Cyber Security for Europe

D3.9

Research challenges and requirements for secure software development

Document Identification	
Due date	31 March 2020
Submission date	31 March 2020
Revision	1.0

Related WP	WP3	Dissemination Level	Public
Lead Participant	DTU	Lead Author	Alberto Lluch Lafuente (DTU)
Contributing Beneficiaries	NEC, CYBER, C3P, CNR, SINTEF, UCY, UPS-IRIT, POLITO	Related Deliverables	D3.1, D4.1, D5.1

Abstract: This document presents deliverable “D3.9: Research Challenges and Requirements for Secure Software Development”. The document focuses on security and privacy issues in the lifecycle of software and describes a set of identified research challenges and required actions. Challenges stem from problems being currently investigated by research groups within CyberSec4Europe, as well as from cybersecurity issues and requirements identified in selected sectors and demonstrators under study in CyberSec4Europe. The document describes short-term research to be carried out within CyberSec4Europe, and hints at long-term research challenges to be addressed after the project, in 5 years and beyond.

This document is issued within the CyberSec4Europe project. This project has received funding from the European Union's Horizon 2020 Programme under grant agreement no. 830929.

The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Executive Summary

Task 3.3 “Software Development Lifecycle” of CyberSec4Europe has as one of its main goals to identify research challenges, requirements and approaches in all stages of the lifecycle of software [CDL19]. This document presents deliverable “D3.9: Research Challenges and Requirements for Secure Software Development”, which aims at supporting such a goal by describing a set of research challenges and approaches related security and privacy issues in the lifecycle of software.

In the preparation of this deliverable, several challenges were considered. A number of them was finally identified and based on several criteria. The main selection criteria were

- Expertise of the research groups participating to Task 3.3, to ensure success in addressing the challenges;
- Relevance of the challenge to the current research activities of the research groups participating to Task 3.3 to improve focus and synergies;
- Relevance to cybersecurity issues and requirements identified in selected sectors and demonstrators within CyberSec4Europe in deliverables “D4.1 - Requirements Analysis from Vertical Stakeholders” [D4.1] and “D5.1 - Requirements Analysis of Demonstration Cases” [D5.1], respectively to enable collaboration with WP5 and WP4;
- Suitability of the research assets identified in an initial phase of WP3 (cf. "D3.1 Common Framework Handbook 1") to be part of the solutions to the challenges to substantiate the results.

Content of the deliverable. The document presents a total of 11 research challenges relevant to the lifecycle of software, summarized in Table 1. Each challenge is presented in its own section with the following structure:

- **Scenario**: A motivating story inspired by the demonstration cases [D5.1] and the verticals [D4.1] under study in CyberSec4Europe. A more explicit connection is then listed in 12A.3.
- **Research Challenge**: The main security and privacy concerns being considered, and a brief account on why current state of practice and research does not offer completely satisfactory solutions.
- **Short-Term Research**: A set of research activities, approaches and assets that will be put forward within CyberSec4Europe in order to address the identified research challenge, and connections to state-of-the-art.
- **Long-Term Research**: Research areas that should be included in research roadmaps and that should be addressed in the future, beyond CyberSec4Europe, in order to fully address the identified research challenge and related future ones.

An appendix provides additional information:

- Appendix 12A.1 relates each short-term research activity to the main responsible partner.
- Appendix 12A.2 provides the full list of research assets mentioned in the chapters. Such assets are software tools and frameworks that will be applied and developed as part of the hereby described research and possibly in other activities of CyberSec4Europe.

- Appendix 12A.3 provides a mapping of research assets into the requirements of the vertical sectors [D4.1] and demonstrators [D5.1]. Such mapping is included for the convenience of the reader, who can find a more detailed description in deliverable “D3.1 Common Framework Handbook 1” [Saa19].
- Appendix 12A.4 provides a mapping of research assets into the building blocks of the global architecture of CyberSec4Europe (described in detail in deliverable “D3.1 Common Framework Handbook 1” [Saa19]).
- Appendix 12A.5 provides a mapping of short-term research activities into the building blocks of the global architecture of CyberSec4Europe.

Table 1. Research Challenges in Software Lifecycle

Challenge 1: Assessing Security and Privacy Through the Life Cycle
- **Short-term research:**
  - Logical foundations of privacy and security
  - Protocol verification
  - Quantitative security
- **Long-term research:**
  - Lightweight formal methods for industrial compliance

Challenge 2 : Assessing Privacy Properties of Complex Systems
- **Short-term research:**
  - Privacy definitions for complex systems
  - Privacy policy languages for complex systems
  - Privacy analyses for complex systems
- **Long-term research:**
  - Holistic approaches to privacy, utility and efficiency

Challenge 3: Privacy protection and User Empowerment in IoT
- **Short-term research:**
  - Identity management for IoT
  - Authentication, Authorization and Access Control for IoT
  - Secure infrastructures for IoT
- **Long-term research:**
  - Privacy protection in untrusted IoT environments

Challenge 4: Securing Unsafe and Legacy Software
- **Short-term research:**
  - Software Hardening
- **Long-term research:**
  - Hardening by design

Challenge 5: Protecting Leaked Credentials
- **Short-term research:**
  - Password hardening
- **Long-term research:**
  - Credentials-protecting authentication

Challenge 6: Secure Access Control in Heterogeneous Systems
- **Short-term research:**
  - Formal verification for secure access control
  - Enriched formal models to cope with security threats
  - Analysis of security issues and synthesis of corrections
- **Long-term research:**
  - Scalable formal models, verification and synthesis
  - Lightweight run-time security verification

Challenge 7: Manageable and Understandable Security Engineering
- **Short-term research:**
  - Formal security requirements
  - Formal threat specification
  - Security patterns
  - Model-based integration & validation of security patterns
- **Long-term research:**
  - Ensure continuous service and minimal maintenance
  - Pivoting attacks

Challenge 8: Unreliable Risk Estimates
- **Short-term research:**
  - Requirements for data-based risk estimates
- **Long-term research:**
  - Risk estimates with dynamic data sources

Challenge 9: Automated and Verified Network Security Configuration in Highly Dynamic Environments
- **Short-term research:**
  - Automated network security functions
  - Performance and scalabilty of automated network security
- **Long-term research:**
  - Automated verified configuration for large-scale networks
  - Autonomic elicitation of security policies

Challenge 10: Scalable and Private Industrial Blockchain
- **Short-term research:**
  - Scalable secure and practical consensus layers
  - Smart contract security
  - Efficient privacy-preserving blockchain protocols
- **Long-term research:**
  - Industry-ready scalable, privacy-preserving blockchains

Challenge 11: Scaling TEEs for Cloud Applications
- **Short-term research:**
  - Protocols for TEEs in cloud-based applications
- **Long-term research:**
  - Adaptable by-design TEEs

How to read the deliverable. The deliverable has been written so that each challenge is described in a standalone chapter that can be read individually by a general audience. Project-specific details that are not essential to understand the main research challenges and approaches (like partners, relations to other project activities, etc.) are provided in the appendices, so to make the chapters more accessible to an audience not familiar with the internal structure of CyberSec4Europe.

The order of the chapters is arbitrary and does not indicate any preferred order of reading. For the convenience of the reader we propose here several ways of reading the chapters based on a set of themes related to the lifecycle of software.

Reading option 1 (Precise definitions of security and privacy and related rigorous analysis and design techniques to support the lifecycle of software). Formal methods constitute a key set of techniques and technologies to provide mathematical rigor to security and privacy properties, and their analysis, and to effectively achieve security-by-design and privacy-by-design. The introduction of such methods in early stages is still posing challenges to software development, as discussed briefly in Chapter 1. Chapter 7 provides a more concrete example by discussing challenges to security requirements engineering and research to support security engineering activities with formal modelling languages and tools. Formal methods have evolved into domain-specific approaches with more developments in certain classes of software, in particular for security and safety-critical components of IT systems. Chapters 1, 6 and 9 focus on several challenges and research activities related to formally verified software in for key components of IT infrastructures such as communication protocols (Chapter 1), communication networks (Chapter 9), and access control systems (Chapter 6). While, traditionally, formal methods have been associated to early stages, their use has been pushed along the entire lifecyle of software. Chapter 6 discusses examples of run-time verification of software. While most of the challenges related to formal methods focus on security, this document includes chapters on the formal treatment of privacy. In particular, Chapters 2 and 3 focus on challenges and research on formal languages and models for privacy, and their analysis techniques.

Reading option 2 (Secure and privacy-respecting software in trusted and untrusted environments). Developing software components with well-assessed security and privacy properties is easier when one can assume that the component will be deployed on an environment made of interacting parties, software platforms and infrastructures that can be trusted. Chapters 10, 11, 3 focus on challenges to the development of trusted execution environments. In particular, Chapter 10 deals with industrial blockchains, which despite of offering a nice trusted execution environment for software based on smart contracts, demand for privacy properties and sizes that require further research developments. Chapter 11, instead, deals with scaling of TEEs for cloud applications. Last, Chapter 3 focuses on the challenges of building secure IoT infrastructures. Chapter 3 deals also with the case in which components are deployed in untrusted environment, with a focus on privacy protection in the IoT. Challenges and research related to the analysis and verification of software in presence of untrusted environment is also covered in Chapters 1, 6, 7 and 9. While most of our document focuses on new software to be developed according to the principles of security-by-design and privacy-by-design, there is still a need to deal with legacy software and with software developed with unsafe technologies. Chapter 4 discusses some challenges and research in that area.

Reading option 3 (Security and privacy properties of software may be quantitative and things can go wrong). The principles of security-by-design and privacy-by-design aim at considering security and privacy goals from the beginning of the lifecycle of software so as to remove vulnerabilities as early as possible. However, in many scenarios, it is more realistic to aim at minimizing vulnerabilities and/or measuring them if removing them is not feasible. Threat and risks analysis techniques are typically used at early stages of software development to identify those vulnerabilities, their likelihood and the effectiveness of countermeasures. Chapters 1 and 8 describe some challenges and research in that area. In particular, Chapter 1 regards the use of formal methods for quantitative security analysis, while Chapter 8 focuses on the challenges to data-driven risk-analysis. Another example where security and privacy are inherently quantitative are notions of privacy such as differential privacy, discussed in Chapter 2. Finally, one needs to assume that things can go wrong and security and privacy failures may happen and need to be addressed. As an example, Chapter 5 discussed the challenge of dealing with leaked credentials.

Document information

Contributors

Name	Partner
Alberto Lluch Lafuente	DTU
Flemming Nielson	DTU
Sebastian Mödersheim	DTU
Anders Schlichtkrull	DTU
Alessandro Sforzin	NEC
Claudio Soriente	NEC
Liina Kamm	CYBER
Rolando Martins	C3P
João Soares	C3P
Luis Antunes	C3P
Luca Durante	CNR
Manuel Cheminod	CNR
Elias Athanasopoulos	UCY
Brahim Hamid	UPS-IRIT
Aida Omerovic	SINTEF
Karin Bernsmed	SINTEF
Per H Meland	SINTEF
Riccardo Sisto	POLITO

Reviewers

Name	Partner
Joao Soãres	C3P
Riccardo Sisto	POLITO
Ahad Niknia (High-level review)	GUF

History

0.01	2020-01-27	Alberto Lluch Lafuente	1st Word-based draft based on internal LaTeX draft.
0.02	2020-01-27	Alberto Lluch Lafuente	2nd draft with fixes and executive summary.
0.03	2020-01-28	Alberto Lluch Lafuente	3rd draft with fixes and minor updates.
0.04	2020-01-31	Alberto Lluch Lafuente	4th draft with sections more homogeneous, fixed typos, revised main references, formatting, and summary.
0.05	2020-02-13	Alberto Lluch Lafuente	5th draft with updates from partners and with extended executive summary. Version submitted for first high-level review.
0.06	2020-02-14	Alberto Lluch Lafuente	6th draft with some minor fixes.
0.07	2020-03-11	Alberto Lluch Lafuente	7th draft with all comments by reviewers and WPL addressed. New appendices added to cope with the reviewers suggestions. Local refences removed.
0.08	2020-03-20	Alberto Lluch Lafuente	8th draft with approval by WPL and reviewers and some minor fixes.
0.09	2020-03-27	Alberto Lluch Lafuente	9h draft with some layout changes as requested by project coordinator.
1.0	2020-03-30	Ahad Niknia	High-level review, final check and preparation for submission

List of Contents

1 Assessing Security and Privacy Through the Entire Software Life Cycle

Scenario

A European consortium develops a new platform to support and promote sharing of medical data. A long, thorough, process involving significant human resources is spent to certify that the platform lives up to the strictest security and privacy regulations and guarantees. European hospitals, pharmaceutical companies, and research institutes start using the platform as an effective tool to boost research and development of new medical treatments. Shortly after the deployment of the platform, a new cybersecurity threat is discovered. The users of the platform demand that the consortium proves that their system still lives up to the security and privacy requirements despite of the new threat. How can the consortium effectively show, at any time, and with reasonable effort, that the platform is secure and privacy-respecting?

