Security Technologies and Methods for Advanced Cyber Threat Intelligence, Detection and Mitigation

Gohar Sargsyan
Dimitrios Kavallieros
Nicholas Kolokotronis
(Editors)

now
the essence of knowledge

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

ISBN: 978-1-68083-834-3
E-ISBN: 978-1-68083-835-0
DOI: 10.1561/9781680838350

Suggested citation: Gohar Sargsyan, Dimitrios Kavallieros and Nicholas E. Kolokotronis (eds.). (2022). *Security Technologies and Methods for Advanced Cyber Threat Intelligence, Detection and Mitigation*. Boston–Delft: Now Publishers

Table of Contents

Dedication

To my family for giving me unconditional love and for always being there for me to support in any circumstances.

–Gohar Sargsyan

To my wife, Eleni and friends who are like family, for all the support and love through all these years.

–Dimitrios Kavallieros

To my wife, Mary, and children, Athanasia and Manos, for their endless love and support.

–Nicholas E. Kolokotronis

Preface

Dear reader,

Dear colleagues in Cyber Security,

It is our pleasure to present this book and say a few words about the strategy of its development and creation. The exponential growth of the Internet interconnectivity has led us to a significant growth of cyber-attack incidents globally often with severe and disastrous consequences. The rapid development of more innovative and effective (cyber)security solutions and approaches became an urgency for the (digital/cyber) security professionals to create solutions to detect, mitigate and prevent from grievous consequences.

Therefore, several years ago we, the editors of this book, came together as a part of core group to brainstorm about creating innovative advanced cyber-threat intelligence, detection and mitigation solutions. The idea was extended into broader domain experts comprising of total nine partner multidisciplinary organisations from seven EU countries, covering all key aspects of a successful programme. This way the project Cyber-Trust (Advanced Cyber-Threat Intelligence, Detection and Mitigation Platform for IoT) was created and submitted to the European Commission's H2020 framework programme for evaluation and potential co-funding. We were thrilled to receive the maximum possible scores for all the evaluation criteria (1. the excellence and the innovative idea, 2. the impact and 3. the implementation plan) for our proposal. This led to an amazing three and a half years of partnership and collaboration journey to execute and deliver results as promised.

As an outcome of these excellent partnership and collaboration efforts, this book is a synergetic product of many different minds coming out from multidisciplinary and multicultural professionals in cybersecurity domain, originating from Cyber-Trust project. The idea to produce this book came up during the third year of

intensive collaboration work within Cyber-Trust focusing from the perspective of the exploitation of the results. And it did not stop there.

We remained open and engaged in more ideas and contributions from our experiences and future professional plans to make sure that this book will be a valuable contribution to research and innovation, science and business on the (digital/cyber) security domain beyond the framework of our current collaboration.

This book provides insights on new security technologies and methods for advanced cyber threat intelligence, detection and mitigation. We cover topics such as cyber-security and AI, cyber-threat intelligence, digital forensics, moving target defense, intrusion detection systems, post-quantum security, privacy and data protection, security visualization, smart contracts security, software security, blockchain, security architectures, system and data integrity, trust management systems, distributed systems security, dynamic risk management, privacy and ethics.

Wishing you interesting reading!

Yours sincerely,

Editors

Gohar Sargsyan
Dimitrios Kavallieros
Nicholas E. Kolokotronis

Executive Summary

The "*Security Technologies and Methods for Advanced for Advanced Cyber Threat Intelligence, Detection and Mitigation*" book builds on the experience of the Cyber-Trust EU project's (grant agreement 786698) methods, use cases, technology development, testing and validation and extends into a broader science, lead IT industry market and applied research with practical cases. Cybersecurity is gaining momentum and is scaling up in very many areas, as this publication will show. We provide new perspectives on advanced (cyber) security innovation (eco) systems covering key different perspectives. How to build and run them from the process and skills perspective is of great importance when developing, applying and scaling up innovative security systems.

This book is comprised of 12 chapters, consisting of independent parts, which provide complete view both on their own and interconnected with relevant parts within the book. Below we briefly summarise the contents of each chapter.

In Chapter 1 the authors cover Design and Architecture considerations for Advanced Cyber-Threat Intelligence, Detection, and Mitigation Platforms. In particular an architectural framework and approach is introduced which guarantees better efficiency.

Chapter 2 explores the procedural aspects detailing how the impact assessment process is organised and takes place inside such complex cybersecurity platforms referring to Cyber-Trust case.

The authors of Chapter 3 outline a system that incorporates and extends current tools and techniques from the Cyber Threat Intelligence life-cycle by providing a holistic view in the Cyber-Threat Intelligence process.

Moving Target Defense techniques for mitigation sophisticated IoT (Internet of Things) is the core of the Chapter 4, which presents an implementation of an intrusion response system. The authors also demonstrate that the evaluation results

showed its high effectiveness against traditional threats, and increased in effectiveness against novel threats.

Chapter 5 is focusing on Cyber-Threat Detection in the IoT. Here the authors present a comprehensive overview of the IoT devices profiling and threat detection solution proposed by Cyber-Trust to tackle the grand challenges of securing the IoT devices' ecosystem. In addition, the effectiveness and performance of the proposed solution are in-depth verified, especially against botnets and Zero-day attacks.

For the Mitigation of Unknown Threats in IoT Honeypots, Machine Learning can be utilized to effectively address the issue, which is described in detail in Chapter 6. The approach introduced in this chapter is novel which detects malicious network traffic that employs a honeypot and machine learning.

In Chapter 7 the authors provide a theoretical support of the recent developments in the area of post-quantum cryptography (PQC) aiming at the incorporation of secure cryptographic primitives to the blockchains. The challenges to both researchers and industry regarding the implementation of postquantum algorithms in blockchain applications are demonstrated.

The authors of Chapter 8 discuss and propose an approach to trust computation in the Internet of things, which synthesizes behavioral, device status and associated risk aspects into a comprehensive trust score, that can be consulted to realize trust-based access control.

Chapter 9 introduces the testing, validation, verification and evaluation methodology that Cyber-Trust project followed during the pilot phase of the project's lifecycle. In a nutshell, the authors present that collecting and analyzing data from pilot activities reveals the satisfaction rate of the stakeholders and the level of system's performance.

From testing and validation moving on into testbeds for business. Chapter 10 is all about Smart Home testbeds for Business. The authors present the results from the emulated, tested SoHo (Smart Home) platform, their exploitation potential in several fields, mainly from a business perspective as well as their impact on business and potential extensions.

For Chapter 11, we have a valuable input from an industry leadership discussing how to secure today's complex digital realities by introducing tested and proven CGI cybersecurity approach for today's modern work environments.

Last but not least, Chapter 12 of this book is about the security and privacy aspects for digital twins, drivers, concerns and recommendations on how to manage risks. Practical cases on point are also provided.

DOI: 10.1561/9781680838350.ch1

Chapter 1

How to Design and Set Architecture for Advanced Cyber-Threat Intelligence, Detection, and Mitigation Platforms

By G. Sargsyan and R. Binnendijk†*

CGI
*gohar.sargsyan@cgi.com
†raymond.binnedijk@cgi.com

This chapter will demonstrate how to design and set architecture for advanced cyber-threat intelligence, detection and mitigation platforms following the example of Cyber-Trust EU research and innovation project [1] applying proven architecture methodology Risk- and Cost-Driven Architecture (RCDA) [2]. The architecture approach RCDA have advantages versus other approaches which helped the consortium partners to agree upon from early stage of platform design and development. According to RCDA principles, the architecture work starts with identifying architectural concerns with the highest impact in terms of risk and cost, and addressing those concerns by making architectural decisions. Hence, this article contains the results of the most impactful architectural decisions made. This has allowed the architecture and requirements processes to mutually benefit from each other's progress, and resulted in good cohesion between requirements and architecture. The price for this cohesion is some rework in maintaining traceability: In this article we introduces the requirements traceability which is based on an early stage requirements and further extended into references to the output of end user requirements and legal, ethical and data protection frameworks.

The concerns with the highest impact in terms of risk and cost identified at the start of the project were especially integration, but also compliance and security.

Integration is a concern because the cyber-treat intelligence, detection and mitigation solution is composed of many separate components which are being developed by various development and research teams. This concern is addressed by shaping a modular architecture composed of various loosely coupled components where the interfaces between these components are shaped via integration guidelines. In addition, the architecture includes the approach chosen to develop or otherwise obtain the deliverable elements that make up the technical solution.

Compliance is an important concern, especially with respect to legal, ethical, social and privacy rules. This concern is mainly addressed in Cyber-Trust use case scenarios, the end user requirements legal and ethical recommendations, and impact assessment.

Security is always a key concern in such complex platforms especially on designing and developing cyber-threat intelligence, detection, and mitigation platforms. We address this concern, aligned with and complementary to Legal and ethical recommendations.

1.1 Background and Driving Forces

By establishing an innovative cyber-threat intelligence gathering, detection, and mitigation platform, as well as, by performing high-quality interdisciplinary research in key areas, the Cyber-Trust project aims to develop novel technologies and concepts to tackle the grand challenges towards securing the ecosystem of IoT devices. It is structured around three pillars: a. key proactive technologies, b. cyber-attack detection and mitigation, and c. distributed ledger technologies, as seen in the Table 1.1 below.

To set up the Cyber-Trust platform design and architecture iterative approach was allied to be able to have the opportunity to validate and learn regarding the architectural decisions made. The following iterative cycles have been applied:

Iterative Cycle 1 – The user requirements and regulatory framework have been set up to pave the way for the system design and architecture. During this phase, emerging trends in cyber-attacks have been identified to guide the definition of use case scenarios and the collection of the end-user requirements and the regulatory framework is being analysed and the impact of the proposed methods to fundamental rights, data protection and privacy is being assessed. The use cases have been identified. Iterative Cycle 1 includes the work packages

- Cyber-threat landscape and end-user requirements;
- Legal issues: data protection and privacy.

Iterative Cycle 2 – Platform design. In this phase, the Cyber-Trust platform reference architecture is created, incorporating inputs from the first phase, translated into technological tools to be built in Iterative Cycle 3. The tools above comprising the integrated platform are being designed and prototyped and the consortium is in the initial stage of the platform design. The design and architecture of the system is implemented under the work package

- Cyber-Trust framework, platform design and architecture.

The main outputs of this phase are the platform's prototype, and its final specifications at the end of the phase which are associated with a milestone Cyber-Trust architecture and design specifications. In this phase initial version of the system design and architecture is set including integration plan. To ensure compliance and security privacy consideration, ethical and legal aspects continue to be active in this phase to review and advice on the requirements.

Iterative Cycle 3 – Refinement of design and platform architecture. In this iterative cycle, the Cyber-Trust platform reference architecture is iteratively being monitored and refined in parallel with the tools development and during the validation of pilots. The tools are being developed and architecture is being refined (if any flaws) or being revalidated throughout the course of software development and integration. When the platform is ready pilots will validate the platform, where design and architecture follows the final stage of revalidation and provision of any input the platform arcitcture may have for more robustness.

In setting up this complex design and architecture of Cyber-Trust we applied iterative approach. Firstly, the initial architecture was delivered. Then feedback was gathered during testing and validation workshops engaging advisory board

Table 1.1. Three pillars of Cyber-Trust.

Key proactive technologies	Attack detection and mitigation	Distributed Ledger Technologies
• cyber-threat intelligence • cyber-threat sharing • reputation/trust management • security games	• advanced targeted attacks • network infrastructure attacks • network visualisation • mitigation and remediation • forensics evidence collection	• registration • update • verification • modelling • consensus • privacy

members and focused expert group. These feedback was processed during Platform reference architecture and design specification. The partners produced a prototype as a draft version of actual working and partly integrated software components. Each technical partner contributed to the development, design and provided, explained and shared technology that will serve as the base building blocks for implementing the Cyber-Trust platform during component and software development cycle. By developing software early in the project, during design and architecture iterative cycle, architecture and implementation was merged early, which provided the advantage of validating and refining the architecture and jumpstart the implementation to be performed during core software development phase.

Mixture of research and development: Key Proactive Technologies and cyber-threat intelligence, Advanced cyber-attack detection and mitigation and Distributed ledger technology for enhanced accountability follow (and partly go parallel) work package Cyber-Trust framework, platform design and architecture activities and aim to implement the solution architecture (Proof of Concept). These implementation activities are comprised of a mixture of research and development activities, were relevant state of art technology is identified, used and extended and new tools are custom developed. The research and technology partners focused closely work together with clear identified roles and responsibilities to ensure efficient, high quality and smooth delivery of Cyber-Trust platform.

1.2 Architecture Approach and Methodology

In this section we will provide brief description of architecture methodology

1.2.1 Risk and Cost Driven Architecture Methodology (RCDA)

The consortium has chosen Risk and Cost-Driven Architecture (RCDA) framework and approach as the key methodology for architecture design. The advantage of applying this method is that it supports architectural decision making throughout the whole design process [3]. Concerns and decisions are weighed throughout the design process and stakeholders' requirements are constantly validated against the design. The design process is iterative to ensure high-quality results. The fact that RCDA is a recognized method in the Open Group Certified Architect program, [4] it is an extra advantage for the project and consortium partners to promote openness and collaboration on the most efficient way of shaping the design and architecture.

RCDA practices were applied while **initially shaping the Cyber-Trust project**.

The following concrete measures were applied:

- The architect is involved during the requirements stage to help and guide aiming at improving the connection between requirements and design.
- Architecture is delivered in two increments, with the ability to verify and learn:
  - The initial architecture [5] is delivered at early stage, to be able to align requirements engineering with software development and to have the opportunity to validate design decisions.
  - This initial architecture is validated by building and testing working software, i.e. rapid prototype [6]
  - The architecture is determined after processing the feedback gathered through initial architecture [5] the rapid prototype assessment [6] and validation and UI mock ups demonstration, assessment and validation [7]
- Architecture focuses on critical design decisions and should not over-specify, and start early in the project, but the architectural work does not stop here. Technical Design and tools selection is performed later in the project during the development stage (Work Packages (WP) 5, 6 and 7 – WP5, 6, 7 in the project Cyber-Trust) and pilots implementation (WP8) which is the final stage towards platform evaluation and validation, therefore, final architecture of the platform. The architect is involved during these work packages where the architecture is validated and elaborated. The architect will help and guide but not lead.
- Legal and ethical recommendations have been provided throughout the iterations of architecture work [3].

