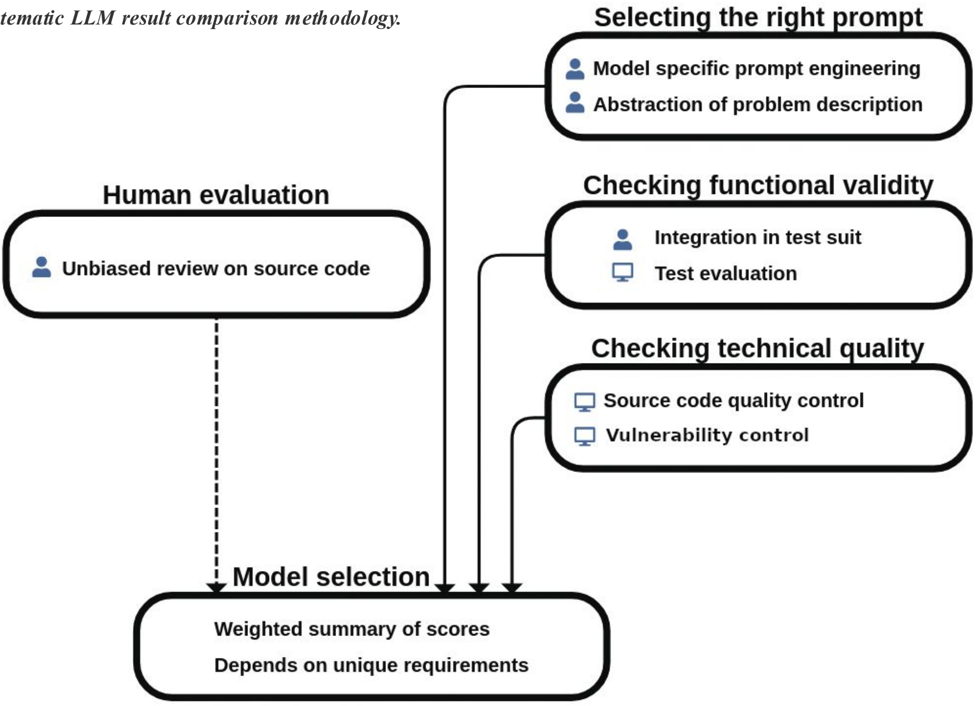
In recent years, Large Language Models (LLMs) such as GPT-4, ChatGPT, and GitHub Copilot have made remarkable strides in various fields, including software development. From generating new code to fixing existing security problems, these models offer promising tools for developers. However, their capabilities are still far from perfect, especially when it comes to real-world applications. We empirically investigated the real-world capabilities of state-of-the-art LLMs in two very common developer tasks: code synthesis and automated vulnerability repair.

We defined and applied a comparison methodology (depicted in Figure 1) to evaluate how state-of-the-art LLMs perform on code synthesis tasks [1]. Code synthesis is the task where, given a natural language specification, a program code is created that implements the defined functionality. Our comparison included ChatGPT and GitHub Copilot, which we evaluated on the Program Synthesis Benchmark [L1]. The functional evaluations showed that out of 25 programming tasks (i.e. algorithmic solutions to a specified problem), ChatGPT could solve 19, while Copilot could solve 13 tasks with a 100% test pass rate on a test suite including 10,000 tests (meaning that a program was considered correct if it passed all the 10,000 tests). Counting the tasks where the generated code achieved at least a 75% test pass rate on the 10,000 cases, we observed that ChatGPT could solve 20, while Copilot could solve 15 programming tasks.

During the human technical evaluation step, which in simpler terms meant to answer questions like "How good and secure is the code?" or "How many problems are present in the generated code?", the evaluations showed that these models tend to make some minor mistakes, but crucial problems, including potential vulnerabilities, are quite rare. These results are encouraging and might lead to the obvious question: if these models perform so well in synthesising good quality and secure code from human specifications, would they be capable of correcting human-written code as well?

We analysed this question by focusing on how well GPT-4 can fix existing problems in software systems, vulnerabilities (i.e. potential security issues) in particular. Unlike code synthesis, which focuses on generating new code from scratch, this study [2] investigates whether GPT-4 can repair flaws in existing

Figure 1: A systematic LLM result comparison methodology.


Selecting the right prompt
Model specific prompt engineering
Abstraction of problem description
Human evaluation
Unbiased review on source code
Checking functional validity
Integration in test suit
Test evaluation
Checking technical quality
Source code quality control
Vulnerability control
Model selection
Weighted summary of scores
Depends on unique requirements

code – an increasingly crucial task in the world of cybersecurity.

Using a dataset of 46 real-world Java vulnerabilities from the Vul4J repository [L2], we tasked GPT-4 with generating mitigation patches for these vulnerabilities. We performed a prompt engineering step to get the most out of the model. We also included the human factor in multiple layers. Firstly, we made evaluations by hand, so there were human ratings of the model's performance. Secondly, we used a benchmark that contained real-life human errors and their patches. The functional tests were also included, as the benchmark contained unit tests for every vulnerability.

We tested the vulnerability-fixing capabilities of the model, and we also tested if the model could be used as a code review assistant. We evaluated how well the model can fix vulnerabilities by providing our predefined prompt with the vulnerable code, and we requested the generation of its fixed version. At this point, we applied the changes and ran the tests. The results (see Figure 2) show that, on average, the model can fix 33.33% of the real vulnerabilities.

We also investigated the use case where we do not ask for a full fix, but only a suggestion. In this case, we asked the model to provide us with textual hints on how to fix a problem besides generating a mitigation patch. We evaluated these suggestions with multiple developers and found that, on average, in 56.6% of the cases, GPT-4 could give developers useful hints on how to fix the source code.

While these two studies focus on different aspects of secure software development – code generation and vulnerability repair – they share a common theme: the importance of human oversight. Both studies make it clear that LLMs like ChatGPT, Copilot, and GPT-4 are not yet capable of replacing human developers. Instead, they serve as powerful assistants that can help automate repetitive tasks, which eventually can lead to higher quality and more secure source code.

Looking at the broader picture, LLMs are making impressive strides in automating software development tasks, but there is still a long way to go. While ChatGPT and Copilot show promise in generating new code, and GPT-4 demonstrates potential in repairing vulnerabilities, neither is ready for full autonomy in real-world software development.

One key area for future research and development is improving the functional and technical validity as well as security of LLM-generated code. As the studies suggest, more sophisticated evaluation frameworks are needed to ensure that the code generated by these models meets both functional requirements and technical best practice. This will likely involve integrating automated tools for static analysis, as well as enhancing the models' understanding of complex software engineering tasks.

Another important direction for future work is better integrating LLM-generated code into existing development and security operations (DevSecOps) workflows. Right now, the code produced by these models often requires significant refinement before it can be deployed. Finding ways to streamline

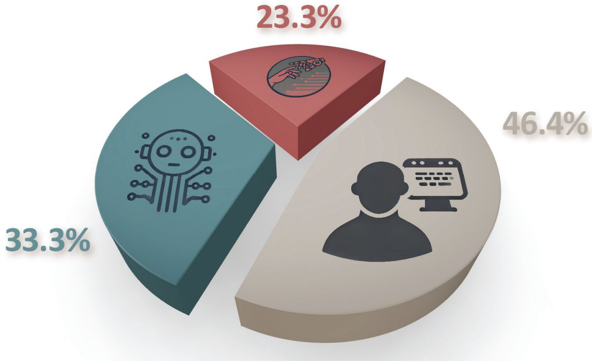


Figure 2: Distribution of vulnerabilities (i) fully fixed (green); (ii) useful fixing guidelines generated (red); (iii) cannot be fixed (grey) by GPT-4.

this process – either by improving the models' output or by developing tools that assist with post-processing – will be critical to making LLMs more useful in real-world development environments.

This research received support from the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory.

Links:
[L1] https://kwz.me/hDv
[L2] https://github.com/tuhh-softsec/vul4j

References:

[1] Z. Ságodi, I. Siket, R. Ferenc, "Methodology for code synthesis evaluation of LLMs presented by a case study of ChatGPT and copilot," IEEE Access, vol. 12, pp. 72303–72316, 2024. doi: 10.1109/ACCESS.2024.3403858

[2] Z. Ságodi, et al., "Reality check: assessing GPT-4 in fixing real-world software vulnerabilities," in Proc. of the 28th Int. Conf. on Evaluation and Assessment in Software Engineering, pp. 252–261, 2024.

Please contact:

Péter Hegedűs, University of Szeged, Hungary
hpeter@inf.u-szeged.hu

AI-Driven Software Security: Vulnerability Detection, Patching, and Anti-Fuzzing

by Aayush Garg, Yuejun Guo and Qiang Tang (Luxembourg Institute of Science and Technology)

Artificial Intelligence (AI) is revolutionizing software security within the DevSecOps framework by embedding automated tools for real-time vulnerability detection, patching, and anti-fuzzing into the development pipeline. The LAZARUS project at the Luxembourg Institute of Science and Technology (LIST) is leading this transformation, leveraging advanced AI models to proactively identify and address security threats before they can be exploited.

AI is transforming the way we approach software security, particularly within the DevSecOps framework, where security practices are integrated throughout the development lifecycle. As software systems become more complex and integral to our daily lives, the need for robust security measures has never been more critical. Traditional security methods, often implemented at the end of the development process, are no longer sufficient to address the sophisticated and fast-evolving threats that developers face today. At the Luxembourg Institute of Science and Technology (LIST), we are addressing these challenges through the pLatform for Analysis of Resilient and secUre Software (LAZARUS) project [L1], which focuses on enhancing software security by embedding AI-driven tools into Continuous Integration/Continuous Deployment (CI/CD) pipelines.

One of the primary innovations of the LAZARUS project is the use of AI for real-time vulnerability detection. By integrating AI models directly into the DevSecOps pipeline, as illustrated in Figure 1, we enable continuous monitoring of code as it is written and integrated. These models, built upon advanced large language models (LLMs) like CodeLLama, using vast datasets, are designed to predict vulnerabilities at the function level, allowing for early detection and immediate action. This proactive approach ensures that vulnerabilities are identified and addressed before they can be exploited, significantly reducing the risk of security breaches [1].

In addition to vulnerability prediction, the LAZARUS project also focuses on automating the patching process. Patching is a critical component of software maintenance, yet it often lags behind the discovery of vulnerabilities due to the manual effort involved. Our AI-driven patching tool, built on advanced AI models like CodeT5, automatically generates patches for identified vulnerabilities. This tool not only ensures that vulnerabilities are quickly remedied but also maintains the functionality of the original code. By automating patch generation and integration into the CI/CD pipeline, we ensure that security patches are applied immediately, reducing the window of opportunity for potential attackers [1].

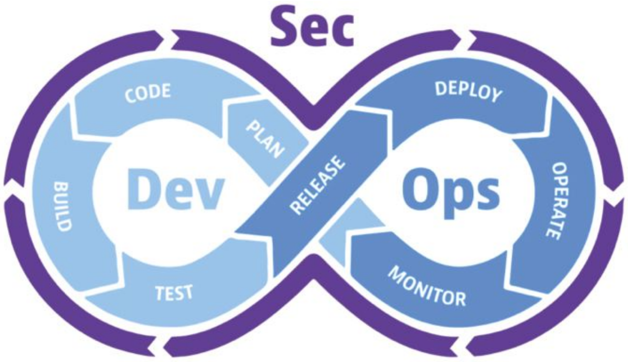


Figure 1: DevSecOps pipeline [L2].

Another significant challenge in software security is defending against fuzzing attacks. Fuzzing is a technique used by attackers to discover vulnerabilities by sending unexpected or malformed inputs to software APIs. In response, the LAZARUS project has developed an AI-based anti-fuzzing tool. This tool utilises Deep Learning (DL) models to classify and identify the origin of fuzzing inputs in real-time, allowing for immediate and targeted defences. By continuously monitoring incoming data and classifying these inputs based on their characteristics, our anti-fuzzing tool effectively neutralises potential threats before they can compromise the system. This capability is crucial in a DevSecOps environment, where security must be both robust and seamless, ensuring that development processes are not disrupted [1].

The integration of these AI-driven tools into the DevSecOps framework not only enhances security but also aligns with the agile, fast-paced nature of modern software development. By automating key security functions, such as vulnerability detection, patching, and defence against fuzzing, we reduce the burden on developers and security teams. This allows them to focus on delivering secure, reliable software without sacrificing speed or innovation.

Looking forward, our efforts will continue to refine these AI models, particularly in handling complex scenarios and to expand the scope to cover a broader range of vulnerabilities and fuzzing techniques. We are also exploring collaborations with other ERCIM institutions to share insights and further develop these security tools, ensuring they remain at the cutting edge of technology. The work being done under the LAZARUS project represents a significant advancement in the way we approach software security within the DevSecOps framework. By embedding AI into every stage of the development process, we are not only enhancing security but also ensuring that it keeps pace with the rapid evolution of software technologies. As these tools continue to evolve, they will become essential components in the toolkit of any organisation committed to maintaining secure and resilient software systems in an increasingly complex digital landscape. For more details on our research and publications, visit the LAZARUS project website here [L1].

Links:
[L1] https://lazarus-he.eu
[L2] https://www.dynatrace.com/news/blog/what-is-devsecops

References:

[1] https://kwz.me/hDj

Please contact:

Qiang Tang, Luxembourg Institute of Science and Technology (LIST), Luxembourg,
firstname.lastname@list.lu

A Proposal for Privacy-preserving Ransomware Detection by means of Federated Machine Learning

by Giovanni Ciaramella (IMT School for Advanced Studies Lucca and CNR-IIT), Fabio Martinelli (CNR-IIT), and Francesco Mercaldo (University of Molise and CNR-IIT)

The academic and industrial research community is exploring various machine learning methods to detect malware, particularly ransomware. However, real-world adoption is hindered by privacy concerns, as malware detection typically requires sending applications to a centralised model. To address this, we propose a privacy-preserving ransomware detection method using federated learning, which trains models locally on edge devices without transferring data. Preliminary experiments on a dataset of 15,000 real-world applications confirm the method's effectiveness.

Over the years, the number of cyber-attacks has increased drastically. As shown in a report published by Statista, approximately 65% of financial organisations worldwide reported experiencing a ransomware attack in 2024, an increase from 64% in 2023 and 34% in 2021 [L1]. For many years, experts and researchers identified various solutions to protect users from attacks. Another issue in recent years is keeping personal data safe. To address that, the European Union introduced the General Data Protection Regulation (GDPR) in 2016 to regulate personal data usage. Nowadays, each technological device needs to handle personal data for several reasons. In 2017, to address the privacy issue, Google introduced the concept of federated machine learning [1]. This consists of training a model using local data, keeping raw data decentralised, and only sharing the model updated to the server.

In this article, we propose a method based on federated learning to classify malware, ransomware, and trusted applications

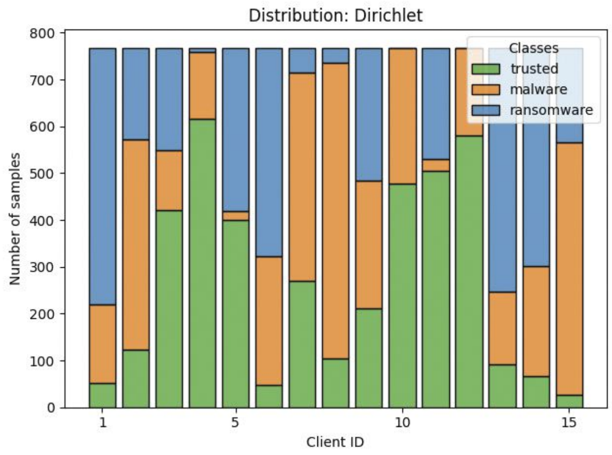


Figure 1: The Dirichlet distribution obtained among 15 different clients.

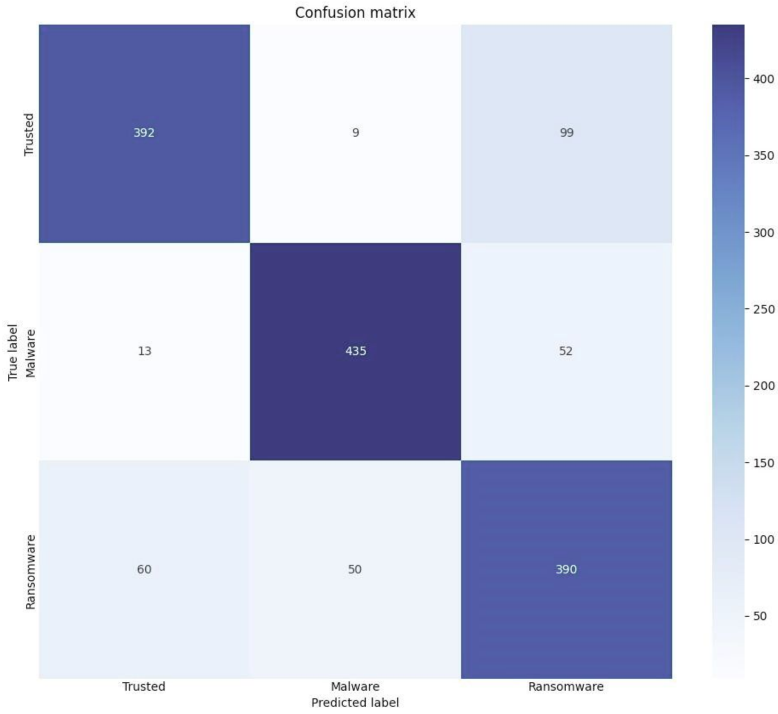


Figure 2: The Confusion matrix.

belonging to the Windows environment. We assessed the effectiveness of the proposed approach by exploiting over 15,000 samples divided into ransomware (5,000), generic malware (5,000), and trusted samples (5,000). The generic malware and ransomware samples were sourced from the VirusShare repository [L2] while trusted samples were collected from libraries (such as DLL files) and executable files from a Microsoft Windows 10 machine. To verify the maliciousness or trustworthiness of the samples, we submitted them to the VirusTotal [L3] and Jotti [L4] web services. As the next step, all executable files were transformed into text files containing the opcodes obtained using the objdump disassembler. To create an image representation, we assigned an RGB value to each opcode within the file and used these values to fill each matrix's pixel. Specifically, we assigned a distinct colour to each opcode for all the applications involved in the experimental analysis. After creating the complete dataset of images, we divided it into training, validation, and testing sets using an 80-10-10 split (12,000-1,500-1,500).

We started the experiment leveraging federated learning adopting the clipping norm aggregator, which limits the influence of individual data points by capping their gradients' norms before averaging them during model training, helping to reduce the impact of outliers. Moreover, in federated learning it is necessary to use a distribution because of the number of clients involved, and to recreate a real-world scenario we used non-IID data, i.e. Dirichlet distribution as shown in Figure 1. This type of distribution turns out to be different from the IID data [2] because it uses different statistical properties for each client, thus recreating a real-life scenario. After deciding on the norm type and distribution type to use, we began conducting experiments, employing a state-of-the-art architecture, i.e. MobileNetV3 [3]. To train our model, we used the "sgdm" optimiser, 15 clients, 20 rounds, and 15 epochs per round on each client. At the end of the experiments, we obtained a model with an accuracy value of 0.824 and a loss of 0.438 during the test phase.

To evaluate the model's performance we computed the confusion matrix shown in Figure 2. The model was able to correctly identify 392 samples as "Trusted", though it misclassified 108 instances, mistakenly labelling 9 as "Malware" and 99 as "Ransomware". In the category named "Malware" the model achieved 435 correct predictions, with minor errors resulting in 13 instances being labelled as "Trusted" and 52 as "Ransomware". The "Ransomware" category also performed strongly, with 390 accurate classifications. However, 110 samples were incorrectly categorised, with 60 samples classified as "Trusted" and 50 as "Malware".

In conclusion, the federated learning application in the malware detection field can be crucial in enhancing detection capabilities by guaranteeing the respect of user privacy. We demonstrated that a model trained and tested by exploiting federated learning can identify malware, ransomware, and trusted Windows applications with interesting performance accuracy. As future work, we plan to consider model explainability with the aim to pinpoint which area of images related to applications are most distinctive for a given prediction.

This work has been partially supported by EU DUCA, EU CyberSecPro, SYNAPSE, PTR 22-24 P2.01 (Cybersecurity) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the EU - NextGenerationEU projects.

Links:
[L1] https://kwz.me/hDq
[L2] https://virusshare.com/
[L3] https://www.virustotal.com/
[L4] https://virusscan.jotti.org/

References:

[1] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in Proc. of Artificial intelligence and statistics, pp. 1273–1282, PMLR, 2017.
[2] G Ciaramella et al., "An approach for privacy-preserving mobile malware detection through federated machine learning," in Proc. of 26th International Conference on Enterprise Information Systems, pp. 553–563, 2024.
[3] A. Howard et al., "Searching for mobilenetv3," in Proc. of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324, 2019.

Please contact:

Giovanni Ciaramella, IMT School for Advanced Studies Lucca, Italy and CNR-IIT, Pisa, Italy
giovanni.ciaramella@imtlucca.it and
giovanni.ciaramella@iit.cnr.it

Vulnerability of Software Package Repositories: PyPI, Maven and npm

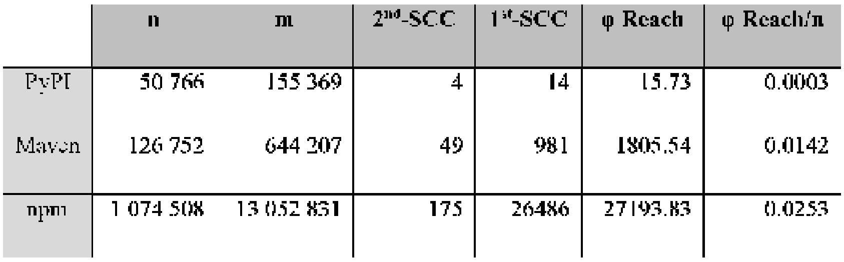
by Daniel Setó-Rey, Carlos López-Nozal, and José Ignacio Santos-Martín (Universidad de Burgos)

The reuse of software by importing packages from repositories is an efficient way to develop software. However, reusing software in this manner introduces vulnerability risks due to transitive dependencies. These vulnerabilities must be measured to identify risks and propose corrective actions.

Software ecosystems are communities formed around programming languages and shared package management tools, enabling developers to create new packages that import and reuse the functionality of others [1]. Development within these ecosystems is efficient because common functionalities only need to be developed, maintained and tested by a single team, rather than having multiple authors reimplement the same functionality. The use of centralised library repositories to reduce development times and costs is widespread across nearly all languages and software projects. However, this efficient approach introduces vulnerability risks, primarily due to transitive dependencies and packages cycles. Because of dependency transitivity, a single defect in the repository can have far-reaching and unpredictable effects on the ecosystem. These defects may result in functional errors or issues with performance and security. The risk can be hard for developers to assess, as they typically import only a small portion of the dependencies.

In [2], we propose, develop, and test a new theoretical model to characterise the vulnerability of package repositories, which are represented as complex networks of dependencies. We define vulnerability as a metric that measures the sensitivity of a package repository to random defects represented by a cost function called φ Reach. We applied this model to three well-known package repositories (PyPI, Maven and npm) to calculate their vulnerability [L1]. The package network and dependency information were constructed using the libraries.io data dump [L2]. Our analysis revealed that the emergence of a large strongly connected component (SCC), a set of mutually dependent packages, is associated with a disproportionate increase in the vulnerability of package dependency networks (see details in Table 1).

Based on the concept of node vulnerability, we define immunisation at a node as any corrective or preventive action taken to eliminate the possibility of it failing or incorporating a defect. The effect of immunising a set of nodes is calculated as the difference between the network vulnerability of the initial network and that of the immunised network. In our experiments [2], we observed that protecting SCC to prevent the introduction or propagation of defects can nearly eliminate the network's vulnerability. However, depending on the number of such packages, this solution may not be practical. Identifying the optimal set of nodes to immunise for the greatest reduction in vulnerability is an NP-hard problem, so heuristics are necessary to find sufficiently good solutions. Figure 1 shows the SCC of the Maven package dependency network, consisting of 981 nodes (0.8% of the network). Immunising the component's cut vertices (351 nodes) can reduce vulnerability by



	n	m	2nd-SCC	1st-SCC	φ Reach	φ Reach/n
PyPI	50 766	155 369	4	14	15.73	0.0003
Maven	126 752	644 207	49	981	1805.54	0.0142
npm	1 074 508	13 052 831	175	26486	27193.83	0.0253

Table 1: Characteristics and vulnerability to failure of reference package dependency networks. n: number of packages, m: number of arcs (dependency relations), 2nd and 1st-SCC: second largest and largest strongly connected component present, φ Reach: vulnerability to failure measured by the Reach cost function and next to it percentage vulnerability in relation to network size (n).

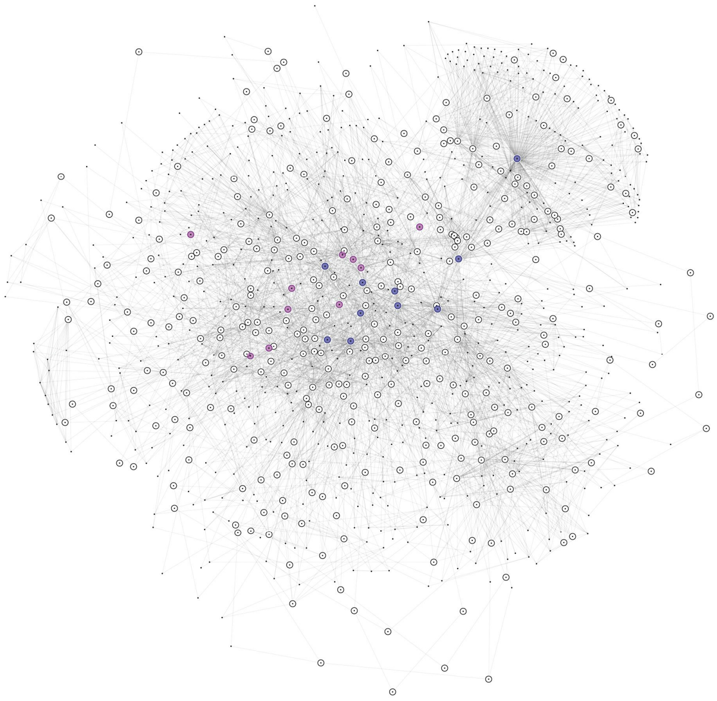


Figure 1: The SCC of the Maven package dependency network consists of 981 nodes, representing 0.8% of the network. The nodes highlighted in white (351 nodes) are the component's cut vertices. The nodes highlighted in blue are the ten cut vertices with the highest out-degree centrality, while the nodes highlighted in red are the ten with the highest betweenness centrality.