Research Challenge

Security and privacy goals cannot always be assessed once and for all. Existing regulations like GDPR depend on "the available technology at the time" [Cou16], and new regulations are likely to be introduced as a response to new threats and societal demands. Moreover, software is continuously subject to updates and replacement of software units in the entire stack. As a consequence, a software system that is deemed secure and privacy-respecting today, may not live up to security and privacy guarantees tomorrow. Security and privacy goals must be considered from the foundation of software systems and must be assessed continuously through their entire life cycle, also after deployment and even after decommissioning. Such assessments can be expensive, time-demanding and unreliable unless they are supported by effective, automated analysis and verification tools based on theoretically well-founded techniques.

Short-Term Research

Software must be developed with proactive security- and privacy-by-design methodologies that consider security and privacy as part of the blueprint in all phases of its life cycle, from conception, design and realisation, to deployment, operation and decommissioning. This requires the development of modelling and programming languages supporting security- and privacy-by-design methodologies, with by-construction and static analysis guarantees [HN19], and of automated tools to verify, measure, assess and monitor security and privacy properties, risks and vulnerabilities along the entire life cycle of software. We plan to conduct research in three key areas of research, namely (i) logical foundations of privacy and security, (ii) protocol verification, and (iii) quantitative security.

Logical foundations of privacy and security. The introduction of regulations like GDPR is making companies more interested in ensuring that their IT systems live up to privacy expectations. The tension between privacy and other security goals like accountability and integrity is giving raise to new challenges, demand for novel approaches. In this area we plan to continue our investigations on logic-based approaches to formalise privacy and provenance [MV19, Llu19] to support well-founded mechanism to asses such properties and guarantee their preservation along the lifecycle of information. We will investigate a new logical based view of privacy goals of distributed systems that allows for more declarative reasoning about privacy than current specifications based on observational equivalence, and semantic foundations of security models and policies for high-level interaction paradigms beyond traditional message-passing and shared memory models, involving not only individual access or point-to-point transfer of resources but also coordinated resource access, and resource aggregation and dissemination. We will consider security aspects included in regulations such as GDPR and we will draw inspiration by challenges from the medical domain as identified in [D5.1].

Protocol verification. A particularly successful area witnessing the feasibility of the vision of formally secure-by-design software is protocol verification. Over the last years effective software tools like ProVerif [Bla01, B+16, Pro], Tamarin [BCDS17, Tam] and OFMC/AIF [BMV05, OFM] have been developed and used to formally prove or disprove the security properties of several cryptographic protocols. Within CS4E we will continue developing techniques for protocol verification, based on OFMC/AIF and theorem provers with the aim to cope, among others, with the curse of dimensionality, for instance by exploring compositional techniques. We will integrate our results on complementary verification methods for security [HM18] as well as our meta-results in compositionality [MSW+19] within logical frameworks such as Isabelle [Isa], allowing for comprehensive real-world models, secure-by-design implementations and a large coverage with automation.

Quantitative security. In many realistic scenarios it is not possible to entirely remove vulnerabilities related to security and privacy and certain vulnerabilities need to be assumed, as for example controlled leakages, provided they are properly measured and understood. Quantitative security is a promising area of research that deals with such situations and where techniques and tools have been developed for various purposes such as time-related information flows and quantitative risk modelling and analysis. Within CS4E we will further develop our approaches to modelling and analysis of risks and vulnerabilities based on probabilistic graphical models of security [AN17, ANP16], to include scalable verification techniques based on stochastic and statistical model checking with tools like BADGraph [BAD,tBL+20], and we will continue our studies of various aspects of security in the context of timed systems [VNNK19, VNN18, NNV17, VNN17], thereby paving the way for investigating how to balance potentially conflicting safety and security demands.

Long-Term Research

The dominating approaches to development are agile and prioritizing fast deployment over security and privacy guarantees. More research is needed to develop tools and techniques to support secure- and privacy-by-design techniques within agile approaches, so that competitiveness and fast deployment are not compromised by security and privacy requirements and so that changes in privacy and security requirements can be efficiently reassessed even after the system has been deployed.

Research efforts are needed to further develop and promote lightweight formal methods that can be gradually applied to increase the levels of assurances obtained. Methods must be developed to support a spectrum of guarantee levels, each providing greater assurance but also requiring more work, at a level more approachable than the common criteria [Com]. The enforcement must be gradual in order not to close the opportunity for SMEs to deliver the software systems used in EU, and appropriate tool support is need. Industrial compliance would be enhanced if GDPR were to be extended with requirements for using formal methods and tools.

2 Assessing Privacy Properties of Complex Systems

Scenario

Data is a treasure trove and having the right data in a right place can bring about enormous value while at the same time posing challenges to privacy. Examples of such tension between functionality and privacy can be found in the medical domain (see the discussion in Ch. 3.6 in Ferreira et al. [D4.1]) and, in particular, in systems like medical exchange platforms (Sforzin et al. [D5.1], Ch. 8). We keep building such complex systems to support the exchange of data between different parties, and to jointly perform data analyses, in order to find the best pharmaceuticals or to design the best medical services. These systems have to find the right balance and present the appropriate choices between the functionality offered to the users, and the privacy offered to the data subjects.

Research Challenge

Complex systems like the ones mentioned above release variously processed data to the involved parties, thereby potentially violating the privacy of data subjects. Methods for assessing privacy properties of complex systems are needed[1]. Currently, there are no good means for the data subjects to communicate, which pieces of their data they consider more or less sensitive, and the release of which pieces could be acceptable for them, perhaps as a trade-off of certain benefits to themselves. We also lack the analysis methods for these computational systems, in order to verify whether and to what extent they satisfy the policies of data subjects. We do have certain definitions for output privacy (in particular, differential privacy [DMNS06], but also the properties similar to k-anonymity [Swe02]) with better or worse compositional properties and thus amenability to automated analysis, but these properties likely do not match well with the statements that data subjects want to make about the use of their data [BMS13].

Short-Term Research

The challenge needs to address research developments in at least the following three areas: (i) privacy definitions for complex systems, (ii) privacy policy languages for complex systems, and (iii) privacy analysis for complex systems.

Privacy definitions for complex systems. We need framework of definitions that allow privacy authorities (including users) to specify their preferences and trade-offs in terms that they are familiar with, and that are significant for them. We plan to start from the derivative sensitivity framework described in [LPP18] and to expand it to support the needed notions. The extended framework shall contain references to different kinds of adversaries, including the Differential

[1] See, e.g, the following cases:
https://www.aki.ee/en/news/estonian- data- watchdog- worried- about- misuse- population- register
https://www.aki.ee/et/uudised/andmekaitse- inspektsioon- lopetas- tallinna- piletimuugisusteemi- jarelevalvemenetl 3
https://healthitsecurity.com/news/the- 10- biggest- healthcare- data- breaches- of- 2019- so- far

Privacy adversary that “knows everything except the data element it wants to guess”, but also weaker ones. Indeed, protection against such a strong adversary may be impossible or impractical, but we would still like to obtain statements about privacy-sensitive systems that are more informative than “does not provide protection against the Differential Privacy adversary”. In particular, we want to characterize the privacy loss through the common practice of the publication of aggregate values, informally argued as privacy-preserving, because each person’s private data will be hidden by other people’s data. In this way, the privacy definitions will also allow us to meaningfully talk about the properties of legacy systems, even those that have not been augmented with any Privacy Enhancing Technology (PET).

Privacy policy languages for complex systems. The data owners and subjects want to decide themselves, which parts of their data they consider more or less sensitive. A language of policies would allow these actors to specify that it is OK to leak certain pieces of their data; or to leak them with certain-sized noise or certain guessing probabilities. The policies refine, what is sensitive and what is not. Preliminary steps in this direction have been done (see e.g. the language for privacy sticky policies for data aggregations in the approach of [KL18] and the references therein) and could be used as inspiration.

Privacy analyses for complex systems. We want to significantly extend the existing analysers of the Pleak [TTY+19] framework, in particular increasing their compositionality and mutual composability. The structure of a privacy analysis should follow the structure of the system, as well as the structure of PETs applied on it. The handling of the latter should be a separate step of the analysis, as much as this is possible. The analyses also have to follow the privacy definitions, and the policies of users. The analyses should either provide worst-case guarantees for the systems, or state what kind of leaks will happen with the dataset that the system is actually run on.

Long-Term Research

In long term, we expect the analyses to become more holistic, considering at the same time the privacy, utility, and efficiency properties of the systems and their trade-offs. We expect there to be automated enhancements for complex systems, mostly concerning the placement of additional PETs, or specifying the details of PETs (e.g. the magnitude of the added noise [DMNS06]). The selection is made based on the results of privacy analyses, and has to consider the trade-offs between privacy gain, utility loss, efficiency loss (all these potentially for each stakeholder of the system), and invasiveness of modifications.

3 Privacy protection and User Empowerment in the Internet of Things

Scenario

Modern cities are facing increasing pressure to become more efficient from economical, societal and environmental standpoints. This has motivated a closer integration between different domains, namely, transportation, energy, management and services. For every stated domain, there exists a vast number of sensors spread within the cities, generating a significant volume of data. Several smart-cities projects have appeared in the past trying to explore this data to promote a more efficient and cost-effective management of the city. Nevertheless, most of these projects failed to succeed in attaining safer cities, as the underlying platforms were not designed by default to be secured and privacy friendly as they should be (see e.g. the requirement analysis in Chapters 3.7 of [D4.1] and Chapter 9 of [D5.1]).

Research Challenge

The ever-increasing volume of data produced by the Internet-of- Things (IoT) undermine some of the fundamental privacy principles, including informative self-determination, data minimization, consent and the rights to individual access [Web15]. In most countries, public and private entities should limit the collection of personal data to the bare minimum necessary towards achieving the intended goals. Furthermore, these entities should also delete the data that is no longer required for the purpose it was collected [WMAH15, PRW+15, PMB+16]. However, a large set of companies and entities are located in countries that do not legally enforce these obligations. This allows large sets of data to be collected and used without respect for those international privacy rights. Therefore it is mandatory to find mechanisms to ensure an adequate level of privacy protection and user empowerment, through new applications and services based on access to personal information, in order to solve the existing tension between legislation and technology [PRW+15, PMB+16].

Short-Term Research

We plan to address some of these challenge by focusing on a set of key research problems: (i) Identity management in the IoT, (ii) Authentication, Authorization and Access Control in the IoT, and (iii) Secure infrastructures for the IoT.

Identity management for the IoT. Given the proliferation of IoT devices and their interconnections, highly scalable identity management solutions/mechanisms are required to manage their identity [ZCDV17]. It should be noted that devices should not only be identified by their attributes, but also based on their contexts (e.g., their physical location, surrounding devices, etc.). Current related work does not bring novel solutions for these issues [HMW+16, SS15, ZCDV17]. The existing limitation of IoT (low power and processing), is well known, which makes PKI not the best approach [NSC+16]. We aim to manage the identity of the things without using PKI, while giving users the possibility to configure their own policies, preferences and terms of adherence, and do it in ways that can be automated for both users and the organizations they engage.

The platform must answer all the questions mentioned above, to be more suitable for low-resources devices.

Authentication, Authorization and Access Control for the IoT. It is becoming increasingly pervasive the requirement of authentication between devices as a way to develop trusted services. However, traditional authentication and authorization methods may not be applicable [ZCDV17]. For instance, authentication and authorization through cryptographically pre-shared keys is not practical or even achievable due to the rapidly growing number of devices, making key management a difficult task.

In this context we envision a scenario where devices belong to specific groups within the same region/area. In this way, local identity providers can be used for managing the identities of the devices, allowing in turn, the creation of distributed trust federations among them. Additionally, devices should not only be able to mutually identify and authenticate themselves among other devices, but should also be able to provide a proof-of-identity when interacting with external actors.

Regarding authorization and control access, heterogeneity and complexity of both IoT devices and networks renders traditional authorization methods impractical and unrealistic [LYZ+17]. For example, the rapidly growing number of devices makes key management a difficult task. Therefore a scalable solution must be attained. While some research has attempted to resolve this problem [SRGCP15], no common agreement is available and is still an open research area.

Secure Infrastructures for the IoT. Current computing infrastructures, such as cloud computing data-centers, need active protection against external threats [SK11]. For that end, network intrusion detection systems (NIDS) are normally deployed to filter outside attacks [MHL94]. However, current NIDS are limited by their scalability, which we aim to address by developing an elastic secure backend for offloading the task of analyzing signatures [PET]. Furthermore, the developing of such systems have to deal with the gap between theory and practice. Namely, the actual implementation of complex algorithms is non-trivial even for the most experienced developer. The use of fault-injection frameworks such as HERMES [MGN+13] is a way to alleviate this problem. We plan to conduct research on the above mentioned areas, and we plan to incorporate them in a middleware layer that should be implemented and evaluated, in the smart city platform of the Municipality of Porto.