During the project, at the highest level of abstraction, the architectural specification process follows a simple **workflow** loop with three steps:

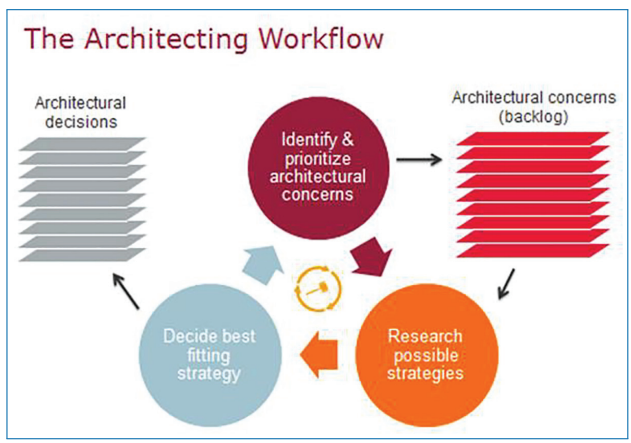


Figure 1.1. RCDA Architectural Micro cycle.

We call this the "Architecture Micro cycle". This workflow loop is driven by a backlog of unresolved architectural concerns, resulting from the ARCHITECTURAL REQUIREMENTS PRIORITIZATION practice. The architectural decisions taken, resulting from the ARCHITECTURAL DECISION-MAKING practice, to address these concerns are added to an ever-growing stack of Architectural Decisions.

This microcycle representation is a severe oversimplification. In real life, the architectural decisions usually affect more than one concern, and can hardly ever be made sequentially. The architect has to make sure that the entire set of decisions maximally supports the entire set of concerns.

In addition to the first two practices (prioritization and decision making) mentioned above, RCDA offers a set of core **practices** that are applied throughout the lifecycle of the project.

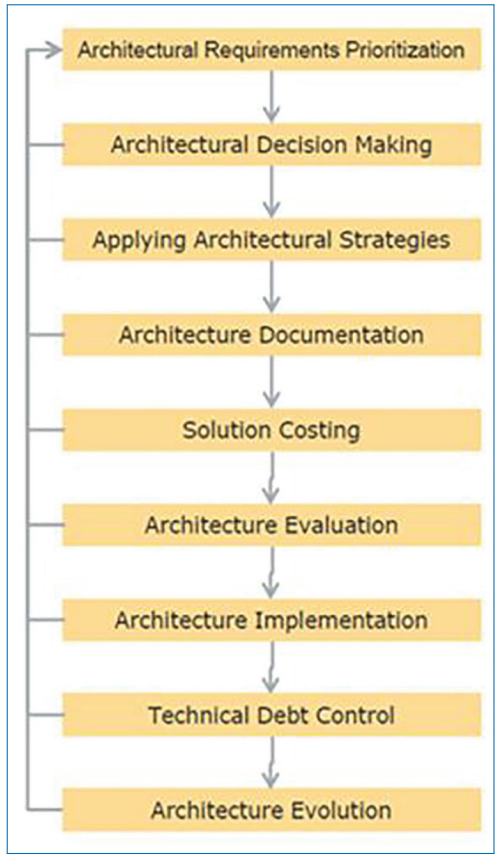


Figure 1.2. RCDA core practices.

For more information about RCDA see [2, 3].

Figure 1.3 shows how RCDA practices are applied within the Cyber-Trust project process.

Practices are applied incrementally, continuously identifying and prioritizing concerns and making (and applying and documenting and validating etc.) decisions to mitigate these concerns.

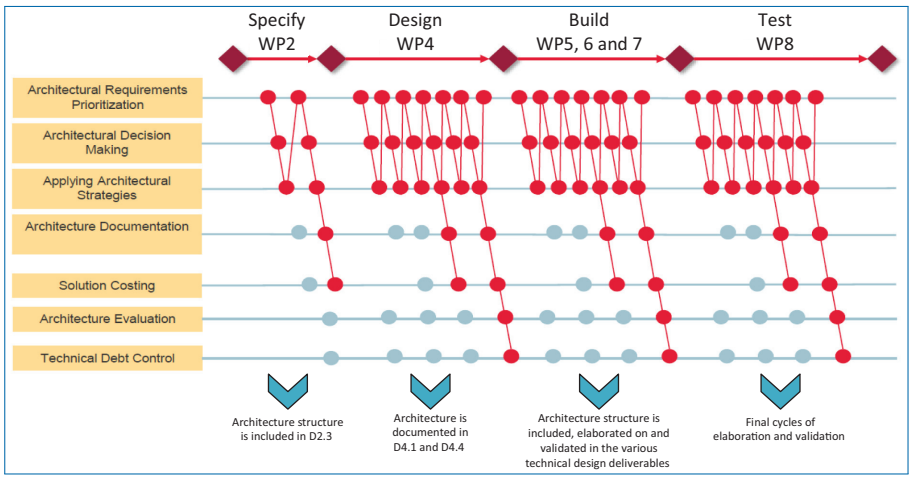


Figure 1.3. Indicative high-level overview of RCDA practices applied within Cyber-Trust project process.

The Cyber-Trust project is mainly based on a traditional, phased approach (waterfall). Although phases overlap and the architect is involved thorough the entire lifecycle, most of the work is performed in the design phase (WP4).

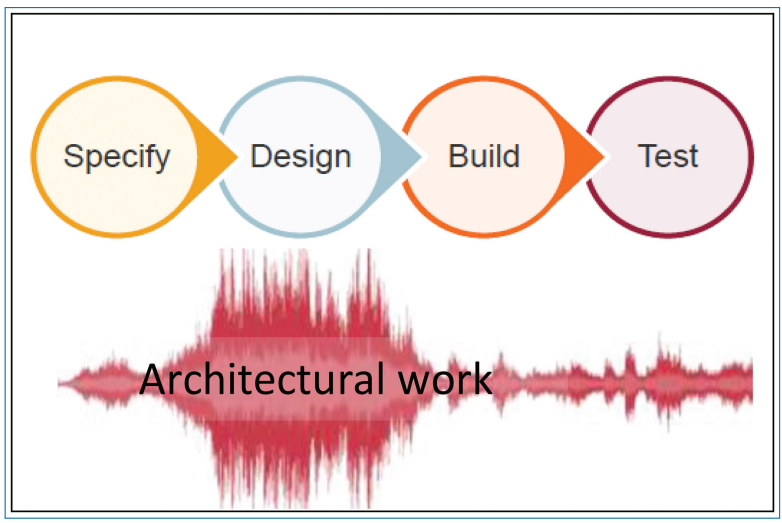


Figure 1.4. Architectural work is done throughout the project lifecycle.

1.2.2 Architecture Views

The Cyber-Trust solution architecture is shaped by the various architectural requirements and decisions made and documented in a set of views (see Table 1.2). These views focus on effectively communicating the architecture to the relevant stakeholders. Around and beyond these views, additional documentation is provided to complete technical systems design.

The views are detailed in subsequent chapters.

Table 1.2. Architectural views.

Architecture views		
Chapter	**View**	**Goal**
2	Context	Describe the high-level solution context.
3	Requirements	To identify, understand and prioritise architecturally significant requirements.
4	Decisions, concerns & deduced architectural requirements	To describe concerns, key decisions and deduced architectural requirements.
5	Operational view	To describe how the system behaves in an operational environment.
6	Delivery breakdown view	To serve as a basis for planning solution delivery.
7	Infrastructure view	To identify and explain hardware, infrastructure software and deployment aspects of the solution.
8	Data view	Describe the data that is relevant and how this data is distributed within the solution.
9	Security view	To describes the set of processes, mechanisms and components used to make the system secure.

1.2.3 Compliance and Security

Compliance is an important concern, especially with respect to legal, ethical, social, privacy rules. This concern is mainly addressed in Cyber-Trust uses case scenarios, Cyber-Trust end-user requirements and especially Legal and ethical recommendations, which explain how, compliance concerns vary based on the use cases, the tools to be developed within the Cyber-Trust project and the architecture will have to be flexible enough to address these variances.

Security is always a key concern in such complex platforms, especially in designing and developing cyber-threat intelligence, detection, and mitigation platforms.

More details will be provided later in this article addressing this concern, aligned with and complementary to legal and ethical recommendations.

The approach that we will apply on designing the Cyber-Trust platform is the following: the requirements (end-user requirements and architectural requirements) should be legally and ethically compliant. A legal and ethical review has been provided by a dedicated expert partner throughout the entire duration since the setting up the system until the validation while designing and developing the system.

1.3 Solution, Context and Overviews

1.3.1 Context

The Cyber-Trust project is built upon three main cyber-security research thrusts, that is key proactive technologies, cyber-attack detection and mitigation, and distributed ledger technologies. The proposed approach aims to capture different phases of a large-scale cyber-attack before and after existing (and possibly unknown) vulnerabilities of devices have been widely exploited by cyber-criminals to launch the attack. Some novel methods and tools will be developed to deal with the fundamental problems of prevention, detection, and mitigation of advanced cyber-attacks involving IoT devices and networks.

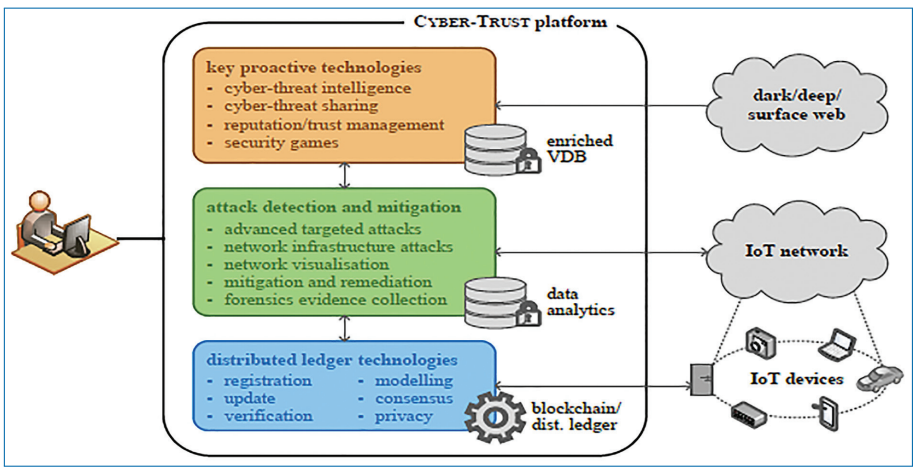


Figure 1.5. High-level solution overview. Source [1].

1.3.2 Static Solution Overview

The CyberTrust solution is assembled by four solution areas, indicated by four colours in the Figure 1.6 below:

(1) Platform containing all central services and data (blue)
(2) A platform with specific ISP services and data (orange)
(3) An application running on smart-phones (green)
(4) An application running on smart-gateways (purple)

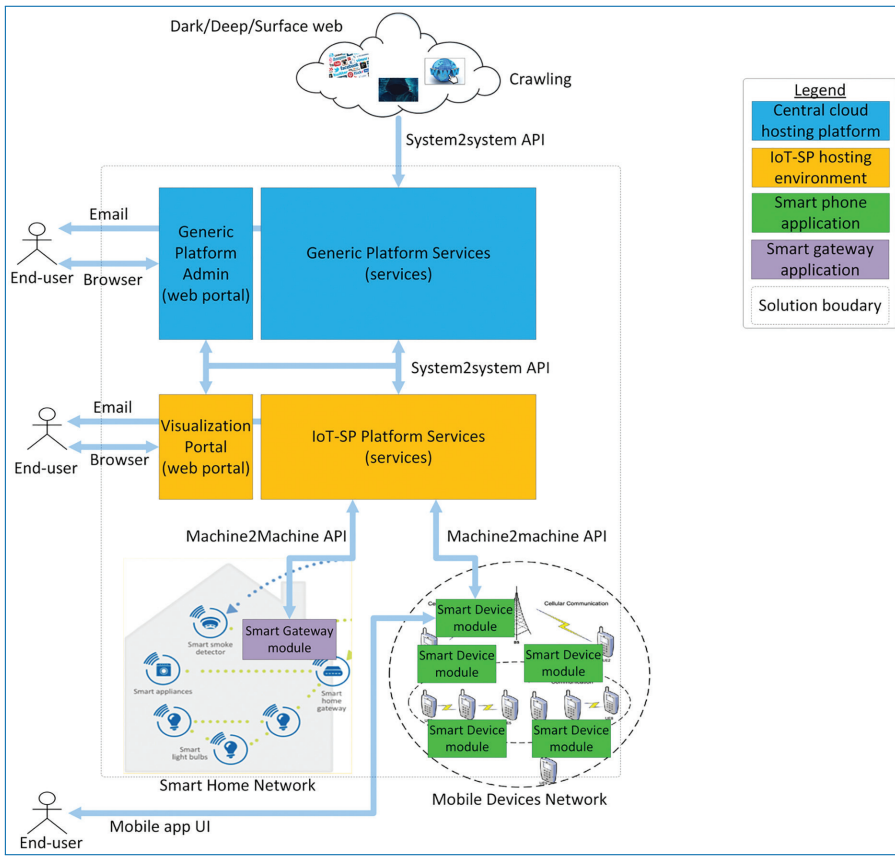


Figure 1.6. Static Solution Overview & Solution Boundary.

1.3.3 Runtime Solution Overview

The figure below shows the four Cyber-Trust solution area at runtime. Each ISP runs its own ISP-services platform, connecting to the various ISP related smart

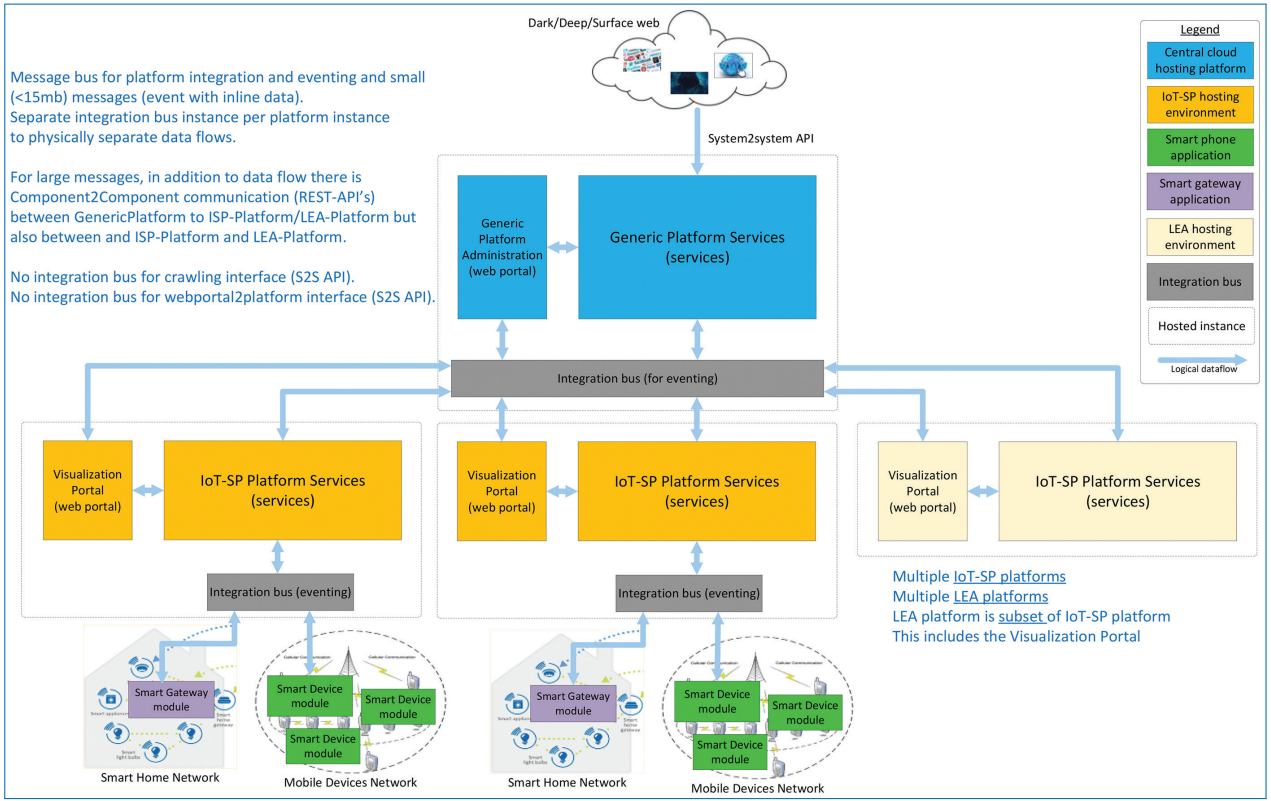


Figure 1.7 Run-time Solution Overview.

gateways and smart phones. If Law Enforcement Agencies (LEA's) join the platform they will run a dedicated platform, which consists of a subset of the ISP-platform.