93.9%. Additionally, Figure 1 highlights the ten cut nodes with the highest out-degree centrality and betweenness centrality, resulting in vulnerability reductions of 13% and 26%, respectively.

We used a variety of techniques to narrow down the sets of important packages related to the network's vulnerability, achieving similar reductions by acting on a smaller number of packages. We demonstrate that selecting the set of strong articulation points (SAP) of the SCC achieves reductions similar to acting on the entire SCC.

This work helps decision-makers in software ecosystems (such as software developers, package developers and package repository managers) in assessing vulnerability risks caused by dependencies on third-party packages. Specifically, it can help:

- package repository managers to establish and follow vulnerability reduction plans
- software developers to evaluate the overall risk associated with the use of a given package
- package developers to assess the overall risk associated with package development in the context of a given repository
- package repository managers to establish measures to reduce or eliminate the occurrence of strongly related components, such as dependency cycle control
- package repository managers to implement immunisation policies to reduce the vulnerability of the network, with different techniques available to find good immunisation target sets.

Links:
[L1] https://doi.org/10.5281/zenodo.7358391
[L2] https://libraries.io/

References:

[1] C. Bogart, et al., "When and how to make breaking changes: policies and practices in 18 open source software ecosystems," ACM Trans. Softw. Eng. Methodol., vol. 30, no. 4, 2021. doi:10.1145/3447245
[2] D. Seto-Rey, J. I. Santos-Martin, and C. Lopez-Nozal, "Vulnerability of package dependency networks," IEEE Transactions on Network Science and Engineering, vol 10, no. 6, pp. 3396–3408, 2023. https://doi.org/10.1109/TNSE.2023.3260880

Please contact:

Carlos López-Nozal, Universidad de Burgos, Spain
clopezno@ubu.es

Uninstallable by Design: The Role of Pre-installed Apps in Android's Security Landscape

by Thomas Sutter (University of Bern and Zurich University of Applied Sciences), Ariane Trammell (Zurich University of Applied Sciences), and Timo Kehrer (University of Bern)

The competitive smartphone market is keen to prevent its intellectual property from being analysed by competitors and the public. As a result, most smartphones are locked when distributed, and anti-reversing techniques are widely used. Consequently, millions of users use smartphones daily without a clear understanding of the software's functionality and purpose. We developed a novel framework, FirmwareDroid, to analyse the security of mobile device firmware.

Nowadays, most smartphones come with a large number of pre-installed mobile apps. For instance, on Android, it is normal to find between 150 and 1,000 pre-installed apps for various purposes. Some of these apps are essential for the system to operate correctly, while others aim solely to enhance the user experience. Android allows the inclusion of third-party apps. Thus, many devices come shipped with apps that some users don't want or need on their devices, such as social media and news apps. These apps are often referred to as "bloatware," as they unnecessarily bloat the device's operating system and cannot be simply uninstalled by users. The main reason why bloatware on Android cannot be uninstalled is a security feature that prevents attackers from modifying system apps on the device.

On Android, the file system where pre-installed apps are stored is read-only, to prevent any modifications to the system. This security feature is fundamental and complements other security mechanisms (e.g. Android Verified Boot) to prevent Man-At-The-End (MATE) attacks. As most Android devices are effectively locked, users cannot remove pre-installed apps from the filesystem without unlocking the bootloader of the device (e.g. by rooting or jailbreaking).

Moreover, the open-source nature of Android allows vendors to modify existing system apps by adding or removing functions. As these modifications are mainly closed-source changes, they might introduce bugs that affect the security of the device. Recent studies have found several security issues in pre-installed apps [1] and identified privacy concerns [2] but could only provide limited insights in terms of analysis methods applied. This demonstrates the need for better testing and verification of pre-installed apps on mobile devices.

Undoubtedly, there is a high level of public interest in knowing what data is collected on smartphones and which vendors follow good software development and security practices. As a result, in an attempt to restore some transparency, we developed a framework called FirmwareDroid [L1] that allows the

Figure 1: Smartphones are distributed with hundreds of pre-installed apps that cannot be uninstalled by users, but are these apps secure?

extraction of pre-installed apps from Android firmware. FirmwareDroid's purpose is to automate the security analysis of pre-installed apps for various purposes, such as detecting vulnerabilities, identifying malware, or privacy issues. In addition, the project makes the analysis more applicable for practitioners and allows researchers to integrate their own testing tools.

With FirmwareDroid, we demonstrated that many pre-installed apps utilise advertising trackers and have a concerning number of dangerous privileges [3]. Currently, FirmwareDroid includes several state-of-the-art static analysis tools, and the framework allows for the extraction and analysis of apps at scale with its multi-core scanning engine. However, the dynamic analysis of pre-installed apps has not been well studied and primarily relies on physical devices for testing, which is cost-intensive, does not scale well, and is challenging due to vendors implementing anti-reversing techniques.

As we continue to improve FirmwareDroid, our next critical objective is to expand its capabilities to include dynamic analysis of pre-installed apps. This enhancement will significantly strengthen the tools available to researchers, enabling them to conduct more comprehensive and detailed studies.

Given the highly fragmented nature of Android, each vendor's custom modifications to the operating system present unique challenges. To address this, we recognise the need for innovative solutions that allow us to test these modifications in real time. Understanding what changes have been made and the reasons behind them is not only valuable for advancing research but also essential for safeguarding the public's digital security.

To achieve this, we are developing a solution that facilitates the dynamic testing of Android pre-installed apps without the need for physical devices. By leveraging advanced emulation technologies, we aim to create a scalable, cost-effective method for analysing these apps in a controlled environment. This development will open new avenues for research and provide greater transparency, ultimately contributing to a safer and more secure mobile ecosystem for everyone. Through this project, we hope to help both the research community and the public regain some transparency and further strengthen the security of mobile devices. FirmwareDroid is open-source and will be continuously developed over the coming years. Contributions to the framework or our research are welcome.

Link:
[L1] https://github.com/FirmwareDroid/FirmwareDroid

References:

[1] M. Elsabagh, et al., "FIRMSCOPE: automatic uncovering of privilege-escalation vulnerabilities in pre-installed apps in Android firmware," in USENIX, 2020.
[2] J. Gamba, et al., "An analysis of pre-installed android software," in IEEE S&P, 2020.
[3] T. Sutter and B. Tellenbach, "IEEE/ACM MOBILESoft," in FirmwareDroid: Towards Automated Static Analysis of Pre-Installed Android Apps, 2023.

Please contact:

Thomas Sutter, PhD Student Software Engineering Group, University of Bern, Switzerland
thomas.sutter@unibe.ch

Ariane Trammell, Head of Information Security Group, Zurich University of Applied Science, Switzerland
ariane.trammell@zhaw.ch

Towards Cyber Security Risk Analysis for Digital Products

by Christophe Ponsard and Jean-François Daune (CETIC)

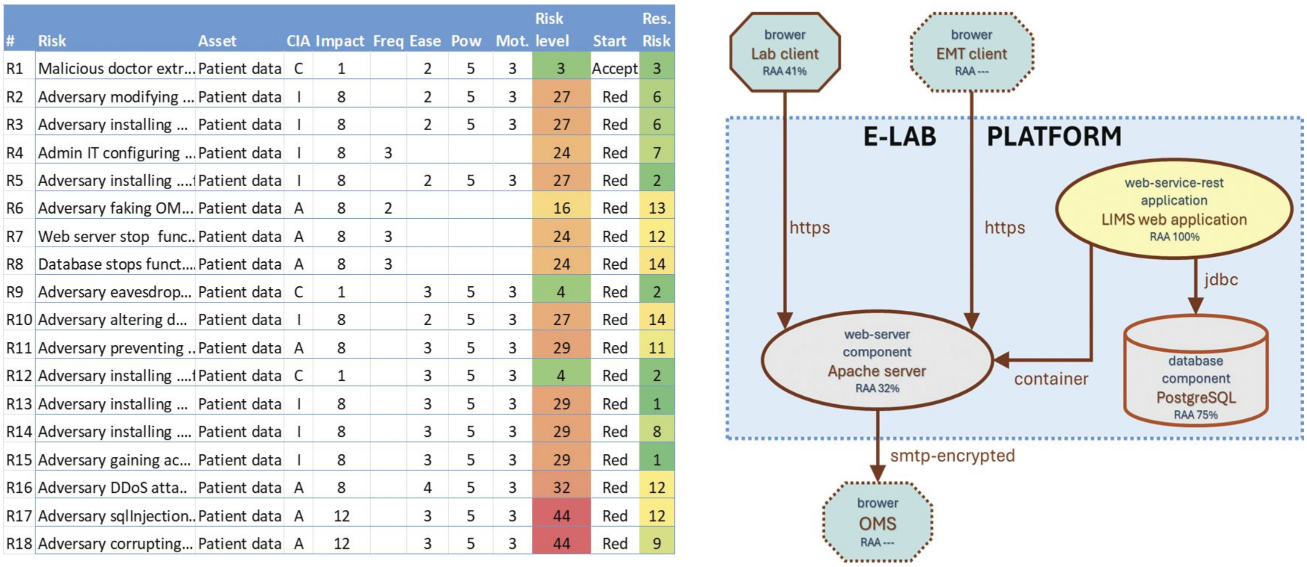
Digital products have become ubiquitous across all domains for everyday activities of both citizens and companies. Providing secure products is required to ensure the organisations relying on them have a minimal attack surface. This article highlights specific needs and our ongoing work to conduct a cyber security risk analysis for a digital product, which is also increasingly required by regulations such as the EU The Network and Information Security Directive (NIS2) or the upcoming Cyber Resilience Act.

Conducting regular cyber security risk analysis inside an organisation is strongly recommended to stay resilient against a quickly evolving threat landscape. It is a mandatory step in many standards and related certifications across many sectors such as ISO27002 (Information Technology - IT), IEC 62443 (Operational Technology - OT) and SAE 21434 (automotive). Those approaches are however performed at a full organisation level and do not focus on the point of view of the editor providing a digital solution. However, securing the supply chain is now part of regulations such as the NIS2 and specific security requirements will also apply to digital products with the upcoming Cyber Resilience Act (CRA).

When investigating available methods for conducting a risk analysis targeting a digital product, two main limitations emerged. First, many methods are focusing on protecting the organisation's business assets through a large set of supporting IT assets. These methods do not take the point of view of a digital product, although they identify potential threats related to their configuration, (mis)use, update and, more recently, development. This typically relies on the ISO27005 iterative risk-oriented approach with methods such as EBIOS (France) [L1], IT-SRM (EU) [L2] or MONARC (Luxembourg) [L3] that we investigated and were also reviewed by ENISA [1]. Second, product-oriented methods focus more on checking the implementation with respect to the security specification by examining the security maturity across the lifecycle and (pen) testing the product (e.g. assurance levels in the common criteria). However, they just assume a prior risk assessment to show the security objectives are addressing identified threats. Those methods can also be cumbersome and have difficulties coping with product evolution, although more incremental processes are being investigated [2].

A product-oriented risk analysis requires a mix of approaches with the product as central asset. It must deal with different aspects such as the interaction with the outer ecosystem, the security guarantee to provide on specific information/processes (including the product itself in integrity and for intellectual property), the inner product architecture, the resulting attack surface through its interfaces, the internal development process and the product dependencies (e.g. through Software Bill of Materials). Additional requirements are to rely as much as possible on available methods and tools in order to be efficient and to stay compliant with existing standards. An architecture modelling foundation also enables a precise description, in-depth analysis (e.g. using recommended threat modelling) and automation for generating artefacts or identifying security measures.

After an in-depth review of the methods mentioned above, we decided to adopt IT-SRM as our core method for several reasons. First, as IT systems are our main focus, it is 27001 compliant but it can also be considered as the basis for specialised domains such as OT or medical software. Second, it is very well documented including catalogues borrowed from other methods such as EBIOS. Finally, it quite systematically supports the review of the required security properties over technical assets to produce risk estimates and, from there, to decide about additional security measures. The method does not provide tool support but can easily be automated using spreadsheets and modelling tools as depicted in Figure 1, which is freely inspired by a real case study.



#	Risk	Asset	CIA	Impact	Freq	Ease	Pow	Mot.	Risk level	Start	Res. Risk
R1	Malicious doctor extr...	Patient data	C	1		2	5	3	3	Accept	3
R2	Adversary modifying ...	Patient data	I	8		2	5	3	27	Red	6
R3	Adversary installing ...	Patient data	I	8		2	5	3	27	Red	6
R4	Admin IT configuring ...	Patient data	I	8	3				24	Red	7
R5	Adversary installing 1	Patient data	I	8		2	5	3	27	Red	2
R6	Adversary faking OM...	Patient data	A	8	2				16	Red	13
R7	Web server stop func..	Patient data	A	8	3				24	Red	12
R8	Database stops funct....	Patient data	A	8	3				24	Red	14
R9	Adversary eavesdrop...	Patient data	C	1		3	5	3	4	Red	2
R10	Adversary altering d...	Patient data	I	8		2	5	3	27	Red	14
R11	Adversary preventing ..	Patient data	A	8		3	5	3	29	Red	11
R12	Adversary installing 1	Patient data	C	1		3	5	3	4	Red	2
R13	Adversary installing ...	Patient data	I	8		3	5	3	29	Red	1
R14	Adversary installing	Patient data	I	8		3	5	3	29	Red	8
R15	Adversary gaining ac...	Patient data	I	8		3	5	3	29	Red	1
R16	Adversary DDoS atta..	Patient data	A	8		4	5	3	32	Red	12
R17	Adversary sqlInjection..	Patient data	A	12		3	5	3	44	Red	12
R18	Adversary corrupting...	Patient data	A	12		3	5	3	44	Red	9

Figure 1: Current ISRM tool support with spreadsheet (right) and Threagile architecture modelling (left).

The next step of our work is to improve tool support to better capture the product architecture and guide the analysis, e.g. inside the Open Source MONARC risk analysis tool based on our previous work [3]. For this, we are also investigating how to enrich the component classification, related risks and countermeasures. Finally, our work is currently being validated on different use cases, e.g. a medical ERP [L4], in the scope of a grand challenge on risk analysis for penetration testing proposed by the Belgian CyberWal platform and driven by industrial needs.

Links:

[L1] https://kwz.me/hDw
[L2] https://kwz.me/hDx
[L3] https://www.monarc.lu
[L4] https://kwz.me/hDz

References:

[1] C. Lambrinoudakis et al, “Interoperable EU risk management framework,” ENISA Report, 2022.

[2] S. Dupont et al, “Product incremental security risk assessment using DevSecOps practices,” SecAssure@ESORICS 2022.

[3] C. Ponsard et al, “Improving cyber security risk assessment by combined use of i* and infrastructure models,” IStar conference, 2021.

Please contact:

Christophe Ponsard, CETIC, Belgium
christophe.ponsard@cetic.be

Challenges for the Secure Integration of Drones into Warehouse Logistics of SMEs

by Peter Kieseberg (St. Pölten UAS), Christoph Kaltenriner (Dataphone GmbH), and Peter Gallistl (Dataphone GmbH)

Drones promise significant benefits for small-and-medium-sized enterprises (SMEs) in warehouse logistics, but integrating them securely into existing systems is a complex challenge. This article explores how SMEs can overcome security risks – including man-at-the-end (MATE) attacks – and operational hurdles to effectively adopt drone technology.

Drones offer a lot of benefits when used for mundane tasks in warehouses, especially with respect to inventorisation. Thus, companies running large warehouses have dived deep into automatisation, optimising their warehouses accordingly with specially designed solutions. This not only includes the integration of robotic hardware, be it ground-based solutions or even drones, but especially rebuilding the warehouse with intelligent shelving systems that communicate with the infrastructure and potentially the mechanical actors. Security is very important in these solutions [1], with a lot of thought put into the respective aspects of the system architecture, including focus on MATE attacks. While these solutions are certainly very promising for large cooperations being able to build large custom-made warehouses, it is infeasible for the standard SME, both, from a cost side, as well as from the fact that SMEs typically cannot shut down their facilities for long in order to update them to this level.

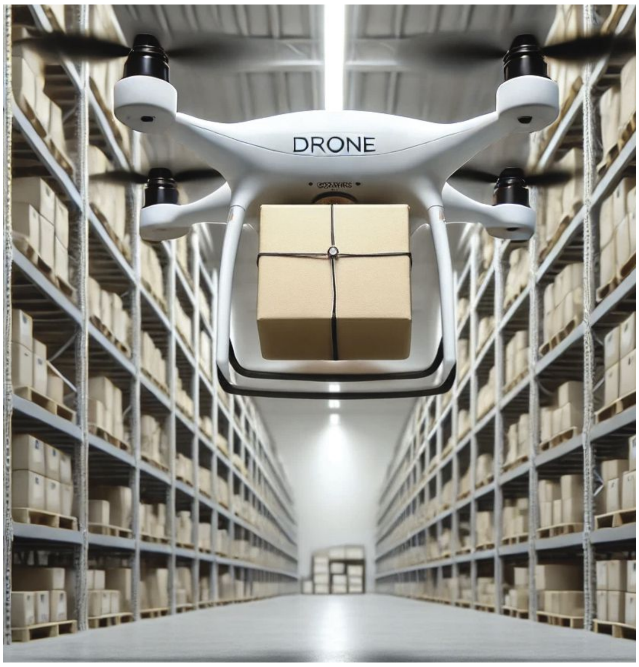


Illustration of a drone in a warehouse.

Thus, intelligent warehouse logistics using drones needs to be able to securely integrate with a lot of legacy equipment, ranging from the very shelves to the software components controlling the warehouse, in order to allow for secure integration. This especially poses problems for current state-of-the-art drone solutions, as these environments are not able to support the drones through communication and shelf-side automation features. Furthermore, the high degree of automation as is pushed forward by big cooperations is not feasible for SMEs, still requiring human workers to fulfil a multitude of tasks. This further complicates the task of drone integration, as human workers might leave goods on the floor, or might not stack shelves exactly enough in order for the drones to follow their flight paths without bumping into them. While this could theoretically be solved using AI methods, recent research has shown that drones have severe problems in similar complex environments [2].

Given the side parameters outlined above, security becomes a major concern, especially considering MATE attacks when comparing the environments to highly professional automated logistic warehouses:

1. The drone control system needs to interact with a multitude of different backend software systems, often of questionable security.
2. Configuration of and interaction with the system needs to be done on a regular basis by a lot of involved persons due to layout changes in the warehouses and the absence of communication with the shelving systems. This stands in stark

contrast to modern warehouses of large companies that are highly optimised for automation.

3. In addition, the typical shelving system prevailing in SMEs has no means of detecting mis-stacking or other errors in the stacking process, thus requiring the drone and its control system to take care of such matters. This adds a layer of complexity that is either ignored and put into the hands of the human workers, or can cause problems for the drones, e.g. through crates invading the drones' designated flight paths. Drones with the necessary sensors on board for dealing with these kinds of problems are currently prohibitively expensive for most SMEs.
4. In addition, warehouses in the SME sector are in general not fully automated, i.e. human workers need to work there too. While side-by-side work with drones is currently out of the question in this price class for safety reasons, so-called mixed environments can be set up by separating the drones from the human workers through shifts, e.g. having the human workers work by day and the drones doing their tasks like inventorisation at night. This, of course, also opens a lot of chances of interference of workers with the drones, either due to carelessness or malicious intent.

In addition, contrary to many other environments, small disturbances can wreak havoc on a company, especially when dealing with Just-in-Time (JIT) logistics. Another MATE-related problem lies in the fact that both logistics software, as well as software controlling the drones, are typically closed source. While the former is often quite old and undocumented, current developments in AI also allow drone control software to evolve very quickly, which makes security testing complex and costly. Furthermore, procurement of systems containing AI components is far from trivial from a security perspective, as shown in [3]. While typical logistics does not fall under the high-risk level of the AI-Act, except of course where critical infrastructure is involved, they might need to follow the NIS and NIS2 regulations, where a lot of legal work needs to be done with respect to training of AI systems and AI risk management.

In summary, drones in SME warehouses offer great opportunities in becoming more competitive and cutting costs for many tasks like inventorisation, yet there are still open challenges with respect to system hardening, especially against MATE attacks.

References:

[1] A. Rejeb, et al., "Drones for supply chain management and logistics: a review and research agenda," Int. J. of Logistics Research and Applications, pp.1–24, 2021.

[2] A. Buchelt, et al., "Exploring artificial intelligence for applications of drones in forest ecology and management," Forest Ecology and Management, vol. 551, p.121530, 2024.

[3] P. Kieseberg, et al., "Security considerations for the procurement and acquisition of Artificial Intelligence (AI) systems," in 2022 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE), pp. 1–7.

Please contact:

Peter Kieseberg, St. Pölten University of Applied Sciences, Austria

Enhancing IoT Security Across the Supply Chain

by Ramon Barakat, Sascha Hackel (Fraunhofer FOKUS) and Miltiadis Siavvas (CERTH)

The Design and Operation of Secure Supply Chain (DOSS) project's Supply Trust Chain idea seeks to truly improve IoT security and trust from design to deployment. The project's goal is to secure the IoT supply chain throughout its lifecycle, ensuring that stakeholders always have access to security-related information.

The DOSS project [L1], under the Horizon Europe initiative, is dedicated to securing the IoT supply chain throughout its entire lifecycle. This project aims to ensure IoT security by monitoring all stages from design to decommissioning, making security-relevant information accessible to stakeholders, and fostering robust communication among various actors.

The Device Security Passport, a machine-processable document that contains pertinent security data for IoT components, such as certificates, SBOM, HBOM, and MUD files, is one of the essential parts of the DOSS project. This document will be stored in a secure, accessible repository, ensuring that security data is both reliable and protected.

Figure 1 illustrates the preliminary architecture of the DOSS project, highlighting the integration of various modules and workflows. This architecture outlines how different components, such as the Component Tester and the Digital Security Twin, interact to ensure comprehensive security validation. By detailing these interactions, Figure 1 provides an overview of the system's design and operational flow, emphasising the project's holistic approach to IoT security.

Component Tester and Vulnerability Prediction

Adopting the Zero Trust principle of "never trust, always verify," DOSS will validate all IoT architecture components at multiple stages. A unified Component Tester will verify the Device Security Passport's content and ensure the security of the IoT components.

The Component Tester will ensure the security of the IoT components including third-party applications (closed- and open-source software) and proprietary developments. It utilises a comprehensive testing and validation methodology, analysing closed-source software using binary code validation techniques and verifying components according to their Device Security Passports. Software security verification methods will be used to evaluate open-source software and proprietary development.

The DOSS project emphasises the importance of Interactive Application Security Testing (IAST) to address the complexities of IoT security. IAST combines the strengths of Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) to provide a more comprehensive security analysis.

In addition to the static analysis, a Vulnerability Prediction module using Large Language Models (LLMs) will be inte-

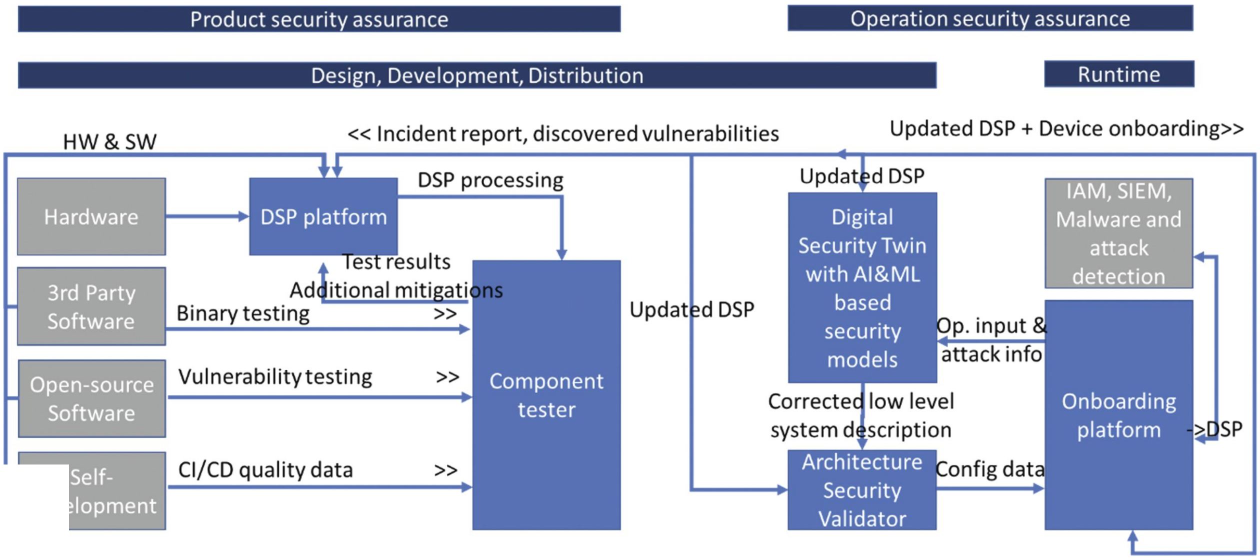

Product security assurance
Operation security assurance
Design, Development, Distribution
Runtime
HW & SW
<< Incident report, discovered vulnerabilities
Updated DSP + Device onboarding>>
Hardware
DSP platform
DSP processing
Test results
Additional mitigations
3rd Party Software
Binary testing >>
Open-source Software
Vulnerability testing >>
Self-
elopment
CI/CD quality data >>
Component tester
Updated DSP
Updated DSP
Digital Security Twin with AI&ML based security models
IAM, SIEM, Malware and attack detection
Op. input & attack info
Corrected low level system description
Onboarding platform
->DSP
Architecture Security Validator
Config data

Figure 1: Overview of the supply chain integration (Supply Trust Chain).

grated. The vulnerability prediction focuses on identifying security hotspots in software components, enabling better allocation of testing resources [2]. DOSS will explore the use of popular pre-trained LLMs for vulnerability prediction, including Generative Pre-trained Transformer (GPT), Bidirectional Encoder Representations from Transformers (BERT), and Bidirectional Auto-Regressive Transformers (BART), which will be fine-tuned using curated vulnerability datasets and compared against traditional text-mining-based models. For enhancing their accuracy, additional modalities will be used as inputs to capture more context from the source code, focusing on text-rich graphs, such as Abstract Syntax Trees (ASTs), Control-flow Diagrams (CFDs), etc. Explainable AI (XAI) techniques will be also leveraged to facilitate vulnerability localisation [3].

An essential aspect of the DOSS CT is the patch validation process. Directed fuzzing techniques are used to ensure that security patches effectively fix vulnerabilities completely without introducing new ones. This technique generates a suite of tests that trigger the identified vulnerabilities in various ways during patch development, ensuring comprehensive coverage and validation.This integrated approach enhances the detection of vulnerabilities and streamlines the remediation process, ensuring that only secure components are integrated into the IoT architecture.