Long-Term Research

Given the volume and rising tide of increasingly sophisticated digital attacks, current environments and platforms are unable to ensure data security, including data integrity and confidentiality [cyb]. This evidences that traditional approaches based on static or passive defense mechanisms are not enough, and that systems have to evolve and offer introspection guarantees for ensuring security even in the presence of intrusions.

Additionally, requirements for computing data containing sensitive information, renders the use of public cloud infrastructures impossible given the inherent trust models. These rely solely on the reputation of providers, and no guarantees of enforcement can be made about possible introspection on the workloads in these scenarios, either from internal actors or due to compromised infrastructure.

Techniques such as homomorphic encryption [G+09, BV14] and multi-party computation [Gol98, Can00] have shown potential for preserving strong levels of privacy when storing and processing data on untrusted parties. However, performance and scalability issues still hinder their feasibility in real-world deployments. This is especially true in contexts where large amounts of data are to be securely stored and processed.

Thus, we envision the need for active or dynamic defense mechanisms (e.g., intrusion tolerant solutions) allowing system to adapt and respond to attacks. However, the development and deployment of highly resilient and secure infrastructures carries massive computational and communicational overhead, as well as configuration and management burdens. Thus, more research efforts are needed in order to achieve the goals of secure cyber infrastructures.

4 Securing Unsafe and Legacy Software

Scenario

Despite a significant evolution in software engineering and programming languages during the last decades, many safety-critical domains still rely on legacy systems or even systems developed with unsafe programming techniques. A paradigmatic example of such domains is the maritime transport domain (see the discussion on security concerns in [D4.1, D5.1].). A very frequent form of software is memory unsafe systems, usually written in C/C++, where code can freely access memory at run-time. This is a benefit for performance and sometimes required for certain functionality, e.g., operating-system kernels need to read and write memory with no constraints. Nevertheless, unconstrained access of memory may have severe consequences during attacks if software contains memory-error vulnerabilities. Several such bugs can be abused and transformed to powerful exploits using inputs, which drive the vulnerable program to perform malicious actions (e.g., download malware) or to exfiltrate sensitive data by over-writing and over-reading its memory intentionally. For example, consider the popular WannaCry ransomware; a program that could randomly compromise vulnerable hosts, and propagate through the network [MP17]. The particular ranswomware did not leverage the human aspect for compromising hosts (i.e., a human that accidentally clicks on a malicious link), but a memory bug found in the Windows kernel, and, in particular, in the Server Message Block (SMB) –a protocol for mounting remote volumes of storage– implementation [MP17].

Research Challenge

Currently, there are different techniques for addressing memory unsafe programs. For instance, by identifying and eliminating memory errors before deploying software or by entirely writing all code using a safe programming language. Such solutions can be carried out during the early stages of the software life cycle. Consider, for example, that identifying and eliminating errors can be done using software testing and formal verification. However, such solutions are less adequate when dealing with already running software and even worse with legacy software where testing, verification and re-development may be difficult. There is an urgent need for Securing Unsafe and Legacy Software of this kind.

Short-Term Research

Within CS4EU we will address this challenge by developing research approaches for software hardening. We elaborate below on the mechanics of software hardening, as well as on the available options we have and we plan to investigate. Additionally, in our short-term research efforts, we will explore and collect the different properties and requirements of a broad set of software programs, since there is currently no universal method for hardening unsafe code.

Software hardening. Software hardening techniques provide an alternative solution to securing unsafe software [HGA+15, vGC+16, PCvdV+17, SKGA16]. We plan to develop techniques to transform software written in memory unsafe systems, in a way that existing vulnerabilities will be hard to be transformed to exploits. This, essentially, means that unknown vulnerabilities will be still in the software. However, exploiting those vulnerabilities will force the program to crash, rather than getting compromised.

In parallel, we will test currently enabled (but experimental) hardening techniques available in state-of-the-art compilers, such as Control-flow Integrity [claa] and Safe Stack [clab], as implemented in Clang.

Software hardening is considered ambitious, since: (a) despite the many available defenses in place software is still exploitable [SPWS13] (e.g., WannaCry [MP17] managed to automatically infect several hosts by leveraging a simple buffer-overflow bug), (b) generic software can be fairly diverse, which makes a global hardening solution unsuitable [SRC+12], (c) security defenses impose severe overheads [vdKAB+18], (d) security defenses can alter the functionality of the software [SPWS13], (e) legacy software may be hard to analyze or change [vGC+16, SKGA16].

In our efforts, within the context of CS4E, we will follow research directions that try to strike a balance between all aforementioned issues. For instance, we will focus on lightweight hardening, which can be applied on certain parts of software and for the most severe vulnerability classes.

Long-Term Research

In the long term, more systematic effort should focus on strengthening software by design. Hardening approaches should be included in state-of-the-art industrial compilers, completely unsafe systems should be reduced (in terms of code base) and more research should be carried out for standardizing these techniques.

5 Protecting Leaked Credentials

Scenario

Several applications, especially web apps, require an authentication component. For realizing this component, text-based passwords are still the dominant option. Beyond the many usability issues associated with handling several text-based passwords, security is also an important dimension. Through the years, a significant amount of on-line services has been compromised and their stored passwords have been leaked. Once the database is compromised, it takes little time for a program to crack the cryptographically hashed (weak) passwords, no matter the algorithm used. The need for protecting credentials once leaked is important for applications that handle sensitive personal data. An example is the privacy-preserving identity management scenario discussed in Deliverable 5.1 [D5.1].

Research Challenge

A significant amount of services has been a victim to a databreach attack. Such attacks have exposed billions of leaked credentials to the public - There is a very well-known public service that collects all leaked credentials and offers users to infer if they are included in the set of victims at https://haveibeenpwned.com.. It is important to stress that such incidents do not affect only small-budget Internet services, but even the very large vendors, which have large budgets for investing in their security. Such vendors include Sony [son], Twitter [twi], and LinkedIn [lin], to name a few.

Judging from those past incidents, it is now well established that protecting the credentials is fairly challenging. The reason, primarily, is that credentials are stored in a medium, which needs to be accessed by application code. Now, application code may have bugs. These vulnerabilities may not be strong enough for offering a complete compromise of the system, however, they may lead to leveraging of the application code for just exposing the passwords. Protecting all application code or separating the code for authentication that needs to access the user's credentials is hard.

Thus, the problem of password leaks has attracted significant attention from both academia and industry. In response to credential leaks, researchers have proposed cryptographic services for hardening all stored passwords [DA19, ECS+15, LESC17, SFSB16]. These services can be enabled for instance to web sites either locally or remotely. In the latter case, the web site leverages a third party to validate each password.

Moreover, these services perform several sessions of cryptographic hashing combined with message authentication codes, which involve additional cryptographic keys that the attacker will unlikely have during a password breach. Thus, the goal of these services is to force adversaries to use the cryptographic service while cracking the passwords. A different direction is to use several services to vet for each other during authentication [KAPK13], incurring reasonable overhead, or inject fake passwords in the database [JR13]. No matter the research direction, all of the aforementioned techniques essentially transform off-line password cracking to on-line. The latter is fairly easy to tackle, for instance, using CAPTCHAs or a limited amount of authentication attempts.

It is important to stress that, as mentioned above, all of these techniques assume that the service will be eventually compromised (due to application bugs) and the passwords will indeed get leaked. However, the leaked information will not be sufficient to reveal the actual credentials. Although these services incorporate elaborate cryptographic schemes for password hardening, it is unclear how easily typical web sites can utilize them without outsourcing the functionality to large providers.

However, despite the many technological investments and research, the protection of leaked credentials is still an open problem. There is a need to develop novel approaches to protecting leaked credentials.

Short-Term Research

Within CS4E we will address the challenge of protecting leaked credentials by developing approaches to build easy-to-enable password-hardening services.

Password-hardening services. In CS4E we will realize easy-to-enable password-hardening services and the respective APIs, which can be applied to commodity web applications, without the need of external entities. In particular, we will leverage modssl-hmac [DA19], which can be easily enabled in any application that runs through the Apache web server. Enabling modssl-hmac is as simple as installing a custom version of modssl and slightly modifying the authentication of the web app. This modification, for instance, in WordPress is less than 50 LoCs. A service like modssl-hmac can assist the web developer in incorporating strong authentication during the development of the application.

Long-Term Research

Someone could argue that credential leaks are possible, since currently authentication mechanisms rely heavily on storing credentials that can fully authenticate anyone that has access to them. In the past, there were protocols proposed that relied on storing part of the information needed (in the form of a cryptographic challenge) to the server [Wu00], nevertheless, their adoption was limited. Today, researchers continue work on realizing systems that divert from the typical storage of re-usable hashed passwords on the server [AMMM18, TPC+]. Unfortunately, such efforts need substantial changes in how authentication works and, in many cases, the interface exposed to the end user may change, imposing further usability challenges. However, we anticipate that the myriads of problems stemming form currently deployed authentication systems will drive research to new mechanisms, where credentials may have an entirely different form than the one we know today.

6 Secure Access Control in Heterogeneous Systems

Scenario

In complex systems where a large number of agents are accessing many different kinds of services provided by a multiplicity of software agents, it is critical to ensure that the logical and physical interactions between these types of agents do not affect negatively the security properties of the overall system. In particular, we consider heterogeneous systems where different types of network architectures and providers are connecting devices of different natures, ranging from standard IT equipment to special purpose embedded devices. Examples of such cases are easily found in industrial systems and critical infrastructures where physical and cyber activities are strictly intertwined. Such a scenario is also reflected in Smart Cities (see Chapter 3.7 of [D4.1]) where human agents perform heterogeneous tasks interacting with complex physical and cyber services provided as a combination of several distributed components.

Research Challenge

In this kind of scenario, the security of the system depends on the security of the single components and is affected by their relationships and combination and by the behavior of the agents accessing them.

One of the main aspects to consider when evaluating the security properties of a system is the correct implementation of access control policies: agents must be able to access only those resources that are explicitly allowed to. This kind of policies are usually enforced by some kind of access control mechanisms that must be correctly designed, implemented and configured. These control mechanisms are then deployed at the component level and at the overall system level.

The challenge is to achieve secure access control in complex heterogeneous systems. In the considered scenarios, the challenge stems from two main considerations: the first is that the heterogeneous nature of the involved components is reflected in a non uniform distribution of capabilities of their security mechanisms. Some elements, such as sensors, have low computational power and other limitations (e.g. low energy consumption constraint) that greatly limit the security mechanisms that can be deployed. The second consideration is that while a resource can behave correctly in isolation, the situation can dramatically change when the same resource is then combined with others. In practice, the challenge here is to be able to analyze and assess the security of the overall system taking into account all its limitations and characteristics including all the possible behaviors of the operating agents.

For example, we can consider a standard office network where we can assume the availability of a centralized authentication system that is able to correctly implement the desired access-control. However, in the general case of heterogeneous systems, most of the time this is not possible as the involved devices may not be able to support and implement the needed security mechanisms or they may implement proprietary ones that cannot be easily integrated.

Of course, the scalability of this kind of analysis is a critical challenge too, as the complexity of the considered system is continuously growing, in sheer number of components but also in their heterogeneity.

Short-Term Research

Within the CS4EU project, we will pursue three directions to address this challenge: (i) use formal verification approaches to secure access control, extending formal models to include other possible causes of security threats in the system, and developing an approach able to propose solutions to the identified problems.

Formal verification for secure access control. The main idea is to evaluate exhaustively all the possible behaviors and actions that the agents in the systems can perform accessing the various resources, and to compare the outcomes of such an evaluation with the correct behaviors as defined in high-level security policies. Discrepancies found in this phase point out flaws in the system, either in the design of the security policies or in the implementation and configuration of the deployed security mechanisms.

Such an exhaustive analysis approach is built on top of a formal model defined and used to describe the system, its components and their relationships. The formal model can then be leveraged using model checking (formal analysis) techniques to build the automaton for each agent (which is the formal representation of all the sequences of actions that he/she can perform). The obtained information is then used to verify the coherence with the security policies.

Some steps in this direction have been already taken and some preliminary results are already available. In particular, a model suitable for the description of both the system components and the security policies have been designed [BDSV15] and used [CDSV15].

Enriched formal models that cope further possible causes of security threats. In particular, in [CDSV17] vulnerabilities of software components have been considered. This is particularly relevant for the scenario considered (including smart cities) because of the large number of different software components and agents. In fact, vulnerabilities can allow some malicious agent to bypass the deployed security mechanisms causing cascading effects possibly leading to critical states. It is worth remarking that in the considered scenarios, cyber (software) vulnerabilities can also affect the physical world because of the complex relations between the services and devices operating in the cyber and physical layers.