Communication is done via messaging through an Integration Bus.

Regarding to Figure 1.7 the following considerations are made

- Message bus for platform integration and eventing and small (<15 mb) messages (event with inline data).
- Separate integration bus instance per platform instance to physically separate data flows.
- For large messages, in addition to data flow there is Component2 Component communication (REST-API's) between GenericPlatform to ISP-Platform/ LEA-Platform but also between and ISP-Platform and LEA-Platform.
- No integration bus for crawling interface (S2S API).
- No integration bus for webportal2platform interface (S2S API).
- Multiple IoT-SP platforms
- Multiple LEA platforms
- LEA platform is subset of IoT-SP platform
- This includes the Visualization Portal

The static and run-time solution overviews together comprise the key architecture of Cyber-Trust platform as cyber-threat intelligence, detection and mitigation platforms. Cyber-Trust platform development any technological advancement or devepThis architecture served as guide and fundamental in for further development of the project all technological works including development

1.4 Conclusions

In this chapter, we presented an approach to setting up the design and architecture of advanced cyber-threat intelligence gathering, detection and mitigation platform. We demonstrated this following the example of Cyber-Trust European Commission H2020 research and innovation project implemented by nine multidisciplinary partners from seven countries bringing together the best practices and experiences coming from the project partners. The architecture approach applied on Cyber-Trust is Risk and Cost Driven Architecture (RCDA) based on advantages versus other approaches that the consortium partners agreed upon at the project initiation stage of the platform development. According to RCDA principles, the architecture work starts with identifying architectural concerns with the highest impact in terms of risk and cost, and addressing those concerns by making architectural decisions. Hence, this chapter contains the results of the most impactful architectural decisions made. This has allowed the architecture and requirements

processes to mutually benefit from each other's progress, and resulted in good cohesion between requirements and architecture. The price for this cohesion is some rework in maintaining traceability: In this article we introduces the requirements traceability which is based on an early stage requirements and further extended into references to the output of end user requirements and legal, ethical and dat protection framewors. We also took into consideration the concerns with the highest impact in terms of risk and cost identified at the start of the project which were especially integration, but also compliance and security.

Acknowledgment


This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] Cyber-Trust Project – https://cyber-trust.eu/ – Advanced Cyber-Threat Intelligence, Detection and Mitigation in Trusted IoT, EU H2020 project grant agreement no. 78669

[2] CGI, Risk and Cost Driven Architecture Methodology (RCDA) – CGI registered IP 2012 (NL and corp).

[3] E.R. Poort, H. van Vliet, "RCDA: Architecting as a risk- and cost management discipline" Journal of Software and Systems, Selected Papersfrom 2011 Joint Working *IEEE/IFIP Conference on Software Architecture* (WICSA 2011), Volume 85, Issue 9, September 2012, pp 1995–2013 https://www.sciencedirect.com/science/article/abs/pii/S0164121212000994

[4] Open Group Certified Architect – http://www.opengroup.org/openca/cert/

[5] R. Binnendijk, G Sargsyan (CGI) et al.: "Architecture and Design Specifications: Initial" Cyber-Trust project Deliverable (D4.1) of Cyber-Trust 2019.

[6] G. Sargsyan, R. Binnenjijk (CGI) et al., "Rapid Prototype Evaluation Results and Assessment" Deliverable (D4.2) of Cyber-Trust 2019

[7] S. Cuomo, S. Naldini et al. (Mathema) "UI mock-ups evaluation, assessment and validation" Deliverable (D4.3) of Cyber-Trust, 2019

[8] P. Quinn, O. Gkotsopoulou (Vrij Universiteit Brussels), "Legal Issues, Data protection and Privacy" – Work Pachage in Cyber-Trust, 2018–2021, "Legal and Ethical Recommendations" Deliverable (D3.3) of Cyber-Trust 2018, "Cyber-Trust Impact Assessment" – 1, 2 Deliverables D3.4 and D3.5, 2020–2021.


IMPACT

DOI: 10.1561/9781680838350.ch2

Chapter 2

The Cyber-Trust Paradigm of Procedural Aspects for Cybersecurity Research Impact Assessment

By O. Gkotsopoulou[*]

Vrije Universiteit Brussel
olga.gkotsopoulou@vub.be

This chapter explores the meta-elements of an impact assessment, what we call the *procedural aspects, before, during and after*. In other words, how the procedure of the impact assessment is organised and takes place inside the Cyber-Trust project. This article concentrates all the experience gained and lessons learnt so far. The structural scheme used in the Cyber-Trust project can serve as a basis for other research project consortia which develop innovative solutions in the field, or as a starting point for discussion as to how to improve and eventually standardise such procedure.

*. This Chapter is based on the blogpost 'Procedural Aspects of an Impact Assessment for Innovative Cybersecurity Systems Research: The Cyber-Trust model' by Olga Gkotsopoulou, dated 3 December 2020, hosted on https://cyber-trust.eu/

2.1 Introduction and Background

The H2020 Cyber-Trust project aims to foster a holistic and novel cyber-threat intelligence gathering, prevention, detection and mitigation platform, to secure the complex and ever-growing smart infrastructure, used by millions of people daily. The project consortium follows the latest technical innovations as well as best practice in the field, observing developments in the applicable legal and regulatory framework and investigating other ethical and societal considerations. In this regard, from its conception, the Cyber-Trust project has established an impact assessment mechanism, with particular focus on data protection and privacy, as a cross-disciplinary exercise among its partners consisting of seven consecutive and strongly inter-connected steps. The mechanism corresponds to a data protection impact assessment as enshrined in Article 35 GDPR but given the complexity of the goal to be achieved, the consortium enhanced the procedure with elements of wider impact assessments including broader ethical and societal considerations.

This chapter explores the meta-elements of an impact assessment, what we call the *procedural aspects, before, during and after*. In other words, how the procedure of the impact assessment is organised and takes place inside the Cyber-Trust project. This article concentrates all the experience gained and lessons learnt so far. The structural scheme used in the Cyber-Trust project can serve as a basis for other research project consortia which develop innovative solutions in the field, or as a starting point for discussion as to how to improve and eventually standardise such procedure.

2.2 The Rationale Behind an Impact Assessment in a Cyber-security Research Project

With the entry into force of the General Data Protection Regulation in 2018, Data Protection Impact Assessments (or in short, DPIAs) became a legal requirement for data controllers regarding specific data processing operations in some contexts. The DPIAs refer to the development or deployment of a new system, product or process regarding the processing of personal data, for instance in a large-scale or a novel manner. They allow to identify risks well in advance and explore risk mitigation strategies.

Impact assessments, however, are not new. Environmental impact assessments have been implemented for years. Organisations have been performing privacy impact assessments, impact assessments from a societal or ethical point of view or even assessments with a particular focus.

2.3 The Rationale Behind an Impact Assessment in Cyber-Trust

A DPIA was considered necessary in the Cyber-Trust context, apart from the fact that it was part of the project's contractual obligations, for two reasons:

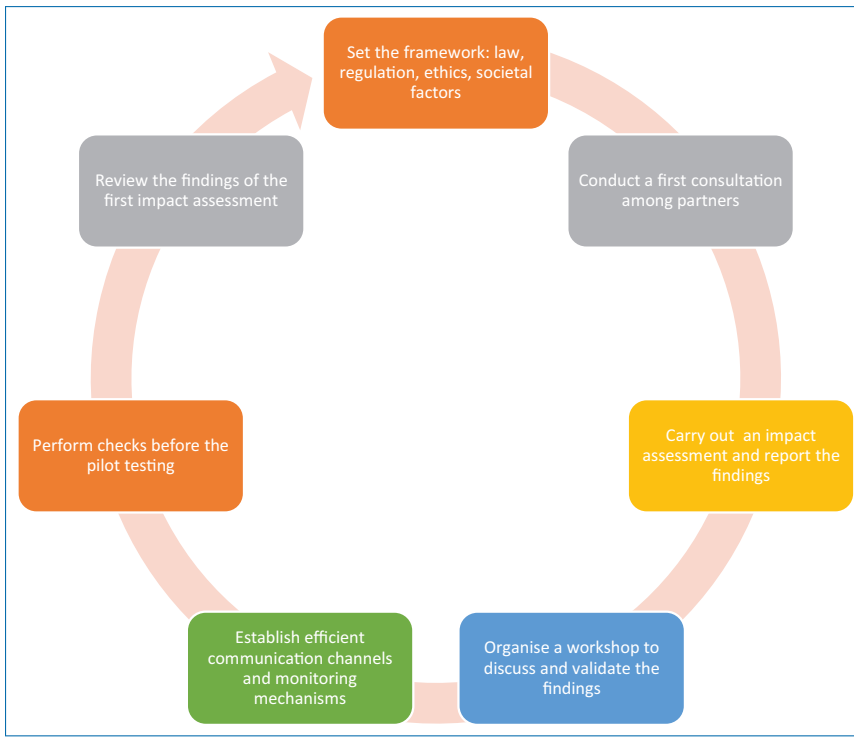
a. with regards to the intended processing after the research, if the system is marketed: as is the case with many cybersecurity systems, when fully operational and deployed, personal data processing may take place on a large scale. This processing quite often will occur with the use of innovative technological solutions. In the Cyber-Trust project, novel technologies include the use of machine learning, Artificial Intelligence and Distributed Ledger Technologies and aim to create a system beyond the current state of the art. Such technologies can involve novel forms of data collection and usage, which may entail a high risk to individuals' rights and freedoms. In addition to that, the system has a complex constellation of engaged actors (users and end-users), ranging from multiple data subjects to telecommunication providers and Law Enforcement Agencies.
b. Intended processing during the research: In the case of the web crawler, personal data might be processed without the provision of a privacy notice directly to the individual. Given that one part of the crawling service will be deployed in a real environment, with little human impact on the choice of websites and links that will be accessed, in particular with the use of Artificial Intelligence, the possibility to crawl even instantly personal data from publicly available sources is not remote. Even though in the Cyber-Trust context, the purpose of the collection is neither the identification and profiling of individuals nor the collection of personal data as such, in the Guidelines of the European Commission concerning ethics and data protection in the Horizon 2020 projects, the use of web crawling is considered as raising ethical concerns and thus, a DPIA is listed as an appropriate tool for the identification of risks and of potential mitigation measures.

2.4 Existing Guidance

The procedural steps intertwin with each other creating a net of information flows inside the consortium, useful for decision and policy making, and a knowledge hub for potential stakeholders who in the future may wish to deploy the system. The article will not present the actual analysis steps that are expected to take place during an impact assessment. As a context dependent process, this can only be defined in case-by-case settings.

Moreover, there is a lot of guidance concerning the substance of an impact assessment. The Article 29 Working Party published in 2017 guidelines on Data Protection Impact Assessment to enable the common interpretation of Article 35 GDPR. National Supervisory Authorities of EU Member States have also published guidelines and templates to assist the data controllers, data processors as well as researchers and manufacturers to document and assess the on-going, planned or envisaged data processing operations. For instance, the French authority (CNIL) has a repository with guidance on its website and even a dedicated software [1]. The Brussels Laboratory for Data Protection & Privacy Impact Assessments at the Vrije Universiteit Brussel has additionally published a series of briefs on the data protection impact assessment process in different languages, providing interactive templates [2]. In principle, a specific methodology is not suggested in GDPR. This allows organisations to use any framework or methodology, as long as it *"describes the nature, scope, context and purposes of the processing; assesses the necessity, proportionality and compliance measures; identifies and assesses risks to individuals; and identifies any additional measures to mitigate those risks*."

2.5 The Seven Steps



2.5.1 First Step: Establishing the Legal and Regulatory Framework at the Start of the Project

The Cyber-Trust consortium is rather inter-disciplinary. Its partners come from academia, business, public administration and carry with them different backgrounds and experiences in: tech, cybersecurity, policy, law, ethics, industry, trade, telecommunications, law enforcement. Therefore, the first step is to bring all these partners to reflect upon the context in which a) the Cyber-Trust research will take place and b) the future Cyber-Trust system will be deployed. In the first few months of the project (first semester) and during the system conceptualisation phase, the partners explored thoroughly the impact of the legal and regulatory framework based on the very rough initial concept of the project. They did so by studying the EU regulation framework and the national laws applicable in the countries where the partners are based and are of utmost importance in case of future release of the system. In the Cyber-Trust context, i.e., in the cybersecurity context, what was particularly reviewed were the data protection and privacy laws, laws governing telecommunications, laws in relation to evidence with particular focus on electronic evidence, regulation in relation to cybercrime, and ad-hoc regulation or policy guidelines with respect to specific technologies deployed during the projects (DLT systems, machine learning, etc). This study led to two written reports [3, 4] establishing basic concepts and building up to complex and niche discussions. In this stage, other legal and ethics requirements were also settled by the consortium, for instance the involvement or appointment of data protection officers per participating entity, the preparation of templates, such as informed consent forms and information sheets for the participation in research and the processing of personal data, whenever necessary and so forth. Those requirements would differ from project to project.