IoT Operation and Supply Trust Chain

DOSS extends its security concept to IoT operation. New devices will be integrated using an automated Onboarding Platform, leveraging the Device Security Passport for configuration.Furthermore, the CT of the DOSS project is designed to be versatile, allowing for use not only during the development phase but also on the operator side. This ensures ongoing security validation.

Infrastructure automation technologies will be used to model various service architectures by importing verified components into a Digital Cybersecurity Twin. High-level system descriptions will be extracted by AI-assisted models to detect threats and suggest defences. The Architecture Security Validator will perform automated architecture validation on these models ensuring also their compliance with industry standards. The tested, modelled, and validated IoT architectures will also be monitored during their operation by the IoTAC platform presented in [1]. The IoTAC platform [L2] will monitor and protect the operational environment, providing feedback to design-time modules and stakeholders through the Supply Trust Chain.

Conclusion

The DOSS project aims to significantly enhance the security and trust of IoT services and architectures. Through its comprehensive approach, DOSS strives to contribute to a more secure and trustworthy IoT ecosystem. The project's findings will be validated in automotive, prosumer cell operation, and smart home domains, ensuring practical relevance and impact.

The DOSS project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101120270.

Links:
[L1] https://dossproject.eu
[L2] https://iotac.eu/

References:

[1] S. Hackel, M. Schneider, and R. Barakat, "Security-by-design IoT development and certification with IoTAC," ERCIM News 134, Jun. 2023. https://ercim-news.ercim.eu/en134/r-i/security-by-design-iot-development-and-certification-with-iotac
[2] I. Kalouptsoglou, et al., "Software vulnerability prediction: A systematic mapping study," Information and Software Technology, vol. 107303, 2023.
[3] M. Fu and C. Tantithamthavorn, "Linevul: A transformer-based line-level vulnerability prediction," in Proc. of the 19th Int. Conf. on Mining Software Repositories, 2022, pp. 608–620.

Please contact:

Ramon Barakat, Fraunhofer FOKUS, Germany
ramon.barakat@fokus.fraunhofer.de

Sascha Hackel, Fraunhofer FOKUS, Germany
sascha.hackel@fokus.fraunhofer.de

Towards Safer Software: Exploring Validation Techniques for Rust Binaries

by Antonis Louka (University of Cyprus), Andreas Dionysiou (Frederick University), and Elias Athanasopoulos (University of Cyprus)

In today's programming landscape, ensuring software security is more critical than ever. Rust, a relatively new programming language, incorporates safety features that produce secure and efficient machine code without relying on runtime support. In our work, developed at the University of Cyprus, we explore how an attacker might deliberately create vulnerabilities in Rust binaries post-compilation, and the need for code validation for such systems.

Current programming systems can be split into safe and unsafe based on how they manage memory. Traditional (unsafe) systems like C/C++ are built using a loose memory management model that depends on the programmer to manage memory. Safe systems on the other hand, such as Java and C#, use heavy runtime support to perform memory management accounting and ensure safety. Rust, a new programming language, enforces memory safety without runtime support, providing a new middle ground between the two aforementioned categories. Other similar efforts are Go, which depends on a lightweight garbage collector, and Swift, which uses automatic reference counting.

Specifically, Rust uses the Ownership, Borrowing, and Lifetime concepts to create Rust-specific rules to automate memory management and avoid temporal-safety bugs like use-after-free (UaF) or double free (DF) bugs. Such rules are enforced by a Rust compiler routine called the "Borrow Checker". Additionally, Rust adds automatic checks in the program that are essentially conditional statements during compilation for avoiding spatial-safety bugs, such as buffer overflows (BO). Overall, Rust's safety measures are enforced during compilation, leaving the binary without any guarantees during execution other than trusting the compiler's output. These measures can be also bypassed using the unsafe keyword that is mainly used to integrate C code in Rust programs.

In our work, we argue that the machine code produced by a Rust compiler should be validated for safety before reaching the end user. Such a validation process is not new, but has been applied in the past when web browsers supported running third-party native code using the NaCl framework [1]. More precisely, we argue that when a binary is submitted to an app store, there should be a validating routine along with code-reviewing mechanisms that assess if the given binary adheres to memory-safety Rust-specific rules.

We demonstrate that an attacker can stealthily modify a Rust binary and recreate temporal and spatial safety bugs. The machine code produced by the Rust compiler can later be altered to introduce such bugs into the binary that reaches the end user. This provides an opportunity for malicious developers to compile an application and stealthily add malicious bugs for later exploitation (see Figure 1).

This is a similar scenario to that depicted by Wang et al. [2], and the creation of Jekyll apps (intentionally malicious applications that bypass the defence mechanisms of the IOS store). In this case (i) the compiler cannot be trusted to be the only entity that ensures memory safety for such systems, (ii) the digital signature measures cannot detect if the application has been tampered with or not since the developer is the actual attacker, and (iii) current review mechanisms cannot capture such alterations as they do not follow Rust's memory safety rules. The

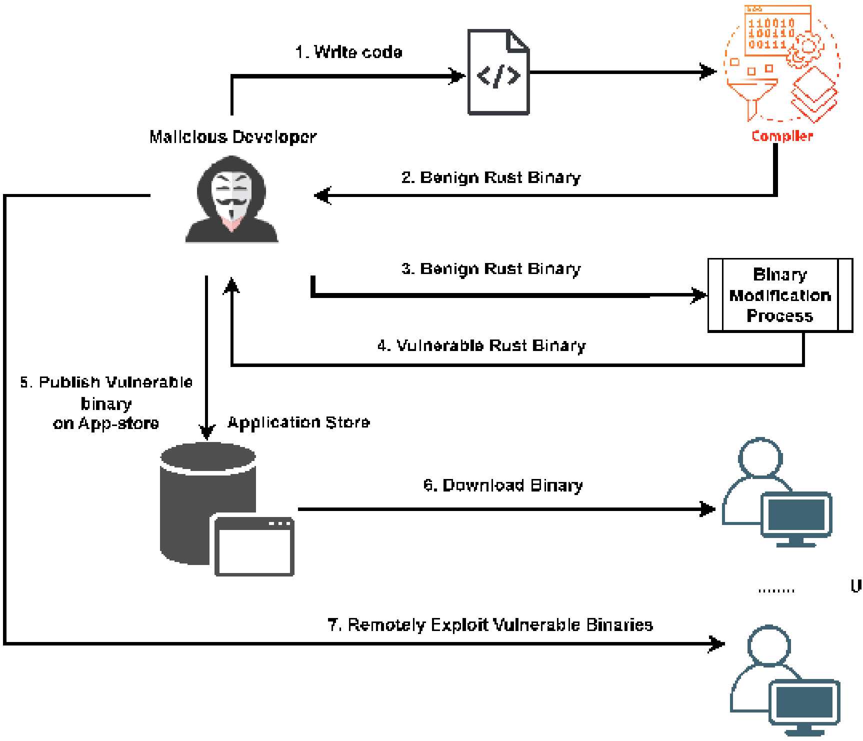


Figure 1: Threat model diagram: A malicious developer creates a benign Rust application that successfully passes the compilation phase. The developer then uses a modification process that alters the binary and creates a bug i.e. BO, UaF making the binary vulnerable. Finally, the developer publishes the vulnerable binary to an application store that does not utilise language-specific validation, bypassing the different reviewing mechanisms. After multiple users download and install the malicious application, the developer can exploit the vulnerable binaries remotely.

aforementioned reasons mandate the creation of validation routines that check specific artifacts in an executable that are language-specific and fundamental to enforce memory safety.

The validation methodology proposed in this work focuses on Rust binaries only, and during our experiments, we demonstrate that certain Rust safety measures, such as checks for BOs, can be more easily manipulated, while others, such as UaF, need more elaborate modifications. We demonstrate that by stealthily changing the conditional statements added by the compiler, traditional BOs can be created and used to corrupt stack control data, i.e. return addresses. We additionally show that fundamental Rust rules enforced by the Borrow Checker can be bypassed to create dangling pointers and UaF bugs.

We develop a preliminary validation framework that uses heuristics and other specific patterns of Rust, i.e. exception-handling patterns and buffer initialisation patterns, and we focus on the validation of buffer-overflow checks and buffer references. Our framework validates buffer-overflow checks by comparing them with a ground truth conditional check (created from analysing the binary). For reference validation, we use data-flow tracking from a given buffer (source) to other aliases (references) while simulating Rust's rules and checking for any violations.

The ability to validate code for such programming systems is crucial not only for scenarios as described in this article. Rust and Go are currently very popular and attractive choices for creating system applications. Many developers have started migrating code from C/C++ to Rust to benefit from memory-safety guarantees. Validating such code is also not trivial as unsafe blocks of code need to be handled differently than regular safe code. This information is not present in a compiled binary, which creates the need to add validation symbols in an executable to assist future validation frameworks. On the other hand, some work has demonstrated that concepts used to automate memory allocation and deallocation may produce bugs, especially when interacting with unsafe Rust code [3]. Validation frameworks may be able to capture such bugs before the application reaches the end user.

Our research introduces an initial validation approach specifically targeting Rust buffers in binaries; however, much work remains to develop a comprehensive validation framework for the new era of programming systems such as Rust.

References:

[1] B. Yee et al., "Native client: a sandbox for portable, untrusted x86 native code," in 30th IEEE Symposium on Security and Privacy, 2009, pp. 79–93.
[2] T. Wang, et al., "Jekyll on iOS: when benign apps become evil," in 22nd USENIX Security Symposium (USENIX Security 13), 2013, pp. 559–572.
[3] M. Cui and Y. Zhou, "SafeDrop: Detecting memory deallocation bugs of rust programs via static data-flow analysis," ACM Transactions on Software Engineering and Methodology, vol. 32, no. 4, pp. 1–21, 2023.

Please contact:

Elias Athanasopoulos, University of Cyprus
athanasopoulos.elias@ucy.ac.cy

The Salto Project: Static Analysis of OCaml Programs by Abstract Interpretation

by Pierre Lermusiaux and Benoît Montagu (Inria)

Functional programming languages, such as OCaml, take advantage of strong static guarantees provided by their type checkers that ensure that well-typed programs cannot "go wrong". However many correctness and safety properties escape the scope of the guarantees provided by the type system. In order to provide some additional guarantees, the Salto project leverages static analysis techniques, mainly based on abstract interpretation, to develop tools that help OCaml programmers increase the confidence in their programs.

With the wide-spread use of cyber-systems, the programs controlling these systems have become the keepers of our personal data. Malfunctioning programs expose all these systems to attacks and faults that can put our data and our lives at risk. Thus, the questions of safety and correctness of the programs and protocols used in critical systems are now essential issues for the future of information technologies. Providing formal guarantees on the behaviour of programs is, however, a challenging problem that often requires advanced scientific knowledge, which limits the adoption of software analysis tools in industry.

While dynamic techniques based on tests and simulations have been preferred so far in industry, the adoption of abstract-interpretation-based tools, such as Astrée [1], a static analyser for C programs, have proven the interest and applicability of static and formal analysis techniques. Recently, NomadicLabs [L3] has shown interest in using software analysis tools to improve the confidence on the large OCaml code base behind the Tezos crypto-currency. This prompted a collaboration between NomadicLabs and Inria to fund the Salto project [L1] that officially started in November 2022.

OCaml [L2] is a functional programming language with automatic memory management, which is developed and maintained by Inria researchers. It's strong type system guarantees that a well-typed OCaml programs are memory safe, i.e. that data is used in a consistent way by the program. However, potential errors, such as mishandled exceptions, arithmetic overflow, out-of-bound accesses, and the detection of undefined behaviours escape the scope of these guarantees. The goal of the Salto project is to define sound static analysis techniques to formally detect such cases.

The formal verification of non-trivial properties of a program is a notoriously undecidable problem, meaning that there is no general method able to systematically prove or disprove semantics properties of a program. To circumvent this formal limitation of static analyses, the abstract interpretation framework proposes to compute an over-approximation of a program's reachable states. An abstract interpreter can do so effi-

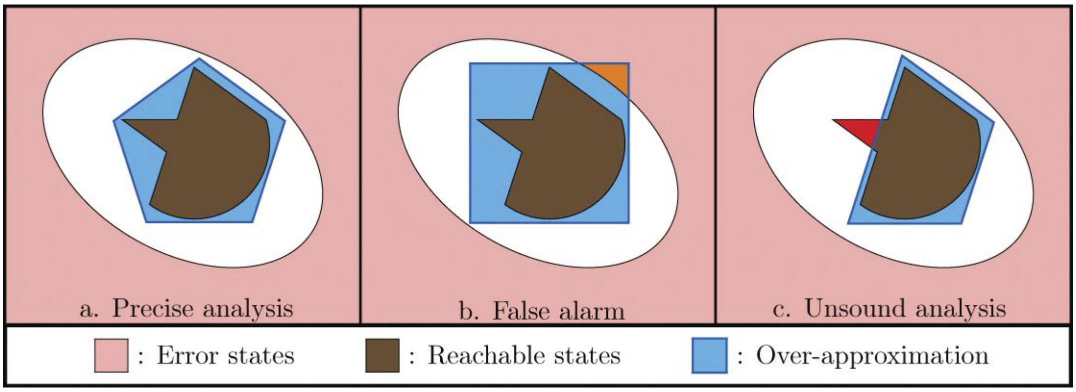


Figure 1: Representation of three possible analysis results.

ciently by relying on the notion of abstract domains that define over-approximations of the sets of possible states of a program, called abstract values, and the abstractions of the semantic operations of a program on these abstract values.

This approach allows abstract interpreters, like Astrée and Salto, to accurately detect all executions that might contradict the desired semantic properties (i.e. the analysis is sound), while sometimes raising false alarms about executions that cannot happen in reality (see Figure 1). The goal of an abstract interpreter is therefore to be as precise as possible to minimise the number of false alarms, while being efficient enough to analyse programs in a reasonable time. In the context of functional languages such as OCaml, applying the abstract interpretation framework is particularly challenging, because data flow and control flow are interdependent: the control flow of programs is dynamically derived from the application of functional terms, whose values generally depend on all evaluations performed before. While other work [3] has relied on techniques such as inference of types and effects to statically analyse OCaml programs, our work on Salto constitutes one of the first efforts to propose a wide-scale analysis of OCaml programs based on abstract interpretation.

To define and implement this static analyser, the Salto project has led to the definition of an intermediate language, where redundant features of the OCaml language are conflated, and disambiguation of pattern-matching is performed. This intermediate language simplifies both the definition of the semantics of programs, and the implementation of our abstract interpreter. The abstract interpreter relies on a novel abstract domain that can represent recursively defined sets of values, including function closures. This abstract domain features a dual representation of abstract values, that can be viewed both as tree terms, which can be hash-consed and memoised to ensure the analysis performance, as well as graphs, in order to implement tree-automaton-inspired operations on the set of values they represent.

The Salto project has reached a first milestone with the implementation of a static analyser for OCaml programs, parameterised over the abstract domains of OCaml primitive datatypes, that can detect uncaught exceptions accurately [2], and arithmetical errors, such as integer overflow. This analyser has been tested on a large test suite, and already supports a large subset of features of OCaml, including dynamically extensible data-types, mutable records, and the module system. The support in Salto of more advanced OCaml features is under active development (such as recursive non-functional terms and modules), that will permit to analyse the Tezos code base ultimately. Finally, we believe that the Salto abstract interpreter constitutes a solid basis to experiment with analyses that detect more advanced security issues in OCaml programs, such as undesirable side effects, resource leaks, or cases where the behaviour of a program may depend on the evaluation order chosen by the compiler.

Links:

[L1] Salto: https://salto.gitlabpages.inria.fr
[L2] OCaml: https://ocaml.org
[L3] NomadicLabs: https://www.nomadic-labs.com

References:

[1] P. Cousot et al: “The ASTREÉ Analyzer,” ESOP 2005.
[2] P. Lermusiaux and B. Montagu, “Detection of uncaught exceptions in functional programs by abstract interpretation,” ESOP 2024.
[3] X. Leroy and F. Pessaux, “Type-based analysis of uncaught exceptions,” ACM TOPLAS, vol. 22 Issue 2, March 2000.

Please contact:

Pierre Lermusiaux, Inria, France
pierre.lermusiaux@inria.fr

Benoît Montagu, Inria, France
benoit.montagu@inria.fr

Generating Mixed Boolean-Arithmetic Expressions through Equality Saturation

by Caroline Lawitschka and Sebastian Schrittwieser (University of Vienna)

We introduce a novel methodology for generating complex and robust Mixed Boolean-Arithmetic (MBA) expressions for various software protection methodologies. Our research specifically focuses on leveraging the concept of equality saturation to create MBA expressions of arbitrary complexity.

Mixed Boolean-Arithmetic (MBA) expressions combine Boolean logic operations such as AND, OR, XOR, and NOT with arithmetic operations like addition (+), subtraction (-), multiplication (*), and division (/). This combination provides a powerful tool for creating complex expressions that can be used in code obfuscation.

Code obfuscation is a broad term encompassing techniques used to make program code more difficult to understand, analyse, and reverse-engineer, thereby protecting intellectual property and preventing software tampering. In the software protection domain, MBA expressions are often used to generate opaque predicates, which always evaluate to either true or false, independently from their inputs. While the software developer knows the evaluation result of the opaque predicate, it is time-consuming for an attacker to figure out its input-independence and the correct evaluation result. In practice, opaque predicates help create fake branches with paths that are never executed at runtime. The effectiveness of MBA in obfuscation techniques has been well-documented in the literature. For instance, Zhou et al. [1] demonstrated how MBA expressions can be utilised for information hiding in software, highlighting the robustness of MBA-based obfuscation methods against various analysis techniques.

However, MBA-based obfuscations are not invincible. Recent years have seen significant efforts to develop methods for breaking and simplifying MBA expressions. These efforts are driven by the need to analyse and understand obfuscated code, whether for debugging purposes, reverse engineering, or detecting malicious software. As obfuscation methods become more sophisticated, so do the techniques for breaking them.

In our research at the Christian Doppler Laboratory for Assurance and Transparency in Software Protection [L1], we focus on a novel methodology for generating strong and verifiable MBA expressions based on the concept of equality saturation. In particular, we build e-graphs for the generation of arbitrarily complex MBA expressions.

In 2009, Tate et al. [2] proposed equality saturation as a new optimisation technique which is based on the e-graph data structure. An e-graph is a graph structure that represents a congruence relation over a set of expressions. It consists of equivalence classes (e-classes), each containing equivalent nodes (e-nodes). E-nodes are unique, and the children of e-nodes are references to e-classes. Term rewriting involves transforming a term based on a set of possible rewriting rules without losing its semantics. E-graphs can efficiently express these equivalent terms.

Equality saturation is a technique for program optimisation based on the e-graph structure. Simply put, equality saturation explores all expressions possible within the existing rewrite rules and then selects the best one (e.g. fastest) for a specific use case based on a cost function. The advantage over iterative term rewriting is that equality saturation can find global optima.

The fundamental idea behind our novel approach is to construct strong opaque predicates – expressions that always evaluate to true or false but are difficult to analyse – from simple

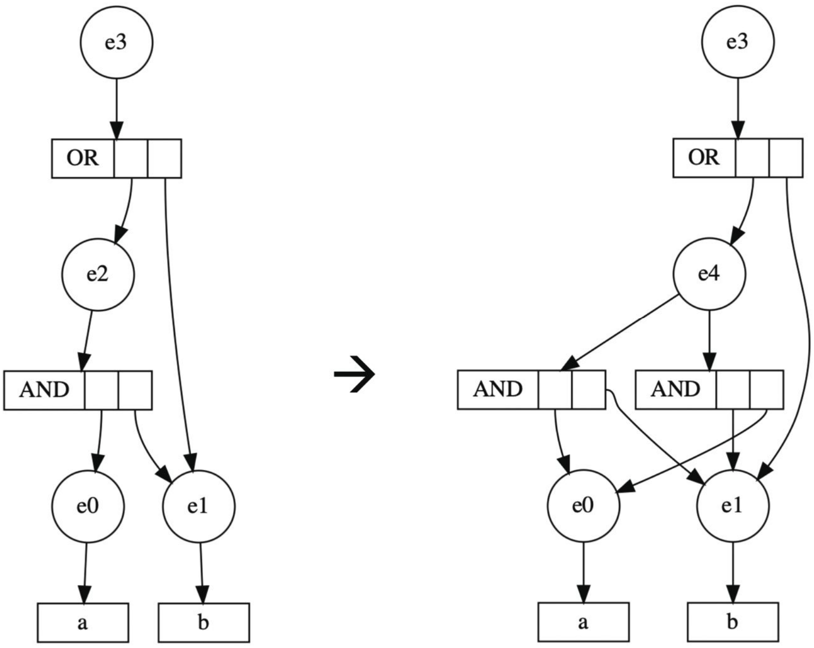


Figure 1: Building an e-graph with one simple rewriting rule.

MBA expressions. Using equality saturation, these simple expressions are transformed into more complex ones that are semantically equivalent but much harder to understand and analyse. This transformation is guided by an e-graph structure, where rewriting rules are applied iteratively to generate increasingly complex expressions. While e-graphs and equality saturation have previously been used for deobfuscation of MBA expressions, we revere this process and generate arbitrarily complex expressions from simple ones.

Our research involved creating a comprehensive set of rewriting rules that describe equivalent expressions, ranging from basic arithmetic properties to more complex transformations. For example, a simple rewriting rule might describe the commutative property of multiplication, where the expression (a * b) can be rewritten as (b * a). A more complex rule could involve the transformation of an expression involving bitwise negation, such as rewriting ~(a + b) as (~a + ~b + 1). These rewriting rules were manually and systematically generated to cover a wide range of possible transformations, ensuring that the resulting MBA expressions are both complex and semantically equivalent to the original expressions.

To create arbitrarily complex MBA expressions starting from a randomly chosen simple expression, an increasingly complex e-graph is generated iteratively by applying these rewriting rules. Figure 1 illustrates the concept using a trivial example. Our starting point for the construction of a complex MBA expression is the term (a & b) | b. The only rewriting rule applied to the e-graph is the commutative rule (a & b) = (b & a). Iteratively, many more rewriting rules are applied until a complex e-graph containing all possible equivalences emerges. Each iteration of this process results in a more complex expression, ultimately leading to an MBA expression that is highly resistant to analysis.

In conclusion, our research introduces a novel methodology for generating complex and verifiable MBA expressions by leveraging the concepts of e-graphs and equality saturation, providing a robust solution for generating arbitrarily complex opaque predicates used for software protection.

Links:
[L1] https://cdl-astra.at

References:

[1] Y. Zhou, et al. " Information hiding in software with mixed boolean-arithmetic transforms," in Int. Workshop on Information Security Applications, pp. 61–75, Springer, 2007.

[2] R. Tate, et al. "Equality saturation: a new approach to optimization", in Proc. of the 36th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 264–276), 2009.

Please contact:

Caroline Lawitschka, Christian Doppler Laboratory for Assurance and Transparency in Software Protection, Faculty of Computer Science, University of Vienna, Austria
caroline.lawitschka @univie.ac.at

GLITCH: Polyglot Code Smell Detection in Infrastructure as Code

by Nuno Saavedra, João F. Ferreira (INESC-ID and University of Lisbon) and Alexandra Mendes (INESC TEC and University of Porto)

GLITCH is a versatile tool designed for detecting code smells in Infrastructure as Code (IaC) scripts across multiple technologies. Developed by researchers from INESC-ID (Lisbon), INESC TEC (Porto), Instituto Superior Técnico / University of Lisbon, and the Faculty of Engineering / University of Porto, GLITCH automates the detection of both security and design flaws in scripts written in Ansible, Chef, Docker, Puppet, and Terraform. By using a technology-agnostic framework, GLITCH aims to improve the consistency and efficiency of code smell detection, making it a valuable resource for DevOps engineers and researchers focused on software quality.

As Infrastructure as Code (IaC) scripts become increasingly critical for automating IT infrastructure, ensuring their consistency and security is essential [1, 2]. Code smells – patterns that suggest potential issues – can lead to vulnerabilities, inefficiencies, and maintenance difficulties. GLITCH addresses these concerns by detecting a wide range of smells across multiple IaC languages, including security, design, and implementation flaws.

The GLITCH project is supported by a collaboration of researchers focused on enhancing software engineering practices. Their combined expertise aims to improve the quality and security of software systems through the development and application of this technology-agnostic tool.

Motivation

As organisations increasingly rely on IaC to automate and manage their infrastructure, the risk of introducing issues or vulnerabilities through poorly written or insecure code has become a significant concern. For example, Facebook's outage in 2021 was triggered by a misconfiguration in their internal backbone network, leading to disruptions in communication between data centres. This issue disconnected Facebook's services and tools, making it impossible to diagnose and resolve the problem quickly. During the day of the outage, shares in the company dropped by nearly 5% and Facebook CEO Mark Zuckerberg's wealth fell by more than $6 billion. According to a report produced by Fortune and Snopes, Facebook lost at least $60 million in advertising revenue. In the developing world, the outage disrupted daily life as Facebook's platforms are key for communication. In conflict zones like Syria, aid workers relied on WhatsApp to share bombing locations for safe travel but were unable to do so during the outage.

Traditional tools for detecting code smells in IaC scripts often focus on specific technologies and are developed independently, leading to inconsistencies and a lack of comprehensive coverage. GLITCH was conceived to address these gaps by of-

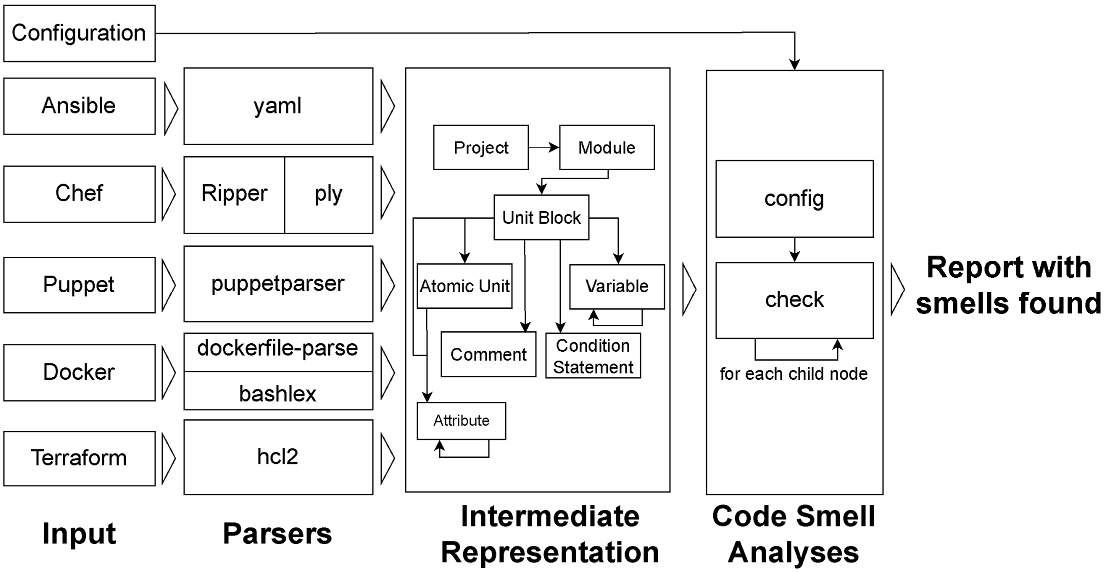


Figure 1: GLITCH's architecture overview.

fering a unified, technology-agnostic approach to code smell detection. By detecting and mitigating code smells early in the development process, GLITCH aims to prevent potential security vulnerabilities, reduce maintenance costs, and enhance the overall quality of IaC scripts.