Analysis and correction of security issues. The other direction concerns the usefulness of the obtained results from the point of view of an administrator that is willing to identify the problems but also to solve them. When discrepancies are found by the analysis, the complete results obtained are usually too complex to be manually analysed and it is difficult to extract a possible solution. To overcome this drawback, some automatic tool able to efficiently design and test the re-configuration of the system, would be very useful. A first approach in this direction has been made in [CDS+19].

Long-Term Research

One of the main problems that will become more urgent in the future, is the sheer number of objects involved in the considered complex systems. Thinking specifically to the Smart Cities examples, not only the number of agents and services involved will continuously increase but also the complexity of their relationships and dependencies in the composition of more and more complex services. In terms of security this will further expand the attack surface of the overall system. Moreover, the complexity of the composition and interactions between agents and resources will possibly hide new forms of cyber attacks and this must be addressed by general enough analysis approaches that, at the same time, ensure scalability and efficiency.

This will exacerbate the already identified problems of the interpretation of the analysis results and the process of finding a possible solution. In this matter, future research activities could focus on the automatic generation of system configurations that, moreover, can be prepared in advance in order to react quickly to adverse events. To support this approach it will also be necessary to develop and further extend the use of formal methods in the verification of the involved models as the effective usage of formal approaches is greatly affected by the increasing complexity of the analysed systems. Another future problem to be addressed is the "live" identification of system conditions and events that could lead to critical states. In the real world, events can happen in fast sequences that should be quickly analysed and addressed in real-time. This will require a balance between a deep enough analysis (to obtain correct and complete results) and a fast one.

7 Manageable and Understandable Security Engineering

Scenario

The shift from traditional computer systems towards the Internet of Things, i.e. devices connected via the Internet, wireless communication or other interfaces requires a reconsideration of secure and trusted systems engineering processes. Especially, the introduction of mobility and its leitmotiv “anytime, anywhere” reinforce this complexity due to the need of enabling various means of connectivity, such as Ethernet, Wi-Fi, 5G and so on. Security is concerned with ensuring a system is protected from accidental or deliberate external attack, some of which may target the system’s reliability. Modern software systems existing in critical domains consist of many distributed and interacting components that rely on extensive communication to achieve their intended functionality. Efforts to secure such systems include securing the underlying infrastructure, the information that they store, use and communicate. Scenarios from safety-critical domains like healthcare [D4.1, D5.1], where humans and IT systems, including wearables, where human life is at risk is a perfect example.

Research Challenge

Security experts, practitioners, and researchers from different international organizations, associations, and academia have agreed that for security, “it’s not just the code.” The most popular and well-known software security vulnerabilities are design issues. From the system developer’s perspective, security issues need to be identified early in the first development steps and at the highest levels, primarily at the architecture design stage, where their semantics are clear. They also need to be mapped to lower levels, where they are enforced by corresponding concrete mechanisms.

There is currently a lack of a methodological tool support for developing both the system architecture and security. This results in handling the architecture and security largely independently, causing in turn extensive rework and thus budget and schedule overruns. Therefore, the design of architecture and security cannot be considered in isolation, but they need to be studied in tandem as development of security and architecture. We aim at developing a design framework for handling the integration of security and architecture, which also semi-automatically supports their evaluation and their subsequent redesign. We provide evidence of its benefits through a high-level integration-based reuse of patterns and model libraries, as well as through applicability using various representative evaluation cases from our research project.

Achieving manageable and understandable security engineering is challenging due to a set of issues. (i) The need to exploit modeling languages and formal methods at a reasonable cost in an industrial context. Contrary to the other aspects, security engineering requires well defined and precise solutions to develop an accurate analysis, for evaluation and / or certification. In addition, the early stages are not suitable for beginners (architects with llittle experience in the engineering of secure systems), given the sensitive and delicate character of these systems. (ii) The need to ensure conformity, validation and certification. Safety-critical systems are usually accredited or certified. We must implement other types of software and means of generating validated artifacts, such as programming language code and certification artifacts, which are capable of producing a restrictive set of artifacts that comply with domain standards. In other words, can we certify an application by using security patterns? Can we show that a system built based on the use of security patterns is secure? Intuitively, we can show that the application includes a pattern able to stop each expected threat by a simple matching of threats to patterns. If we have a pattern for each threat, we can consider the system secure at the model level.

Short-Term Research

We will address these challenges to make the security engineering process manageable and understandable using novel methods and tools for engineering secure systems using patterns, models and formal reasoning that ensure that system security solutions are built by design. We use Model- Driven Engineering (MDE) [Sel03] abstraction mechanisms to define and handle software architecture maturity properties, threats and requirements through a metamodel that unifies those concepts. In the context of our work, formal methods [RG13] will be used for the precise specification of security and architecture. These techniques will be used to support validation of security and architecture properties during the pattern-based security engineering process, including properties of compositions of security properties. In particular, we introduce a new formal modeling paradigm for software system security engineering within a pattern-based approach as a foundation for novel security engineering practices. In particular we plan to advance the state of the art in security for systems engineering in three relevant areas: (i) security properties, (ii) threats and (iii) patterns. We employ the MDE and Domain Specific modeling [FR05] technologies and attempt to add more formality to improve parts of the system design.

Formal security requirements. The modeling of basic concepts related to security and software architecture requirements are established and well-known. For instance, frameworks for security properties [JR08, FGR10, HW18] have been used to express security requirements in terms of a set of desirable security properties (i.e., positive statements). Security mechanisms are then introduced according to expected security properties. The development of security property modeling methodologies has helped to inform the development of secure system designs and architectures. Moreover, existing security property classification references (e.g., CIA) are meant to assist in understanding the potential security requirements landscape of a system under development by enabling security analysts to categorize, sort, and consolidate security requirements for system assets. A rigorous treatment of security properties needs to be based on clear formal semantics that enable system developers to precisely specify security requirements and the appropriate design solutions to fulfill them. Formal frameworks for security properties are established, but are not integrated with system engineering processes. Hence, security property modeling and analysis has to be considered at some point in model-based system development processes.

Formal threat specification. The existence of security threats in the designs of safety-critical systems can significantly impact their safe and reliable operation. Detecting and stopping advanced and persistent security threats is one of the major challenges in computer-based systems. As such, many security research and development programs have sought to develop more rigorous and systematic methodologies for designing and evaluating software systems early in their development in an effort to built-in security. Challenges regarding ways to identify, analyze, and prepare for threats, mitigate vulnerabilities, and minimize impact and consequences need to be addressed. Threats need to be precisely specified before a tool can manipulate them, and though several approaches for threat specification have been proposed, they do not provide the scalability and flexibility required in practice. This work aims to address this challenge by developing an integrated approach for threat specification, detection, and treatment during the software architecture design time. In the context of our work, we propose to use formal methods for the precise specification and analysis of security architecture threats as properties of a modeled system.

Security patterns. A way to provide for the unification of security and other architectural aspects, in the presence of a myriad of implementation details, is to use abstraction. In particular, patterns provide a good means of abstraction. Patterns are encapsulated solutions to recurrent system problems. They define a vocabulary that concisely expresses requirements and solutions as well as provide a communication vocabulary for designers. The description of architectures and best practices (i.e., security solutions) using patterns makes them more understandable and usable, whilst providing guidelines for design and analysis. Further, architectural patterns were studied extensively for security and dependability in isolation in the context of the TERESA FP7 project and could be worth extending for healthcare with a focus on the interplay between privacy, security and the system architecture. We employed a model-driven engineering methodological approach associated with a pattern-based approach to support the development of secure software systems [HW18]. We will develop an extendible design language by providing semi-formal descriptions and formal semantics for modeling security solutions as patterns. The language must capture the core elements of the pattern to support its (a) precise specification, (b) appropriate selection and (c) seamless integration and use. The first aspect is related to pattern definition, whereas the second and third aspects are related to the problem definition (threats and desired security properties).

Model-based integration and validation of security patterns. The combination of semi-formal modelling and formal modelling to specify security patterns and to prove their targeted system security properties enabled the development of an accurate analysis [HGF16], for evaluation and/or certification. We will develop a process to support integration of security solutions as patterns in systems engineering that allows to transfer verified rules for patterns to system models and to validate correct (and secure) integration on the level of the model also during refinement. First, we build on the semantics of security by translating the pattern modeling language to existing formal languages. Then, we can show that a system satisfies some predefined security properties using the output of the formal specification and validation of patterns' solutions. It has to be ensured that the assumptions used for proving the correctness of the pattern are indeed satisfied by the particular environment of the application. This is the basis of the correct integration of patterns. The goal is to increase the confidence in the system and build highly secure software systems.

We will develop a process to support integration of security solutions as patterns in systems engineering that allows to transfer verified rules for patterns to system models and to validate correct (and secure) integration on the level of the model also during refinement.

Long-Term Research

There is a need to ensure continuous (safe) service, and minimal maintenance costs. In a context of fast changing cybersecurity threats, and ever emerging vulnerabilities, long-lived safety-critical systems require a high level of maintainability. Maintenance in secure conditions (e.g., security patch installation) should be possible without having to re-accredit or re-certify the system. Currently some security patches on systems are simply skipped because modifying the code, or updating an anti-virus database, would require running the accreditation or certification process anew. This situation is not sustainable. Architectural patterns may be suggested for the maintenance under security conditions (MSC) of safety-critical systems.

With the expected results, it may also be possible to further study the relationship between threats that have been detected in the software architecture model to determine whether the existence of one threat facilitates the existence of another. As we have seen in the past, adversaries often will exploit one vulnerability (which leads to a threat) in order to exploit another vulnerability to escalate their reach in the system. The formal setting of the specification and detection of threats approach will provide many essential mechanisms that will enable this kind of reasoning to aid in identifying the root causes leading to the existence or emergence of detected threats. Such a capability is expected to have a major impact on the time and resources required to treat treats in software systems.

8 Unreliable Risk Estimates

Scenario

In 2017, a collision between an American military vessel and a civilian container ship killed 10 sailors. The investigations following the event revealed that part of the accident was due to the pilots failure to understand how the touch screen-driven integrated bridge and navigation system worked. Further, the federal safety investigators found that when the system was in computer-assisted manual mode, watch standers behind other stations could unintentionally and unilaterally take over steering control [NoM]. The story illustrates that digitalization of safety critical systems comes with a risk. In this case, the accident was caused by a unintended failure, but we can very well imagine even more severe incidents when we take the increasing cyber threat picture into account. Chapter 3.5 of the Cybersec4Europe deliverable 4.1 [D4.1] dicusses in detail security concerns in the maritime domain.

Research Challenge

There is a large body of knowledge, data and statistics available for risk assessment when the events that are taken into account are caused by random failures. Hardware failures are statistically predictable, and methods exist to assess the reliability and security of software, under the assumption that the software will not be updated or changed. In contrast, cybersecurity risk estimates today tend to be based on gut feeling and best guesses; the main reasons being the lack of relevant publicly available historical data, the ever- increasing threat picture, the constant disclosure of new vulnerabilities and the subsequent roll-out of patches to address them, and the difficulties to foresee security incidents that have simply never occurred before. Improved justification and traceability of cybersecurity risk estimates can be achieved through data-driven decisions. This is, however, not straight-forward to achieve. With evolving technology and constantly emerging attack methods (and motivations), basing security decisions on past incidents is typically referred to as driving by looking in the rear-view mirror and cannot be considered reliable.

To provide better and more accurate estimates of cybersecurity risks, we will need to apply data-driven models that are looking into the future, rather than providing a historical picture. More specifically, we need to understand what types of empirical data that can be used as relevant input to cyber risk models. We also need to survey the possible sources of such data, and to investigate how the data can be aggregated and combined to form indicators of cyber security risks. Such knowledge and such models will be a fundamental part of assessing cybersecurity risks for evolving systems.Typical sources of the empirical data include: logs, statistics, measurements, expert judgments and thought experiments. They are often combined in order to instantiate the risk indicators and obtain the estimates.

Traceability of the cybersecurity risk estimates to the sources of the empirical data, also helps the decision maker to assess the validity of the risk model. For this, a notion of the quality of the data sources and their significance for the risk estimates, would be needed. Typically, logs, statistics, measurements, expert judgments and thought experiments may, to varying degree, be combined in order to instantiate the risk indicators and obtain the estimates.

The risk models and the estimates annotated to them need to be well documented, justified and modifiable. It is therefore essential to keep track of the relationship between the cybersecurity risk estimates and the sources of the empirical data that the estimates are (fully or partially) derived from. This relationship is what we refer to as traceability between the cybersecurity risk models and the empirical data. In the context of usage of the risk models, such a traceability is expected to help the decision maker to assess the validity of the risk model. The validity assessment will be based on several factors, such as: representativeness of the data sources, quality of the data, significance of the indicators being instantiated by using the data, as well as the way the indicators are being interpreted and presented.