2.5.2 Second Step: First Wide Consultation Among Partners to Define Together the Way Forward

In the beginning of the second semester, and after the partners had thoroughly studied the legal and regulatory framework, the first consultation among all technical partners took place. The key partners were identified with the help of the Project Coordinator and the Technical Manager. Those partners were invited to complete a brief questionnaire about the concept of the component they were developing. The main aim was to have a first impression of the desired design and gather concerns or questions thereof, that have emerged based on the study of the legal and regulatory framework. The result of this consultation was the drafting of a first set of general and more concrete recommendations to assist key partners further with

their concepts and designs [5]. During this period, a number of *ad-hoc* bilateral meetings took place. This process coincided with the discussions about the initial architecture and the partners assessed the need for an impact assessment. At this stage, the partners also proposed the impact assessment methodology and established its reporting procedure.

2.5.3 Third Step: Carrying Out, Completing and Reporting About the Impact Assessment

In parallel with the intense negotiations for the finalization of the system architecture, the partners engaged in an extensive dialogue about how to better incorporate the recommendations provided in Step 2, into their envisaged work. The partners were again invited to complete individual, tailor-made written questionnaires for their components, assessing each of them separately but also in the context of the overall system. In practice, the partners were invited to elaborate further on their initial concerns and questions, as well as to explicitly state the benefits of the proposed solutions.

Those questionnaires included open questions, common for all the components as well as specific questions, tailor-made for particular components. This exercise consists of two steps: first, the partners visualise the component they develop, their research needs, the data processing operations they plan and explain how they aim to remain compliant during the project, taking a look at the requirements of each data protection principle; second, the partners demonstrate how they envisage their component to correspond in general to data protection principles, in case of possible future commercialisation. In other words, the assessment referred: (a) the intended data processing which would take place during the project; and (b) to the intended data processing of a novel technological system which is likely to be used by different data controllers to carry out different processing operations.

Due to the disciplinary variance, the partners also created a glossary of often-used terms (for instance, what is a data subject, what is the difference between the right to privacy and the right to data protection, etc). The consortium was invited to ponder upon which information to collect and why, whether that information include any personal data and why those data are necessary for the purpose they have in mind, under which legal basis and for how long they plan or envisage to store those data.

Timing, precision and flexibility are key here: Although partners were provided with initial questionnaires, through continuous interaction some questions were refined and new questions were added or dropped. All questionnaires made clear from the start, in contact with the Technical manager and the Project coordinator, who is in charge of providing a response; in other words, the technical partners

having a leading role in the design of a particular data processing operation and the non-technical partners who should be consulted due to the weight of their expertise in the project. In some occasions, partners were encouraged to consult external experts and their own Data Protection Officers.

Depending on the system in question – as often will be the case for cybersecurity systems, the procedure of mapping all the data processing operations from the user interface until all the backend sources and databases, may be dynamic, lengthy, highly collaborative, rather interactive, intense and resource-demanding. This is why, it is advised to initiate it as soon as possible and in any case before the intended processing. It is to be noted that this procedure is not a one-time exercise but as living instrument will take place alongside the planning, development, validation and actual implementation phase.

The outcome of this initial process in the Cyber-Trust case was a written report, which consisted of summaries of all partners' responses, a set of guidelines per component, a data processing matrix per component and a risk assessment matrix per component and for the overall project. The full questionnaires as filled in by the partners were also added as Annex at the end of the written report, in case partners wish to search for a clarification or for details not included in the main report, in line with transparency requirements.

2.5.4 Fourth Step: Workshop to Discuss and Validate the Impact Assessment Outcomes

After the completion of the first impact assessment and the publication of the outcomes, an *ad-hoc* workshop was organised in plenary to discuss the impact assessment outcomes and draw attention to the key decision makers inside the consortium. The primary aim of the workshop was to reflect upon and clarify common misconceptions that were observed during the impact assessment procedure, to recall the legal and ethical requirements and ultimately to examine the substantial scope and outcomes of the first impact assessment and evaluate its procedural aspects. The workshop was also the starting point for the preparation of the consequent review of the impact assessment to be completed at the end of the project and coincided with the preliminary deliberation of the system workflows.

2.5.5 Fifth Step: Continuous Communication During the Development

From the beginning of the project and throughout its whole duration, the non-technical partners have been participating in regular managerial and technical meetings and have been monitoring the development process. All partners have been

encouraged to contact the legal partners when they have questions or concerns, and the legal partners in turn follow the legal and regulatory developments and provide updates when a change in a law with a potential impact for the Cyber-Trust system occurs, or new case law emerges. Multiple discussions among individual partners, the Technical Manager and the Project Coordinator, have led to the drafting of collective papers and books, investigating inter-disciplinary topics of global interest. Such topics include, but are not limited to: data protection by design for cybersecurity systems in smart homes, privacy preserving mechanisms in Distributed Ledger Technology systems, privacy and data protection in the Internet of Things ecosystem and so forth. Those initiatives do not only improve the understanding of the consortium towards complex issues, but additionally further advance debates in the field, mobilising the attention of researchers, stakeholders and citizens with the organisation of public seminars and events, as well as forming synergies with other research projects. Moreover, an important element in the Cyber-Trust project is that, in order to ensure that the impact with respect to the legal and regulatory framework will be effectively taken into consideration, the consortium has additionally established a number of so-called 'legal and ethics' Key Performance Indicators (KPIs). For example, the partners have to work towards the realisation of a specific KPI which establishes the minimum number of privacy-preserving measures the system should include by default.

2.5.6 Sixth Step: Check Before the Pilots

Before the pilots, key partners were invited to perform a final check that all conditions in relation to compliance were met. This includes having readily available important documentation, such as research participants information sheets and consent forms, resuming and completing communication with their Data Protection Officers or Ethics committees and receiving any kind of necessary permissions or authorisations as well as reviewing and finalising the data flows.

2.5.7 Seventh Step: Review and Second Assessment Report

Near the end of the project life cycle, a review of the impact assessment report is planned. The aim of the review is to assess the efforts of the partners to incorporate the outcomes of the first impact assessment during the design and actual implementation in pilot-testing, conduct a comparative risk assessment based on the initial risk assessment matrix and reflect upon any new issues which potentially emerged due to technical or regulatory updates, in the meantime between

the first and the second report. During the review, given the maturity of the pilot results, the consortium will first examine whether more components (compared to the first report) should be assessed or whether components which were excluded from the first report should be now assessed. During the review, the consortium will also aim to address issues during Step 4, for instance further improving the understanding between the technical and non-technical partners with the expansion of the established glossary and optimising the methodology. Targeted, tailor-made questionnaires will be used again at this stage and bilateral discussions with the partners will take place. The results will be compiled in a written report, which along with the technical documentation, will accompany the final Cyber-Trust platform in case of potential marketing. This documentation will allow interested stakeholders and future data controllers to understand the benefits and risks of the platform and perform their own assessment, having a solid basis as a starting point.

2.6 Lessons Learnt

Of paramount importance is planning ahead, starting early enough, including a first outline in the research proposal. Then, as this is a horizontal procedure, the proper tools and mechanisms (e.g., questionnaires, repositories, glossaries, reports) should be identified and used to keep the consortium informed and engaged throughout the project life cycle.

2.7 Concluding Remarks

To sum up, even though structures for an impact assessment may show similarities, for most part they remain tailor-made for each project or system and their particular needs, as well as for the decision making they correspond to. The same goes for the procedural aspects. As we saw, the procedural aspects of an impact assessment are equally important to the substance of it, with regards to its effective and efficient completion and regular review. Here we presented the procedural approach adopted by the Cyber-Trust project, which constitutes a complex cross-disciplinary system with diverse beneficiaries, breaking down into seven steps. In long-term, impact assessments can have further benefits, including broader compliance and assistance with demonstrating accountability and enhancing trust towards individuals and users.

Acknowledgment


This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] CNIL, *Privacy Impact Assessment (PIA)*, available at: https://www.cnil.fr/en/privacy-impact-assessment-pia

[2] Kloza Dariusz and others, *Data Protection Impact Assessment in the European Union: Developing a Template for a Report from the Assessment Process* (2020) 52, available at: https://osf.io/preprints/lawarxiv/7qrfp/

[3] Gkotsopoulou, O. and Quinn, P. (eds), *D3.1 Regulatory framework analysis,* August 2018, available at: https://cyber-trust.eu/wp-content/uploads/2020/02/D3.1.pdf

[4] Gkotsopoulou, O. and Quinn, P. (eds), *D3.2 Legal analysis of the use of evidence material*, October 2018, available at: https://cyber-trust.eu/wp-content/uploads/2020/02/D3.2.pdf

[5] Gkotsopoulou, O. and Quinn, P. (eds), *D3.3 Legal and ethical recommendations,* October 2018, available at: https://cyber-trust.eu/wp-content/uploads/2020/02/D3.3.pdf


cyper threat intelligence

DOI: 10.1561/9781680838350.ch3

Chapter 3

Cyber-Threat Intelligence

By P. Koloveas[*], *T. Chantzios*[†], *C. Tryfonopoulos*[‡] *and S. Skiadopoulos*[§]

University of the Peloponnese
[*]pkoloveas@uop.gr
[†]tchantzios@uop.gr
[‡]trifon@uop.gr
[§]spiros@uop.gr

In today's world, technology has become ever-present and more accessible than ever via a plethora of different devices and platforms ranging from company servers and commodity PCs to mobile phones and wearables, used for interacting with and interconnecting a wide range of stakeholders such as households, organizations and critical infrastructures. The volume and variety of the different operating systems, the device particularities, the various usage domains and the accessibility-ready nature of the platforms creates a vast and complex threat landscape that is difficult to contain. Trying to stay on top of these evolving cyber-threats has become an increasingly difficult task, and timeliness in the delivery of relevant cyber-threat related information is essential for appropriate protection and mitigation. Such information is typically leveraged from collected data, and includes zero-day vulnerabilities and exploits, indicators (system artifacts or observables associated with an

attack), security alerts, threat intelligence reports, as well as recommended security tool configurations, and is often referred to as *Cyber-Threat Intelligence* (CTI) and entails the collection, analysis, leveraging, management and sharing of huge volumes of data. In this chapter, we outline INTIME, a system that incorporates and extends current tools and techniques from the CTI life-cycle by providing a holistic view in the Cyber-Threat Intelligence process. Through this process the reader will be able to (i) identify a number of modern tools and technologies related to the CTI life-cycle mentioned above, (ii) detect issues and research challenges that are involved in the design of key technologies for pre-reconnaissance Cyber-Threat Intelligence, and (iii) plan follow-up activities that will allow the adoption of the latest advances in the field.

3.1 Introduction

Over the years cyber-threats have increased in numbers and sophistication; adversaries now use a vast set of tools and tactics to attack their victims with their motivations ranging from intelligence collection to destruction or financial gain. Thus, organisations worldwide, from governments to public and corporate enterprises, are under constant threat by these evolving cyber-attacks. Lately, the utilisation of Internet-of-Things (IoT) devices on a number of applications, ranging from home automation to monitoring of critical infrastructures, has created an even more complicated cyber-defence landscape. The sheer number of IoT devices deployed globally, most of which are readily accessible and easily hacked, allows threat actors to use them as the cyber-weapon delivery system of choice in many of today's cyber-attacks, ranging from botnet-building for Distributed Denial-of-Service (DDoS) attacks to malware spreading and spamming.

Trying to stay on top of these evolving cyber-threats has become an increasingly difficult task, and timeliness in the delivery of relevant cyber-threat related information is essential for appropriate protection and mitigation. Such information is typically leveraged from collected data, and includes zero-day vulnerabilities and exploits, indicators (i.e., system artifacts or observables associated with an attack), security alerts, threat intelligence reports, as well as recommended security tool configurations, and is often referred to as *Cyber-Threat Intelligence* (CTI). To this end, with the term CTI we typically refer to any information that may help an organisation identify, assess, monitor, and respond to cyber-threats. In the era of big data, it is important to note that the term intelligence does not typically refer to the data itself, but rather to information that has been *collected, analysed, leveraged* and *converted* to a series of actions that may be followed upon, i.e., has become *actionable*.

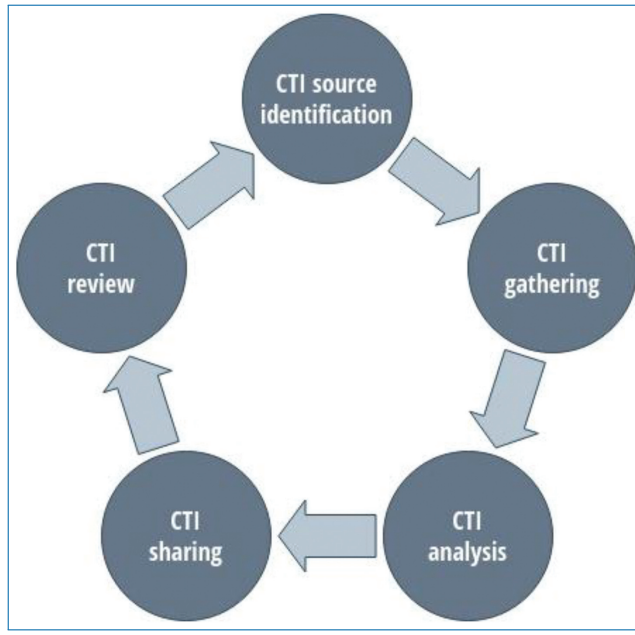


Figure 3.1. The CTI life-cycle.

The CTI cycle, illustrated in Figure 3.1, is the process of generating and evaluating CTI. The first step of this process is CTI source identification. It pertains to the identification of threat information that needs to be collected from monitoring devices, feeds, and security repositories to support decision-making and raise cyber-security awareness. The next step, namely CTI gathering, is the collection of the necessary data from the identified sources, along with the tools for extracting a wide variety of information, like tactical and strategic. This process is not a one-time action, but it is be performed in a continuous manner. The main goal at this stage is to collect as much information as possible and allow correlations and further analysis. The third step is CTI analysis and is built upon the information that has been collected; it includes both automated and human-driven analysis. The fourth step is CTI sharing to the relevant stakeholders, i.e., the entities that can utilize the generated intelligence, in a form that they find to be appropriate, useful, and in many cases actionable. This makes sharing highly-dependent on the audience (e.g., tactical, operational, and strategic level). CTI review (also referred to as CTI feedback), which is the last step in the above process, constitutes the key to the continuous improvement of the generated intelligence.