GLITCH is open-source [L1] and its development began in 2022. As the project progressed, it expanded to include design and implementation smells and extended its support to Docker and Terraform [2]. Preliminary results have already demonstrated GLITCH's effectiveness in detecting a broad range of code smells. The project is ongoing, with continuous improvements being made to enhance its capabilities and extend its applicability to new IaC technologies.

Aim and Techniques Employed

By providing a unified approach to code smell detection, GLITCH seeks to reduce the effort required to develop and maintain secure and high-quality IaC scripts. Additionally, the project aims to create large datasets of IaC scripts and code smells that can be used by researchers and practitioners to further advance the field of software quality and security.

GLITCH approaches code smell detection by transforming IaC scripts into an intermediate representation. This representation captures common concepts across different IaC technologies, enabling the development of code smell detectors that are not tied to any specific language or tool. The framework uses both rule-based and algorithmic approaches to detect various smells. For security smells, a rule-based approach is used, while design and implementation smells are detected using more complex algorithms. The tool traverses the intermediate representation of the scripts, applying multiple analyses to identify all potential code smells. Figure1 shows GLITCH's architecture overview.

Academically, it contributes to the field of software engineering by providing a novel, unified framework for code smell detection in IaC scripts. Practically, it offers a valuable tool for DevOps engineers, system administrators, and organisations that rely on IaC to manage their infrastructure. By improving the security and quality of IaC scripts, GLITCH helps organisations prevent potential vulnerabilities and reduce the costs associated with maintaining complex infrastructure systems.

Future Activities

The team aims to refine the existing detection mechanisms to improve their precision and recall. Another significant future activity is the creation of oracle datasets for design and implementation smells, similar to those already developed for security smells. These datasets will be used to evaluate the effectiveness of GLITCH and to facilitate further research in the field. Current efforts also include automated program repair of IaC scripts at the level of GLITCH's intermediate representation.

In conclusion, GLITCH represents a significant step forward in the field of software security, particularly in the context of IaC. By providing a comprehensive, technology-agnostic tool for code smell detection, GLITCH helps ensure that IaC scripts are secure, consistent, and maintainable, ultimately contributing to the integrity and reliability of the software systems they support.

Link:
[L1] https://github.com/sr-lab/GLITCH

References:

[1] N. Saavedra and J. F. Ferreira, "GLITCH: automated polyglot security smell detection in infrastructure as code," in Proc. of the 37th IEEE/ACM Int. Conf. on Automated Software Engineering, pp. 1–12, 2022.
[2] N. Saavedra, et al., "Polyglot code smell detection for infrastructure as code with GLITCH," in Proc. of the 38th IEEE/ACM Int. Conf. on Automated Software Engineering (ASE), pp. 2042–2045, IEEE, 2023.

Please contact:

João F. Ferreira, INESC-ID and IST, University of Lisbon, Lisbon, Portugal
joao@joaoff.com

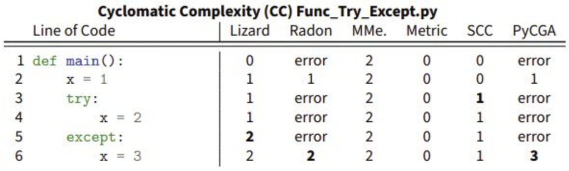
Comparability of Software Metrics and Estimating the Strength of Software Protections

by Patrick Kochberger, Philipp Haindl (St. Pölten University of Applied Sciences), Matteo Battaglin and Patrick Felbauer (University of Vienna)

In this project, we investigate how different tools measure code complexity of software protections, revealing significant variations in their results. While simpler metrics like lines of code (LOC) often produce similar outcomes, more advanced metrics such as cyclomatic complexity (CC) and maintainability index (MI) show major differences across tools. These discrepancies highlight the need for better methodologies when assessing the effectiveness of obfuscation techniques for protecting software.

In software development and software and program protection, techniques for estimating the complexity of source and/or program code are an important tool. On the one hand, when writing code, keeping it simple, easy to understand and employing efficient programming paradigms is a key requirement for long-term maintainability. On the other hand, in software protection, the goal is to make it as hard as possible to understand programs to increase the resilience against intellectual theft, tampering and exploitation. In this and the ongoing work at the St. Pölten University of Applied Sciences and the Christian Doppler Laboratory AsTra [L1], we focus on advancing the methodological foundations of software protection, particularly against adversary-at-the-end attacks, by studying and comparing code complexity metrics. Code metrics [1] are functions used for measuring certain properties of software. They are objective, reproducible and quantifiable measurements to gauge characteristics, e.g. the complexity of code. The resulting information is useful in a variety of applications of software development, including quality assurance, estimating performance, and costs. Specifically, these metrics are used to identify code which is overly complex, and not maintainable.

In software protection [2], code obfuscation is used to protect benign applications against attacks such as illegal distribution, intellectual theft, tampering, and exploitation. The idea is to intentionally increase the complexity of code in such a way that some functionality is more difficult to detect and/or understand. The goal is to make analysis of the application more time-consuming and difficult. To assess obfuscation techniques and obfuscation and deobfuscation tools and evaluate them against each other, one possible option is to compare the code complexity by measuring and calculating code complexity metrics.



Cyclomatic Complexity (CC) Func_Try_Except.py

Line of Code		Lizard	Radon	MMe.	Metric	SCC	PyCGA
1	`def main():`	0	error	2	0	0	error
2	`    x = 1`	1	1	2	0	0	1
3	`    try:`	1	error	2	0	**1**	error
4	`        x = 2`	1	error	2	0	1	error
5	`    except:`	**2**	error	2	0	1	error
6	`        x = 3`	2	**2**	2	0	1	**3**

Figure 1: A small excerpt from our results on how six of the tools measure the cyclomatic complexity when facing a try-except construct in Python.

In a first step towards comparable measurements, we performed a study on five different groups of code complexity metrics: Lines Of Code (LOC), Cyclomatic Complexity (CC), Halstead (Hal), Maintainability Index (MI), COnstructive COst MOdel (COCOMO), implemented and measured by 12 different tools. The LOC-related metrics quantify the size of a program by counting the number of lines. Each concrete metric and implementation has its own rules regarding blank lines, comment lines, import lines, and non-executable lines. The CC measures the amount of decision logic based on the software's control flow graph. The Hal software metrics are a set of measurements based on the number of distinct operators and operands. The Halstead Difficulty is related to the complexity of the program. The MI is a weighted combination of Hal, CC, and LOC metrics. The COCOMO estimates software project effort and costs in person-months. The model relies on additional information, aside from the source code, such as the type of software project and its cost drivers to calculate its results. We surveyed the availability and differences among several code metrics across individual implementations in several tools. For a detailed investigation, we constructed a dataset containing different test cases and implemented an analysis framework which feeds the program code as a whole or piece-by-piece to the tools calculating the metric. In the piece-by-piece mode (e.g. line-by-line in Figure 1) the framework helps to find out which individual lines of code affect the overall metric score.

Our results indicate that the comparability of the tools' results varies significantly depending on the specific metric in question. For example, for the relatively simple LOC and source lines of code (SLOC) metrics, the tools calculated the most similar values, although even with these metrics, some tools had significant divergence. For the other studied metrics, some of the results were so different as to render them incomparable. Therefore, it is essential to not only specify the metric but also the tool which performed the measurements.

We plan to analyse even more code complexity metrics and want to evaluate their performance and capability in evaluating the strength of software protections. The goal is to find and develop measurements to compare code on different levels (source, intermediate, and binary code) to estimate the strength of protections before and after the compilation step as well as the effect of different compilers and optimisation passes.

Link: [L1] https://cdl-astra.at

References:

[1] A. S. Nuñez-Varela, et al., "Source code metrics: a systematic mapping study," J. of Systems and Software, vol. 128, pp. 164–197, 2017, doi: 10.1016/j.jss.2017.03.044.

[2] S. Schrittwieser, et al., "Protecting software through obfuscation: can it keep pace with progress in code analysis?," ACM Computing Surveys, 2017. https://doi.org/10.1145/2886012

Please contact:

Patrick Kochberger, St. Pölten University of Applied Sciences, Austria, patrick.kochberger@fhstp.ac.at

Interactive Fuzzing Reveals Zero-Day Vulnerabilities in Several MQTT Brokers

by Steffen Lüdtke, Roman Kraus and Martin Schneider (Fraunhofer FOKUS)

The growing number and diversity of cybersecurity attacks pose a challenge for developing secure systems, particularly in an age where many systems are connected to the internet. To facilitate early vulnerability detection, we propose an interactive fuzzing technique which employs grammars and supports genetic algorithms to interact with the SUT. This technique can generate inputs for specific attack scenarios, which are useful for finding new vulnerabilities and for assessing the completeness of patches. The proposed technique found zero-day vulnerabilities in established MQTT brokers within a few minutes.

Fuzzing and Genetic Algorithms

Fuzzing is a testing technique where a system under test (SUT) is confronted with randomly generated inputs to test for its robustness and security. Grammar-based fuzzing is a form of fuzzing which uses grammars to support the input generation [1]. Grammars can specify the structure of an input format, messages and message sequences and thus raise the probability of creating syntactically valid inputs and interactions with the SUT. This increases the likelihood of testing deeper functionalities compared to fully random input generation. Nevertheless, the random nature of fuzzing makes it difficult to efficiently generate inputs which fulfil a specific goal, e.g. inputs which particularly stress the resources of a SUT. Therefore, inputs for a given attack scenario (e.g. denial-of-service attacks) cannot be efficiently generated.

Genetic algorithms are an optimisation technique inspired by natural evolution. The idea is to start with a random set of solutions for a problem and to improve it through an iterative process of selection, recombination, and mutation. In the selection process, a subset of individuals is chosen based on a problem-specific fitness function to create the next generation (e.g. the top 30%). Their traits are then exchanged (recombination) and mutated to populate the next generation. This initially keeps advantageous traits and then develops them via recombination and mutation.

Genetic algorithms have been previously employed with fuzzing to, e.g. increase the code coverage. We employ genetic algorithms to create inputs and optimise them towards user-defined vulnerability indicators. This allows us to efficiently create inputs for specific attack types, which can be useful for finding new vulnerabilities and for assessing the quality of patches. We achieve this by specifying observable symptoms of a successful attack on the SUT instead of specifying the characteristics of the inputs, which are often more difficult to determine in advance.

MQTT and Mosquitto

Our case study is the Eclipse Mosquitto broker [L1]. This is a broker for the MQTT protocol, which is a prominent protocol for data exchange (e.g. telemetry) in the Internet of Things (IoT). MQTT implements a publish-subscribe model. This means that clients can publish messages under a specific topic to a broker which then forwards these messages to clients which have subscribed to that topic. Eclipse Mosquitto is a popular MQTT broker, which is implemented in C. It is therefore potentially vulnerable to memory errors like use-after-frees, segmentation faults or memory leaks. Mosquitto's wide usage, which includes industry settings, makes it a relevant case study.

Architecture

Figure 1 provides an overview of our fuzzing architecture. The test case generation is done by Fuzzino [L2], a fuzzing library developed by Fraunhofer FOKUS. It receives as input ABNF grammars, which describe individual MQTT message types and valid message sequences. Fuzzino uses these grammars to derive test cases, which each constitute one sequence of MQTT messages. Within the grammars we use a custom meta language, which models relationships that normally cannot be expressed with context-free grammars. For example, we use symbols which are resolved to dynamically calculated length fields or which define implicit relationships between symbols (e.g. between a password flag and a password field). This increases the validity of inputs we generate and thus increases the scope of functionalities we can test.

Once Fuzzino has derived an initial generation of test cases, it starts executing them against the MQTT broker, e.g. Mosquitto. For this, it first contacts a supervisor that starts and

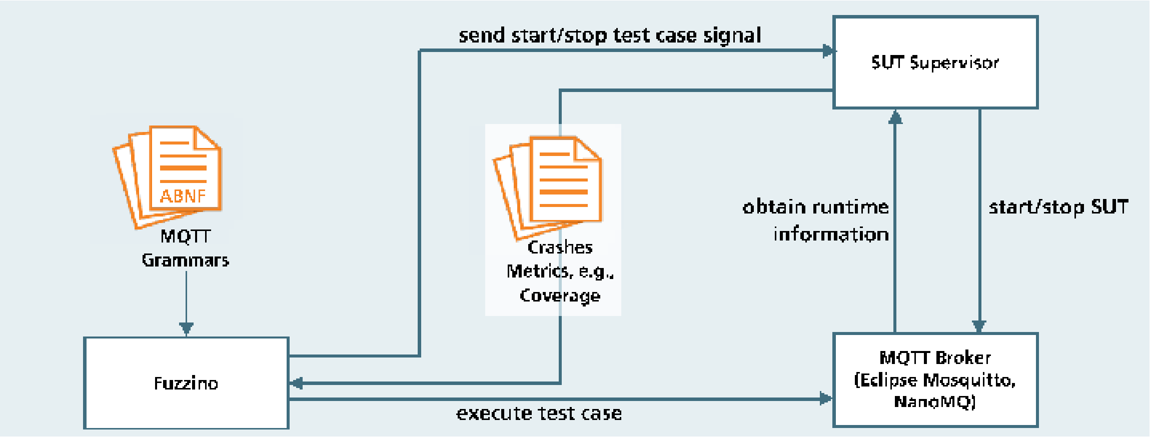

send start/stop test case signal
SUT Supervisor
ABNF
MQTT Grammars
Crashes Metrics, e.g., Coverage
obtain runtime information
start/stop SUT
Fuzzino
execute test case
MQTT Broker (Eclipse Mosquitto, NanoMQ)

Figure 1: The fuzzing architecture that includes Fuzzino, the supervisor for the SUT and the SUT, showing the interaction between these components.

stops the MQTT broker before and after each test case. This is necessary to match findings to individual test cases and to detect errors which only become evident on termination (e.g. memory leaks). We collect vulnerability metrics for each executed test case. This might be, e.g. the time which the broker needs to respond or the code coverage we achieve. Some of these metrics can be measured directly (e.g. the response time), while others require additional monitoring systems or instrumentations. Memory errors for instance are detected with the help of AddressSanitizer, which is an instrumentation that monitors memory accesses at runtime. The recorded metrics and findings are reported back to Fuzzino via the supervisor. Fuzzino then uses this data to calculate its fitness scores, produce the next generation and repeat this process until a predefined number of generations is reached.

Evaluation

Our evaluation on Mosquitto suggests that our approach can be effective at systematically increasing targeted vulnerability indicators compared to fuzzing without genetic optimisation, e.g. in terms of the response time. Moreover, it could also be used to support the development of patches by producing diverse inputs for a vulnerability to ensure that the patch inhibits all ways a vulnerability may be triggered. For this, we use a fitness function which targets locations that trigger the vulnerability while also rewarding coverage diversity.

Furthermore, unguided, i.e. non-genetic fuzzing, was also effective. Namely, it found a bug within the current release of Mosquitto (2.0.18) in just a few minutes, which was confirmed to be a zero-day vulnerability. Furthermore, it found two unknown bugs on NanoMQ [L3], another MQTT broker we analysed. This highlights that unguided fuzzing can be especially useful if you want to broadly search for new vulnerabilities without specific attack scenarios or metrics in mind.

Conclusion

The combination of grammar-based and genetic fuzzing is a powerful technique which can support testing systems with specific attack scenarios in mind. This might be the simple optimisation towards certain vulnerability indicators or more complex scenarios where a test suite is derived based on a specific vulnerability. Unguided fuzzing can complement this approach as a technique to analyse the security of a system without other goals in mind.

Links:
[L1] https://mosquitto.org
[L2] https://www.fokus.fraunhofer.de/de/sqc/security_testing
[L3] https://nanomq.io

Reference:

[1] S. Lüdtke, et al., "Attack-based automation of security testing for IoT applications with genetic algorithms and fuzzing," IEEE QRS, doi: 10.1109/QRS-C55045.2021.00023

Please contact:

Martin Schneider, Fraunhofer FOKUS, Berlin, Germany
martin.schneider@fokus.fraunhofer.de

Roman Kraus, Fraunhofer FOKUS, Berlin, Germany
roman.kraus@fokus.fraunhofer.de

Verifying Code Correctness of Protected Software through Translation Validation

by Sebastian Schrittwieser (University of Vienna)

Software protection has evolved over the past three decades, but ensuring the correctness of protective code transformations remains a challenge. Our novel approach breaks down complex obfuscation techniques into smaller, manageable components and implements them as compiler passes. By using translation validation, the correctness of each transformation is ensured, resulting in more reliable and robust software protections.

Over three decades of software protection research and development have brought various protection techniques against code analysis, modification, and theft of binary code. However, a significant challenge is verifying the correctness of protective code transformations. While a lot of basic research has been done in the past in compiler correctness to ensure that functionality in binary programs matches the functionality described in the source code, there is no methodological basis for ensuring the correctness of code transformations for software protection.

Usually, software protections, such a code obfuscation, are applied at source code stage. However, in 2015 it was first shown that applying software protections directly in the compiler is feasible. Using compile-time obfuscation, we are able move the question of correctness of code transformations into a research area where we can build on a substantial body of prior work and methodologies. With Alive [L1] Lopes et al. introduced a refinement checker for the LLVM compiler infrastructure in 2015. The basic idea of so-called translation validation [1] is to compare an unoptimised code block with an optimised version and either prove that the optimised version is a refinement of the unoptimised one, or that the refinement fails in at least one particular case using concrete input values as a counterexample. In the past, Alive was successfully used to identify several bugs in optimisation passes of LLVM, and developers of optimisations are encouraged to validate them before committing them to the LLVM code base.

In our research at the Christian Doppler Laboratory for Assurance and Transparency in Software Protection [L2], we build upon this solid foundation for translation validation to verify the correctness of obfuscating passes in LLVM. One major research challenge is reducing the complexity of transformations. While, in theory, Alive covers a large number of language constructs of the LLVM intermediate representation (IR), there are certain language constructs that are known to prevent the underlying SMT solver from solving the query in a reasonable time frame. These include floating points, wide integer divisions, and some complex memory operations. Thus, atomic implementations of typical real-world obfuscation techniques such as virtualisation or control flow flattening would generate code transformations that are too complex for

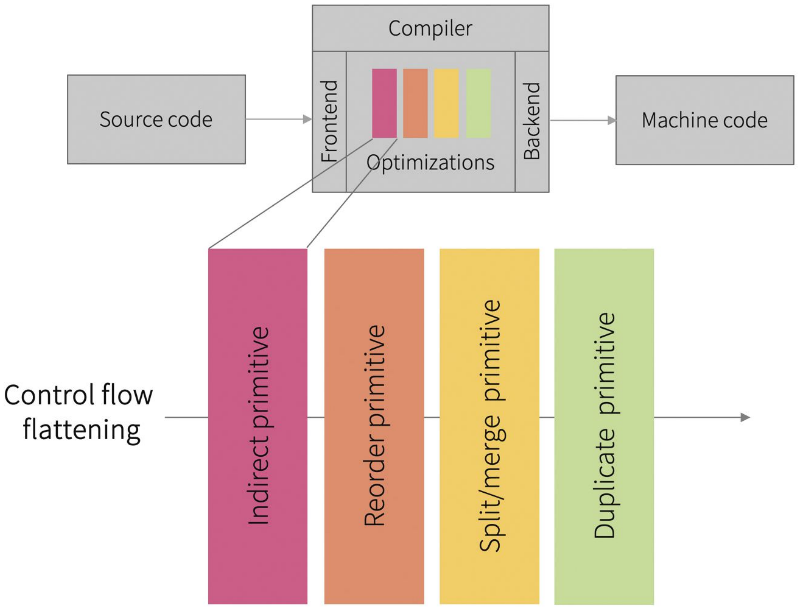


Figure 1: Control flow flattening obfuscation constructed from multiple compiler passes.

translation validation. For this reason, we deconstruct complex software protection methods into their fundamental protection principles, which can be implemented as separate compiler passes for which translation validation is feasible. Nagra and Collberg [2] have defined eleven software protection primitives that formalise basic protection principles. These primitives range from covering an object worth protecting with another one (cover primitive) to duplicating an object to artificially increase the set of objects to be analysed (duplicate primitive) to detecting and responding to code tampering (detect-responds primitive). Starting from these eleven primitives, we create a set of basic obfuscation passes for LLVM and use a refinement checker for translation validation. Each pass can be parameterised to concretise its functionality. For example, for the detect-responds primitive different methods for the detection of code tampering can be selected such as calculating hash values of code sections.

These compiler passes can now be combined in any way to construct strong software protections. Figure 1 shows the basic concept using the example of the control flow flattening obfuscation technique. In this technique, the control flow graph of a program is modified so that all basic blocks are at the same hierarchical level. After executing a basic block, the control flow is directed to a central dispatcher, which determines the next basic block based on the current state of the program and jumps to it. Rather than implementing this technique atomically, we assemble it from a combination of small compiler passes whose functionality is further specified by parameterisation. The indirect primitive is used to construct the dispatcher. The reorder primitive changes the order of the basic blocks so that no conclusions can be drawn about the functionality of the compiled binary based on local proximity of basic blocks in the code. Finally, the split/merge and duplicate primitives can be used to further split larger basic blocks and duplicate blocks to increase the total number of blocks and thus make reverse-engineering more time consuming.

Currently, most of our transformations are deterministic, with the notable exception of the reordering primitive, which allows the target structures to be randomly shuffled. Usually, determinism is a desirable goal for compilers. However, for binary software protections that do not perform inherently irreversible code transformations, it may be useful to promote code diversity through randomness. The verification of code correctness is not affected by the use of randomness, since translation validation always validates concrete instances of code transformations using refinement checkers instead of validating the implementation of the transformation pass. In our future work, we thus intend to investigate which forms of code diversity can be implemented and what effect they have on the strength of the resulting software protection.

Links:
[L1] https://alive2.llvm.org
[L2] https://cdl-astra.at

References:

[1] A. Pnueli, M. Siegel, and E. Singerman, “Translation validation,” in Tools and Algorithms for the Construction and Analysis of Systems: 4th International Conference, TACAS'98, held as part of the ETAPS'98 Lisbon, 1998, Proc. 4, pp. 151–166, Springer.
[2] J. Nagra and C. Collberg, “Surreptitious software: obfuscation, watermarking, and tamperproofing for software protection,” Pearson Education, 2009.

Please contact:

Sebastian Schrittwieser, Christian Doppler Laboratory for Assurance and Transparency in Software Protection, Faculty of Computer Science, University of Vienna, Austria
sebastian.schrittwieser@univie.ac.at

A Framework for the Analysis of Physical Unclonable Function Interfaces

by Chenglu Jin (CWI) and Marten van Dijk (CWI and Vrije Universiteit)

For a long time, process variations in the manufacturing of computer chips has been a big hurdle for producing high-quality products. However, one can turn these imperfections caused by the process variations into something good: into unique random functions that are impossible to clone even by the original manufacturer. At CWI in Amsterdam, we are building a solid foundational understanding of these security primitives and bringing them closer to practice. This could be very interesting for use in computer systems and embedded systems, like cloud servers and controllers in critical infrastructures.

A Physical Unclonable Function (PUF) is a hardware security primitive, and its input-output behaviour (also called Challenge-Response Pairs or CRPs) depends on the uncontrollable and unclonable manufacturing process variation introduced in the manufacturing process of a semiconductor chip [1]. Hence, even the same manufacturer cannot physically clone a fabricated PUF device with the same behaviour.

One classical PUF design is Arbiter PUF (originally introduced by Marten van Dijk and his collaborators), as shown in Figure 1. It consists of a chain of switch components and ends with an arbiter. Each switch component propagates the two inputs on the left to the two outputs on the right via either the crossed paths or the parallel paths, depending on the value of the top selection bit input. Since the delay difference between the selected pair of paths is designed to be as small as possible, the main factor that will affect the delay difference is the manufacturing process variations. To evaluate an Arbiter PUF, the user first provides challenge bits to the selection bits of the switch chain and then sends a pulse signal from the left end of the switch chain. Every challenge-response pair should only depend on the process variation in the delay difference selected by the challenge bits.

This unclonable feature allows PUF to enable many security applications. The most traditional use case of PUF is device authentication. Since there are no two PUFs in the world that have the exact same input-output behaviour, a verifier is able to remotely check whether the counterparty holds the PUF by checking whether the prover can send the responses to a few random challenges chosen by the verifier. The verifier just needs to compare the responses from the prover with the responses stored in its CRP database collected when it holds the PUF. If the prover does not hold the PUF at the moment, the prover will not be able to generate the correct responses, even if it had access to the PUF before. This is because the possible challenge space of the PUF is extremely large; without knowing which challenges will be selected by the verifier beforehand, the probability of the dishonest prover successfully predicting the used challenges would be extremely low. Thus, PUF provides a very lightweight and reliable method to perform device authentication in practice. Moreover, PUFs can be used to construct secure cryptographic protocols, like key agreement, oblivious transfer, and bit commitment.

PUF is also very useful in providing architectural support for software security. For example, PUF can be used for key management and remote attestation. PUF-based key management has a unique advantage over traditional key management, which stores secret keys in a digital memory. A digital memory may be vulnerable to probing attacks. A PUF-based key management scheme uses the responses of a PUF as the secret keys, while only the corresponding challenges need to be stored in a digital memory, which can be public information known by attackers. The secret keys will only be revealed during the runtime after the PUF has been fed with the stored challenges. Additionally, one can further extend this idea to construct secure remote attestation protocols, where a verifier will be able to check whether the outsourced code and data are executed on the authenticated processor embedded with a genuine PUF.