However, regardless of their source and quality, the empirical data used to derive risk indicator values will almost always be imperfect and imprecise. In order to facilitate comprehensibility of the risk models and make their validity assessable by a human user, we also need proper means for expressing the uncertainty of the risk estimates. There are many ways of expressing the risk estimates, and the literature distinguishes generally between the possibilistic and probabilistic approaches [BPZ08, Par96, FKH+07].

Another distinction is between the two types of uncertainty, namely epistemic (due to lack of of knowledge or information about the system) and aleatory (due to inherent randomness of the system or variability of the usage profile) [DKD09]. The aleatory uncertainty is irreducible even by additional measurements. Aleatory uncertainty is typically represented by continuous probability distributions and forecasting is based on stochastic models. Epistemic uncertainty, on the other hand, is reducible, non-stochastic and of discrete nature. The epistemic uncertainty is therefore best suited for possibilistic uncertainty representations. For a detailed classification of the types and sources of imperfect information, along with a survey of methods for representing and reasoning with the imperfect information, see Parsons [Par96].

Being of a discrete nature, the epistemic uncertainty should be handled by a purely possibilistic representations, that is, the representations where pieces of information take the form of fuzzy sets [Zad78] of possible values. The merit of this framework lies in its simplicity, which enables incomplete probabilistic information on the real line to be encoded in the form of fuzzy intervals [DP87]. Selection of the appropriate approach for uncertainty handling implies finding the right balance between the practical applicability on the one hand and the functional properties of the approach on the other hand [OKS12].

In the case of the cybersecurity risk models, we argue that the epistemic uncertainty is the most relevant and the dominating one. Cybersecurity risk models are namely characterized by rather discrete and non-stochastic changes. In majority of the cybersecurity risk models, aleatory uncertainty is negligible in terms of magnitude and impact, while the epistemic one is crucial. It is therefore the epistemic uncertainty we focus on when dealing with the parameters on the cybersecurity risk models. An additional reason is the higher comprehensibility of the possibilistic uncertainty expressions for the human users.

Short-Term Research

To address the above described challenge we will investigate research approaches in the area of requirements for data-based risk estimates.

Requirements for data-based risk estimates. In contrast to historical data and guesswork, Anderson et al. [ABCM08] suggest to use forward-looking indicators as an alternative source of decision data, and in our work we are augmenting threat models with such indicators in order to anticipate potential crises. An indicator can for instance be observations of ongoing mechanisms and trends within the cybercrime markets as suggested by Pfleeger and Caputo [PC12]. We will, however, need to formalize this into quantifiable values for use within threat models. Our initial approach will be based on extrinsic economic models where cybercriminal must weigh the benefits and costs to decide whether to commit a crime [CD95, Ksh06].

Our research will, in addition to reviewing the relevant data sources and their applicability for evaluation of the cybersecurity risk indicators, also propose two distinct sets of requirements, namely (1) Requirements for efficiently selecting and combining the adequate data sources as well as assessing their reliability, in the context of instantiation of cybersecurity risk indicators., and (2) Requirements to practically useful approach(es) for expressing uncertainty of the risk estimates.

Long-Term Research

Security risk indicators can only be as as good as their validity and reliability. Particularly indicator definitions, usage of the indicators, as well as the quality of indicator data sources are subject to validity threats. We therefore need to provide explicit and structured guidelines for assessing the validity and reliability of the indicators. There should also be established guidelines for improving the levels of validity and reliability of the indicators in general and their data sources in particular. The guidelines should comprise methods of maintaining, removing and adding indicators in the risk models, as well as ways of interpreting the indicator values. The guidelines should be targeted towards pre-defined stakeholder groups, i.e. the major roles being involved in developing and using the evidence-based security risk models.

9 Automated and Verified Network Security Configuration in Highly Dynamic Environments

Scenario

A European service provider company operating in the field of smart cities, stimulated by the requirements for higher dynamicity coming from its customers, has decided to introduce automation of network management, based on the emerging network softwarization paradigm. Two main innovative technologies are introduced. Network Functions Virtualization (NFV) [MSG+16], which allows the company to substitute hardware components with software modules that can be deployed on general-purpose servers with agility, and Software-Defined Networking (SDN) [KRV+15], which allows the company to control and dynamically adapt its network architecture by means of software. Even though these technologies already provide ways to ease the work of the company's network administrator, all the tools that he or she exploits in this virtualized environment – i.e., NFV and cloud orchestrators, such as Open Baton and Kubernetes – mainly target general network management, without providing enough automation of network security management. As a consequence, the network administrator of this company must manually create the virtual services and configure each network security function. This manual approach is, however, leading to a number of drawbacks. Firstly, the time required for a manual configuration is very high: the administrator has to carefully analyze the security requirements of the service users and then manually access to each function to install the proper configuration. Secondly, human operations are difficult and error-prone: in a distributed network architecture, the introduction of anomalies in the configuration of the functions is difficult to avoid, opening the path to several cybersecurity attacks. Because of these reasons, the service provider network has already been victim of a number of attacks; this should not be surprising, because in the world misconfiguration has actually become the third most frequent cause of breaches for attacks, with an increasing trend from the previous years [Ver19]. Similar scenarios may occur in smart city environments like the ones described in Chapter 9 of [D4.1], where solutions that make it easy to protect and isolate parts from vulnerabilities are envisaged.

Research Challenge

In highly dynamic networked virtualized systems, such as the ones described in the scenario, the traditional manual approach to configuration is no longer feasible because new solutions must be found very quickly and, at the same time, configuration errors cannot be tolerated because of the demand for high security assurance. Recent studies show, in fact, that automation itself can improve cyber resilience – i.e., the ability to prevent, detect, contain and respond to a cyberattack – [Pon19]. This would be enabled by a policy-based approach [WSS+01]; in particular, the most interesting operation is policy refinement, through which the allocation and configuration of each function is automatically computed to fulfill a set of specified policies. In order to get high security assurance levels, a fully automated solution could also exploit automated formal methods, so as to finally generate configurations that are also formally verified.

Recently, some research has been carried out about automation of network security management. In this context, two main areas have been investigated in literature: the first one is automatic security service composition (e.g. [HLL18, RA17]), while the second one is automatic configuration of security functions (e.g. [BMNW04, ABR+14]). These two areas are complementary: after the automatic creation of the security service structure (security function composition), then the security configuration of each component should be computed. In both cases, however, the state-of-the art is still far from a fully automated solution that considers all the main network security functions (firewalls, VPNs, IDSs, etc.) and that performs policy refinement in a formally correct way on real-size networks in reasonable time.

The main challenge is that providing automatically a formally verified solution is very complex, especially when the complexity of the network increases. Of course, this complexity contrasts with the need for fast operation.

Short-Term Research

The formal verification problem, i.e. checking the correctness of a given security configuration, which is less difficult than the synthesis problem addressed by this challenge, can be solved with reasonable computation resources, even for large networks [SVJ+15, SPNR16]. The expected trend of research for the next few years is to make also the automated search of a solution feasible for dynamic virtualized networks, at least for the most common security functions. Within CS4E, we are committed to work on this challenge with research in two directions, namely (i) automation of network security functions and (ii) performance and scalability of such automations.

Automated network security functions. First, we plan to address the most common security functions, i.e. packet-filtering firewalls, for which we plan to overcome some of the current limitations of the state-of-the-art. Some first steps have already been done with the definition of a framework (VEREFOO) for automatic allocation and configuration of packet-filtering firewalls [VER, BMS+19]. Research is also moving on to extend the automated methodologies to other network security functions. This would firstly involve other filtering functions such as web application firewalls, deep packet inspectors or anti-spam filters, so that a full automation of filtering-based security can be achieved. Then, the immediate next step would be to address other security function types, starting from intrusion detection systems, able to detect incoming attacks that are not beforehand blocked by the filtering functions. In this context, the goal would be to perform the policy refinement of an all-round security architecture, facing the increasing heterogeneity of security functions.

Performance and scalability of automated network security. Another research direction that is being pursued is to improve the performance and scalability of the automated approaches, so as to be able to apply the technique to larger networks in reasonable time. This represents a critical requirement for introducing automation in network security, because whenever a new policy is added or modified, the reaction time should be limited. A possible trend to make this possible is to develop methodologies that, given a set of policies, can identify the components of the security architecture that are affected by the upcoming changes and their dependencies with the others. Thus, refinement would be limited to a subset of the functions, speeding up the refinement process. New challenges can be faced in this context: for instance, some strategies could be designed to minimize the number of changes in the existing configuration or the time required to change all the function configurations. In particular, the automatic computation of the function configurations is not a fast operation, because it is typically performed from scratch without considering a partial existing configuration. Another possible direction is to invesigate heuristic approaches or machine intelligence approaches, combined with formal verification ones, with the idea of de-coupling the search for a solution, which could be done with the former techniques, from the formal correctness checking, which could be done by network configuration verification techniques.

Long-Term Research

The improvements on performance and scalability that are possible in the short term will not probably lead to techniques capable to deal with very large networks, such as the ones that can be created today in large data centers. In the long term, we expect a continuation of the research along this line, in order to finally achieve this goal. When considering large cloud-based networks, another aspect which research will need to address in order to achieve full network security autonomy is the automatic elicitation of security policies. Currently, an automatic process still needs interaction with a human being from which it receives the security policies that must be fulfilled by the computed configurations. Instead, in the future, an autonomic process could extract the information needed for policy refinement from the network itself, thus closing an action-reaction loop that would not involve external interventions anymore. This would require the definition of intrusion prevention methodologies that would be able to perform the so-called policy discovery – i.e., extraction of policies from network monitoring –. New algorithms based on machine learning and artificial intelligence could be defined to perform the autonomic reconfiguration of the security service whenever the statistics computed from the extracted information would characterize an ongoing cyber-attack. Alongside with this achievement, a fully autonomic platform should be also capable of keeping safe all the service functionalities even in short periods where some security defenses have been temporarily compromised, until a full reconfiguration does not confine the danger.

10 Scalable and Private Industrial Blockchain

Scenario

Supply-chains are complex systems moving products or services from suppliers to customers. Supply-chain processes unfold over a multitude of stages and geographical locations, making it very hard to trace particular events and investigate incidents. It also makes it more difficult to track the ownership of goods and inventory at each step. Furthermore, transactions between these companies usually involve manual paper transfer of records (orders, invoices, etc.), a costly bureaucracy subject to human errors, losses, damages, thefts, and frauds. This inherent complexity only leads to economical losses, inefficiencies, and delays that will upset both a company's health and its customers' satisfaction. Customers have no reliable to way to verify and validate the value of the goods that they purchase, because of a lack of transparency and prices that do not reflect the true costs of production. In some extreme cases, there might be serious legal consequences. In a hard to manage supply-chain it is hard to detect illicit activities such as counterfeiting, or forced labor in factories.

Research Challenge

Existing blockchain deployments have shortcomings that researchers must address [LSF+17]For example, supply chains comprise many organizations willing to share information with a restricted number of partners. However, in existing blockchain architectures, transactions and their information are public. Encrypting transactions is insufficient because it still allows everyone to know when an exchange occurred.

Another key problem issue is scalability. Permissionless blockchains' (e.g., Bitcoin) scalability is excellent, but they sacrifice the network's throughput (e.g., Bitcoin achieves 7 transactions per second). On the other hand, permissioned blockchains can reach a higher throughput at the expense of scalability. The challenge here is to design an architecture able to scale to thousands of nodes without sacrificing throughput.

Finally, supply chains comprise private organizations and service providers that want to oversee their business networks and be able to grant or deny access to their services.

Approaches to achieve scalable and private industrial blockchains need to be investigated.

Short-Term Research

Within CS4EU we plan to address those challenges with research activities in three areas: (i) scalable secure and practical consensus layers, (ii) smart contract security, and (iii) efficient privacy-preserving blockchain protocols.

Scalable secure and practical consensus layers. Existing secure consensus layers for permissioned blockchains are not production ready [LSF+17]. Namely, their $O(n^2)$ performance makes them unsuitable to scale to modern networks comprising thousands of nodes. Additionally, their code is complex and deployment unfriendly, because of the algorithm's implementation many components, such as the main protocol, the fall-back protocol, the transition protocol, the topology change mechanism, etc. More research is needed to develop secure and practical consensus layers able to scale.

Smart contract security. Smart contracts are an important part of permissioned blockchains, because they allow organization to code their business policies and run them on the blockchain. For this reason, smart contracts are targets of cyber attacks as recent research has shown [RLK+19]. However, existing security solutions for smart contracts are unable to detect all attacks. Moreover, it is impossible to update the smart contracts' code once they are deployed on the blockchain. The direct consequence is that we have best practices that can find attacks but cannot patch them, thus leaving smart contracts vulnerable. The project short term goals should therefore include smart contract security.