To support the CTI life-cycle outlined above, Koloveas *et al.* presented the INTIME [1]; an integrated framework for Threat Intelligence Mining and Extraction that encompasses key technologies for pre-reconnaissance CTI *gathering, analysis, management* and *sharing* through the use of state-of-the-art tools and

technologies. INTIME is an approach that *holistically* supports the complete CTI lifecycle via an integrated, simple-to-use, yet extensible framework and supports the task of gathering, consolidating and managing CTI from deep web forums or marketplaces and clear web social platforms, leveraging this information to identify emerging threats, zero-day vulnerabilities and new exploits to IoT devices. The main objective of this chapter is to provide an overview of the architecture and implementation of the various tools, methods and algorithms utilised, developed, and tested in INTIME. More specifically, we focus on INTIME's components that support:

- Deciding if a crawled website contains useful CTI; this is achieved by ranking the collected content to assess its relevance and usefulness to the task at hand.
- Extracting CTI from the collected content that was classified as useful, by resorting to state-of-the-art natural language understanding and named entity recognition techniques.
- Managing and sharing collected CTI via a combination of custom-made and widely adopted, state-of-the-art solutions that allow the exploration, consolidation, visualization, and seamless sharing of CTI across different organizations.

INTIME has been entirely designed on and developed by relying on open-source software including an open-source focused crawler, an open-source implementation of word embeddings for the latent topic modeling, open-source natural language understanding tools, and open-source datastores for the storage of the topic models and the crawled content.

3.2 INTIME Architecture

INTIME's architecture consists of three major components, namely (a) Data Acquisition, (b) Data Analysis and (c) Data Management and Sharing. The *Data Acquisition* module is responsible for the monitoring and crawling of various web resources. This task is achieved by employing traditional crawling and scraping techniques, along with machine learning-assisted components to direct the crawl to relevant sources. Although this module can easily extract information from specific well-structured sources, further analysis is required when it comes to the web content crawled from unstructured or semi-structured sources. To further analyse the gathered content, the *Data Analysis* module hosts two machine learning-based submodules, the *Content Ranking* submodule, which acts as an internal filter that ranks the data according to their relevance to the topic at hand, and the *CTI Extraction* submodule, that employs several information extraction techniques to extract useful

information from the webpages that were deemed relevant. The idea behind this two-stage approach stems from the inability of a simple crawler to accurately model the openness of the topic. The difficulty emerges from websites that, although relevant to the topic (e.g., discussing IoT security in general), have no actual information that may be leveraged to actionable intelligence (e.g., do not mention any specific IoT-related vulnerability). After the analysis, the extracted information is passed to the last module, named *Data Management and Sharing*, which hosts and dispenses all the Cyber-Threat Intelligence that the system collects. This architecture was initially developed by Koloveas *et al.* [2] and was focused on the crawling and ranking tasks. Later, it has been extended to its present state through the INTIME framework [1].

Noteworthily, Machine Learning and Deep Learning have a central role in our architecture, as the entire *Data Analysis* module is built upon Deep Learning techniques such as Word Embeddings (content ranking) and Named Entity Recognition (CTI extraction). Also, the *Data Acquisition* module utilises traditional Machine Learning algorithms to classify content gathered by the *Crawling* and *Social Media Monitoring* submodules. In Figure 3.2, every module where Machine Learning methods are present, is enclosed in dashed lines.

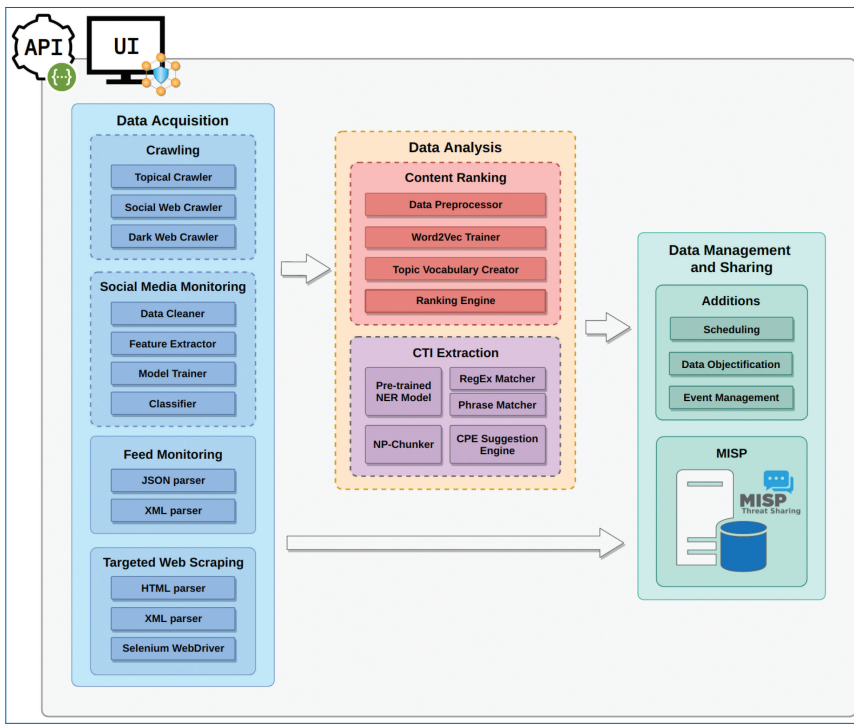


Figure 3.2. A high-level view of the system's architecture.

3.3 Data Acquisition Module

Currently useful cyber-security related information that may be leveraged to actionable intelligence may be found in a vast variety of different online sources ranging from technical and security-focused blogs in the clear web, discussions between experts in specialised security forums, social media content, to underground dark web hacker forums and marketplaces selling cybercrime tools and zero-day vulnerabilities and exploits. To cover this widespread need for data acquisition, our architecture provides a flexible yet powerful Data Acquisition Module that is conceptually separated into four distinct submodules:

1. The **Crawling** submodule allows users to easily setup and deploy automated data collection crawlers that are able to navigate the clear, social, and dark web to discover and harvest content of interest. The Crawling submodule allows the user to select between a wide variety of options including focused (also referred as topical) crawling directed by appropriate machine learning methods, downloading of entire domains based on powerful, yet easy to setup in-depth crawlers, TOR-based dark web spidering, and semi-automated handling of authentication methods based on cookie management. After collecting the content of interest, the users may then use rest of the modules provided by our architecture to further process it to extract useful CTI from it. The Crawling submodule is discussed in more detail in Section 3.3.1.
2. The **Social Media Monitoring** submodule allows users to monitor popular social media streams for content of interest; to do so it utilises publicly available APIs from social platforms and provides a pre-trained, ready-to-use set of classification algorithms that may be used to distinguish between relevant and non-relevant content. The Social Media Monitoring submodule is elaborated on in Section 3.3.2.
3. The **Feed Monitoring** submodule allows the users to monitor structured JSON or RSS-based data feeds from established sources such as NIST, while allowing them to modify several monitoring parameters like the monitoring interval and the type of objects they are interested in (e.g., CVEs, CPEs, or CWEs).
4. The **Targeted Web Scraping** module provides access to structured data from reputable sources that do not provide a data feed capability. Inclusion of such sources is out-of-the-box for the end user, however due the nature of the web scraping task, incorporating new ones includes a certain level of technicality. To support this process, our architecture offers a pre-installed set of tools that may be used to assist the programmer, including standard HTML parsing, XPath querying and JavaScript handling tools and libraries.

All the data extracted from the *Crawling* and *Social Media Monitoring* submodules, are stored in an internal NoSQL database (MongoDB), where they are analysed by the *Content Ranking* and *CTI Extraction* submodules of the *Data Analysis* module. Afterwards, they are sent to the *Data Management and Sharing* module in the form of structured CTI. Notice that data collected by the *Feed Monitoring* and *Targeted Web Scraping* submodules from the structured data sources are directly stored to the *Data Management and Sharing* module since they require no further processing.

In the following sections, we elaborate further on the submodules that were outlined above.

3.3.1 The Crawling Submodule

The Crawling submodule implements several distinct services that may be invoked by the users to initiate automated data collection on a wide variety of online sources in the clear, social, or dark web; the underlying crawling infrastructure is built on NYU's ACHE crawler.[1]

The *focused crawling* functionality uses the *SMILE Page Classifier* [3], which uses a Machine Learning text classifier, trained by a selection of positive and negative examples of webpages, to direct the crawl towards topically relevant websites (in our case websites with content relevant to cyber security). This functionality can also be assisted by the *SeedFinder* [4] sub-component, which can aid the process of locating initial seeds for the focused crawl; this is achieved by combining the classification model with a user-provided query relevant to the topic.

In-depth crawling is essentially a domain downloading operation based on the ACHE crawler that traverses a specific domain (like a forum or a website) in a breadth-first search manner and download all webpages therein. To direct the crawl to specific parts of the domain, *regex-based filters* are used; these filters provide blacklisting and whitelisting functionality to direct the crawler away from and towards respectively specific sections of the domain. In this way the user may instruct the crawler to avoid downloading non-informative pages (e.g., members areas, login or help pages) or to actively direct it to specific discussion threads in a forum.

Dark web crawling is also supported by INTIME's architecture. This functionality relies on the utilisation of TOR proxies to visit the user-specified onion links. Note that the user is not required to have any experience in this procedure, as all required actions (i.e., joining the TOR network, using the proxy, initialising the crawler) are fired automatically via internal API calls.

1. https://github.com/ViDA-NYU/ache

Any authentication issues that may arise during crawling are resolved via manual user login (the first time the crawler encounters an authentication barrier) and session cookie storage for all subsequent crawler visits.

3.3.2 The Social Media Monitoring Submodule

The Social Media Monitoring submodule focuses on real-time event detection from social streams using state-of-the-art tools from the data science domain to automatically classify posts as related or unrelated to a user-defined topic. To gather data from social media streams the submodule utilises the provided social platform APIs; the user is able to specify a set of social media accounts and/or a set of keywords that are of interest and the content collection mechanism will retrieve (in a recurring publish/subscribe fashion) all content posted from those accounts or matching the provided keywords.

Subsequently, the user is able to classify the retrieved content as related or unrelated to the task by simply selecting among various popular classification algorithms including (multinomial) Naive Bayes, K-Nearest Neighbors, decision trees, random forests, logistic regression, SVMs, as well as proven deep learning models like Convolutional Neural Networks. All classification and machine learning algorithms come pre-trained on real-world data and with default parameter setups for security classification tasks, but users may modify both the training data and setup parameters to fit their specific classification needs. The above process is streamlined to be usable out-of-the-box, but the advanced user may also customize all parts of the process, including content acquisition from social media without an API, introduction of other classification or machine learning algorithms, and task-specific algorithm training.

3.3.3 Submodules for Monitoring Structured Sources

Apart from the unstructured and semi-structured data that are gathered by the functionalities mentioned above, our system can also be supplemented by structured data from reputable sources of CTI. Such sources can be divided in two main categories. The first category provides structured data feeds of the information collections. The second category does not provide data feeds but exposes the contents of their database on web-based UIs in a structured manner. Our architecture provides functionalities to extract information from both categories.

For the first category, the system utilises standard JSON/XML parsing techniques with variable monitoring periods dependent on the data feed's update

frequency. Such sources include NVD[2] and JVN[3] vulnerability data stores, which provide their data in JSON and XML data feeds respectively. For the second category, several scraping techniques have been implemented, providing a flexible set of tools to account for the different types of websites where such information exists. These techniques range from standard HTML parsing and XPath querying, to sophisticated WebDrivers for automatic form manipulation and dynamic pop-up dismissal. Sources in this category include KB-Cert[4] and VulDB[5] vulnerability data stores and Exploit-DB,[6] which is a CVE-compliant archive of public exploits and the corresponding vulnerable software.

As previously mentioned, the data acquired from these types of sources are inserted directly into the *Data Management and Sharing* module without passing through the *Data Analysis* modules, since they are already structured in the desired CTI form.

3.4 Data Analysis Modules

Deciding if a collected website contains useful Cyber-Threat Intelligence is a challenging task, given the typically generic nature of many websites that discuss general security issues. To tackle this problem, we created an additional processing layer that initially ranks the collected content to assess its relevance and usefulness to the task at hand *(Content Ranking submodule)* and then attempts to extract actionable CTI from the highest ranked documents *(CTI Extraction submodule)*.

3.4.1 The Content Ranking Submodule

The idea behind our ranking approach was to represent the topic as a vocabulary distribution by utilising distributional vectors of related words; for example, a topic on IoT security could be captured by related words and phrases like “Mirai botnet”, “IoT”, or “exploit kits”. Such salient phrases related to the topic may be obtained by un-/semi-supervised training of latent topic models over external datasets such as IoT and security related forums. In this way, we are able to capture semantic

2. https://nvd.nist.gov/
3. https://jvndb.jvn.jp/en/
4. https://www.kb.cert.org/vuls/
5. https://vuldb.com/
6. https://www.exploit-db.com/

dependencies and statistical correlations among words for a given topic and represent them in a low-dimension latent space. To do so, we used Word2Vec [5]; a shallow, two-layer neural network that can be trained to reconstruct linguistic contexts and map semantically similar words close on the *embedding space*. Each word in the embedding space is represented as a *word embedding*. Those word embeddings can capture the relationship between the words in the dataset, making vector arithmetic possible. The above-described method, along with a method to map the words to our topic, which will be discussed later, could help us create a *Topic Vocabulary*.

Topic Vocabulary. To train the Word2Vec model, we had to create an appropriate dataset for the Content Ranking task. Our dataset had to contain the common vocabulary that is utilised when the topics of *IoT* and *Security* are being discussed. To capture this vocabulary, we resorted to a number of different discussion forums within the Stack Exchange ecosystem. To this end, we utilised the *Stack Exchange Data Dump*[7] to get access to IoT and security-related discussion forums including *Internet of Things*,[8] *Information Security*,[9] *Arduino*,[10] and *Raspberry Pi*.[11] The last two were selected because they are the most prominent devices for custom IoT projects with very active communities, so their data would help our model to better incorporate the technical IoT vocabulary. The utilised data dumps contain user discussions in Q&A form, including the text from posts, comments and discussion-specific tags in XML format. The posts and comments were used as the main input for the model.

On many cases, the words of the trained model were too generic or off-topic, thus, there was the need for a method that would remove those words, to create a smaller, more robust, topic-specific vocabulary. To do so, we utilised the extracted *tags* and augmented them with the set of N most related terms in the latent space for each tag. Table 3.1 shows an example of the most relevant terms to the *DDoS* user tag, for $N = 5, 10, 15$.

Ranking Engine. Since useful CTI manifests itself in the form of cyber-security articles, user posts in security/hacker forums, or advertisement posts in cybercrime marketplaces, it can be also characterised as distributional vectors of words. That way, we can compare the similarity between the distributional vectors of the

7. https://archive.org/details/stackexchange
8. https://iot.stackexchange.com/
9. https://security.stackexchange.com/
10. https://arduino.stackexchange.com/
11. https://raspberrypi.stackexchange.com/

Table 3.1. Most Relevant Terms for Tag "DDoS".

Rank	Term
#1	volumetric
#2	dos
#3	flooding
#4	flood
#5	sloloris
#6	denial_of_service
#7	cloudflare
#8	prolexic
#9	floods
#10	aldos
#11	slowloris
#12	Ip_spoofing
#13	loic
#14	drdos
#15	zombies

harvested content and the given topic to assess the relevance and usefulness of the content.