All of the above use cases have demonstrated the promising potential of the PUF technology, and the security of these protocols can be reduced to the security of the PUF itself. However, attackers could also build a mathematical clone of a

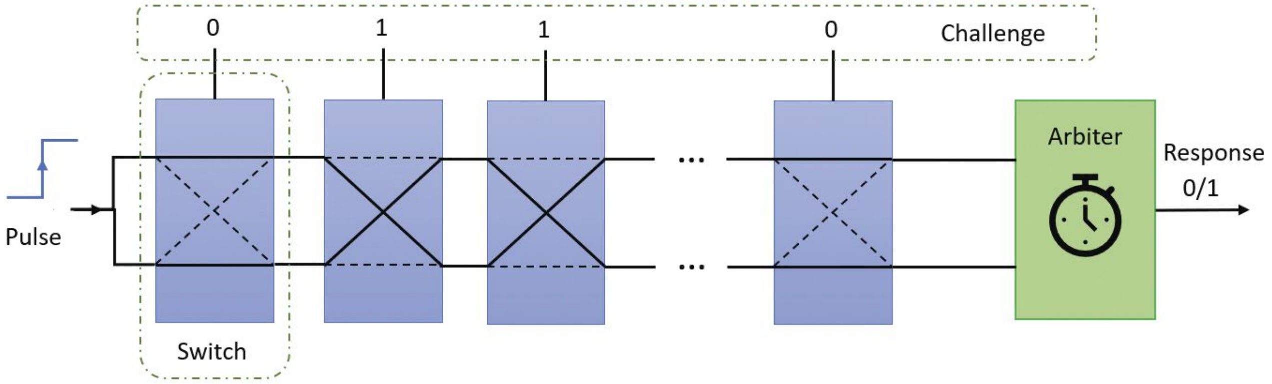


Figure 1: The basic operation of an Arbiter Physical Unclonable Function. Physical properties of chips like these are unique. Their uniqueness can be used to add security to critical infrastructure systems.

PUF that can predict all of its CRPs. Even with some of the state-of-the-art PUF designs, advanced modelling attacks can still achieve substantially better accuracy (e.g. 75%) than random guesses (50%). So, until now, this type of attack has significantly undermined the security of the PUF and any security protocols or mechanisms built on it.

To solve this problem, we would want PUFs to have the same behaviour as a physical random oracle. A random oracle is supposed to generate an independent random value output from a uniform distribution when given a new and unseen input, and if it receives an input that has been queried before, it should reply with the same output as last time. However, PUF evaluations are sensitive not only to the process variations but also to measurement noise. Thus, sometimes, even the same challenge will lead to different responses on the same PUF. To overcome the reliability issue, one needs to post-process raw PUF responses before revealing them to the adversary. One popular post-processing method is called fuzzy extractor, and it is designed to extract a reliable secret string from a fuzzy input, which is the raw PUF responses [2]. However, to keep the security of the whole system, the computation inside the fuzzy extractor has to be kept secret from the adversary.

In our recent work, we introduce a theoretical framework that does not require any confidential digital computing while we can still prove rigorous statements about the bit security of a system that interfaces with the PUF [3]. The framework is even secure when the adversary has a prediction model (e.g. with a 75% accuracy). In particular, we proved the bit security of a PUF-based random oracle construction. This merges the PUF framework with fuzzy extractors. This gives hope for us to rigorously analyse the security of all the systems and protocols built on top of PUF.

Link:
[L1]: https://www.cwi.nl/en/groups/computer-security/

References:

[1] B. Gassend, et al. “Silicon physical random functions,” in Proc. of the 9th ACM Conference on Computer and Communications Security, 2002.
[2] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” Advances in Cryptology-EUROCRYPT 2004: Int. Conf. on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2004. Proceedings 23, Springer, 2004.
[3] M. van Dijk and C. Jin. “A theoretical framework for the analysis of physical unclonable function interfaces and its relation to the random oracle model,” J. of Cryptology, 36.4 (2023): 35.

Please contact:

Chenglu Jin, CWI, the Netherlands
Chenglu.Jin@cwi.nl

Marten van Dijk, CWI, the Netherlands
Marten.van.Dijk@cwi.nl

Enhancing Software Security in Hardware SoC Environments: A Heterogeneous Approach

by Radhen Hendarmawan (RISE)

In the rapidly evolving world of embedded systems, ensuring robust software security within System-on-Chip (SoC) environments is essential. At RISE, we explore a heterogeneous approach using Field-Programmable Gate Arrays (FPGAs) and develop toolkits to streamline hardware acceleration, offering software developers powerful solutions to bolster security and performance.

The rise of embedded systems across various industries has led to the widespread adoption of System-on-Chip (SoC) architectures, which integrate multiple components such as CPUs, memory, and custom hardware blocks onto a single chip. These systems, while compact and efficient, present unique challenges, especially in terms of software security. As cyber threats continue to grow in sophistication, securing SoC environments against vulnerabilities like Man-At-The-End (MATE) attacks, malware, and hardware trojans has become critical.

Traditional software-based security measures often struggle to keep up with the demands of SoC environments, particularly due to constraints related to power, processing capability, and the heterogeneous nature of these systems. To address these challenges, researchers at RISE have been investigating a heterogeneous approach to software security, leveraging FPGAs for their flexibility, performance, and energy efficiency. Additionally, we have developed a toolkit and framework that simplifies the process of creating hardware accelerators, enabling software developers to rapidly prototype and implement security solutions with minimal expertise in hardware complexity.

Why a Heterogeneous Approach?

SoC systems typically combine CPUs, GPUs, and custom hardware blocks, necessitating a multifaceted approach to security. Integrating FPGAs into these systems provides several advantages:

1. Performance Efficiency: FPGAs excel in executing cryptographic algorithms, such as the Advanced Encryption Standard (AES), much faster than traditional processors due to their parallel processing capabilities. This is crucial for applications that require real-time data protection, such as secure communications and digital rights management.
2. Energy Efficiency: Power consumption is a key concern in embedded systems, particularly for battery-operated devices. FPGA-based implementations of security algorithms have demonstrated significantly lower power consumption compared to software-based solutions on CPUs or ARM processors, making them ideal for energy-constrained environments.
3. Flexibility and Reconfigurability: Unlike fixed-function hardware, FPGAs can be reprogrammed to adapt to evolv-

ing security threats or to implement different security protocols as needed. This reconfigurability allows for the dynamic adaptation of security measures, providing a robust defence against emerging cyber threats.

Toolkits and Frameworks for Rapid Prototyping

One of the challenges in leveraging FPGA technology is the complexity involved in designing hardware accelerators, which often requires specialised knowledge in hardware design and troubleshooting. To overcome this barrier, our team at RISE has developed a toolkit and framework that facilitates the rapid prototyping of hardware accelerators using high-level synthesis (HLS). This HW-SW co-design solution is designed to help software developers create Hardware Intellectual Property (IP) and accelerators without needing extensive expertise in hardware complexity.

Our toolkit streamlines the process of developing FPGA bitstreams by abstracting the low-level hardware details and providing a user-friendly interface for defining and configuring hardware functions. This allows developers to focus on software security algorithms and leave the intricacies of hardware design to the automated tools. The framework supports various high-level programming languages, enabling seamless integration with existing software workflows and reducing the time required to bring new security features to market.

Case Study: AES Implementation on FPGA

To demonstrate the effectiveness of our approach, we conducted a study comparing the performance, resource utilisation, and power consumption of AES implementations on different hardware platforms: CPU, ARM, and FPGA. The results were compelling:

- Performance: The FPGA implementation of AES128 achieved encryption latencies of 20ms, compared to 250ms on ARM and 100ms on CPU. This performance boost is essential for applications where speed is critical, such as in secure real-time communications.
- Resource Utilisation: While the FPGA required more resources, such as Look-Up Tables (LUTs) and Block RAMs (BRAMs), the trade-off was justified by the significant gains in performance and energy efficiency. Our toolkit played a crucial role in optimising resource usage, enabling efficient implementation without sacrificing performance.
- Power Consumption: The FPGA implementation consumed less power during encryption operations compared to both ARM and CPU platforms, underscoring its suitability for energy-sensitive applications like IoT devices.

Future Directions and Implications

The integration of FPGAs in SoC environments represents a significant advancement in software security. By leveraging their unique capabilities and our rapid prototyping toolkit, we can develop security solutions that are not only more efficient but also more adaptable to the ever-changing cyber threat landscape. Looking ahead, our research will focus on expanding the toolkit to support additional security mechanisms, such as secure key management and real-time intrusion detection, within FPGA-based SoCs.

This heterogeneous approach to software security holds immense potential across a wide range of applications, from consumer electronics to critical infrastructure. As cyber threats continue to evolve, the ability to dynamically adapt security measures will become increasingly important. At RISE, we are committed to advancing this field and providing the tools and solutions that will help safeguard the next generation of embedded systems.

Links:
[L1] Advanced Encryption Standard (AES) Overview: https://kwz.me/hDV
[L2] Introduction to FPGAs: https://kwz.me/hDW

References:

[1] Xilinx, “Vivado design suite user guide: high-level synthesis,” UG902, 2023.
[2] M. Tehranipoor and F. Koushanfar, “A survey of hardware trojan taxonomy and detection,” IEEE Design & Test of Computers, vol. 27, no. 1, pp. 10–25, 2010.

Please contact:

Radhen Hendarmawan, RISE, Sweden
hendarmawan@rise.se

Side-Channel Resistant Applications through Co-designed Hardware/Software: the SCRATCHS Project

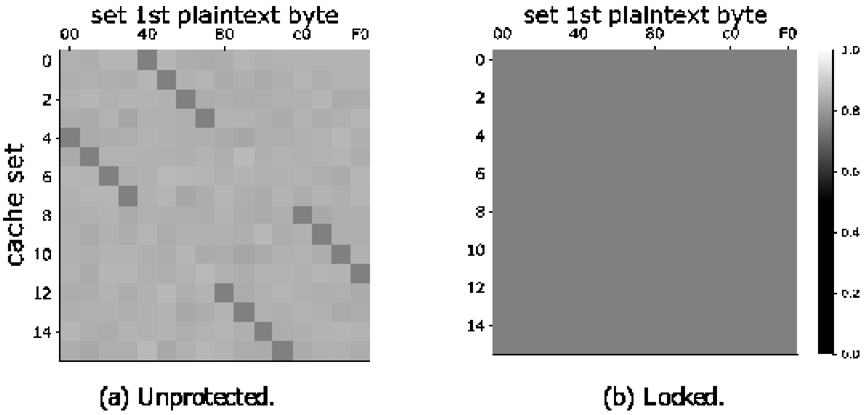
by Frédéric Besson, Célia Le Du (Inria), and Pierre Wilke (Centrale Supélec Rennes)

Hardware and software solutions for protectecting against timing side-channel attacks are effective in securing sensitive data, but significantly impact the performance of the programs they protect. The SCRATCHS project, which stands for Side-Channel Resistant Applications through Co-designed Hardware/Software, aims to combine security and execution speed by developing a new hybrid protection solution based on hardware and software co-design.

Observing fingerprints left on a building’s keypad to guess the entry code, or analysing the sound of keyboard keys to deduce a typed password are examples of side-channel attacks. These attacks are numerous and extend into more advanced cybersecurity contexts. This type of attack involves extracting sensitive information by observing and interpreting signals emitted by a computer system. Attackers use information such as the execution time of a program, its power consumption, or the state of the memory cache to deduce cryptographic information, such as encryption keys.

The SCRATCHS project aims to create a new protection solution that combines hardware and software solutions, focusing particularly on cache-based timing side-channels. This solution would ensure security while maintaining high performance.

Started in 2021, this project is supported by the LabEx CominLabs and brings together researchers located in Rennes, Lorient, and Brest, working in the Lab-STICC and IRISA research laboratories. These researchers are affiliated with institutions such as INRIA, CentraleSupélec, ENSTA Bretagne, and the University of South Brittany. SCRATCHS benefits from collaboration among experts in the fields of formal methods, software-hardware interface security, hardware design, and microarchitecture.



(a) Unprotected.

(b) Locked.

Figure 1: For a given secret key (0x42), the hit rate for each cache set (on the Y-axis) is shown relative to the first byte of the plaintext (X-axis). In the unprotected case, discernible patterns allow information about the secret key to be recovered. In the locked case, the hit rate remains constant, making the attack ineffective. (Figure adapted from [1]).

Hardware and Software Countermeasures

The main types of hardware countermeasures against cache-based side-channel attacks are randomisation and partitioning. In the first case, the address-to-cache-set mapping may be randomised. Randomisation alters the links between the cache areas that are accessed and the memory areas visited in the sensitive application. The mapping between memory areas in the application and the cache areas is randomised. However, this random mapping needs to be changed regularly. In the second case, the cache can be partitioned into multiple security domains, preventing attackers from inspecting and modifying the state of the cache corresponding to the domain where the secure data applications reside. These approaches induce overhead and slow down the programs by protecting all data, including that which does not require protection.

Software countermeasures, on the other hand, operate independently of the underlying hardware and thus do not rely on the mechanisms of the hardware in question. Constant-time programming, which could be a solution, forbids memory accesses based on secret data as well as conditional branches with conditions dependent on a secret. Although effective, this type of programming is prone to errors and also induces overhead.

The SCRATCHS Solution: a Fine-grained Dynamic Partitioning Mechanism via Cache Locking

SCRATCHS involves the co-design of a RISC-V processor and a compiler toolchain to ensure the security of sensitive data while maintaining optimal program execution speed. This hybrid solution aims to enable communication between software and hardware components to reduce the risk of secret data recovery only when necessary.

To put this into practice, dynamic partitioning is used, where the sensitive program can instruct the system to always keep sensitive data in the cache, a process known as locking. As long as this data is locked, it cannot be evicted from the cache, and all accesses to it are necessarily cache hits. Access to sensitive data will thus always take the same quick time, preventing the attacker from using access time measurements to decode secret data. Figure 1 illustrates a PRIME+PROBE [3] analysis after one round of AES. This shows that our locking mechanism prevents the attacker from learning information about the secret key. Through this process, the necessary data is locked in the cache until operations are completed, and the rest of the cache remains available for other applications, maintaining optimal performance.

Technically, this involves modifying the processor and the cache. Lock and unlock commands are added to the instruction set architecture, specifically the RISC-V ISA [1]. Meanwhile, mathematical calculations on the security of this solution have provided formal proof of this co-design approach [2].

The SCRATCHS project will conclude in 2025 and will give way to the follow-up action LockOS. The goal of LockOS will be to implement the developed solutions on more complex processors with different privilege levels while adding operating system support. The aim is to create a demonstration to prove that the developed protections can achieve a high level of security while maintaining strong performance.

References:

[1] N. Gaudin et al., “A fine-grained dynamic partitioning against cache-based timing attacks via cache locking,” ISVLSI, 2024.

[2] J.-L. Hatchikian-Houdot et al., “Formal hardware/software models for cache locking enabling Fast and Secure Code,” ESORICS, 2024.

[3] F. Liu et al, “Last-level cache side-channel attacks are practical,” SP 2015.

Please contact:

Frédéric Besson, Inria, France
frederic.besson@inria.fr

Protecting Cryptographic Material in Ethereum Blockchain Clients Using an Open-source Secure Element

by Mario de la Haba Navarro (Decentralized Security), Pablo Sánchez-Serrano (University of Malaga), and Isaac Agudo (Decentralized Security and University of Malaga)

A recurring security issue in software is protecting cryptographic material, especially for cloud-hosted or internet-facing applications, where the risk of key compromise is higher. One solution is using a Hardware Security Module (HSM), which secures keys and performs cryptographic operations without exposing them. However, HSMs are typically closed source, making security evaluations difficult, and may not support the latest cryptographic methods. For over three years, we've been developing an open-source modular platform to build an HSM for Ethereum clients, a crucial part of the blockchain infrastructure.

Relying on security through obscurity is an approach that can present significant risks. Security by obscurity is based on the premise that a system is more secure if the details of its implementation are unknown. However, this approach can be problematic because if implementation details are discovered, the security of the system may be compromised. There are many examples in the literature of closed systems that were immediately broken once the source code was recovered, but even without the source code, a side-channel can compromise the security of the system. In September 2024, a researcher from NinjaLab published EUCLEAK [L1], an attack that affects ECC keys in the Infineon SLE78 secure element. This is the main component on many different hardware authentication tokens, such as the Yubikey 5 tokens, one of the most widespread FIDO hardware tokens. This is not an isolated example – in 2017, researchers from Masaryk University presented ROCA [R1], an attack affecting the Infineon M7892 B11 chip due to a vulnerability in another closed-source library, this time affecting RSA keys. This shows that despite the risk, the industry is still supporting closed-source and hardware products.

The use of a Hardware Security Module (HSM) is widespread in the modern internet, as many cloud providers already provide some kind of key-management systems based on an HSM. Also, the use of HSM is a requirement in many critical systems, e.g. Root Certification Authorities. However, in the current blockchain security landscape there are not many examples of the use of HSMs, and most of them focus on protecting the private keys of the user wallets [2], leaving the keys of the validator nodes aside. We focused on the blockchain scenario because a blockchain client is a heavily exposed piece of software that is commonly run in the cloud. Also, there is an inherent economical risk associated with blockchain clients, because they need to have access to the cryptographic keys that are used to move funds, so an attacker compromising the machine could use them to empty the victim's wallet.

If we focus on the Ethereum blockchain, there is another incentive for trying to come up with an open-source HSM – the cryptographic algorithms they use are evolving faster than commercial products. Current consensus protocol makes use of Boneh-Lynn-Shacham (BLS) signatures, which is not a widely used industry standard and hence not supported by traditional HSMs. To the best of our knowledge there is only one HSM manufacturer that claims to support this type of signature [L2], but as with many HSMs, its implementation is closed source. The next generation consensus protocol will probably make use of Zero Knowledge Proof (ZKP) protocols, so the HSM should evolve in sync with the network. Then it is not only a matter of having a more transparent HSM but also a more agile one.

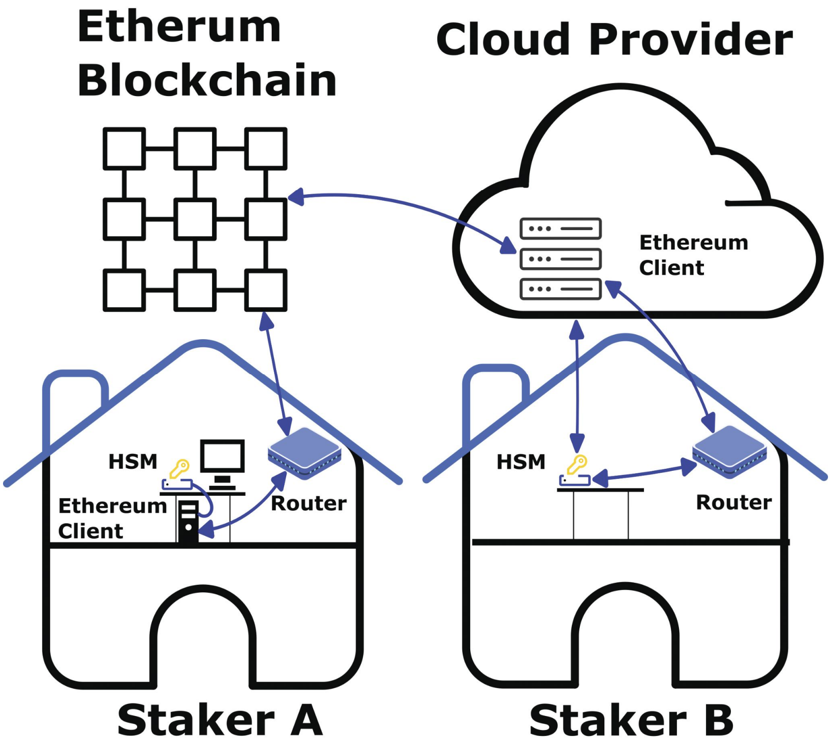


Figure 1: Different deployment scenarios for the HSM in Ethereum.

Our goal in this project was to provide the base for an open-source HSM, using a low-cost hardware board, so that anyone can use it with their Ethereum node, without the cost of the hardware being a limitation. This means that developers and security experts can examine the code, identify potential vulnerabilities and contribute to improving the security of the implementation. In addition, we implemented our code using a hardware development kit that provides an open hardware design, allowing the community to also improve the hardware part. In Figure

1 we can see the two different scenarios that we have considered when developing our project: a node operator with physical access to the node (Staker A), and a node operator running the node in the cloud (Staker B). In the first case, the HSM would be physically connected to the node, whereas in the second case the HSM would need to connect to the node over the internet, using either a cellular connection or the internet connection of the node operator. For each scenario we tested a different hardware. We chose Nordic Semmiconductors Development Kits as they provide a wide range of boards based on the ARM Cortex-M33 CPU (e.g. nRF5340 DK, nRF9160 DK and rRF7002 DK), with full support for ARM TrustZone technology. Also, they provide an excellent SDK based on the open-source RTOS Zephyr OS [L3], clear and concise documentation and a very strong community.

This project was funded by the Ethereum Foundation[L4] and has been developed in collaboration between Decentralized Security [L6] and NICS Lab [L7]. All the code and more technical details can be found in the Github repository [L5].

Links:
[L1] https://ninjalab.io/eucleak/
[L2] https://kwz.me/hDM
[L3] https://zephyrproject.org/
[L4] https://kwz.me/hDQ
[L5] https://github.com/decentralizedsecurity/bls-hsm
[L6] https://decentralizedsecurity.es
[L7] https://www.nics.uma.es

References:

[1] M. Nemec et al., “The return of coppersmith’s attack: practical factorization of widely used RSA moduli,” ACM CCS , 2017. doi: 10.1145/3133956.3133969

[2] W. Shbair, E. Gavrilov, et al., “HSM-based key management solution for Ethereum blockchain,” in IEEE Int. Conf. on Blockchain and Cryptocurrency, 2021.

Please contact:

Isaac Agudo, Founder of Decentralized Security, Researcher at NICS Lab, University of Málaga, isaac@uma.e

CALL FOR PAPERS

ACM Digital Threats: Research and Practice

Special Issue on Offensive and Defensive Techniques in the Context of Man At The End (MATE) Attacks

Guest Editors:
- Michele Ianni,
  University of Calabria, Italy,
  michele.ianni@unical.it
- Sebastian Schrittwieser,
  University of Vienna, Austria,
  sebastian.schrittwieser@univie.ac.at

MATE (Man-At-The-End) is an attacker model where an adversary has access to the target software or hardwareenvironment of his victim and the ability to observe and modify it in order to extract secrets such as cryptographickeys or sensitive information, possibly with the subsequent goal of altering code integrity or inserting backdoors,among others. A typical example of such a scenario is the case of an attack on a stolen smartphone or againstsoftware leveraging protection to hide sensitive data or intellectual property.

The main focus of the special issue on Offensive and Defensive Techniques in the Context of Man At The End (MATE)Attacks is on new models and techniques to defend software from tampering, reverse engineering, and piracy aswell as to the development of new attack strategies that highlight the need of more complete defenses. We includeboth offensive and defensive techniques because of their close and intertwined relationship depending on theattack scenario: indeed, reverse engineering is defensive when the goal is to analyze obfuscated malware, but it isoffensive when it is used to steal intellectual property and assets in legitimate software. Likewise, obfuscation isdefensive when it aims for protecting a legitimate asset against reverse engineering, while it is offensive if it is usedto hide that malware is embedded in an application. Both scenarios are of practical relevance, and therefore thespecial issue on Offensive and Defensive Techniques in the Context of Man At The End (MATE) Attacks includes allattacks on/defenses of the confidentiality and integrity of software applications and assets embedded therein andexposed to MATE attacks. In such scenarios, attackers have full control over, and white-box access to, the softwareand the systems on which they attack the software in their labs.

Strongly encouraged are proposals of new, speculative ideas, metrics, tools, and procedures for evaluating tamperproofing, watermarking, obfuscation, birthmarking, and software protection algorithms in general. Assessment of new or known techniques in practical settings and discussions of emerging threats, and problems are expected. Likewise, reverse engineering of low-level constructs such as machine code or gate-level circuit definitions through static and dynamic analysis is geared to recover information to determine the intent of programs and understand their inner workings as well as for classifying them with respect to similar known code (which is typically malicious). The special issue on Offensive and Defensive Techniques in the Context of Man At The End (MATE) Attacks welcomes original work on the formal investigation of software protection, where formal methods are used to better understand the nature, relations, potentialities, and limits of software security techniques.

Important Dates
- Submissions deadline:
  30 November 2024
- First-round review decisions:
  15 March 2025
- Deadline for revision submissions:
  30 April 2025
- Notification of final decisions:
  30 June 2025
- Tentative publication:
  30 September 2025.

Please contact:

Michele Ianni, University of Calabria, Italy, michele.ianni@unical.it

Sebastian Schrittwieser, University of Vienna, Austria,
sebastian.schrittwieser@univie.ac.at

For more information about topics and submission, see https://kwz.me/hDR

Connected Aquaponics: Sustainable Agriculture through Industry 5.0 Technologies and Circular Economy Principles

by Rafael Kupsa, Amin Anjomshoaa and Markus Tauber (Research Studios Austria)

The EDEN project is at the forefront of sustainable agriculture, integrating Industry 5.0 technologies to improve upon aquaponics systems. This article explores how the project is developing innovative frameworks to maximise resource efficiency, optimise food production, and foster community engagement securely.

Aquaponics is a closed-loop system where the waste produced by fish is used as nutrients for plants, which in turn purify the water for the fish. This symbiotic relationship drastically reduces the need for water and land compared to traditional agriculture. EDEN is a two-and-a-half-year research project that began in April 2023, in collaboration with Research Studios Austria FG, the University of Applied Sciences St. Pölten, the Austrian Institute of Technology, BEIA International, and Andersfarm, an Austrian aquaponic expert, with the goal of enhancing aquaponics systems in two major steps: First, by creating hardware and software capable of intelligently optimising aquaponics systems using autonomous control loops, the Internet of Things (IoT), and machine learning (ML) technology, and second, by providing a secure platform for connecting multiple systems, allowing them to benefit from each other without compromising their data, if they so choose. As the project progresses, it holds the potential to transform aquaponics into a scalable and economically viable solution for food production, contributing to global efforts to address the challenges of food security and sustainability.

Smart Optimisation in Aquaponics

Initially, the focus is on developing a software framework that integrates IoT and ML technologies within an autonomic control loop. This framework utilises the MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) loop, a well-established approach in autonomic computing that enables systems to self-manage by continuously monitoring their environment, analysing data, planning actions, executing them, and updating their knowledge base. This continuous cycle ensures that the system can adapt to changing conditions in real-time [1,2].

In the context of aquaponics, IoT sensors collect critical data such as water quality, temperature, and humidity. The data is then processed and analysed by ML algorithms, which can detect anomalies or predict potential issues. When irregularities are detected, the system can alert users about them and, in some cases, automatically initiate corrective actions via IoT actuators. Thus, the system can be helpful in optimising key parameters such as water quality, nutrient levels, and environmental conditions in aquaponics systems (see Figure 1).