Efficient privacy-preserving blockchain protocols. Third parties connecting to a blockchain platform need to preserve their privacy. Existing solution focus on enforcing privacy via cryptographic protocols, such as multi-party computation or homomorphic encryption. However, cryptography is ill suited because of the computational overhead, central authorities, and limited expressiveness that its protocols come with. Therefore, short term research should also focus on privacy-preserving, efficient protocols that allow any number of parties to interact with the blockchain.

Long-Term Research

The research goals listed above are fundamental building blocks to a secure blockchain system. On the long run, the goal is to combine those building blocks in a unique blockchain solution ready to tackle industrial challenges. We will also focus on how to make the blockchain compliant with important EU regulations. For example, the immutability of the blockchain's ledger does not sit well with the EU General Data Protection Regulation (GDPR). The blockchain is able to securely handle users' data, but it needs to do it according to EU regulations, or industries and citizens will not adopt it. A possible solution could be storing only *metadata* on the blockchain, but we will investigate more to find the best possible solution(s).

11 Scaling TEEs for Cloud Applications

Scenario

Smart cities are expected to generate a vast amount of data from a multitude of heterogeneous sensors [D5.1]. Creating value out of such data require distributed algorithms and considerable computing resources. If computing resources are not available “in-house”, the cloud is an effective alternative to process large amounts of data with a small investment. However, data generated by sensors in smart cities is likely to carry sensitive information that should not be disclosed to a third-party data processor such as a cloud provider.

Research Challenge

Trusted Execution Environments (TEE) allow running private computations in untrusted environments. As such, they represent a promising solution to the problem of cloud-based data processing in scenarios where data to be processed must be kept hidden from the host. Nevertheless, current TEEs are not compatible with cloud deployments where elasticity and resources consolidations are key to the cloud provider business [GHX+17]. In particular, applications running in the cloud are dynamically assigned resources depending on the current load. Further, the cloud minimizes the number of used resources by co-locating different application on the same hosts. Thus, running an application in the cloud requires that the cloud provider be able to add or remove instances of an application depending on the current load, and to migrate instances of an application across machines. Current TEEs simply do not allow such operations.

Short-Term Research

In the context of this CS4EU we plan to design protocols that enable the use of TEEs in cloud-based applications.

Protocols for TEEs in cloud-based applications. In particular, we plan to focus on scalability and elasticity. The current deployment model of modern TEE requires the application owner to carry out the deployment and initialization of its code. In particular, application code is deployed “secret-less” in a TEE whereas secret material is provisioned by the application owner after the latter has established a secure channel with the TEE-deployed application. Such a deployment paradigm is in sharp contrast with usual cloud operations where the cloud provider must be able to add and remove application instances on the fly. Throughout the project we will design protocols to enable a cloud provider to perform elastic deployment of TEE applications without sacrificing their security. The goal is to design a lightweight mechanism that provides application owners with the current security provisions of modern TEE, while enabling cloud providers to keep operational costs low [SKLF19].

Long-Term Research

Short-term research must cope with the design of available TEEs and provide ad-hoc solutions to its shortcomings. Long-term research should address the weak points of current TEE “by design”. In particular, during the long term we plan to design novel TEEs that keep the expected property of security and privacy while, at the same time, offer enhanced flexibility to be used in different scenarios, from stand-alone client machines to massive cloud deployments. Cache reservation mechanisms will be explored as well as hardware-software co-design, with the goal of mitigating side-channels while minimizing the trusted computing base. Alongside “clean-slate” TEE desing, we will also explore minimal set of changes required in available TEEs in order to minimize vulnerabilities. This activity will provide guidelines to manufacturers to design future revisions of their TEEs.

12 References

[ABCM08] Ross Anderson, Rainer Bohme, Richard Clayton, and Tyler Moore. Security economics and the internal market. Study commissioned by ENISA, 2008.

[ABR$^+$ 14] Pedro Adao, Claudio Bozzato,G. Dei Rossi, Riccardo Focardi, and Flaminia L. Luccio. Mignis: A semantic based tool for firewall configuration. In IEEE 27th Computer Security Foundations Symposium, CSF 2014, Vienna, Austria, 19-22 July, 2014, pages 351–365, 2014.

[AMMM18] Shashank Agrawal, Peihan Miao, Payman Mohassel, and Pratyay Mukherjee. Pasta: Password-based threshold authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, pages 2042–2059, New York, NY, USA, 2018. ACM.

[AN17] Zaruhi Aslanyan and Flemming Nielson. Model checking exact cost for attack scenarios. In Matteo Maffei and Mark Ryan, editors, Principles of Security and Trust - 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, volume 10204 of Lecture Notes in Computer Science, pages 210–231. Springer, 2017.

[ANP16] Zaruhi Aslanyan, Flemming Nielson, and David Parker. Quantitative verification and synthesis of attack-defence scenarios. In IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016, pages 105–119. IEEE Computer Society, 2016.

[B$^+$ 16] Bruno Blanchet et al. Modeling and verifying security protocols with the applied pi calculus and proverif. Foundations and Trends in Privacy and Security, 1(1-2):1–135, 2016.

[BAD] BADGraph. Available at https://github.com/BADGraph/ BADGraph/wiki.

[BCDS17] David A. Basin, Cas Cremers, Jannik Dreier, and Ralf Sasse. Symbolically analyzing security protocols using tamarin. SIGLOG News, 4(4):19–30, 2017.

[BDSV15] Ivan Cibrario Bertolotti, Luca Durante, Lucia Seno, and Adriano Valenzano. A twofold model for the analysis of access control policies in industrial networked systems. Computer Standards & Interfaces, 42:171 – 181, 2015.

[Bla01] Bruno Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In 14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11-13 June 2001, Cape Breton, Nova Scotia, Canada, pages 82–96. IEEE Computer Society, 2001.

[BMNW04] Yair Bartal, Alain Mayer, Kobbi Nissim, and Avishai Wool. Firmato: A novel firewall management toolkit. ACM Trans. Comput. Syst., 22(4):381–420, November 2004.

[BMS13] Jane Bambauer, Krishnamurty Muralidhar, and Rathindra Sarathy. Fool's gold: an illustrated critique of differential privacy. Vanderbilt Journal of Entertainment & Technology Law, 16:701, 2013.

[BMS$^+$ 19] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov. Towards a fully automated and optimized network security functions orchestration. In 2019 4th International Conference on Computing, Communications and Security (ICCCS), pages 1–7, Oct 2019.

[BMV05] David A. Basin, Sebastian Mödersheim, and Luca Viganò. OFMC: A symbolic model checker for security protocols. Int. J. Inf. Sec., 4(3):181–208, 2005.

[Bow] BowTiePlus. Available at https://github.com/KDPRO-SINTEF/BowtieTool

[BPZ08] P. Baraldi, I. C. Popescu, and E. Zio. Predicting the time to failure of a randomly degrading component by a hybrid monte carlo and possibilistic method. In 2008 International Conference on Prognostics and Health Management, pages 1–8, Oct 2008.

[BSA+19] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Sok: Consensus in the age of blockchains. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, October 21-23, 2019, pages 183–198, 2019.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. SIAM Journal on Computing, 43(2):831–871, 2014.

[Can00] Ran Canetti. Security and composition of multiparty cryptographic protocols. Journal of CRYPTOLOGY, 13(1):143–202, 2000.

[CD95] JR Clark and William L Davis. A human capital perspective on criminal careers. Journal of Applied Business Research (JABR), 11(3):58–64, 1995.

[CDL19]. R. A., Chivers, H. Danezis, G. Lupu, E., and A. Martin, “Chapter 16: Secure Software Lifecycle. The cyber security body of knowledge,” 2019, version 1.0. [Online]. Available: https://www.cybok.org/

[CDS$^+$ 19] Manuel Cheminod, Luca Durante, Lucia Seno, Fulvio Valenza, and Adriano Valenzano. A comprehensive approach to the automatic refinement and verification of access control policies. Computers & Security, 80:186 – 199, 2019.

[CDSV15] M. Cheminod, L. Durante, L. Seno, and A. Valenzano. Semiautomated verification of access control implementation in industrial networked systems. IEEE Transactions on Industrial Informatics, 11(6):1388–1399, Dec 2015.

[CDSV17] Manuel Cheminod, Luca Durante, Lucia Seno, and Adriano Valenzano. Detection of attacks based on known vulnerabilities in industrial networked systems. Journal of Information Security and Applications, 34:153 – 165, 2017.

[claa] Clang - control flow integrity. https://clang.llvm.org/docs/ControlFlowIntegrity.html, last accessed in November 2019.

[clab] Clang - safestack. https://clang.llvm.org/docs/SafeStack. html, last accessed in November 2019.

[Com] Common criteria for information technology security evaluation. Available at http://www.commoncriteriaportal.org.

[COR] CORAS. Available at https://www.uio.no/studier/emner/matnat/ifi/INF5150/h06/undervisningsmateriale/060930.CORAS-handbook-v1.0.pdf

[Cou16] Council of European Union. Council regulation (EU) no 2016/679, 2016. https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1564672107598&uri=CELEX:32016R0679.

[cyb] 2018 cyber incident & breach trends report — internet society. https://www.internetsociety.org/resources/ota/ 2019/2018-cyber-incident-breach-trends-report/. Accessed: 2019-12-4.

. Afonso Ferreira and other authors. D4.1 - Requirements Analysis from Vertical Stakeholders. Technical report, CyberSec4Europe (www.cybersec4europe.com), 2019. Available at https://cybersec4europe.eu/wp-content/uploads/2019/ 11/D4.1-Final-WithAnnex.pdf.

[D5.1] Alessandro Sforzin and other authors. D5.1 - Requirements Analysis of Demonstration Cases - Phase 1. Technical report, CyberSec4Europe (www.cybersec4europe.com), 2019. Available at https://cybersec4europe.eu/wp-content/uploads/2019/11/ D5.1-Requirements-Analysis-of-Demonstration-Cases.pdf.

[DA19] Constantinos Diomedous and Elias Athanasopoulos. Practical password hardening based on tls. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pages 441–460. Springer, 2019.

[DKD09 Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? Structural safety, 31(2):105–112, 2009.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings, volume 3876 of Lecture Notes in Computer Science, pages 265–284. Springer, 2006.

[DP87] Didier Dubois and Henri Prade. The mean value of a fuzzy number. Fuzzy sets and systems, 24(3):279–300, 1987.

[ECS+ 15] Adam Everspaugh, Rahul Chaterjee, Samuel Scott, Ari Juels, and Thomas Ristenpart. The pythia PRF service. In 24th USENIX Security Symposium (USENIX Security 15), pages 547–562, Washington, D.C., 2015. USENIX Association.

[FGR10] A. Fuchs, S. Gürgens, and C. Rudolph. A Formal Notion of Trust – Enabling Reasoning about Security Properties. In Preceedings of Fourth IFIP WG 11.1 International Conference on Trust Management, volume 321, pages 200–215. Springer, 2010.

[FKH+ 07] Scott Ferson, Vladik Kreinovich, Janos Ha jagos, William Oberkampf, and Lev Ginzburg. Experimental uncertainty estimation and statistics for data having interval uncertainty, 2007.

[FR05] R. B. France and B. Rumpe. Domain specific modeling. Software and System Modeling, 4(1):1–3, 2005.

[G+ 09] Craig Gentry et al. Fully homomorphic encryption using ideal lattices. In Stoc, volume 9, pages 169–178, 2009.

[GHX+ 17] Jinyu Gu, Zhichao Hua, Yubin Xia, Haibo Chen, Binyu Zang, Haibing Guan, and Jinming Li. Secure live migration of SGX enclaves on untrusted cloud. In 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017, Denver, CO, USA, June 26-29, 2017, pages 225–236. IEEE Computer Society, 2017.

[Gol98] Oded Goldreich. Secure multi-party computation. Manuscript. Preliminary version, 78, 1998.

[HGA+ 15] Istvan Haller, Enes Göktas, Elias Athanasopoulos, Georgios Portokalidis, and Herbert Bos. Shrinkwrap: Vtable protection without loose ends. In ACSAC, pages 341–350. ACM, 2015.

[HGF16] B. Hamid, S. Gürgens, and A. Fuchs. Security patterns modeling and formalization for pattern-based development of secure soft- ware systems. Innovations in Systems and Software Engineering, Springer, 12(2):109–140, 2016.

[HLL18] Zheng Hao, Zhaowen Lin, and Ran Li. A sdn/nfv security protection architecture with a function composition algorithm based on trie. In Proceedings of the 2Nd International Conference on Computer Science and Application Engineering, CSAE ’18, pages 176:1–176:8, New York, NY, USA, 2018. ACM.

[HM18] Andreas Hess and Sebastian Mödersheim. A typing result for stateful protocols. In 31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018, pages 374–388. IEEE Computer Society, 2018.

[HMW+ 16] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, and B. Gabrys. The security challenges in the iot enabled cyber- physical systems and opportunities for evolutionary computing other computational intelligence. In 2016 IEEE Congress on Evolutionary Computation (CEC), pages 1015–1021, July 2016.