To do so, we employ the *Ranking Engine* sub-component. This component first creates the *Topic Vector*, by utilising the resulting Topic Vocabulary and then creates a *Post Vector* for each post entry in the crawled collection.

The *Topic Vector* $\vec{T}$ is constructed as the sum of the distributional vectors of all the topic terms $\vec{t_i}$ that exist in the topic vocabulary, i.e.,

$$\vec{T} = \sum_{\forall_i} \vec{t_i}$$

Similarly, the *Post Vector* $\vec{P}$ is constructed as the sum of the distributional vectors of all the post terms $\vec{w}_j$ that are present in the topic vocabulary. To promote the impact of words related to the topic at hand, we introduce a topic-dependent weighting scheme for post vectors in the spirit of [6]. Namely, for a topic T and a post containing the set of words $\{\vec{w}_1, \vec{w}_2, \ldots\}$, the post vector is computed as

$$\vec{P} = \sum_{\forall_j} \cos(\vec{w}_j, \vec{T})\vec{w}_j$$

Table 3.2. Relevance Score computation.

Excerpt from: www.iotforall.com/5-worst-iot-hacking-vulnerabilities
The Mirai Botnet (aka Dyn Attack) Back in October of 2016, the largest DDoS attack ever was launched on service provider Dyn using an IoT botnet. This led to huge portions of the internet going down, including Twitter, the Guardian, Netflix, Reddit, and CNN.
This IoT botnet was made possible by malware called Mirai. Once infected with Mirai, computers continually search the internet for vulnerable IoT devices and then use known default usernames and passwords to log in, infecting them with malware. These devices were things like digital cameras and DVR players.
Relevance Score: 0.8563855440900794

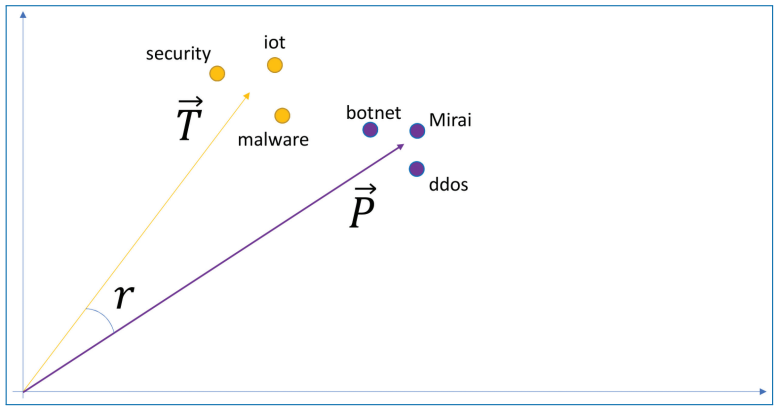


Figure 3.3. Theoretical visualization of the computation process.

Finally, after both vectors have been computed, the *Relevance Score* r between the topic T and a post P is computed as the cosine similarity of their respective distributional vectors in the latent space

$$r = \cos(\vec{T}, \vec{P})$$

Having computed a relevance score for every crawled post in our datastore, the task of identifying relevant/useful information is trivially reduced to a mixture of thresholding and top-k selection operations.

Table 3.2 displays an example of the process followed by the component. Figure 3.3 shows a theoretical visualization of the computation process.

3.4.2 The CTI Extraction Submodule

After the *Content Ranking* component decides which of the collected websites are more likely to contain Cyber-Threat Intelligence, our system has to be able to extract that CTI. To do so, we employ several mechanics such as Named Entity

Recognition with learned and Regex-based entities, Dependency Parsing to identify exploits, malware and vulnerabilities based on the structure of documents, as well as, a novel CPE suggestion engine for aiding semi-automated linking to known platform/vulnerability naming schemes.

Named Entity Recognition. The primary technique that was used for the task was Named Entity Recognition (NER). This technique can identify specific entities that have the potential to lead to CTI discovery.

Instead of training a NER model with entity-annotated data from scratch, a pre-trained model was used to detect generic entities that were not strictly limited to the topic of CTI.

To assist the pre-trained model on finding more entities related to the topic, a *Phrase Matcher* functionality was used. The Phrase Matcher can perform partial and full matches to unique multi-word phrases and map them to specified named entities. The phrases that we imported to the model were full names of companies/organisations and products extracted from JVN and were mapped to the ORG and PRODUCT entities (Table 3.3).

Apart from the entities that the pre-trained model was able to identify, several domain-specific entities were also introduced. These entities were inserted to the NER pipeline by defining Regular Expressions for each one, via the *Regex Matcher* functionality.

Table 3.3 shows the entities that INTIME is able to identify, along with the mechanisms responsible for the identification. Figure 3.4 shows some identified entities on a sample text.

CPE Suggestion Engine. In the previous section, we outlined the process of extracting Named Entities from unstructured text documents in an attempt to identify Cyber-Threat Intelligence. While this is an important task on its own, the extracted information is still largely unstructured and as the *Data Management and Sharing* module already contains large amounts of verified and structured CTI, a mechanism that would help security experts link the newly discovered CTI to existing events, would be beneficial to the entire CTI pipeline.

The most obvious entities that we could use to map the newly-found data to the structured CTI are "*CVE*" and "*CPE*". However, non-technical users do not tend to use these types of identifiers when they converse in the context of web forums, etc., so the likelihood of encountering them in significant enough numbers is low. Because of that, a hybrid solution was devised, the *CPE Suggestion Engine*, which we will describe below.

Although *CVE* and *CPE* entities are very rare in a free-text setting, *Product* entities appear with high frequency on the relevant gathered texts. Consequently, a product database was used to create a recommendation engine, which by utilising

Table 3.3. Supported entity types.

Rank	Term	Source
PERSON	People, including fictional	Pre-trained model
ORG	Companies, agencies, institutions, etc	Pre-trained model, PhraseMatcher
PRODUCT	Objects, vehicle, foods, etc.	Pre-trained model, PhraseMatcher
DATE	Absolute or relative dates or periods	Pre-trained model
TIME	Times smaller than a day.	Pre-trained model
MONEY	Monetary values.	Pre-trained model, RegexMatcher
CVE	Common Vulnerabilities and Exposures (CVE) identifier.	RegexMatcher
CPE	Common Platform Enumeration (CPE) identifier.	RegexMatcher
CWE	Common Weakness Enumeration (CWE) identifier.	RegexMatcher
CVSS2_VECTOR	Common Vulnerability Scoring System (CVSS) v2.	RegexMatcher
CVSS3_VECTOR	Common Vulnerability Scoring System (CVSS) v3.0-v3.1.	RegexMatcher
IP	IP address.	RegexMatcher
VERSION	Software version.	RegexMatcher
FILE	Filename or file extension.	RegexMatcher
COMMAND/ FUCTION/ CONFIG	Shell command/code function/configuration setting.	RegexMatcher

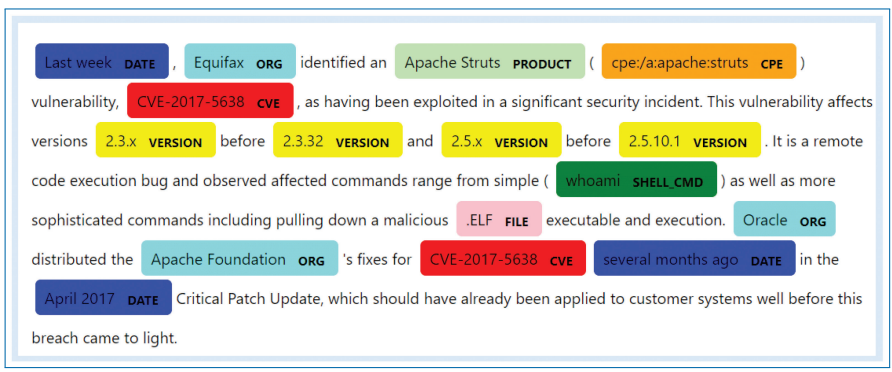


Figure 3.4. Identified entities.

text retrieval methods, suggests the most likely CPEs associated with a particular *Product* entity. Those suggestions get added to the object that gets sent to the *Data Management and Sharing* component, where a security expert can evaluate the suggested CPEs to see if they actually matched an existing Event, and subsequently, perform the linking of the Objects when necessary.

Suggestions were preferred against exact string matching on Product entities mainly due to the fact that in a free-text setting, a user might abbreviate a part of the product, use only the common popular name of it *(e.g., "Struts" instead of "Apache Struts")*, or simply make a spelling mistake. To present accurate suggestions, the problem was approached by performing *fuzzy text search*. To that end, the *CPE Suggestion Engine* uses n-grams, a common method for calculating text similarity. Initially, the n-grams for each product in the database get generated and indexed. Then, a query for each discovered Product entity is performed, and by using MongoDB's Text Search Operator, the module compares the similarity of the query's n-grams to the indexed n-grams. In the end, the top 10 results are returned, sorted by text match score.

NP-Chunking. For the final part of the CTI Extraction, a *Dependency Parser* was used to perform the task of "Noun Phrase Chunking" (NP Chunking). NP Chunking the subset of Text Chunking that deals with the task of recognizing non-overlapping text parts that consist of noun phrases (NPs).

While most of the CTI that we can expect to discover can be effectively modelled to the Named Entity Recogniser, some domain-specific concepts cannot be adequately defined as named entities. Such concepts include types of attacks and system vulnerabilities, exploit names, malware names, etc. They could be added to the Phrase Matcher as terminology lists, but due to the dynamic way that such concepts are described in non-technical texts, the effectiveness of the system would not be satisfactory.

After a thorough observation of the collected data, we discovered a common pattern, that these concepts are innately expressed as Noun Phrase chunks. For example, phrases such as "database injection vulnerability", "brute-force attack" and "privilege escalation exploit" are all NPs that can be classified as Cyber-Threat Intelligence, and we would not be able to identify them with our pre-existing infrastructure.

To this end, as part of our CTI Extraction module, we have implemented an NP Chunker that detects all the NP chunks found in a document and groups them in an object called **HIGHLIGHTS**.

For instance, on the document presented in Figure 3.4, the **HIGHLIGHTS** would be the following:

- "Apache Struts vulnerability",

- "remote code execution bug", and
- "April 2017 Critical Patch Update"

This method greatly assists the security experts to quickly identify whether a document contains actionable CTI, or link it to existing CTI objects from various sources.

3.5 Data Management and Sharing

In this section we present the *Data Management and Sharing* component, which is a full-stack solution, aiming to provide a complete proactive methodology for the tasks of CTI management and sharing. The component is able to store CTI from various sources, merge artifacts that concern information about the same CTI, and inter-correlate similar CTI. After storing all gathered CTI, the *Data Management and Sharing* component is able to present all stored information in human-readable formatting, through the MISP web-application. The interface enables users to further edit, analyze, and enrich the stored CTI. Finally, through the utilization of MISP, it enables the sharing of the stored CTI, in both human and machine-readable formats.

In the following sections, we provide an overview of the component architecture (Section 3.5.1), and a brief description of MISP (data model, sharing properties, and functionalities), in Section 3.5.2. Then, in Section 3.5.3 we present the MISP implementation and customizations within the *Data Management and Sharing* component. Finally, we describe the component's functionality in Section 3.5.4.

3.5.1 Component Overview

First, we have identified different CTI sources, which are vulnerability and exploit databases, containing analyzed CTI, in the form of vulnerability and exploit reports. These reports mainly consist of a plethora of useful and actionable intelligence about the vulnerabilities and exploits, such as a description of the vulnerability at hand, an exploit proof-of-concept, a list of the affected products' configurations (CPEs), metrics that provide an impact factor for the affected product (CVSS), publication and modification dates, references to similar reports, and a unique identifier that has been assigned to the vulnerability at hand (CVE ID). However, while the aforementioned sources often provide reports about the same unique CVE ID, these tend to differ. This happens due to the dynamicity of available information at the time of the analysis. Thus, analyses that occurred at a different time, may provide different metrics in the final reports. To overcome this

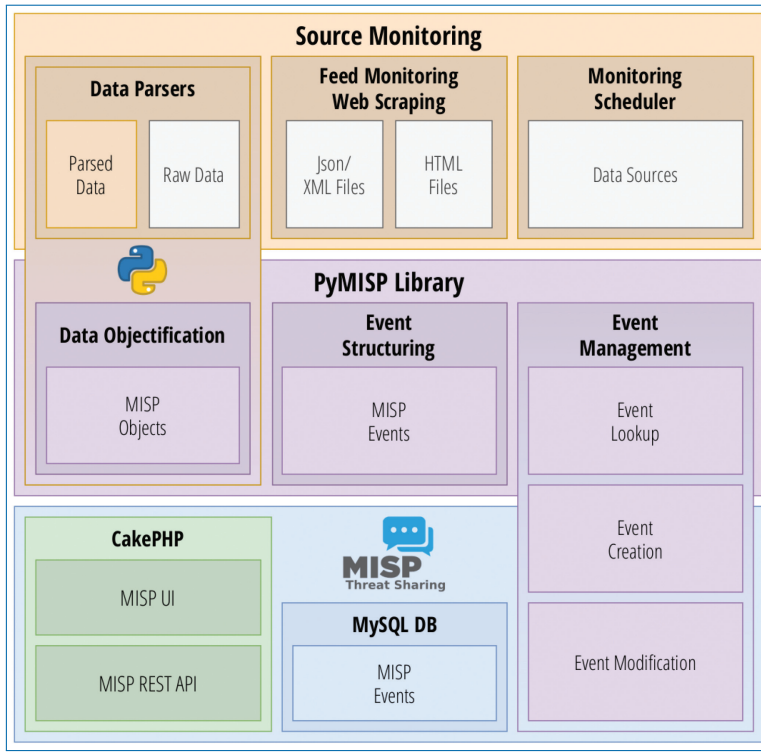


Figure 3.5. Data Sharing and Management architecture.

issue, we gather all publicly available reports from these sources, we parse them, one-by-one, to extract the CTI provided, using the *Feed Monitoring* and *Targeted Web Scraping* modules. Then, we store the parsed CTI, in a clustered manner, with regard to the unique CVE ID encompassed. The selected platform for storing and disseminating the gathered CTI is MISP. These clusters are called *events* in the MISP platform and the clustering of the reports occurs at the event management phase illustrated in Figure 3.5, in which we present an abstracted view of the component architecture. MISP provides the information stored in its database, in both human and machine-readable formats, and allows users to access it either through a GUI or via a REST API. Finally, MISP has implemented various tools, available in the GUI, that enable UI users to review CTI gathered and eliminate false positives or comment on the artifacts, and further analyze and enrich CTI through correlation processes.