One practical example of adaptive control within the EDEN project's aquaponics system is the regulation of pH levels. IoT sensors continuously monitor pH levels, while ML algorithms predict deviations from the optimal range. When an issue is detected, the system can alert the user of it and, depending on the setup, automatically adjusts the water's pH by adding necessary solutions with a pump, maintaining a healthy environment for both fish and plants. This automation reduces manual intervention, ensuring a more efficient and sustainable operation.

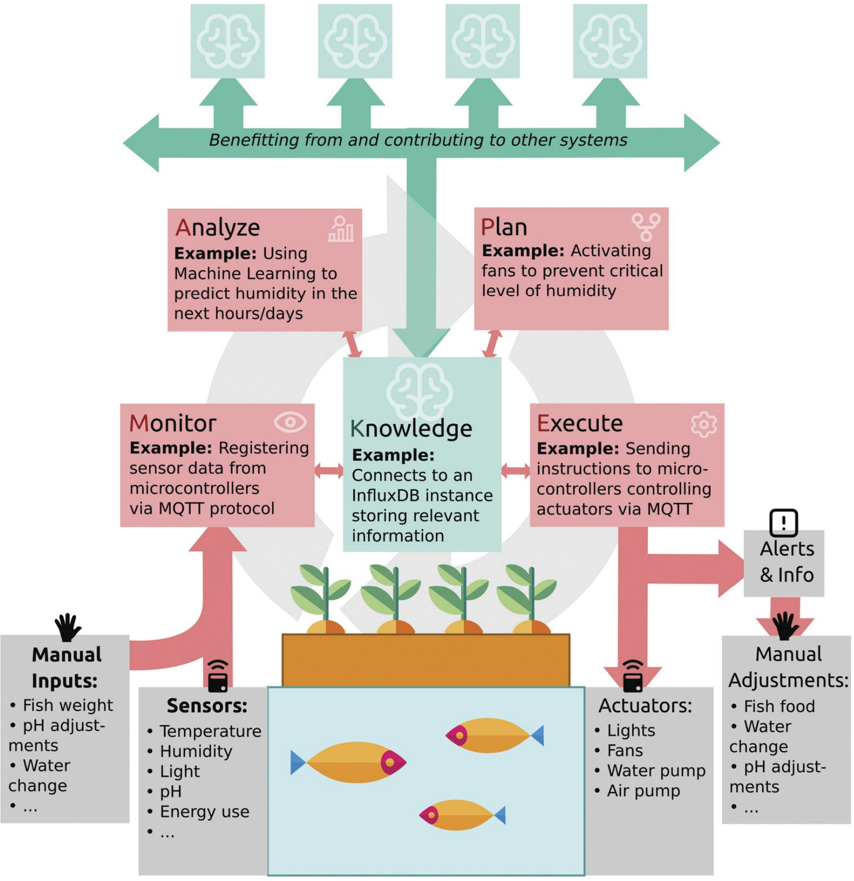

Benefitting from and contributing to other systems
Analyze
Example: Using Machine Learning to predict humidity in the next hours/days
Plan
Example: Activating fans to prevent critical level of humidity
Monitor
Example: Registering sensor data from microcontrollers via MQTT protocol
Knowledge
Example: Connects to an InfluxDB instance storing relevant information
Execute
Example: Sending instructions to micro-controllers controlling actuators via MQTT
Alerts & Info
Manual Inputs:
• Fish weight
• pH adjust-ments
• Water change
• ...
Sensors:
• Temperature
• Humidity
• Light
• pH
• Energy use
• ...
Actuators:
• Lights
• Fans
• Water pump
• Air pump
Manual Adjustments:
• Fish food
• Water change
• pH adjust-ments
• ...

Figure 1: MAPE-K framework architecture for smart control of an aquaponics system.

Networked Systems for Community Benefit

Beyond optimising individual aquaponics systems, the EDEN project seeks to create a network of interconnected systems that share data, resources, and knowledge. This networked approach aligns with the principles of Industry 5.0, which emphasise collaboration and circularity and is modelled after Anjomshoaa and Curry's [3] maturity levels of knowledge sharing and reuse in smart environments. Users will be able to connect via the project's website [L1], currently under development, at three levels:

1. Data level: The first level of community engagement involves data sharing among aquaponics operators. Users can upload data from their systems to a central platform, where it can be accessed by others. This collective data pool provides valuable insights and enables operators to learn from each other's experiences.
2. Stream level: At a more advanced level, the EDEN project envisions real-time data streaming, where aquaponics systems are continuously connected to the central platform. This allows for immediate sharing of data and fosters a collaborative environment where operators can benefit from the collective knowledge of the community.
3. Service level: The EDEN project also plans to facilitate the sharing of ML models among operators. By contributing their models to a shared repository, users can improve the accuracy and applicability of their systems. This federated learning approach ensures that all participants benefit from the collective advancements in technology, without compromising the privacy or security of their individual data.

This approach allows users to decide what and how much they want to share about their systems and how to contribute to improving models and data control. For example, an operator can provide the data of their pH sensors and their manual pH regulation while others can use those to initialise an automated pH-regulation model with a pH pump. The trained model could then be provided back to the user community. This collaborative approach accelerates knowledge sharing, allowing the entire network to benefit from improved performance and resource efficiency.

As the EDEN project promotes interconnected systems and shared data, ensuring the security and privacy of the data is crucial. This can be addressed by implementing robust encryption protocols for data transmission, secure authentication methods for user access, and providing options for users to anonymise their data before sharing it. Users who do not want to share data at all due to privacy concerns can still contribute by sharing models within the federated learning approach.

Links:
[L1] https://eden.researchstudio.at

References:

[1] S. Maksuti, O. Schluga, G. Settanni, M. Tauber, and J. Delsing, "Self-adaptation applied to mqtt via a generic autonomic management framework," in 2019 IEEE Int. Conf. on Industrial Technology (ICIT), pp. 1179–1185.
[2] S. Maksuti, M. Zsilak, M. Tauber and J. Delsing, "Security and autonomic management in system of systems," Infocommunications Journal, vol. XIII, no. 3, pp. 66–75, Sept. 2021. doi: 10.36244/ICJ.2021.3.7
[3] A. Anjomshoaa and E. Curry, "Transfer learning in smart environments," Machine Learning and Knowledge Extraction, vol. 3, no. 2, pp. 318–332, 2021.

Please contact:
Rafael Kupsa, Research Studios Austria Forschungsgesellschaft mbH, Austria
rafael.kupsa@researchstudio.at

An Innovative Approach to Supporting Startups in Greece and Southern Europe

by Panagiotis Konstantinopoulos, Vasileios Loukopoulos, Dionysia Mylona, Maria Veneri, and Konstantinos Bastas (Patras Science Park S.A.)

The Greek experience from operation of Science and Technology Parks (STPs) has demonstrated that the most important factors affecting operation of an STP are capacity building, and lack of funding for support of a wide range of services. A novel mechanism that will act as a Distributed Local Cluster is proposed for application in Balkan and Southern European countries.

Mainstream current support services [1] for startups range from training/mentoring, seed funding, acceleration, supply of workspaces and admin support. However, reflecting on the starting of the development of Science and Technology Parks in Europe (in the 1970s), the situation was much simpler, and the range of offered services, beyond incubation, was limited. (Figure 1)

Patras Science Park (PSP) [L1] was established on the outskirts of the port city of Patras, at the centre of an RTD and innovation ecosystem, following the model of Incubator. Over the years, PSP extended its range and developed a number of added-value services. Two major problems have been identified concerning these services, related to sustainability and capacity building:

- Sustainability of operations, beyond the main incubation activities, is offered spontaneously, implemented mostly through funded projects.
- Capacity-building of existing staff, tenant startups' and other SMEs' key personnel is obstructed, due to lack of funding for the Greek STPs.

It is notable that these problems are common for most Southern European countries and, even more intensely, in the area of the Balkans, where lack of relevant infrastructure for supporting startups is more visible than in most EU member states.

Step 1 – The SPREAD2INNO Project

The gap between the Northern and Southern European countries or between the strong and moderate or emerging innovators has been identified at EU level, as well as its adverse consequences (HORIZON-EIE-2021-SCALEUP-01-01 Call [L2]).

PSP participated in the above Call, as partner in a consortium consisting of leading organisations in the area of acceleration services towards startups, namely, Fondazione E. Amaldi (I), as Project Coordinator, Fraunhofer IPK (D), Teseas (I), Cleantech (BG), Minds & Sparks GmbH (A), including the Brussels-based European Business Angels Network (EBAN).

SPREAD2INNO [L3] brought together different innovation stakeholders – incubators, accelerators, innovation businesses, research organisations, consultancies and networks. The goal was to implement a scalable, replicable and holistic two-year programme for provision of high-quality business services, and collaboration between different entities of well-connected and less-connected regional innovation ecosystems, with immediate impacts at the local and EU level. In this context, the partnership implemented a set of activities such as six local events [2] including training courses, development of an online platform as a single point of reference for information exchange between different innovation ecosystems, two intensive training academies, and a one-week business case and investor training for six finalists.

Two out of the six local events were implemented in Greece:

- The first one took place on 24 and 25 May 2023, in Thessaloniki, co-organised by PSP and EBAN, in parallel with the EBAN Annual Congress for 2024.
- The second one took place on 26 and 27 November 2023, in Patras, in parallel with the 8th Annual Patras IQ Fair.

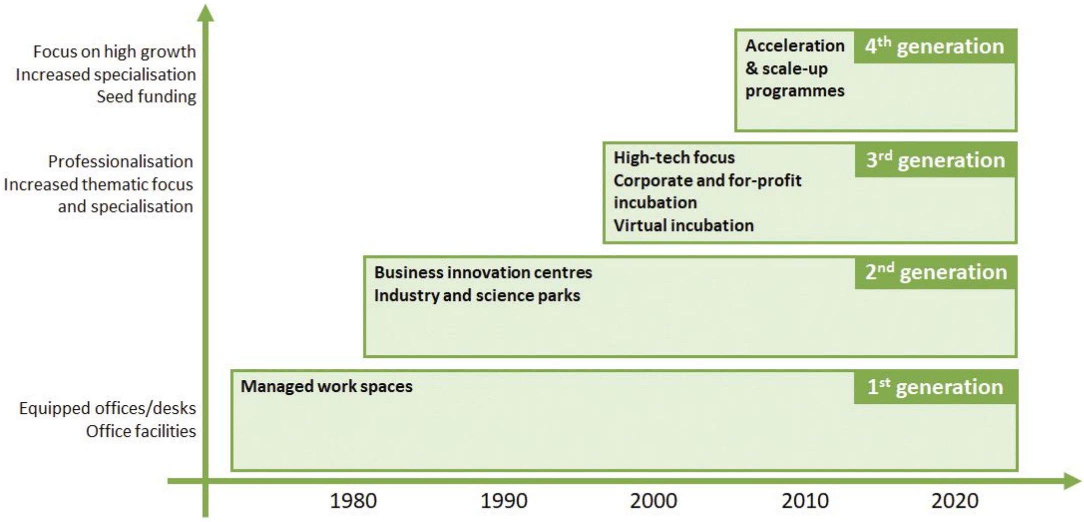


Figure 1: Historical account of supporting services for startups.

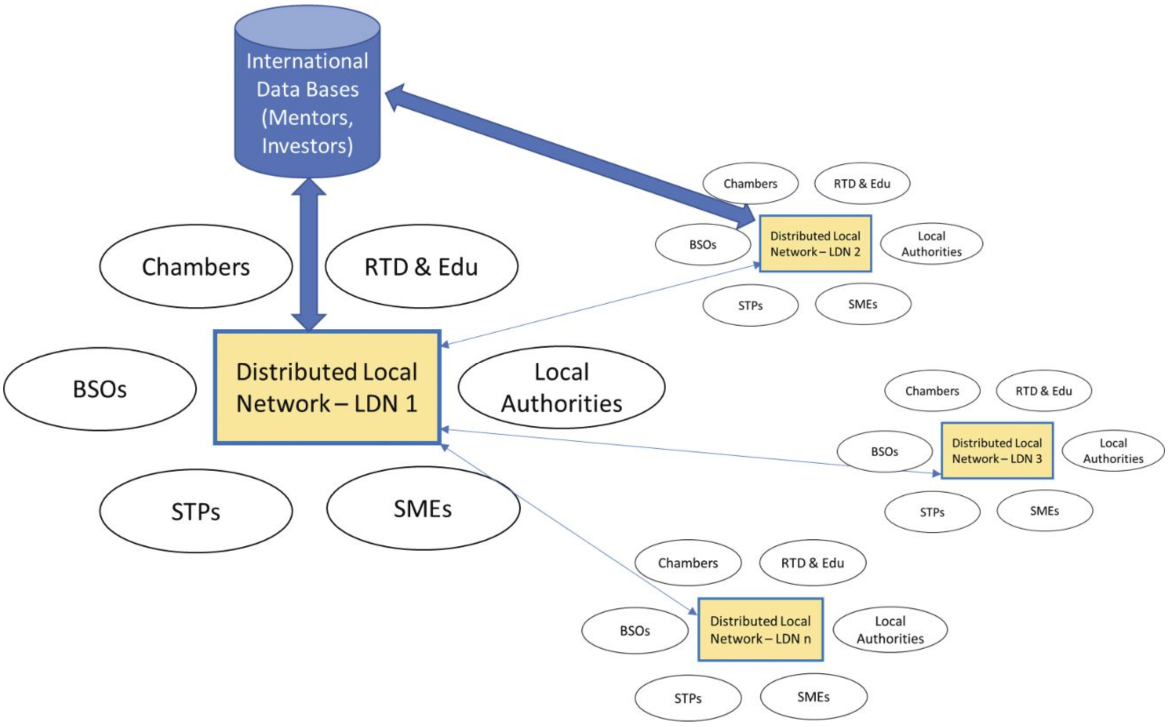

International Data Bases (Mentors, Investors)
Chambers
RTD & Edu
BSOs
Distributed Local Network – LDN 1
Local Authorities
STPs
SMEs
Distributed Local Network – LDN 2
Distributed Local Network – LDN 3
Distributed Local Network – LDN n

Figure 2: Novel mechanism for supporting startups.

The two academies were implemented in Bologna (I) and Sofia (BG), in parallel with WMF 2024 and Spinoff Bulgaria 2024, respectively, with participation of a total of five Greek startups.

Two Greek startups were selected for participation in the final step, i.e. the Startup Week one-week training and personalised mentoring, which will conclude the SPREAD2INNO project activities.

Step 2 – The Distributed Local Networks (DLNs)

During implementation of the SPREAD2INNO project, it became evident among the PSP team that the task of supporting startups should be viewed in a more holistic manner, considering local weaknesses, and combining efforts and resources between all relevant, as well as interested, local stakeholders. The idea was to form "Distributed Local Networks-DLNs" that could cooperate, creating formal or informal networks, which could undertake jointly the task of providing support services to startups and SMEs. This way, the spectrum of services could be extended, the associated costs for the stakeholders could be lowered, and the startups could source the required services locally.

Testing of this novel methodology has already started by PSP, and will be demonstrated in nine different regions, in Greece, Italy, Slovenia, Croatia, Montenegro, Bosnia & Herzegovina, Serbia and Albania, with funding from the Interreg ADRION programme (project A4SUSTINNO).

Step 3 – Distributed Local and Interregional Networks

At the end of the A4SUSTINNO project, the local networks (DLNs) will take advantage of existing links with investors and support organisations, through the SPREAD2INNO activities, and especially the EBAN and other pan-European networks, creating a network, which will be supported by dedicated web platform for information, matching, and diagnosis of needs and weaknesses (Figure 2).

Links:
https://psp.org.gr/
https://kwz.me/hDI
https://www.spread2inno.eu/

References:

[1] "Business incubation – from startup to scaleup, A Policy Brief from the Policy Learning Platform for a smarter Europe," Apr 2024.

[2] G. D'Agostinis, L. Scatena, E. Lombardi, K. Singer-Coudoux, H.N. Buxmann, P. Konstantinopoulos, D. Haskovic, and I. Todorova, "Challenges, disparities and risks of the European innovation system: the SPREAD2INNO project to bridge the gap," 74th International Astronautical Congress (IAC), Baku, Azerbaijan, 2–6 Oct 2023.

Please contact:

Panagiotis A. Konstantinopoulos, Patras Science Park, Greece
pkonst@psp.org.gr

Boosting MATE Security through Small Language Models

by Luca Caviglione (CNR-IMATI), Gianluigi   (CNR-ICAR), Massimo Guarascio (CNR-ICAR), and Paolo Zicari (CNR-ICAR)

Modern Man-At-The-End (MATE) attacks can take advantage of Language Models to produce deceptive information or malware variations. Fortunately, they can also be used to implement advanced defensive techniques. In this vein, we present the use of Small Language Models to reveal fraudulent communications or to automatically generate test cases for preventing the exfiltration of data through tampered MQTT brokers.

The typical MATE adversary is supposed to have an unrestricted access to the target. In modern heterogeneous scenarios, the attacker can interact with software artifacts (e.g. libraries, firmware and digital images) as well as with a multitude of hardware devices, such as IoT nodes or industrial-class assets. Therefore, available offensive tools and techniques are almost boundless, ranging from debuggers to virtualisation frameworks. With the advent of generative AI, the ability of the adversary has expanded, making MATE attacks difficult to prevent and mitigate. For instance, Large Language Models (LLMs) can be used during the reconnaissance stage to gain insights from traffic dumps, execution logs, or documents for implementing a business process or instructing operators. LLMs can also be used to weaponise threats as they have proven to be effective in synthetising ad-hoc phishing mails or in producing malicious code, such as SQL injections (e.g. [1] and the references therein). Fortunately, LLMs are emerging as valuable defensive tools, especially for anticipating attackers. For instance, AI fuzzers can test libraries, configuration files and protocol specifications effectively [1], especially in preventing some manipulations that a MATE adversary is likely to perform when accessing the target.

Aiming to mitigate the impact of MATE attacks, we showcase two major results leveraging Small Language Models (SMLs). Unlike LLMs, they are trained on smaller, more specific, and cleaner datasets. This allows them to achieve better performances than LLMs for specific tasks without the need for supercomputers with huge computational resources, which is an unfeasible requirement for SMEs.

The first result considers the usage of SMLs for mitigating possible phishing attacks or communications crafted to induce human operators to support the attacker, for instance to grant access to an infrastructure. To this end, the threat actor could

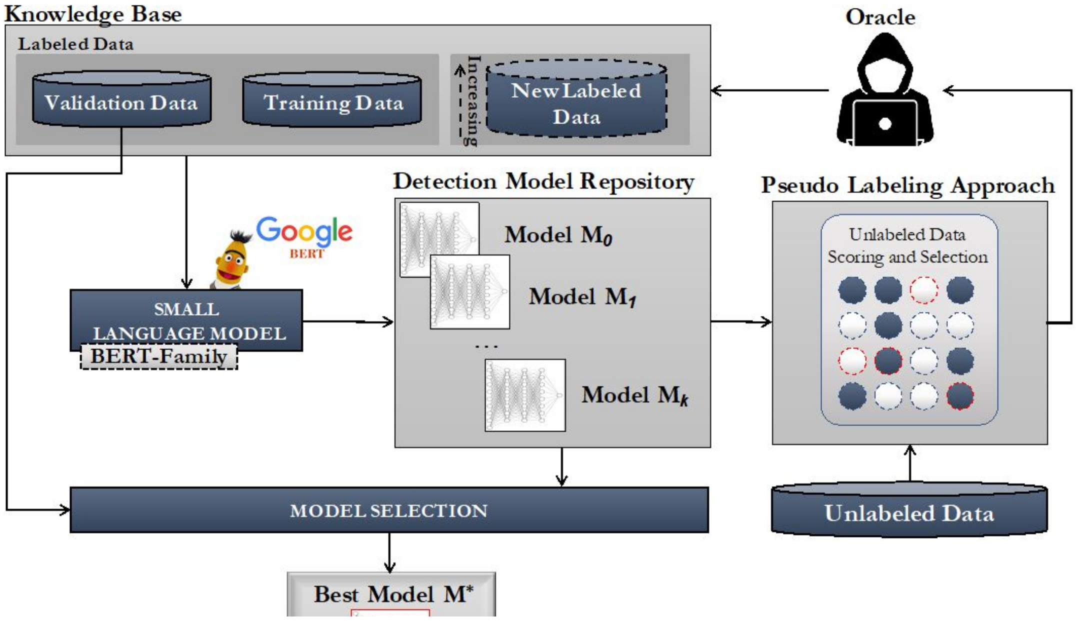


Figure 1: Learning process of the Detection Model and interaction with the expert.

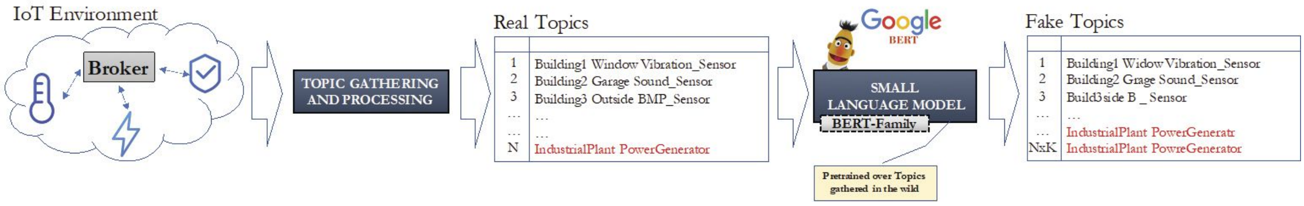

IoT Environment
Broker
TOPIC GATHERING AND PROCESSING
Real Topics
1 Building1 Window Vibration_Sensor
2 Building2 Garage Sound_Sensor
3 Building3 Outside BMP_Sensor
... ...
... ...
N IndustrialPlant PowerGenerator
Google
BERT
SMALL LANGUAGE MODEL
BERT-Family
Pretrained over Topics gathered in the wild
Fake Topics
1 Building1 Widow Vibration_Sensor
2 Building2 Grage Sound_Sensor
3 Build3side B _ Sensor
... ...
... IndustrialPlant PowerGeneratr
NxK IndustrialPlant PowreGenerator

Figure 2: The pipeline for generating "fake" MQTT topics for the creation of covert communications within real MQTT brokers. In red are the topics used as an example for describing the attack template.

collect information on the used software, vendors, CVEs, or supported hardware to produce "fake news", e.g. misleading security bulletins. As a countermeasure, we devised a human-in-the-loop semi-supervised framework for detecting ad-hoc malicious information in a challenging scenario where small amounts of labelled data are available for the training stage. It leverages the iterative Active Learning (AL) process for integrating newly labelled data. Specifically, it uses a "small" instance of a Bidirectional Encoder Representation from Transformers (small BERT) model within an AL scheme where novel labelled data is acquired incrementally, with the help of a human expert, and used to fine-tune the model. The learning process introduced above is depicted in Figure 1. In more detail, the initial set of training data is used to build a preliminary version of the classifier. This training set is expanded iteratively by automatically selecting a small subset of uncertain unlabelled data instances, which are passed to a human expert who will confirm or reject the model classification. By adding these newly labelled instances to the original training set, a new version of the classifier is trained at each iteration of the AL procedure. This framework has been tested for detecting fake news (e.g. gossip and politics) and has demonstrated its effectiveness with improvements in accuracy up to 10%.

The second result for limiting MATE attacks employs SMLs to help the security expert reveal the presence of a malicious actor trying to exfiltrate data through an MQTT broker, which is a core building block of many smart homes and an important "glue layer" to gather IoT measurements for feeding digital twins of cities and industrial settings. A malicious actor could collect MQTT topics of the victim (e.g. a smart building) through a MATE attack and then craft "fake" entries to implement a wide array of information-hiding-capable offensive schemes. Such bogus entries can be used to conceal unauthorised transmissions or encode stolen data, e.g. the creation of a specific topic could signal infected IoT nodes to activate. To prevent such an attack scheme, together with colleagues from the University of Pavia, we used a small BERT to capture all the nuances of the topics made available on a specific broker, e.g. all the measurements belonging to a set of IoT sensors grouped under the umbrella name IndustrialPlant/PowerGenerator. The model can then be used to fuzz-test detection metrics, defensive tools, or best practices used within an organisation to define naming conventions. Figure 2 depicts the proposed approach. Starting from real topics gathered "in the wild" the small BERT can produce "fake" but realistic topics to check the resilience of security policies against information hiding [3]. As an example, a properly trained SLM can automate the exploration of all the possible permutations used by an attacker to encode data (e.g. in the case-lowercase sequence of the letters composing a topic) or determine whether fake entries such as "IndustrialPlant/PowerGeneratr" or "IndustrialPlant/PowreGenerator" are correctly handled at runtime.

The two results are the outcome of research activities carried out by a joint group of scientists from the Institute for High Performance Computing and Networking and the Institute for Applied Mathematics and Information Technologies, within the framework of the National Recovery and Resilience Plan [L1]. Our ongoing research aims at improving the overall security posture of modern software ecosystems when exposed to MATE attacks, e.g. to prevent the abuse of digital images for cloaking malicious payloads or to protect software artifacts via watermarks.

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

Link:
[L1] https://serics.eu

References:

[1] J. Wang, L. Yu, and X. Luo, "LLMIF: augmented Large Language Model for fuzzing IoT devices," in Proc. of the 2024 IEEE Symposium on Security and Privacy, pp. 196-209, San Francisco, CA, USA, 2024.

[2] F. Folino, et al., "Towards data- and compute-efficient fake-news detection: an approach combining active learning and pre-trained language models," SN Computer Science, Vol. 5, pp. 470:1–470:18, 2024.

[3] C. Cespi Polisiani, et al., "Mitigation of covert communications in MQTT topics through small language models," submitted to EuroCyberSec 2024 co-located with the 32nd International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Krakow, Poland, 2024.