[HN19] Michael Huth, Flemming Nielson. Static Analysis for Proactive Security. Computing and Software Science 2019: 374-392.

[HW18] Brahim Hamid and Donatus Weber. Engineering secure systems: Models, patterns and empirical validation. Computers & Security, 77:315–348, 2018.

[Isa] Isabelle. Available at https://isabelle.in.tum.de/.

[JR08] J. Jürjens and R. Rumm. Model-based Security Analysis of the German Health Card Architecture. Methods of Information in Medicine, 47(5):409–416, 2008.

[JR13] Ari Juels and Ronald L. Rivest. Honeywords: Making password- cracking detectable, 2013.

[KAPK13] Georgios Kontaxis, Elias Athanasopoulos, Georgios Portokalidis, and Angelos D. Keromytis. Sauth: Protecting user accounts from password database leaks. In Proceedings of the 2013 ACM SIGSAC Conference on Computer &#38; Communications Security, CCS ’13, pages 187–198, New York, NY, USA, 2013. ACM.

[KL18] Linas Kaminskas and Alberto Lluch-Lafuente. Aggregation policies for tuple spaces. In Giovanna Di Marzo Serugendo and Michele Loreti, editors, Coordination Models and Languages - 20th IFIP WG 6.1 International Conference, COORDINATION 2018, Held as Part of the 13th International Federated Conference on Distributed Computing Techniques, DisCoTec 2018, Madrid, Spain, June 18-21, 2018. Proceedings, volume 10852 of Lecture Notes in Computer Science, pages 181–199. Springer, 2018.

[KRV+ 15] Diego Kreutz, Fernando M. V. Ramos, Paulo Jorge Esteves Ver´ıssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1):14–76, 2015.

[Ksh06] Nir Kshetri. The simple economics of cybercrimes. IEEE Security & Privacy, 4(1):33–39, 2006.

[LESC17] RussellW.F.Lai,ChristophEgger,DominiqueSchr¨oder,and Sherman S. M. Chow. Phoenix: Rebirth of a cryptographic password-hardening service. In 26th USENIX Security Symposium (USENIX Security 17), pages 899–916, Vancouver, BC, 2017. USENIX Association.

[LSF+17] Wenting Li, Alessandro Sforzin, Sergey Fedorov, Ghassan O. Karame. Towards scalable and private industrial blockchains. In Proceedings of the 2017 ACM Workshop on Blockchain, Cryptocurrencies and Contracts. April 2017, pages 9-14, 2017

[lin] Hacker Posts 6.4 Million LinkedIn Passwords. http://www.technewsdaily.com/7839-linked-passwords-hack.html.

[Llu19] Alberto Lluch-Lafuente. A framework for provenance-preserving history distribution and incremental reduction. In Michele Boreale, Flavio Corradini, Michele Loreti, and Rosario Pugliese, editors, Models, Languages, and Tools for Concurrent and Distributed Programming - Essays Dedicated to Rocco De Nicola on the Occasion of His 65th Birthday, volume 11665 of Lecture Notes in Computer Science, pages 471–486. Springer, 2019.

[LPP18] Peeter Laud, Alisa Pankova, and Martin Pettai. Achieving differential privacy using methods from calculus. CoRR, abs/1811.06343, 2018. To appear at PET Symposium 2020.

[LYZ^{+} 17] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal, 4(5):1125–1142, 2017.

[MGN^{+} 13] Rolando Martins, Rajeev Gandhi, Priya Narasimhan, Soila Pertet, Antonio Casimiro, Diego Kreutz, and Paulo Verıssimo. Experiences with fault-injection in a byzantine fault-tolerant protocol. In ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, pages 41–61. Springer, 2013.

[MHL94] Biswanath Mukherjee, L Todd Heberlein, and Karl N Levitt. Network intrusion detection. IEEE network, 8(3):26–41, 1994.

[MP17] Savita Mohurle and Manisha Patil. A brief study of wannacry threat: Ransomware attack 2017. International Journal of Advanced Research in Computer Science, 8(5), 2017.

[MSG^{+} 16] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. Network function virtualization: State-of-the-art and research challenges. IEEE Communications Surveys and Tutorials, 18(1):236–262, 2016.

[MSW^{+} 19] Sebastian Mödersheim, Anders Schlichtkrull, Georg Wagner, Ste fan More, and Lukas Alber. TPL: A trust policy language. In Weizhi Meng, Piotr Cofta, Christian Damsgaard Jensen, and Tyrone Grandison, editors, Trust Management XIII - 13th IFIP WG 11.11 International Conference, IFIPTM 2019, Copenhagen, Denmark, July 17-19, 2019, Proceedings, volume 563 of IFIP Advances in Information and Communication Technology, pages 209–223. Springer, 2019.

[MV19] Sebastian Mödersheim and Luca Vigano, Alpha-betaprivacy. ACM Trans. Priv. Secur., 22(1):7:1–7:35, 2019.

[NNV17] Flemming Nielson, Hanne Riis Nielson, and Panagiotis Vasilikos. Information flow for timed automata. In Luca Aceto, Giorgio Bacci, Giovanni Bacci, Anna Ing´olfsd´ottir, Axel Legay, and Radu Mardare, editors, Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday, volume 10460 of Lecture Notes in Computer Science, pages 3–21. Springer, 2017.

[NoM] https://www.wired.com/story/no-more-screen-time-navy- reverts-physical-throttles/. Accessed: 2019-11-07.

[NSC^{+} 16] Antonio L. Maia Neto, Artur L. F. Souza, Italo Cunha, Michele Nogueira, Ivan Oliveira Nunes, Leonardo Cotta, Nicolas Gentille, Antonio A. F. Loureiro, Diego F. Aranha, Harsh Kupwade Patil, and Leonardo B. Oliveira. Aot: Authentication and access control for the entire iot device life-cycle. In Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, Sen- Sys ’16, pages 1–15, New York, NY, USA, 2016. ACM.

[OFM] OFMC/AIF tool suite. Available at http://www2.compute.dtu. dk/~samo/.

[OKS12] Aida Omerovic, Amela Karahasanovic, and Ketil Stølen. Uncertainty handling in weighted dependency trees: A systematic literature review. In Dependability and Computer Engineering: Concepts for Software-Intensive Systems, pages 381–416. IGI Global, 2012.

[Par96] Simon Parsons. Current approaches to handling imperfect information in data and knowledge bases. IEEE Trans. on Knowl. and Data Eng., 8(3):353–372, June 1996.

[PC12] Shari Lawrence Pfleeger and Deanna D Caputo. Leveraging behavioral science to mitigate cyber security risk. Computers & security, 31(4):597–611, 2012.

[PCvdV+17] Andre Pawlowski, Moritz Contag, Victor van der Veen, Chris Ouwehand, Thorsten Holz, Herbert Bos, Elias Athanasopoulos, and Cristiano Giuffrida. MARX: uncovering class hierarchies in C++ programs. In 24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26 - March 1, 2017, 2017.

[PLE] PLEAK. Available at https://pleak.io/home

[PMB+ 16] Charith Perera, Ciaran McCormick, Arosha K Bandara, Blaine A Price, and Bashar Nuseibeh. Privacy-by-design framework for assessing internet of things applications and platforms. In Proceedings of the 6th International Conference on the Internet of Things, pages 83–92. ACM, 2016.

[Pon19] Ponemon Institute. The fourth annual study on the cyber resilient organization, 2019.

[Pro] ProVerif: Cryptographic protocol verifier in the formal model. Available at https://prosecco.gforge.inria.fr/personal/ bblanche/proverif/.

[PRW+ 15] Charith Perera, Rajiv Ranjan, Lizhe Wang, Samee U Khan, and Albert Y Zomaya. Big data privacy in the internet of things era. IT Professional, 17(3):32–39, 2015.

[RA17] Mohammad Ashiqur Rahman and Ehab Al-Shaer. Automated synthesis of distributed network access controls: A formal framework with refinement. IEEE Trans. Parallel Distrib. Syst., 28(2):416–430, 2017.

[Saa10] Antonio Skarmeta et al. D3.1 Common Framework Handbook 1, available at https://cybersec4europe.eu/wp-content/uploads/2019/11/D3.1-Common-Framework-Handbook-1_submitted.pdf

[RG13] M. Rodano and K. Giammarco. A formal method for evaluation of a modeled system architecture. Procedia Computer Science, 20:210 – 215, 2013.

[RLK+19] Michael Rodler, Wenting Li, Ghassan O. Karame, Lucas Davi. Sereum: protecting existing smart contracts against re-entrancy attacks. In Proceedings of the 2019 Network and Distributed System Security Symposium, NDSS 2019, San Diego, CA, USA, February 24-27 2019.

[Sel03] B. Selic. The Pragmatics of Model-Driven Development. IEEE Software, 20(5):19–25, 2003.

[SEM] SEMCO (www.semcomdt.org).

[SFSB16] Jonas Schneider,Nils Fleischhacker,Dominique Schroeder, and Michael Backes. Efficient cryptographic password hardening services from partially oblivious commitments. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pages 1192–1203, New York, NY, USA, 2016. ACM.

[SK11] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications, 34(1):1–11, 2011.

[SKGA16] Pawel Sarbinowski, Vasileios P. Kemerlis, Cristiano Giuffrida, and Elias Athanasopoulos. Vtpin: Practical vtable hijacking protection for binaries. In Proceedings of the 32Nd Annual Conference on Computer Security Applications, ACSAC '16, pages 448–459, New York, NY, USA, 2016. ACM.

[SKLF19] Claudio Soriente, Ghassan Karame, Wenting Li, and Sergey Fedorov. Replicatee: Enabling seamless replication of SGX enclaves in the cloud. In IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019, pages 158–171. IEEE, 2019.

[SOB] SOBEK. Sobek, Android security by introspection,
https://www.dcc.fc.up.pt/~rmartins/papers/sobek.pdf

[son] Sony Hacked Again, 1 Million Passwords Exposed.

[SPNR16] http://www.informationweek.com/security/attacks/ sony-hacked-again-1-million-passwords-ex/229900111.

[SPWS13] Radu Stoenescu, Matei Popovici, Lorina Negreanu, and Costin Raiciu. Symnet: Scalable symbolic execution for modern networks. In Proceedings of the 2016 ACM SIGCOMM Conference, SIGCOMM '16, pages 314–327, New York, NY, USA, 2016. ACM.

[SRC$^+$ 12] Laszlo Szekeres, Mathias Payer, Tao Wei, and Dawn Song. Sok: Eternal war in memory. In 2013 IEEE Symposium on Security and Privacy, pages 48–62. IEEE, 2013.

[SRGCP15] Ina Schaefer, Rick Rabiser, Dave Clarke, Lorenzo Bettini, David Benavides, Goetz Botterweck, Animesh Pathak, Salvador Trujillo, and Karina Villela. Software diversity: state of the art and perspectives, 2012.

[SS15] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. Computer networks, 76:146–164, 2015.

[SVJ$^+$ 15] S. Singh and N. Singh. Internet of things (iot): Security challenges, business opportunities reference architecture for ecommerce. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pages 1577–1581, Oct 2015.

[Swe02] Serena Spinoso, Matteo Virgilio, Wolfgang John, Antonio Manzalini, Guido Marchetto, and Riccardo Sisto. Formal verification of virtual network function graphs in an sp-devops context. In Service Oriented and Cloud Computing, pages 253–262. Springer, 2015.

[SYS] Latanya Sweeney. k-anonymity: A model for protecting privacy.

[Tam] International Journal of Uncertainty, Fuzziness and Knowledge- Based Systems, 10(5):557–570, 2002.

[tBL+20] M. ter Beek, A. Legay, A. Lluch Lafuente, and A. Vandin. Variability meets Security: Quantitative Security Modeling and Analysis of Highly Customizable Attack Scenarios. In *Proceedings of the 14th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS'20), Magdeburg, Germany* (M. Acher and M. Cordy, eds.), ACM, New York, 2020.

[TDD+18] Petar Tsankov, Andrei Marian Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Bu ̈nzli, and Martin T. Vechev. Securify: Practical security analysis of smart contracts. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018, pages 67–82, 2018

[SYSVER] SYSVER System Verifier (see [CDS+ 19] and [CDSV17])

[TPC+] Tamarin prover. Available at https://tamarin-prover.github.

[TTY+ 19] Aivo Toots, Reedik Tuuling, Maksym Yerokhin, Marlon Dumas, Luciano Garc ́ıa-Ban ̃uelos, Peeter Laud, Raimundas Matulevicius, Alisa Pankova, Martin Pettai, Pille Pullonen, and Jake Tom. Business process privacy analysis in pleak. In Reiner H ̈ahnle and Wil M. P. van der Aalst, editors, Fundamental Approaches to Software Engineering - 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, volume 11424 of Lecture Notes in Computer Science, pages 306–312. Springer, 2019.