3.5.2 MISP

As outlined in the literature [7, 8], MISP takes the lead in the platforms' race, as the most suitable platform for the purposes of the CTI life-cycle support. Thus, it is the

platform of choice for the CTI management and sharing of INTIME. Specifically, the *Data Management and Sharing* module uses, extends and enhances MISP, in order to enrich its storing capabilities with additional context. In the rest of this section, we will describe the essential details of MISP, that regard its (i) data model, (ii) CTI sharing properties and features, and (iii) its additional features.

The main objective of the MISP data model is to have a minimum viable format, which can be extended, according to the needs of additional complexity, instead of trying to capture all possible future requirements in advance. A new entry in MISP is called an *event* object, which is defined by a set of characteristics, along with all kinds of respective descriptions for indicators, including attachments. These characteristics are called *attributes* in MISP, and they provide all useful information to the event, such as an IoC date, threat level, comments, organization that created it, and so on. *Attributes* are mainly described by two fields: *category* and *type*. The main difference is that the *category* field describes what the attribute represents, such as network activity, financial fraud, while the *type* field describes how the attribute represents the chosen category. For example, an *attribute type* might be a checksum, a filename, a hostname, an IP-address, and so on. The actual payload of the attribute is stored in the *value* field.

Any CTI artifact, such as a CVE ID of a vulnerability, is stored in the MISP database in the form of *attributes*. Multiple attributes can be grouped to form an *object*, which forms a bigger CTI artifact, like a vulnerability report. Both attributes and objects must be attached to *events*, which basically serve as the records of the artifacts' storage. Finally, MISP enables an event to be correlated with other events, through matching techniques over their attributes. Each *correlation* that may occur between *events* serves as a bond, which also indicates the matching *attribute*. In Figure 3.6, we present an abstract overview of the database schema part, which is used for storing the CTI.

Specifically:

- The *events* table is a meta-structure scheme, where attributes, objects and meta-data are embedded to compose a sufficient set of indicators, that is able to describe a specific case, like a vulnerability report. An event can be composed from an incident, a security analysis report or a specific threat actor analysis. The meaning of an event derives solely from the information embedded within it. In our case, one event is a collection of objects that are used to describe the CTI artifacts.
- *Objects* serve as a contextual bond between a list of attributes within an event. Their main purpose is to describe more complex structures than can be described by a single attribute. Each object is created using an *Object Template*

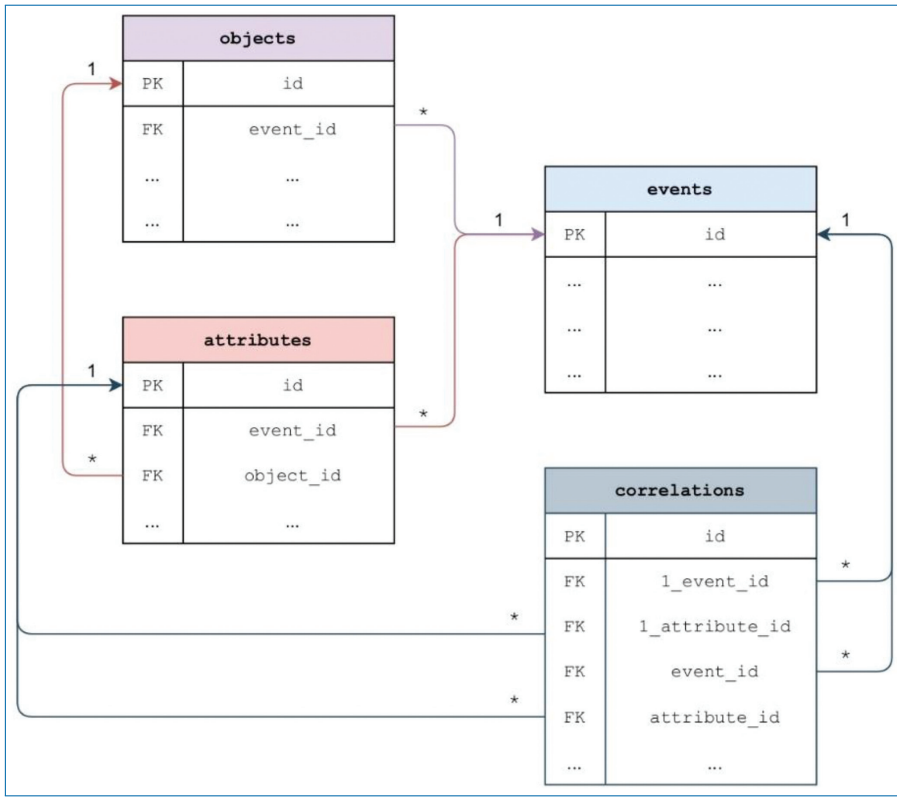


Figure 3.6. MISP database schema abstracted overview.

and carries the meta-data of the template used for its creation within. Objects belong to a meta-category and are defined by a name.

- *Attributes* are used to describe the indicators and contextual data of an event. The main information contained in an attribute is formed by *category-type-value* triplets, where the category and type give meaning and context to the value. Through the various category-type combinations, a wide range of information can be conveyed.
- *Correlations* serve as a bonding system between the stored events. Their main purpose is to describe any artifacts' matching that may have occurred between the events through the MISP Correlation Engine.

With regard to the sharing model of MISP, there are two main aspects. First, MISP enables its users to select the *sharing level* of the information stored in the MISP DB. For example, the sharer can disseminate the information at hand with a specific organization, a community of organizations, interconnected communities, all participants of MISP, or even define a sharing group manually. The next main

aspect of MISP, is the *proposals* feature. While the modification of events is only permitted to member of the creating organization, proposals allow users to make suggestions for changes to an event, created by another organization. A proposal is reported back to the original creator of the event, who may accept the change or discard it. Then, the outcome of the creator's decision will be propagated to all interconnected instances. An example of this feature is the reporting of false positives to the event creator, asking for an error correction. Finally, MISP is able to provide any information stored in its database, in both human and machine-readable formats, and allows users to access it either through a GUI or via a REST API, with respect to the aforementioned aspects of its sharing model.

Furthermore, MISP provides various complementary features, including:

PyMISP[12]: A python library for the implementation of MISP API. PyMISP provides users with fetching, adding, updating, deleting and searching capabilities over the stored events/attributes or *samples*.

The free-text import tool. It enables users to copy and paste raw data (in free-text format) into a single data field, that through a heuristic algorithm matches the attributes. The resulting attributes are then presented to the user who proceeds to validate the findings.

MISP tagging mechanism. It enables users to define customizable tags, through which they can later filter the events and classify the encompassed information. Furthermore, the tags can also be exportable, hence allowing the reusing of the same tags from other MISP instances.

MISP taxonomies. A *taxonomy* is a triplet of tags, which is described by a *namespace*, a *predicate* and a *value*. Through the utilization of taxonomies' repository, organizations have a common format for describing incidents. Furthermore, if the predefined taxonomies do not fit the description of an event, users can define their own.

MISP instances' syncing. MISP is provided with a synchronization protocol, which supports four main features; *pull*, *push*, *cherry-picking*, and the *feed* system. The *pull* feature allows a MISP instance to discover available and accessible events on a connected instance and download any new or modified events. The *push* mechanism allows a MISP instance to convert events to a JSON format that is transferable to remote instances. The *cherry-picking* feature is an alternative to the pull method, which allows users to decide which events should be pulled to the local instance. Finally, the *feed* mechanism allows a MISP instance to generate a dump

12. https://pymisp.readthedocs.io/en/latest/

of JSON files, which derive from a selection of events that an organization was to publish. Then, the output can be served via a web server, through which other MISP instances can access and retrieve the contents via the UI, similarly to the cherry-picking.

MISP sightings. MISP provides a sighting system, which allows users to react on attributes on an event. Originally, it was designed to provide an easy method for users to verify a given attribute, hence raising its credibility. Later, sightings have been improved to provide a method to signal false positives, but also to give an expiration date for some attributes. As stated previously, MISP Sightings are a way for users to state that they have seen or noticed an attribute and also confirm its validity. An attribute may be spotted several times by the same user, and thus a single user can use sighting several times on a single attribute. Sometimes, some attributes may be considered as false positives, and similarly to the previous case, users can signal a single attribute as a false positive several times. There is also the case of some attributes being valid for a certain period of time (for instance, in case of a phishing campaign that is assumed to be up for only one week). In this case, users can assign an expiration date to an attribute, but this time, there can only be one valid expiration date per organization of the MISP instance.

A particularly interesting additional feature of MISP is its correlation engine, which encompasses all the correlations between attributes and more advanced correlations like fuzzy hashing correlation (e.g., ssdeep) or CIDR block matching. Correlations can be both enabled or disabled, for each event per attribute. The *value* field of the attribute is the main payload of the attributes, which is described by the category and type columns, and it is used by the correlation engine to find relations between events. Specifically, after each event creation, the correlation engine of MISP scans through the database for matches of the event's correlatable attributes, with regard to their category and type. For each match, MISP proceeds to store two correlation entries in the database; one that points from the recently created event, to the previously stored, and one that points to the recently created event, from the previously stored, through their unique event IDs, along with their corresponding attribute unique IDs.

3.5.3 MISP Implementation and Customization

To fully accommodate MISP to our needs, we make use of the platform's provided tools to define custom objects that are able to fully encompass the CTI artifacts of the monitored sources. To best describe the artifacts that result from the parsing procedure of our system, we need to store them in MISP in the most suitable

objects; *vulnerability*[13] *and weakness*.[14] Additionally, MISP provides a method for creating custom MISP objects, which we use to create two custom objects for our component; namely, the *vuldb-vulnerability* and *expdb-poc* objects, which enrich the attributes of *vulnerability* and *exploit-poc*[15] objects respectively. Finally, we created one additional custom object *(crawled_obj),* that is able to encapsulate any possible artifact deriving from the *Crawling* and *Social Media Monitoring* submodules, as they derive from the *Data Analysis* and the *CTI Extraction* tasks.

Vulnerability objects describe CVEs, which refers to published, unpublished, or under review vulnerabilities for software, equipment or hardware. Specifically, *vulnerability* objects are able to describe CVE entries, with attributes that regard publication/modification dates, references, vulnerable configurations (in the form of CPEs), description and summary of the vulnerability, CVSS metrics, and of course, the CVE ID.

Weakness objects describe CWEs which refer to usable, incomplete, draft or deprecated weaknesses for software, equipment or hardware. CWE serves as a common language, a measuring technique for security tools, and as a baseline for weakness identification, mitigation, and prevention efforts. Such objects contain attributes that describe the corresponding CWEs, such as description, name, and status of the weakness, and the CWE ID.

The *vuldb-vulnerability* object is an enriched version of the *vulnerability* object, for CVEs. Particularly, it provides all proper attributes to store supplementary CTI parsed from vulnerability-oriented sources, such as the price estimations, CVSS strings from external sources (NVD, Vendor, Researcher), and exploitability and remediation statuses.

The *expdb-poc* object is a differentiated version of the *exploit-poc*[3] object, describing a proof-of-concept or an exploit of a vulnerability. This object has often a relationship with a CVE entry, via a CVE ID reference. The difference between *expdb-poc* and *exploit-poc* is that we created a *credit* field for *expdb-poc*. Furthermore, instead of downloading and storing all exploit proof-of-concepts, we point towards the link of the PoC raw code, through *references*.

The *crawled_obj* object describes CTI that may result from the *Crawling* and *Social Media Monitoring* submodules, through the *Data Analysis* and *CTI Extraction* procedures. First, the object stores several attributes that refer to meta-data about the crawling, such as the crawled document's *id, discovery timestamp, title, raw text,* and *source URL,* along with their corresponding *MD5 hashes*. Additionally,

13. https://www.misp-project.org/objects.html#_vulnerability

14. https://www.misp-project.org/objects.html#_weakness

15. https://www.misp-project.org/objects.html#_exploit_poc

it stores crawling meta-data like the *id* the *type* of the crawler that has discovered the document, a *relevance score* assigned to the document by the *Content Ranking* submodule, and a *highlight* identified by the *CTI Extraction* submodule. Then, the rest of the CTI artifacts deriving from the *CTI Extraction* submodule are stored in the corresponding fields of the defined object, and they may be *vulnerable configurations, CVEs, CWEs, organizations, products, versions and possible CPEs, CVSS metrics, files, IPs, commands, functions, configs, money values, dates* and *timestamps.*

Finally, all MISP Objects that were used in our system contain a *credit* field, which we used to store the source of the parsed CTI, using unique string identifiers for each source.

3.5.4 Component Functionality

In this section we describe the *Data Management and Sharing* component's functionality. Specifically, we describe the *source monitoring* procedure, which is responsible for periodically gathering CTI from our monitored sources. Then, we describe the data management procedure, which we particularly designed to (i) structure incoming CTI into the suitable objects *(object structuring),* (ii) check whether any incoming CTI is indexed by our component or not *(event lookup),* (iii) cluster objects into the corresponding CTI entries *(event creation),* (iii) manage updates and modifications of the stored CTI *(event modification).* Finally, we present the MISP functionalities implemented for the intercorrelation procedure of the indexed CTI (*events' correlations),* and the *CTI sharing and reviewing*.

Source Monitoring. During this phase, our system uses the *Feed Monitoring* and the *Targeted Web Scraping* modules, in order to extract the encompassed CTI. The monitored sources can be divided in two categories. The first category contains sources that provide structured data feeds (in JSON and XML formats) of their information collections. For this category, we use the *Feed Monitoring* module, which proceeds to extract CTI through JSON/XML parsing techniques. The rest of the monitored sources belong to the second category, which refers to sources that do not provide data feeds, but expose the contents of their database on web-based interfaces, in a structured manner. For this category we implemented standard scraping techniques like XPath querying and HTML parsing, through the *Targeted Web Scraping* module. The *source monitoring* procedure is executed with an adjustable monitoring period, which can be instructed in the *Monitoring Scheduler* module.

Object Structuring. After extracting all actionable CTI from the parsing procedures described in the previous section, our component proceeds to structure it in the format of the suitable MISP objects, in accordance to the objects described in

Section 3.5.3, with the use of the PyMISP library (as presented in Figure 3.5). To achieve that, the component generates the MISP objects in JSON format, as triplets of the attributes' *field, value, comment,* with the values extracted from the parsing phase. The *comment* field is used to store enriching information to the *value.* In example, declaring the source of a reference, whether it is from the affected vendor, or from another vulnerability notes' source.

Event Management. The event management phase executes in parallel to the source crawling and parsing phase as described in the previous section. What actually happens during this phase, is either the creation of new events, each time new CTI arrives to the *Data Management and Sharing* component, or the modification of previously stored events, due to updated CTI artifacts. This is also, the phase during which the clustering of the gathered CTI occurs. In the following sections, we describe the process followed in order to achieve that.