Please contact:

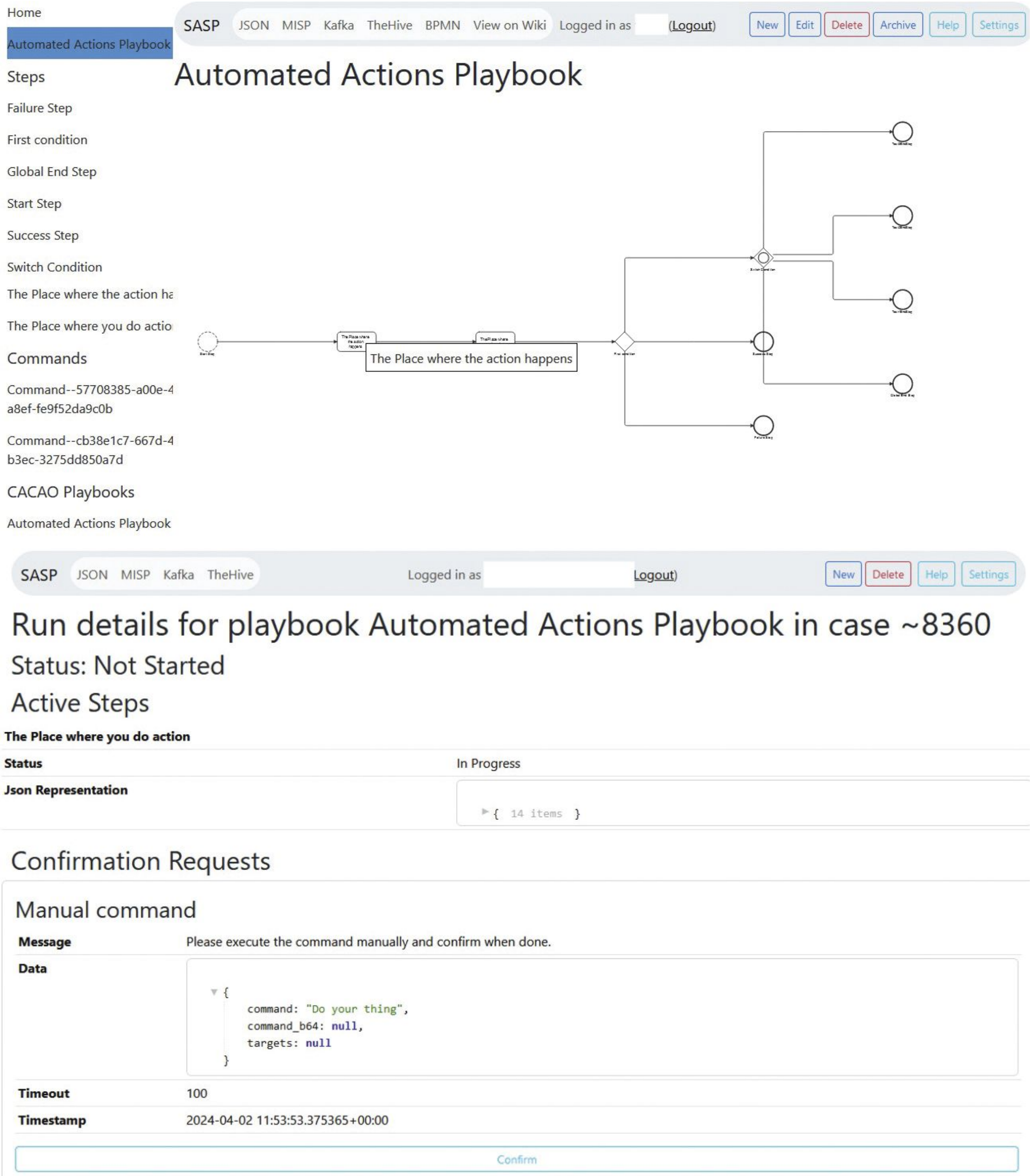
Gianluigi Folino, CNR-ICAR, Italy
gianluigi.folino@icar.cnr.it

Strengthening Cyber Defence through Cooperative Development and Shared Expertise in Incident Response Playbooks

by Mehdi Akbari Gurabi, Lasse Nitz, Charukeshi Mayuresh Joglekar, and Avikarsha Mandal (Fraunhofer FIT)

As cyberattacks evolve and become more complex, defenders require advanced tools for effective incident response. In the H2020 projects, SAPPAN and CyberSEAS, we conducted research to develop a cybersecurity playbook management system that provides a robust framework for creating, maintaining, and sharing standardised incident response procedures. The pilot validation shows how the system can be streamlined into current cybersecurity operations, towards compliance with the latest security recommendations and directives.

In today's continuously evolving digital landscape, enhancing cybersecurity practices is more critical than ever. This dynamic environment necessitates a robust and adaptive ap-



Automated Actions Playbook

Run details for playbook Automated Actions Playbook in case ~8360

Status: Not Started

Active Steps

The Place where you do action

Status	In Progress
Json Representation	▸{ 14 items }

Confirmation Requests

Manual command

Message	Please execute the command manually and confirm when done.

Data	▾{
	command: "Do your thing",
	command_b64: null,
	targets: null
	}

Timeout	100
Timestamp	2024-04-02 11:53:53.375365+00:00

Figure 1: Exemplary tool features in the UI include (top) a visual representation of the playbook in BPMN with an interactive modification option by clicking on the steps, and (bottom) an overview of the running step of a playbook with a command prompt for automatic execution or manual confirmation of action success.

proach aligned with the cybersecurity recommendations and guidelines such as the National Institute of Standards and Technology (NIST) incident response life cycle. The relevance of standardised incident response procedures is amplified by the introduction of Collaborative Automated Course of Action Operations (CACAO), a new standard on how playbooks can be designed and shared. Additionally, the recent implementation of the Network and Information Security Directive (NIS2) aims to enhance cybersecurity measures across the European Union, broadening its scope to include more sectors and imposing stricter obligations. The NIS2 directive highlights the need for comprehensive and standardised incident response procedures to enhance the collaboration between stakeholders. The general nature of the directive requires specific tools that can effectively implement its guidelines.

Our playbook management tool SASP [1] aims to support the transition to NIS2 [2] by offering a structured and practical approach for collaborative incident response [3] and enabling standardised reporting. It provides a cooperative environment to enhance cybersecurity resilience through the cross-European exchange of expertise and experiences.

SASP research and development began with SAPPAN [L1], where we developed advanced cybersecurity solutions focused on privacy-preserving sharing of cybersecurity threats intelligence. SAPPAN focused on facilitating sharing and automation, utilising advanced data analysis and machine learning to enable efficient response and recovery. It supports both single and cross-organisational collaboration through privacy-preserving data processing and sharing, enriching situational awareness, facilitating intrusion detection, and enabling knowledge sharing while maintaining data confidentiality. Building on this groundwork, we continued the development of SASP in the CyberSEAS [L2], refining and expanding our tool's capabilities, integrating it with other incident handling solutions, and validating it in a pilot within the energy sector. The CyberSEAS project aims to improve the resilience of Electrical Power and Energy Systems (EPES) by protecting them from disruptions caused by complex attack scenarios.

The SASP playbook management tool is designed for creating, maintaining, sharing, visualising, and exporting cybersecurity playbooks [1]. It features a user interface for creating playbooks, viewing them in Business Process Model and Notation (BPMN), exporting in JSON format, and sharing playbooks. By supporting the OASIS CACAO playbook format, we ensure playbooks are machine-readable and standardised. Figure 1 shows the interface of the tool for managing and execution of playbooks. Our tool offers a comprehensive suite of functionalities designed to manage machine-readable playbooks efficiently as listed here:

- Management: Adding, modification, deletion, import/export of playbooks.
- Sharing: Playbooks can be shared via the Malware Information Sharing Platform (MISP) using the already defined MISP object for playbook sharing in different formats. They further can be shared via Kafka between different instances of the SASP tool.
- Automated BPMN graph generation: Visualisation in BPMN format with interactive editing options for steps.
- Sanitisation: Ensures the removal of sensitive information from playbooks based on manually included tags using the Traffic Light Protocol (TLP).
- Versioning system: Facilitates continuous improvement of playbooks and getting access to the previous versions.
- Rich variety of supported workflow elements: Elements such as loops, and exclusive branching are supported by the vocabulary, which enhances the flexibility and functionality of playbooks.
- Integration: Integration with TheHive and Cortex for automated execution of incident response actions.

SASP has been tested and validated in real-world environments. This includes a system prototype demonstration by a Security Operations Centre (SOC) responsible for securing IT/OT systems, and a European national Computer Emergency Response Team (CERT). During the piloting phase, various methods for playbook management and sharing were utilised to establish standardised procedures for handling well-known attack scenarios, emphasising governance aligned with NIS2 requirements [2]. The tool integrates with MISP for incident reporting. This integration enables automated sharing of incident indicators, regulatory information, and playbook data. The playbook data suggests appropriate responses to incidents and adheres to the recommended national standard format for incident reporting.

As a demonstration scenario, the SOC shared a specific playbook via MISP (e.g. phishing, malware, or ransomware). This playbook was reviewed, generalised, and sanitised by the European national CERT before being disseminated to relevant stakeholders. Additionally, the SOC shared Indicators of Compromise (IoCs) via MISP, referencing the related playbook to enhance understanding and resolution of the incident. The created MISP event was subsequently shared with relevant recipients through the national CERT's MISP instance.

Input for this validation has been collected through a structured self-assessment questionnaire, a semi-structured pilot partners interview session, and observations and comments from validation partners. The high quality of the feedback and interview answers emphasised the tool's success in improving incident response knowledge management, enhancing collaborative incident response and facilitating information sharing among cybersecurity entities. This validation process confirms the tool's readiness for Technology Readiness Level 7 (System prototype demonstration in an operational environment) and its potential to enhance cybersecurity practices significantly.

We are considering open-sourcing [L3] our tool to enhance community engagement and collaborative improvement. For future research directions, we focus on automation processes for incident response and enhance the tool's capabilities by integrating the tool with de facto tools and services used in incident handling. Additionally, manually creating structured playbooks from legacy unstructured or semi-structured ones can be exhaustively time- and resource-intensive for security operators. Therefore, we aim to develop AI-assisted automation for translating legacy playbooks into standardised, machine-readable formats.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 833418 (SAPPAN) and No 101020560 (CyberSEAS).

Links:
[L1] SAPPAN website: https://sappan-project.eu
[L2] CyberSEAS website: https://cyberseas.eu
[L3] SASP GitHub repository:
https://github.com/Fraunhofer-FIT-DSAI/SASP

References:

[1] M. Akbari Gurabi et al., "SASP: a semantic web-based approach for management of sharable cybersecurity playbooks," in ARES, 2022.
[2] M. Akbari Gurabi et al., "Requirements for playbook-assisted cyber incident response, reporting and automation," Digital Threats: Research and Practice, 2024.
[3] L. Nitz et al., "From Collaboration to Automation: A Proof of Concept for Improved Incident Response," ERCIM news 129, 2022.

Please contact:

Mehdi Akbari Gurabi, Fraunhofer Institute for Applied Information Technology (FIT), Germany
mehdi.akbari.gurabi@fit.fraunhofer.de

Unlocking the Future: A Cloud-Based Artificial Intelligence Access Control System

by Hamidreza Yaghoubi, Navtaj Randhawa (University of Applied Sciences Burgenland), and Igor Ivkić (University of Applied Sciences Burgenland and Lancaster University, UK)

Traditional access control systems, such as key cards, PIN pads, and physical keys, face challenges in scalability, security, and user experience in today's digital world. We present a cloud-based entry system using Raspberry Pi hardware and Amazon Web Services (AWS) technologies like Lambda, Simple Storage Service (S3), and Rekognition. This solution (AWSecure Entry System) enhances security, streamlines authentication, and increases operational efficiency.

Traditional security systems are increasingly challenged by the rapid evolution of digital threats and often rely on outdated technologies that lack the flexibility and scalability required in today's dynamic security environment. Moreover, physical keys and passwords are increasingly vulnerable to theft, counterfeiting and hacking. Furthermore, managing these systems remains labour-intensive and error-prone, leading to potential security breaches with significant financial and reputational consequences. In response to these vulnerabilities, the integration of Artificial Intelligence (AI) into security systems offers a transformative solution. AI enhances monitoring, data analysis, and threat response, especially when combined with Internet of Things (IoT) devices that continuously collect and process environmental data [1, 3]. To address the need for more flexible, scalable, and secure solutions, this article presents an Amazon Web Services (AWS)-based entry system that uses AI facial recognition technology for authentication and access control. This approach not only reduces the risk of identity theft and errors, but also ensures a faster and more secure authentication process, highlighting its potential to improve security in critical areas.

The proposed system (AWSecure Entry System) uses an IoT device (Raspberry Pi with a built-in camera and display) to capture user images and send them to the cloud. The Raspberry Pi acts as an edge device, coordinating interactions between the user interface and the AWS cloud services. To accomplish this, the Raspberry Pi uses an Application Programming Interface (API) gateway [L1] to send the captured user images to the cloud, where Lambda functions manage the flow of data and trigger the appropriate security mechanisms. One of these mechanisms is AWS Rekognition [L2], which performs real-time facial recognition by comparing the captured images with those already securely stored in S3 [2]. Another mechanism (AWS DynamoDB) [L3] provides a low-latency database for storing and retrieving user credentials. The following figure provides an overview of the proposed AWSecure Entry System architecture from the edge device (Raspberry Pi) to the cloud (AWS):

To ensure reliability and efficiency, the system shown in Figure 1 was subjected to a rigorous performance analysis, assessing facial recognition accuracy, lighting conditions, distance and angle variability. This comprehensive evaluation confirmed the robustness and adaptability of the system, demonstrating its ability to meet the stringent security requirements of modern professional environments.

The evaluation of the AWSecure Entry System was conducted through the following series of structured test scenarios to assess its performance in a variety of real-world conditions:

- Scenario 1 – Registered vs. Unregistered User Access: testing system access control for registered and unregistered individuals.
- Scenario 2 – Performance Under Different Lighting Conditions: evaluating the system's facial recognition in bright, dim and completely dark environments.
- Scenario 3 – Facial Recognition with Face Rotations: assessing system accuracy with users facing the camera at different angles (e.g. direct, 45 degrees, 90 degrees).

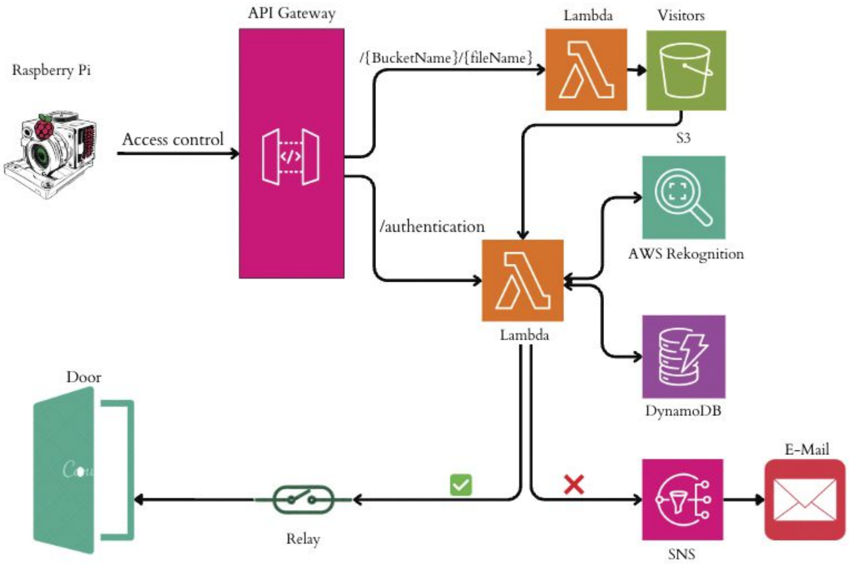


Figure 1: AWSecure Entry System Architecture from the Edge (Raspberry Pi) to the Cloud (AWS).

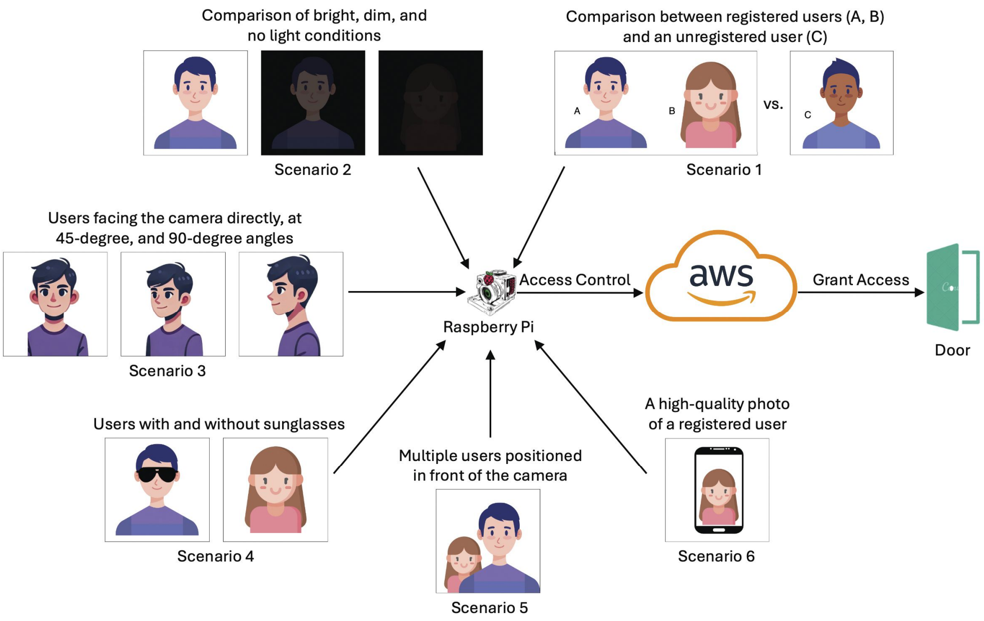

Comparison of bright, dim, and no light conditions
Scenario 2
Comparison between registered users (A, B) and an unregistered user (C)
A
B
vs.
C
Scenario 1
Users facing the camera directly, at 45-degree, and 90-degree angles
Scenario 3
Raspberry Pi
Access Control
aws
Grant Access
Door
Users with and without sunglasses
Scenario 4
Multiple users positioned in front of the camera
Scenario 5
A high-quality photo of a registered user
Scenario 6

Figure 2: Evaluation of different Facial Recognition Test Scenarios using Raspberry Pi and AWS under different conditions, resulting in Door Access Granted based on user identification and matching criteria.

- Scenario 4 – Recognition with Accessories: testing the system's ability to recognise users wearing accessories such as sunglasses.
- Scenario 5 – Multi-user Recognition: evaluating how the system handles multiple users facing the camera simultaneously.
- Scenario 6 – Spoofing Test: testing the system's resilience to spoofing attempts using photos or similar deceptive techniques.

Figure 2 shows the scenarios used for the evaluation of the AWSecure Entry System.

The system successfully demonstrated its basic functionality by accurately granting access to registered users and denying access to unregistered individuals (Scenario 1). When tested under different indoor lighting conditions, the system reliably recognised users in both bright and dim environments, but failed in complete darkness, highlighting the need for adequate lighting for effective facial recognition (Scenario 2). The system also demonstrated high accuracy when users were looking directly at the camera, but struggled with significant face rotations, particularly at 45- and 90-degree angles (Scenario 3). In scenarios where users were wearing accessories (sunglasses), the system maintained its recognition ability, albeit with a slightly lower similarity score (Scenario 4). In addition, when multiple registered users were looking at the camera at the same time, the system correctly prioritised and granted access to the user closest to the camera (Scenario 5). However, a critical vulnerability was identified during the spoofing test, where the system incorrectly granted access based on a high-quality photo, indicating a susceptibility to such attacks (Scenario 6).

The results of the evaluation show that while the AWSecure Entry System performs effectively in most scenarios (Scenarios 1-2 and 4-5), particularly under controlled lighting conditions and with minor facial obstructions, it faces challenges with extreme face angles and is vulnerable to spoofing attempts (Scenarios 3 and 6). These results highlight the need for further refinement, in particular to improve the system's ability to handle different face orientations and its resistance to spoofing attacks. Despite these limitations, the PoC evaluation provides valuable insights into the system's strengths and weaknesses of the system, providing a solid foundation for future development. To ensure the robustness and reliability of the system in real-world deployments, additional layers of security, such as liveness detection and multi-angle facial recognition, should be incorporated. These enhancements would address the identified vulnerabilities and contribute to a more secure and reliable access control solution.

Links:

[L1] https://kwz.me/hDA
[L2] https://kwz.me/hDD
[L3] https://kwz.me/hDF

References:

[1] R. Anand et al., Integration of IoT with Cloud Computing for Smart Applications, CRC Press, 2023. https://books.google.at/books?id=zK3GEAAAQBAJ

[2] V. Sharma, " Object detection and recognition using Amazon Rekognition with Boto3," in 6th International Conference on Trends in Electronics andInformatics (ICOEI), pp. 727–732, 2022. https://doi.org/10.1109/ICOEI53556.2022.9776884

[3] A. Nag, J N Nikhilendra, and M. Kalmath, " IOT based door access control using face recognition," in 3rd Int. Conf. for Convergence in Technology (I2CT), pp. 1–3, 2018. https://doi.org/10.1109/I2CT.2018.8529749

Please contact:

Hamidreza Yaghoubi ,University of Applied Sciences Burgenland, Austria, 2310781014@fh-burgenland.at

Navtaj Randhawa, University of Applied Sciences Burgenland, Austria, 2310781020@fh-burgenland.at

Engineering Secure, Trustworthy, and Ethically Sound AI-Based Computer Systems

by Yasin Ghafourian (Research Studios Austria), Markus Tauber (Research Studios Austria), Germar Schneider, and Andrea Bannert (Infineon Technologies Dresden GmbH & Co. KG), Olga Kattan (Philips), and Erwin Schoitsch (Austrian Institute of Technology)

To promote AI adoption in industrial cyber-physical systems (CPS) within Industry 4.0 and 5.0, it's crucial to develop tools that support the AI lifecycle and address knowledge gaps in standards among developers. Tailored guidance is needed to ensure AI solutions in safety-critical industries are trustworthy and ethically compliant. Given the fragmented standardisation landscape for CPS, this paper proposes an ethical compliance checklist and a self-assessment tool using large language models (LLMs) to help users navigate standards, close knowledge gaps, and ensure human-centred, legally compliant AI applications.

IoT and AI significantly enhance industry competitiveness, success, and sustainability by enhancing supply chain agility and fostering environmental awareness. However, while digitalisation offers numerous advantages, integrating AI into industrial processes, especially in Europe, raises ethical and legal concerns, posing risks for individuals and society [1].

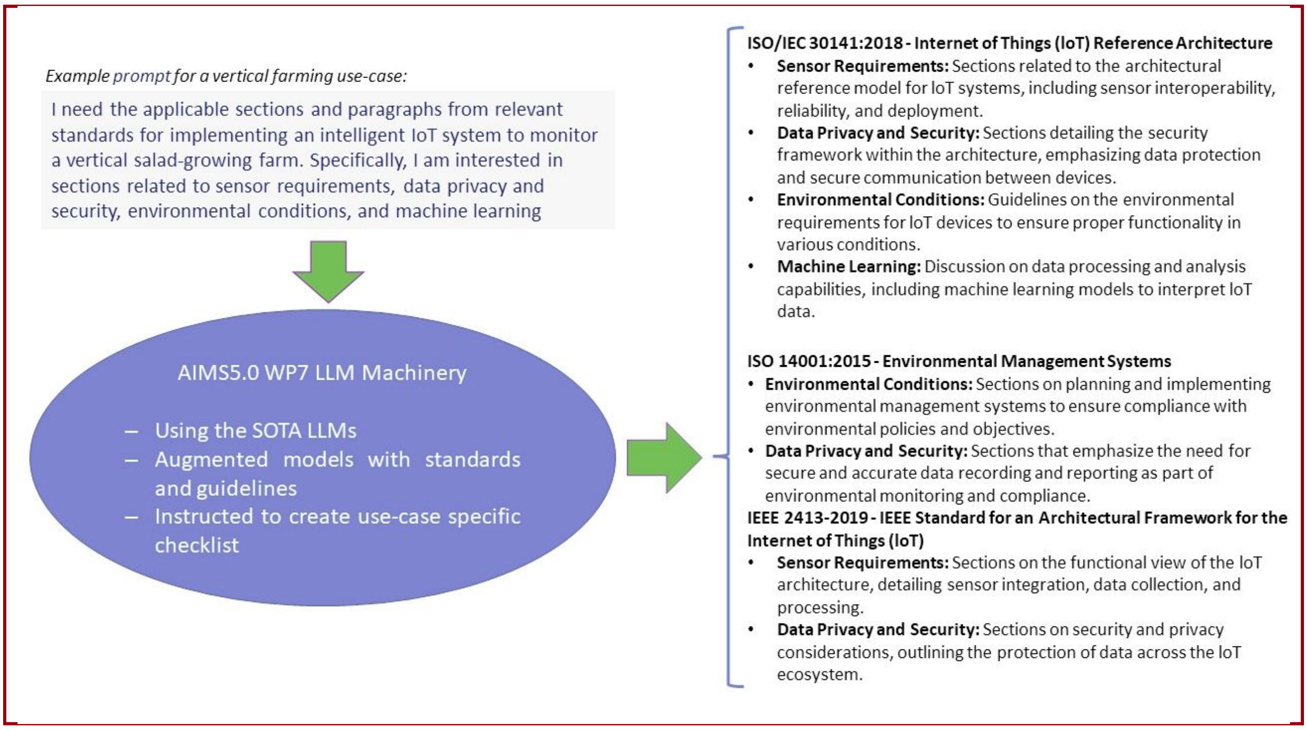
Beyond technical aspects like AI's trustworthiness and reliability, addressing ethical considerations is crucial when introducing AI into industry. Unfortunately, unlike the transparent impact of automation on workplaces, there are few internal regulations governing AI's ethical handling within industrial companies. The problem with the use of AI is that algorithmic decisions are typically opaque. One example is algorithms, which make it possible to digitise tasks within quality control without the need for humans to make important decisions. This would result in jobs being cut or decisions being made that are based solely on algorithms and could lead to errors if these algorithms were wrong or the environment were to change.

The opaque use of data in AI can compromise fairness and accuracy in employee performance appraisals, especially when sensors like cameras monitor workers in production environments. To prevent these issues, AI developers should rigorously evaluate their systems using a comprehensive checklist, allowing them to inform management and work councils. This ensures employee safety and well-being through timely and appropriate actions, including prompt company agreements.

Ethical Compliance Checklist

Given the challenge that AI often enters production invisibly through various projects, it is crucial to define company agreements on the use of AI at the earliest stages. Additionally, the ethical compliance checklist should incorporate mechanisms that identify and address knowledge gaps in awareness about these agreements. By doing so, the checklist can offer tailored explanations or resources to ensure that all stakeholders, regardless of their initial knowledge level, fully comprehend the ethical implications of AI use in their projects.