[twi] Twitter detects and shuts down password data hack in progress. http://arstechnica.com/security/ 2013/02/twitter-detects-and-shuts-down-password- data-hack-in-progress/.

[TPC18] Giannis Tzagarakis, Panagiotis Papadopoulos, Antonios A Chariton, Elias Athanasopoulos, and Evangelos P Markatos. Øpass: Zero-storage password management based on password reminders. In EuroSec 2018

[vdKAB+18] Erik van der Kouwe, Dennis Andriesse, Herbert Bos, Cristiano Giuffrida, and Gernot Heiser. Benchmarking crimes: an emerging threat in systems security. arXiv preprint arXiv:1801.02381, 2018.

[VER] VEREFOO. Available at https://github.com/ netgroup-polito/verefoo.

[Ver19] Verizon. Data Breach Investigations Report, 2019.

[vGC+ 16] Victor van der Veen, Enes G ̈okta ̧s, Moritz Contag, Andre Pawloski, Xi Chen, Sanjay Rawat, Herbert Bos, Thorsten Holz, Elias Athanasopoulos, and Cristiano Giuffrida. A Tough call: Mitigating Advanced Code-Reuse Attacks At The Binary Level. In Proc. of IEEE S&P, pages 934–953, May 2016.

[VNN17] Panagiotis Vasilikos, Flemming Nielson, and Hanne Riis Nielson. Time dependent policy-based access control. In Sven Schewe, Thomas Schneider, and Jef Wijsen, editors, 24th International Symposium on Temporal Representation and Reasoning, TIME 2017, October 16-18, 2017, Mons, Belgium, volume 90 of LIPIcs, pages 21:1–21:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[VNN18] Panagiotis Vasilikos, Flemming Nielson, and Hanne Riis Nielson. Secure information release in timed automata. In Lujo Bauer and Ralf Ku ̈sters, editors, Principles of Security and Trust 7th International Conference, POST 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, volume 10804 of Lecture Notes in Computer Science, pages 28–52. Springer, 2018.

[VNNK19] Panagiotis Vasilikos, Hanne Riis Nielson, Flemming Nielson, and Boris K ̈opf. Timing leaks and coarse-grained clocks. In 32nd IEEE Computer Security Foundations Symposium, CSF 2019, Hoboken, NJ, USA, June 25-28, 2019, pages 32–47. IEEE, 2019.

[Web15] Rolf H. Weber. Internet of things: Privacy issues revisited. Computer Law and Security Review, 31(5):618 – 627, 2015.

[WMAH15] Bruce D. Weinberg, George R. Milne, Yana G. Andonova, and Fatima M. Hajjat. Internet of things: Convenience vs. privacy and secrecy. Business Horizons, 58(6):615 – 624, 2015. SPECIAL ISSUE: THE MAGIC OF SECRETS.

[WSS+ 01] Andrea Westerinen, John Schnizlein, John Strassner, Mark Scherling, Bob Quinn, Shai Herzog, An-Ni Huynh, Mark Carlson, Jay Perry, and Steven Waldbusser. Terminology for policy-based management. RFC, 3198:1–21, 2001.

[Wu00] T. Wu. The srp authentication and key exchange system, 2000.

[Zad78] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. F uzzy sets and systems, 1(1):3–28, 1978.

[ZCDV17] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos. Security and privacy for cloud-based iot: Challenges. IEEE Communications Magazine, 55(1):26–33, January 2017.

[ZCW$^+$ 14] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, ChiaWei Hsu, Chong-Kuan Chen, and Shiuhpyng Shieh. Iot security: ongoing challenges and research opportunities. In 2014 IEEE 7th international conference on service-oriented computing and applications, pages 230–234. IEEE, 2014.

[ZXD+18] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: a survey. IJWGS, 14(4):352–375, 2018

Appendices

A.1 Main CyberSec4Europe Partner Responsible for Each Short-Term Research Activity

Short-Term Research Activity	Main Partner
Logical foundations of privacy and security	DTU
Protocol verification	DTU
Quantitative security	DTU
Privacy definitions for complex systems	CYBER
Privacy policy languages for complex systems	CYBER
Privacy analyses for complex systems	CYBER
Identity management for IoT	C3P
Authentication, Authorization and Access Control for IoT	C3P
Secure infrastructures for IoT	C3P
Software Hardening	UCY
Password hardening	UCY
Formal verification for secure access control	CNR
Enriched formal models to cope with security threats	CNR
Analysis of security issues and synthesis of corrections	CNR
Formal security requirements	IRIT
Formal threat specification	IRIT
Security patterns	IRIT
Model-based integration & validation of security patterns	IRIT
Requirements for data-based risk estimates	SINTEF
Automated network security functions	POLITO
Performance and scalabilty of automated network security	POLITO
Scalable secure and practical consensus layers	NEC
Efficient privacy-preserving blockchain protocols	NEC
Protocols for TEEs in cloud-based applications	NEC

A.2 Full List of Research Assets

- BadGraphs [Bad]
- BowTiePlus [Bow]
- CORAS [COR]
- HERMES [MGN+13]
- OFMC/AIF [OFM]
- PLEAK [PLE]
- SEMCO [SEM]
- SOBEK [SOB]
- SYSVER [SYSVER]
- VEREFOO [VER]

Extended descriptions of the assets are available in Section 5.2 of D3.1 [Saa10].

A.3 Mapping of Assets into Demostrators and Vertical Sectors

	Open Banking	**Supply Chain Security Assurance**	**Maritime Transport**	**Medical Data Exchange**	**Smart Cities**
BadGraph				MD-SP07 MD- SP02 MD- SP03	
BowTiePlus			MT-U01 MT-OP01		
CORAS			MT-U01 MT-OP01		
OFMC/AIF				MD-SP07 MD- SP02 MD- SP03	
PLEAK			MT-UC1 MT-UC2 MT-UC3 MT-UC4	MD-UC1 MD-UC2 MD-UC3	
SEMCO				SP01 SP02 SP03 SP04 O01 O02 SPL01 LR01	
HERMES	OB-OP02	SCH-OP02			
SOBEK	OB-SP11 OB-SP14 OB-SP23 OB-LR01 OB-LR04				
SYSVER				MT-OP03	SMC-SP05 SMC-OP01 SMC-OP04
VEREFOO	OBSP21 OBSP22				SMCF02 SMCF16 SMCSP05 SMCU01

This is an excerpt of the original table that can be found in Section 6.2 of D3.1 [Saa10].

A.4 Mapping of Assets into Building Blocks of the CyberSec4Europe Global Architecture

The below figure is taken from Section 4.3 of D3.1 [Saa10]. It illustrates the global architecture of CyberSec4Europe and indicates how the tasks of WP3 are mapped into the building blocks of the architecture. A detailed description of the architecture and of the figure can be found in D3.1 [Saa10].

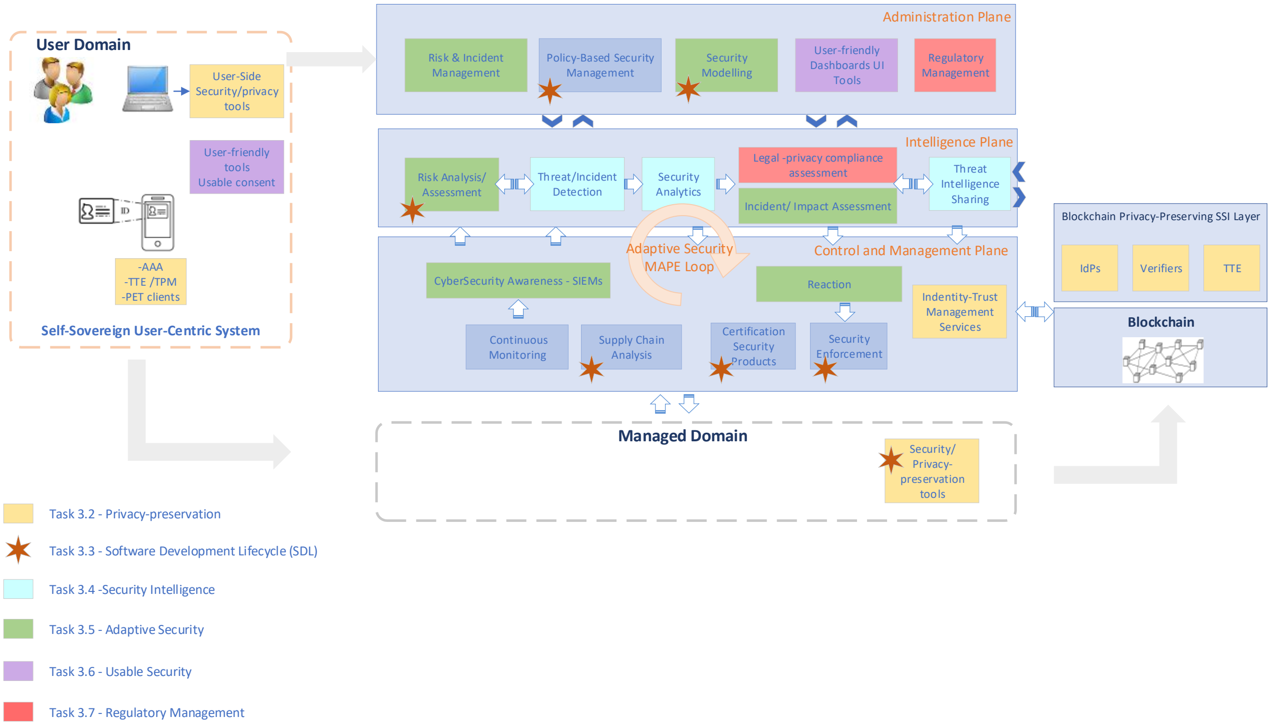
User Domain
User-Side Security/privacy tools
User-friendly tools Usable consent
-AAA
-TTE /TPM
-PET clients
Self-Sovereign User-Centric System
Administration Plane
Risk & Incident Management
Policy-Based Security Management
Security Modelling
User-friendly Dashboards UI Tools
Regulatory Management
Intelligence Plane
Risk Analysis/ Assessment
Threat/Incident Detection
Security Analytics
Legal -privacy compliance assessment
Incident/ Impact Assessment
Threat Intelligence Sharing
Adaptive Security MAPE Loop
Control and Management Plane
CyberSecurity Awareness - SIEMs
Reaction
Indentity-Trust Management Services
Continuous Monitoring
Supply Chain Analysis
Certification Security Products
Security Enforcement
Blockchain Privacy-Preserving SSI Layer
IdPs
Verifiers
TTE
Blockchain
Managed Domain
Security/ Privacy-preservation tools
Task 3.2 - Privacy-preservation
Task 3.3 - Software Development Lifecycle (SDL)
Task 3.4 -Security Intelligence
Task 3.5 - Adaptive Security
Task 3.6 - Usable Security
Task 3.7 - Regulatory Management

Figure 1 - CyberSec4Europe global architecture and building blocks per WP3 task

While the above figure relates Task 3.3 with the building blocks of the arcthiecture, the below table provides a more detailed mapping, relating research assets with building blocks.

	Policy-based Security Management	Security Modelling	Risk Analysis / Assessment	Supply Chain Analysis	Certification Security Products	Security Enforcement	Smart Security / Privacy-preserving tools
BadGraph		X	X				
BowTiePlus		X	X				
CORAS		X	X				
OFMC/AIF					X	X	
PLEAK							X
SEMCO	X	X					
HERMES						X	
SOBEK						X	
SYSVER	X						
VEREFOO	X						

A.5 Mapping of Research into Building Blocks of the CyberSec4Europe Global Architecture

The following table enriches the mapping of research activities within the global architecture of CyberSec4Europe (cf. D3.1 [Saa10]), relating short-term research activities with the building blocks of the global architecture.

	Policy-based Security Management	Security Modelling	Risk Analysis / Assessment	Supply Chain Analysis	Certification Security Products	Security Enforcement	Smart Security / Privacy-preserving tools
Logical foundations of privacy and security		X		X			
Protocol verification				X	X		
Quantitative security		X	X				
Privacy definitions for complex systems		X	X				
Privacy policy languages for complex systems							X
Privacy analyses for complex systems							X
Identity management for IoT						X	
Authentication, Authorization and Access Control for IoT						X	X
Secure infrastructures for IoT						X	
Software Hardening						X	
Password hardening							X
Formal verification for secure access control	X						
Enriched formal models to cope with security threats		X					
Analysis of security issues and synthesis of corrections	X						
Formal security requirements	X	X					
Formal threat specification	X	X					
Security patterns	X	X					
Model-based integration & validation of security patterns	X	X					
Requirements for data-based risk estimates		X	X				
Automated network security functions	X						
Performance and scalabilty of automated network security	X						
Scalable secure and practical consensus layers				X			X
Efficient privacy-preserving blockchain protocols				X		X	X
Protocols for TEEs in cloud-based applications						X	