There are already many national and EU laws, guidelines and standards in place. Some examples include the General Data Protection Regulations, ISO/IEC JTC 1/SC 42/WG 3 on trustworthiness, the Assessment List for Trustworthy Artificial Intelligence (ALTAI)[L1], the General Equal Treatment Act, the Works Constitution Act, the Working Hours Act, the



Example prompt for a vertical farming use-case:

I need the applicable sections and paragraphs from relevant standards for implementing an intelligent IoT system to monitor a vertical salad-growing farm. Specifically, I am interested in sections related to sensor requirements, data privacy and security, environmental conditions, and machine learning

AIMS5.0 WP7 LLM Machinery

- Using the SOTA LLMs
- Augmented models with standards and guidelines
- Instructed to create use-case specific checklist

ISO/IEC 30141:2018 - Internet of Things (IoT) Reference Architecture

- **Sensor Requirements:** Sections related to the architectural reference model for IoT systems, including sensor interoperability, reliability, and deployment.
- **Data Privacy and Security:** Sections detailing the security framework within the architecture, emphasizing data protection and secure communication between devices.
- **Environmental Conditions:** Guidelines on the environmental requirements for IoT devices to ensure proper functionality in various conditions.
- **Machine Learning:** Discussion on data processing and analysis capabilities, including machine learning models to interpret IoT data.

ISO 14001:2015 - Environmental Management Systems

- **Environmental Conditions:** Sections on planning and implementing environmental management systems to ensure compliance with environmental policies and objectives.
- **Data Privacy and Security:** Sections that emphasize the need for secure and accurate data recording and reporting as part of environmental monitoring and compliance.

IEEE 2413-2019 - IEEE Standard for an Architectural Framework for the Internet of Things (IoT)

- **Sensor Requirements:** Sections on the functional view of the IoT architecture, detailing sensor integration, data collection, and processing.
- **Data Privacy and Security:** Sections on security and privacy considerations, outlining the protection of data across the IoT ecosystem.

Figure 1: A vision of our self-assessment tool.

Occupational Health and Safety Act, and the Occupational Safety Act [L2] [L3].

However, these laws alone are not sufficient for companies if they cannot demonstrate that AI methods used in various projects or directly in production might result in privacy breaches and/or violation of an existing legal framework. In general, health and safety issues are subject to co-determination in companies.

The idea is to use a checklist at the start of the project, during the first milestone, to identify key concerns. If necessary, this allows for defining measures with employees to prevent occupational health and safety issues and to create human-centred workplaces where AI is used for human benefit.

An initial version of the checklist includes the following points:

1. Does the project store or process sensitive personal data (e.g. address, salary, gender) or private information (e.g. social networks, hobbies) about the employee?
2. Does the project enable or expand performance and behaviour monitoring (e.g. cameras at workstations, reporting, time recording)?
3. Does the project impact the employee's working hours / work content and/or work organisation (e.g. work intensification, underload/overload, shifting of work, new or different tasks, monotony of tasks, transfer to another position if necessary)?
4. Does the project impact occupational health and safety (e.g. working under full protection, heavy lifting, noise pollution, psychological risks)?
5. Is there planned use of AI and/or algorithms that may impact employees?

Companies should provide the completed checklist to the project leader, manager, and relevant parties, such as worker councils, to ensure informed decision-making. If any points are affirmed during the project, the works council must be involved to protect co-determination rights, and define measures to ensure health and safety while mitigating AI-related risks.

Self-assessment Tool

To address the hidden impacts of AI algorithms on human health and safety rights while ensuring compliance with AI laws, we propose an AI-based self-assessment tool. This tool automates compliance checks and adapts to users' knowledge, bridging gaps to enhance adherence to guidelines. Unlike manual, subjective compliance checks, this tool uses LLMs to generate customised checklists based on relevant standards. Figure 1 illustrates our vision for this tool, highlighting its potential to streamline ethical compliance in AI applications. Building on the goals discussed in this position paper, we will develop a regulation-aware AI model that not only leverages LLMs for compliance but also integrates relevance models to assess and address knowledge gaps among developers and users. By fine-tuning the AI model based on standards, regulations, and ethical guidelines while simultaneously bridging these knowledge gaps, the self-assessment tool will provide more personalised and effective guidance, ensuring that users of varying expertise can achieve a deeper understanding and more reliable compliance with ethical standards.

This paper is supported by the AIMS5.0 project. AIMS5.0 is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Funding Authorities from involved countries under grant agreement no. 101112089.

Links:

[L1] https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

[L2] https://artificialintelligenceact.eu/de/das-gesetz/

[L3] https://www.bundesanzeiger.de/pub/en/start/

References:

[1] C. Huang, et al., "An overview of artificial intelligence ethics," IEEE Transactions on Artificial Intelligence, vol. 4, no. 4, pp. 799–819, 2023.

[2] P. Moertl and N. Ebinger, "The development of ethical and trustworthy AI systems requires appropriate human-systems integration: A white paper," InSecTT, White Paper, 2022. [Online]. https://www.insectt.eu/wp-content/uploads/2022/11/Trustworthiness-Whitepaper-InSecTT-Format-v02-1-1.pdf

Please contact:

Yasin Ghafourian, Research Studios Austria, Austria
yasin.ghafourian@researchstudio.at

Markus Tauber, Research Studios Austria, Austria
markus.tauber@researchstudio.at

Identity and the Web

by Simone Onofri (W3C)

Digital Identities have been in development for decades. As governments increasingly consider becoming providers and consumers of these technologies, they now more than ever have the potential to change the web and the concept of identity as we know it.

The world of Digital Identities has its origins many years ago when there were only centralised Identity Models. In recent years, federated Identity Models have spread to all sectors, from social networks and education to enterprises and governments. Federated Identity Models allow users to authenticate across different systems or platforms using a single set of credentials, often managed by a third party. Additionally, several projects have implemented the new paradigm of decentralised identity, where users have a digital wallet and control over their identity.

This has caught the interest of governments worldwide, which are designing wallets and decentralised identities for citizens, intending to have more secure digital credentials. Digital credentials, which can be more privacy-preserving than physical ones, allow users to control the amount of personal information shared, only disclosing necessary details, unlike physical documents that often reveal more data than needed.

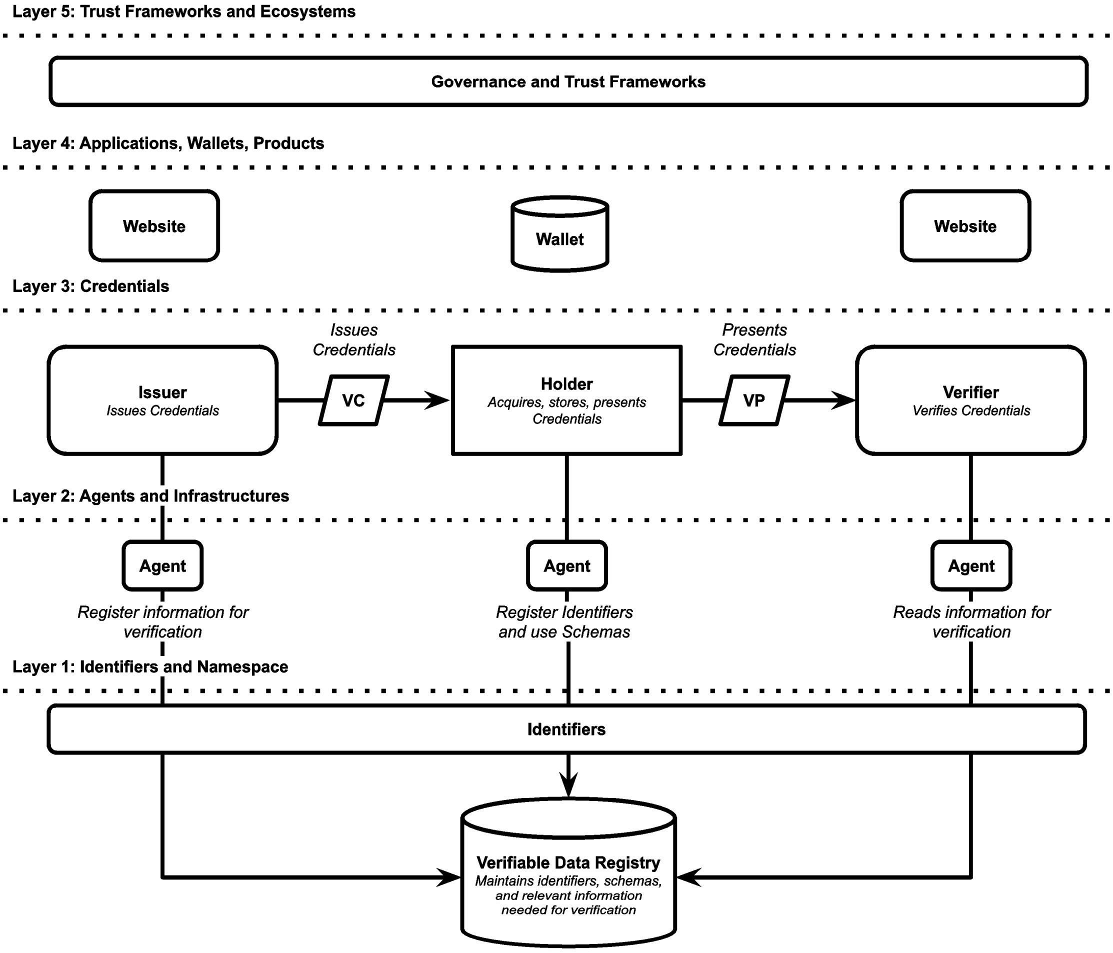

Layer 5: Trust Frameworks and Ecosystems
Governance and Trust Frameworks
Layer 4: Applications, Wallets, Products
Website
Wallet
Website
Layer 3: Credentials
Issues Credentials
Presents Credentials
Issuer
Issues Credentials
VC
Holder
Acquires, stores, presents Credentials
VP
Verifier
Verifies Credentials
Layer 2: Agents and Infrastructures
Agent
Agent
Agent
Register information for verification
Register Identifiers and use Schemas
Reads information for verification
Layer 1: Identifiers and Namespace
Identifiers
Verifiable Data Registry
Maintains identifiers, schemas, and relevant information needed for verification

Figure 1: Decentralised identity architecture.

Given the scope and scale of this innovation, Digital Identities are significantly impacting the web and, in particular, privacy and human rights, altering the assumptions and the balance that have shaped the web ecosystem. There is a large number of stakeholders, including governments, implementers, privacy and human rights advocates [1]. Considering the number of technologies available and their respective standards development organisations (SDOs), the document “Identity and the Web” [L1] was published in August 2024.

The document provides an overview of Digital Identities, focusing on decentralised identities and their impact on the web and users, to understand the possible threats and how to mitigate them, both from a technology and governance perspective.

The document also highlights several areas where standardisation, guidelines, and interoperability could play a crucial role in managing these changes [2]. These include enabling passwordless credentials for authentication and payments, and facilitating federated identity on the web without relying on third-party cookies. It also emphasises the importance of modelling security, privacy, and human-rights threats associated with decentralised credentials (Figure 1). Additionally, standardising digital credentials on the web can help mitigate issues like surveillance, censorship, intrusion, and discrimination, while ensuring interoperability. Addressing these threats requires careful consideration at both technological and governance levels.

The paper also proposes different use cases of the systemic impact on both the market side and the human side [L2].

In conclusion, standards are crucial for driving innovation and mitigating threats. Coordinated efforts among SDOs, people, and governments are needed to ensure Digital Identities are beneficial, balancing both technology and governance. Additionally, the impact on security, privacy, and human rights should be closely monitored, with threat modelling as a key tool.

Links:

[L1] https://www.w3.org/reports/identity-web-impact/

[L2] https://www.w3.org/reports/identity-web-impact/#uses-cases

References:

[1] UN Office of the High Commissioner for Human Rights, “Human rights and technical standard-setting processes for new and emerging digital technologies: report of the Office of the United Nations High Commissioner for Human Rights,” 2023. https://digitallibrary.un.org/record/4031373?v=pdf

[2] H. Flanagan, “Identity on the Web,” 2024. https://www.w3.org/2024/04/AC/talk/identity

Please contact:

Simone Onofri, W3C Security Lead
simone@w3.org

Advancing Research: The Role of the EOSC and EOSC Support Office Austria

by Katharina Flicker (EOSC Support Office Austria, SBA Research, TU Wien), Stefan Hanslik (BMBWF), Tereza Kalová (Vienna University Library, University of Vienna)

In 2015, the vision of a European federated and open multi-disciplinary environment, known as the European Open Science Cloud (EOSC), was born. EOSC represents a significant leap towards more integrated and accessible scientific resources across Europe. In this context, the EOSC Support Office Austria (EOSC SOA) was launched to coordinate Austria's contributions to EOSC's implementation. If you are interested in exploring how EOSC and EOSC SOA are shaping the future of research and why they are crucial for the ERCIM community, continue reading.

The vision of EOSC was brought forth in order to help researchers, innovators, companies and citizens find and reuse data, tools and services by developing a Web of FAIR data and services [L1]. While the EOSC Associa tion was established to guide this initiative on the European level [L2], many countries are laying the foundations for national nodes, ensuring that national needs are incorporated into the strategic planning of EOSC (e.g. via the SRIA [L3]). Against this background, the EOSC Support Office Austria (EOSC SOA) was formed in Austria in 2021 by ACOnet as the Mandated Organization, commissioned by the Federal Ministry of Education, Science and Research (BMBWF). Its establishment also involved CCCA, FAIR Office Austria, JKU, Natural History Museum Vienna, TU Wien, TU Graz and the University of Vienna. Since then, numerous institutions and initiatives have joined the growing EOSC SOA community: the Academy of Fine Arts Vienna, Ludwig Boltzmann Gesellschaft, Open Knowledge Maps, UBIFO, the University of Graz, Vetmeduni and WU Wien. This demonstrates the broad scope of organisations for whom the EOSC is of core interest and their commitment to collaborating to prepare for and benefit from these developments.

EOSC SOA aims to coordinate Austrian contributions in the implementation of EOSC while fostering EOSC readiness in terms of infrastructure and staff training at partner institutions. Additionally, it supports the development of Austrian Open Science policies [L4]. A broad variety of activities are employed to achieve this aim. These include EOSC SOA Working Groups [L5] (WGs), the EOSC Café [L6], participation in EOSC-related projects such as EOSC Focus [L7] and Skills4EOSC[L8], participation in both former [L9] and new [L10] EOSC Association Task Forces as well as regular exchange with both national and European stakeholders and within the EOSC Association. To share information with partner institutions, as well as to bridge between EOSC and GAIA-X, two similar initiatives that are both highly relevant to research-performing organisations, ERCIM established a working group. The goals and activities of these groups are as follows.

Group photo, 3rd General Assembly of the EOSC Support Office Austria / Austrian EOSC Mandated Organisation, in the Museum of Natural History Vienna. 18 October 2023. Photo: ©NHM Wien, Alice Schumacher.

EOSC SOA WGs aim to progress topics relevant to EOSC. To achieve this, they organise events and workshops, conduct surveys, or draft recommendations on particular subjects. They are time-bound and focus on specific topics. External experts can be invited if needed. Currently, there are five WGs: Austrian Country Report, Collections, Researcher Engagement in Austria, Technical Infrastructure, and Training.

As one of the working tools, the EOSC Café serves as a collaborative and open forum with the EOSC SOA, bringing together experts from both within and outside the initiative. These experts come from various disciplines of science and research, research policy, society, and the industry. They provide the EOSC Support Office with valuable external insights and perspectives while supporting EOSC-related activities both domestically and across Europe. EOSC Café participants meet up at least three times per year.

The ERCIM working group initially convened during the ERCIM spring meetings in 2022. Several participants met to explore the opportunities and impact of EOSC and Gaia-X, aiming to ensure future developments are beneficial to ERCIM members. The working group usually meets during the spring and fall meetings to exchange updates on recent events. During this period, members conducted informal surveys on the perception and evaluation of both EOSC and Gaia-X within their institutions.

It became evident that while both initiatives lacked sufficient tangible results to be actionable at the management level, their significance was widely acknowledged, highlighting the need for regular information exchange. Against this background, it was decided to closely monitor their developments and prepare to be ready for upcoming advancements. As EOSC and GAIA-X gain momentum with the development and rollout of services, a major impact on research-performing organisations is expected. Thus, early alignment, the establishment of national support structures, as well as the creation of contact points at research institutes, will facilitate readiness for these developments. This approach aims to effectively integrate, operate, and utilise these initiatives to the benefit of researchers and their research outcomes.

Links:
[L1] https://kwz.me/hDX
[L2] https://eosc.eu/
[L3] https://kwz.me/hDZ
[L4] https://eosc-austria.at/
[L5] https://eosc-austria.at/working-groups/
[L6] https://eosc-austria.at/eosc-austria/governing-bodies/
[L7] https://eosc.eu/eosc-focus-project/
[L8] https://www.skills4eosc.eu/
[L9] https://kwz.me/hFC
[L10] https://eosc.eu/eosc-task-forces/

Please contact:
EOSC Support Office Austria
contact@eosc-austria.at

Preliminary Call for Participation:

AIOTI Workshop on Semantic Interoperability for Digital Twins

Laying the foundations for the next generation of information systems

Inria Campus, Sophia Antipolis, France, 5-6 February 2025

We invite you to the AIOTI Workshop on Semantic Interoperability for Digital Twins, aimed at advancing the field of semantic interoperability and laying the groundwork for the next generation of information systems. This workshop is focused on finding practical solutions for real-world industrial challenges. Discussions will involve a mix of invited talks, selected presentations, breakout sessions, and community-submitted presentations, all aimed at addressing industrial needs for semantic interoperability. The workshop will provide ample opportunities for real discussions.

Goal of the Workshop

What do we want to get out of this workshop?

We want to advance the area of semantic interoperability by:
- Reaching a common understanding of the current state of the art
- Showing success stories and best practices

Identifying what is missing and what new technologies can bring
- Concluding with creating recommendations and/or a roadmap composed of clearly identified next steps

Scope

The workshop wants to focus on solutions for industry rather than an academic event, along with encouraging greater take-up of semantic technologies, including controlled vocabularies, taxonomies, and ontologies. Semantic interoperability is essential for supply chains as well as for repairs and recycling in relation to the plans for the circular economy as part of the EU's Green Deal.Topics

The workshop will cover the following topics:

- State of the Art: Establishing a shared understanding of the current landscape of semantic interoperability.
- Success Stories and Best Practices: Showcasing successful implementations and identifying common practices, e.g. at GS1 and eclass.
- Challenges and New Technologies: Exploring what is missing and the potential contributions of emerging technologies, including AI and digital twins. Discussing metrics for evaluating the maturity of vocabularies, taxonomies, and ontologies
- Roadmap and Next Steps: Concluding with recommendations for advancing the field and proposing a roadmap for future development.

At the time of this publication, the workshop is in the preparatory phase. If you are interested in participating, please contact Dave Raggett at dsr@w3.org.

Call for Proposals

Dagstuhl Seminars and Perspectives Workshops

Schloss Dagstuhl – Leibniz-Zentrum für Informatik is accepting proposals for scientific seminars/workshops in all areas of computer science, in particular also in connection with other fields.

If accepted, the event will be hosted in the seclusion of Dagstuhl's well known, own, dedicated facilities in Wadern on the western fringe of Germany. Moreover, the Dagstuhl office will assume most of the organisational/ administrative work, and the Dagstuhl scientific staff will support the organizers in preparing, running, and documenting the event. Thanks to subsidies the costs are very low for participants.

Dagstuhl events are typically proposed by a group of three to four outstanding researchers of different affiliations. This organizer team should represent a range of research communities and reflect Dagstuhl's international orientation. More information, in particular details about event form and setup, as well as the proposal form and the proposing process, can be found on

https://www.dagstuhl.de/dsproposal

Schloss Dagstuhl – Leibniz-Zentrum für Informatik is funded by the German federal and state government. It pursues a mission of furthering world class research in computer science by facilitating communication and interaction between researchers.

Important Dates

- *Current submission period:* October 15 to November 1, 2024
- *Seminar dates:* Between November 2025 and September 2026 (tentative)
- *Next submission period:* April 1 to April 15, 2025.

W3C@30

The first of October 2024 marked the 30th anniversary of the World Wide Web Consortium (W3C). To mark this milestone, W3C hosted W3C@30, aligning it with the TPAC 2024 conference in Anaheim, CA, USA, where the web standards community gathered.

There, 30 years of W3C and web advancements were celebrated, highlighting W3C's global impact, and looking ahead to the Web's future. The Web has become a central part of daily life for billions, and the Web Consortium has been instrumental in shaping its evolution over the past 30 years.

ERCIM became the European host of W3C in 2003. Today, ERCIM is the European partner of W3C.

More information:
W3C@30: https://kwz.me/hFh
W3C 30th anniversary clip: https://kwz.me/hFi
Press release: https://kwz.me/hFs

Towards a Shared AI Strategy for European Digital Science Institutes and Organisations

Generative AI is in a revolutionary phase of development. This has and will continue to have profound impact on society as well as science. It serves as enabler but also presents inherent challenges. It is important to develop a realistic understanding on the transformative nature of generative AI for the sciences and society, and how to responsibly use this revolutionary technology. As a network of European centres of excellence in digital technology, the ERCIM institutes are well positioned to contribute to this discussion within a national and European context.

Report on the 2024 ERCIM Visionary Event "Challenges and Opportunities of Foundational Models and Generative AI for Science and Society"
Experts from across Europe gathered on 16 April 2024 in Brussels, at the Maison Irène et Frédéric Joliot Curie, to discuss the rapidly evolving landscape of generative AI and large language models (LLMs) during the ERCIM visionary event titled "Challenges and Opportunities of Foundational Models and Generative AI for Science and Society." The event featured focused sessions covering the current state of affairs, an overview of the European political landscape, and the impact of LLMs on science and society. The day concluded with a panel discussion on the role of digital sciences in addressing the opportunities and challenges posed by LLMs and generative AI.

The authors of the report, who were also the event's organisers, documented the discussions, opinions, and statements made during the event, highlighting key insights.

The report is divided into two parts: the first part focuses on LLMs and AI at the EU level, along with specific technical perspectives and applications. The second part presents broader views on the impact of AI on society and science, along with reflections from the panel discussion. These sections offer several key takeaways and high-level recommendations distilled by the authors.

The report that aims to help institutes and organisations to develop an organisational understanding and possibly a strategy, to promote the responsible use and development of generative AI.

The report is available for download at:
https://www.ercim.eu/publications/strategic-reports

In Memoriam: Prof. Dr. rer. pol. Matthias Jarke (1952–2024)

It is with deep sadness that we announce the passing of Prof. Matthias Jarke, a visionary computer scientist who made lasting contributions to both the Fraunhofer-Gesellschaft and ERCIM. His work left a significant impact on applied research in information systems, data management, and digital innovation.

Prof. Jarke was deeply committed to the Fraunhofer-Gesellschaft, serving as the Director of the Fraunhofer Institute for Applied Information Technology FIT from 2000 to 2021. Under his leadership, Fraunhofer FIT became a leading institution in human-computer interaction, requirements engineering, and data warehousing. He successfully bridged academia and industry, establishing Fraunhofer FIT as a major player in digital transformation.

From 2010 to 2015, Prof. Jarke chaired the Fraunhofer ICT Group, one of the largest divisions within the Fraunhofer-Gesellschaft. His leadership furthered Fraunhofer's mission of advancing applied research across Europe and globally. Prof. Jarke also left a profound mark on ERCIM. From 2008 to 2017, he represented the Fraunhofer-Gesellschaft on the ERCIM Board of Directors, serving as Vice-President in 2008 and 2009. His commitment to collaborative research in computer science and applied mathematics strengthened ERCIM's role in advancing international cooperation.

In addition to his work at Fraunhofer, Prof. Jarke held the Chair of Information Systems and Databases at RWTH Aachen University from 1991 to 2021. He played a pivotal role in founding the Bonn-Aachen International Center for Information Technology (b-it), promoting interdisciplinary education in computer science and business. He also helped advance global education as the founding dean of the Faculty of Engineering and Computer Science at GUTech in Oman.

Prof. Jarke's foundational research in requirements engineering, meta-modelling, and data warehousing laid the groundwork for modern software systems. His achievements earned him numerous accolades, including being named a Fellow of the ACM and a member of acatech, the German National Academy of Science and Engineering.

As a mentor, Prof. Jarke was devoted to nurturing young scientists, many of whom now hold leading positions in academia and industry. His legacy of mentorship, leadership, and scientific excellence will continue to inspire future generations.

Prof. Matthias Jarke will be remembered for his visionary leadership and the deep impact he had on the global research community. He will be greatly missed by his family, colleagues, students, and all who knew him.

ERCIM – the European Research Consortium for Informatics and Mathematics is an organisation dedicated to the advancement of European research and development in information technology and applied mathematics. Its member institutions aim to foster collaborative work within the European research community and to increase co-operation with European industry.

ERCIM is the European Partner of the World Wide Web Consortium.

Consiglio Nazionale delle Ricerche
Area della Ricerca CNR di Pisa
Via G. Moruzzi 1, 56124 Pisa, Italy
www.iit.cnr.it

Centrum Wiskunde & Informatica
Science Park 123,
NL-1098 XG Amsterdam, The Netherlands
www.cwi.nl

Fonds National de la Recherche
6, rue Antoine de Saint-Exupéry, B.P. 1777
L-1017 Luxembourg-Kirchberg
www.fnr.lu

Foundation for Research and Technology – Hellas
Institute of Computer Science
P.O. Box 1385, GR-71110 Heraklion, Crete, Greece
www.ics.forth.gr

Fraunhofer ICT Group
Anna-Louisa-Karsch-Str. 2
10178 Berlin, Germany
www.iuk.fraunhofer.de

INESC
c/o INESC Porto, Campus da FEUP,
Rua Dr. Roberto Frias, nº 378,
4200-465 Porto, Portugal
www.inesc.pt

Institut National de Recherche en Informatique
et en Automatique
B.P. 105, F-78153 Le Chesnay, France
www.inria.fr

I.S.I. – Industrial Systems Institute
Patras Science Park building
Platani, Patras, Greece, GR-26504
www.isi.gr

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering, N 7491 Trondheim, Norway
http://www.ntnu.no/

RISE SICS
Box 1263,
SE-164 29 Kista, Sweden
http://www.sics.se/

SBA Research gGmbH
Floragasse 7, 1040 Wien, Austria
www.sba-research.org/

Eötvös Loránd Research Network
Számítástechnikai és Automatizálási Kutató Intézet
P.O. Box 63, H-1518 Budapest, Hungary
www.sztaki.hu/

University of Cyprus
P.O. Box 20537
1678 Nicosia, Cyprus
www.cs.ucy.ac.cy/

Institute for Software Engineering and Software Technology
"Jose María Troya Linero", University of Malaga
Calle Arquitecto Francisco Peñalosa, 18, 29010 Málaga
https://gp.uma.es/itis

Universty of Warsaw
Faculty of Mathematics, Informatics and Mechanics
Banacha 2, 02-097 Warsaw, Poland
www.mimuw.edu.pl/

VTT Technical Research Centre of Finland Ltd
PO Box 1000
FIN-02044 VTT, Finland
www.vttresearch.com