Event Lookup. First of all, in order to determine whether the CTI which arrived, is uncatalogued by the system or not, our component queries the MISP instance, with the CTI's unique identifier at hand. So, through the use of PyMISP, the component queries MISP, for any event that regards the currently parsed CTI's unique ID, by looking into the events' *info* field, which is used to store such identifiers. The result of the query can lead to two possible outcomes; (a) the parsed CTI ID is not already stored, and therefore a new event should be created, or (b) the parsed CTI ID exists, and therefore one or more existing events should be modified. For the second case, the component returns the corresponding MISP Event in JSON format, through PyMISP, and it also temporarily stores the corresponding MISP Event ID, as it is stored in the MISP instance.

Event Creation. If the parsed CTI is unindexed, then through PyMISP, the component follows a three-step approach, to catalogue it. First, it generates a new event in the MISP instance, setting the event's *info* field, to match the parsed unique CTI ID. Then, it generates the required MISP Objects (with regard to the specifications of each monitored source), from the constructed JSON structures of the object structuring phase. Additionally, the generated objects' validity is checked both locally, through the PyMISP library's objects' definitions, and externally, through a PyMISP request of the MISP instance objects' definitions. Both definitions must be the same for this step to succeed, and they are expressed in the form of JSON files, in the PyMISP library's files and the MISP instance's files. Finally, it attaches the generated MISP Objects to the event that was generated in the first step, on the MISP instance. An overview of the generated event through the MISP UI is presented in Figure 3.7(c).

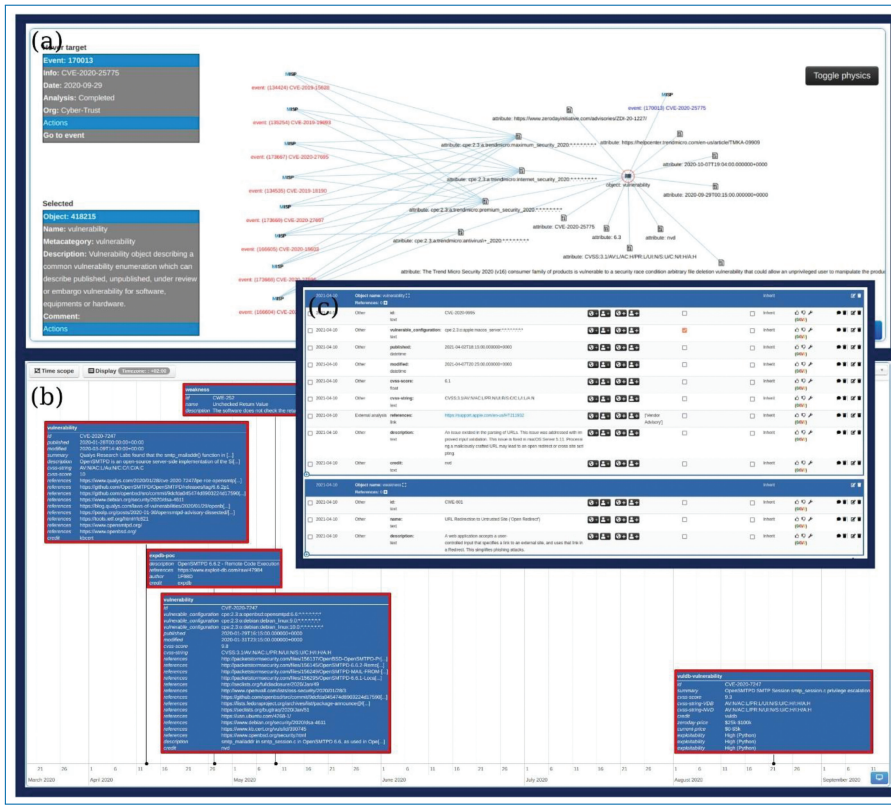


Figure 3.7. MISP events view; (a) events' correlation graph, (b) events' timeline, (c) events' objects and attributes view.

Event Modification. An event modification may occur in two cases; the system parsed CTI which is (a) unstored by the component, but it regards an existing CVE ID entry (which happens due to overlapping CTI from different sources), (b) an updated version of previously stored CTI. Any modification that occurs during this phase, makes use of the previously stored MISP Event, which derives from the *Event lookup* phase, through its CTI ID, that points on the MISP instance, through PyMISP, the event that is going to be modified.

Regarding the first case, the component checks the *credit* field of each object within the events at hand. If there is no match, it proceeds to generate the required MISP Objects, and then it attaches them to the existing event. To achieve that, the component checks the *credit* field of each object within the event at hand.

For the second case, similarly to the previous case, the component generates the corresponding MISP Objects and checks the *credit* field of each object within the event at hand. If there is a match, it proceeds to check the *modified* attribute of the matching object, which regards the *modification date* of the CTI encompassed. If

the *modification date* of the newly parsed CTI is more recent than the previously stored one, the component deletes the stored object, and proceeds to attach the newly generated object, to the MISP Event at hand.

Any modifications or additions of CTI artifacts in the MISP events, appear on the MISP event view, through the events' timeline (Figure 3.7(b)).

Event's Correlations. Finally, it is important to note that, after each event creation/modification, the component proceeds to recalculate the correlations, through the MISP Correlation Engine (presented in Section 3.5.2), since there is a possibility that the newly stored CTI may regard the same affected products, as other events. After this process, the event at hand points to all related events, as depicted in Figure 3.7(a).

CTI Sharing and Reviewing. In parallel to the gathering of all publicly available CTI from the monitored sources, our system is also able to proceed with the CTI sharing and reviewing phase. The sharing of the encompassed CTI may occur in two ways. The first, is to share CTI through the sharing features of MISP, as described in Section 3.5.2. The second method, is to query the component through the provided MISP REST API, using the required authorization credentials. In the following section, we provide a detailed overview of how this may be achieved.

MISP REST API: RESTful Searches. As mentioned earlier, MISP provides the option to search its embedded database, via the provided REST API. Moreover, it is able to export CTI in various CTI sharing standards such as JSON, XML, OpenIOC, Suricata, Snort, STIX, and more. Thus, it is possible to query the MISP REST API, for information regarding a specific entry, and receive a response in the requested format. For these purposes, there are two REST endpoints; one that regards information on event level, and one for the attribute level. In the first case, a user may retrieve all related CTI to the posed query, while in the second case, the user may retrieve all related attributes of the stored CTI, which match the posed query (e.g., a vulnerability's description). Both of these endpoints use the POST HTTP method to query the MISP REST API. Additionally, both endpoints enable users to pose constraints to the requested CTI, such as *dates*, *values (which may also contain wildcards with the use of the "%" character)*, *pagination of the results*, and more. Finally, MISP provides an automation functionality, which is designed to automatically feed other tools and systems with the data of the MISP repository. To make this functionality available for automated tools, an authentication key is used. Thus, in order to gain access to the REST API of MISP, the users should include their uniquely generated key (as a header in the POST request).

Figure 3.8. MISP Sightings mechanism as provided in the MISP UI on the events view.

CTI Reviewing Through MISP Sightings. To this end, for the reviewing of the encompassed CTI, the proposed component utilizes the MISP Sightings mechanism (described in Section 3.5.2), which allows users to declare whether an artifact is true positive or false positive, with regard to the vulnerabilities and exploits stored in MISP. The sightings mechanism for the reviewing of the stored CTI, can be used through the MISP UI on the events view, as highlighted in Figure 3.8.

3.6 Conclusions

In this chapter, we focused on facilitating the CTI life-cycle, by utilizing the appropriate open-source tools, for automating the CTI gathering and sharing tasks. We have presented INTIME, a solution that provides an end-to-end CTI management platform that is able to support the collection, analysis, leveraging and sharing of CTI via an integrated, extensible framework. We presented the architectural solutions behind the proposed system, discussed the individual module technologies and provided details on the module orchestration.

Acknowledgment

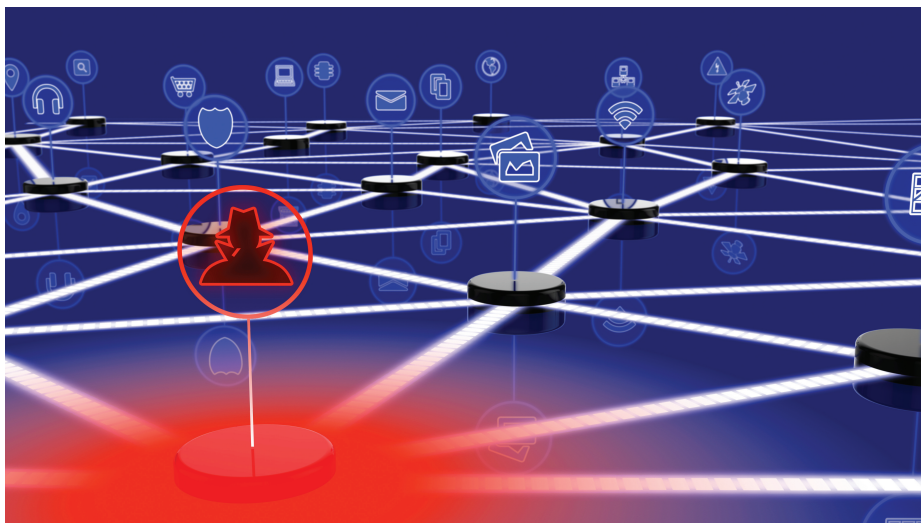
This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] P. Koloveas, T. Chantzios, S. Alevizopoulou, S. Skiadopoulos, C. Tryfonopoulos. *INTIME: A Machine Learning-Based Framework for Gathering and Leveraging Web Data to Cyber-Threat Intelligence.* Electronics 2021, 1, 5, March 2021, doi: 10.3390/electronics1010005

[2] P. Koloveas, T. Chantzios, C. Tryfonopoulos, S. Skiadopoulos. *A crawler architecture for harvesting the clear, social, and dark web for IoT-related cyber-threat intelligence.* In 2019 IEEE World Congress on Services (SERVICES), 2642, 3–8. IEEE, 2019

[3] H. Li. Smile. https://haifengl.github.io, 2014
[4] K. Vieira, L. Barbosa, A.S. da Silva, J. Freire, E. Moura. Finding seeds to bootstrap focused crawlers. World Wide Web 2016, 19, 449–474, doi: 10.1007/s11280-015-0331-7
[5] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. *Distributed Representations of Words and Phrases and Their Compositionality.* Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS, 2013, 2, 3111–3119
[6] J.A. Biega, K.P. Gummadi, I. Mele, D. Milchevski, C. Tryfonopoulos, G. Weikum. *R-Susceptibility: An IR-Centric Approach to Assessing Privacy Risks for Users in Online Communities.* In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016, SIGIR '16. pp. 365–374
[7] A. de Melo e Silva, J.J.C. Gondim, R. de Oliveira Albuquerque, and L.J. García-Villalba. *A Methodology to Evaluate Standards and Platforms within Cyber Threat Intelligence*. Future Internet, 2020, 12(6):108, doi: 10.3390/fi12060108
[8] T. Chantzios, P. Koloveas, S. Skiadopoulos, N. Kolokotronis, C. Tryfonopoulos, V. Bilali, D. Kavallieros. *The Quest for the Appropriate Cyber-threat Intelligence Sharing Platform.* Proceedings of the 8th International Conference on Data Science, Technology and Applications, DATA 2019, Prague, Czech Republic, July 26–28, 2019, 369–376, doi: 10.5220/0007978103690376
[9] C. Wagner, A. Dulaunoy, G. Wagener, A. Iklody. *MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform*. In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA; pp. 49–56.



DOI: 10.1561/9781680838350.ch4

Chapter 4

Moving–Target Defense Techniques for Mitigating Sophisticated IoT Threats

By K.-P. Grammatikakis, I. Koufos† and N. Kolokotronis‡*

University of the Peloponnese
*kpgram@uop.gr
†ikoufos@uop.gr
‡nkolok@uop.gr

Securing the constantly evolving IoT threat landscape is a challenging problem, with severe consequences when not tackled appropriately. In response to that challenge, the field of moving-target defense has developed, to address these threats by utilizing game-theoretic approaches to respond to them while maintaining a high level of availability. This work presents an implementation of an intrusion response system, which uses a Bayesian attack graph to model the complex state of the network and its hosts, and a partially observable Markov decision process to choose optimal mitigation actions. In order to cope with novel and unknown network attacks, like zero-day exploits, an alert management policy was added to focus the POMDP on the current state of the network and provide short-term mitigation actions. Finally, the system was evaluated against five scenarios (Mirai,

Zeus, zero-day, 10 malicious traffic replays, and BlackEnergy) executed in a simulated SOHO environment. Evaluation results showed its high effectiveness against traditional threats, and a slight increase in effectiveness against novel threats.

4.1 Introduction

In recent years, the constantly evolving threat landscape has seen an increasing number of cyber-attacks [1], with network-level attacks, botnets, and malicious software becoming more and more sophisticated over time. Furthermore, these well-understood threats were joined by zero-day attacks (i.e., the exploitation of undisclosed and unpatched vulnerabilities) which by their nature pose a greater threat to the security of computing networks due to the lack of information about them.

The detection of such threats is not trivial, because defenders often find themselves evaluating the security state of their networks through noisy information sources—like log servers (from which the distinction of security events from a torrent of insignificant ones may be difficult) or noisy alerts from intrusion detection systems (i.e., with an unacceptably high number of false positives/negatives). Current mitigation techniques, often relying on human intervention (i.e., incident response teams) or on existing network and host-based controls (e.g., firewalls or antimalware solutions), have proven to be inadequate in terms of coverage. Moreover, such solutions usually do not take service availability into consideration before acting—for instance, inaccurate firewall rule application during an attack may cause more damage than the attack itself, as the availability of critical systems or resources may be severely harmed. In addition, antimalware solutions often fail to protect against a large number of unknown or recent threats, while also requiring human interaction to apply mitigation measures.

More advanced defensive solutions have been developed with a twofold aim: to hinder the progression of an attack, and to gain a better understanding of the attacker's tools and methods. These solutions often interact with the attacker by changing the structure of the network, or present more attractive targets to distract from other network systems. For example, honeypots achieve this by deploying decoy vulnerable services, while honeynets deploy attractive-looking systems as red herrings to distract the attacker. However, even these fail against skilled attackers which are able to identify and avoid them.

Expanding on the idea of interacting with the attacker, *moving-target defense* (MTD) techniques were developed to optimally respond to adapting and complex threats. The main objective of MTD techniques is to affect changes to the network structure (or attack surface) in order to minimize the attacker's reconnaissance

ability, as well as to respond to threats while maintaining an acceptable level of service availability. The current landscape in game-theoretic MTD approaches [2–6] is quite promising but displays contrasting approaches in terms of attack modeling, with some of the works showing inefficiencies in either time efficiency, adaptability, or in the response selection options. Furthermore, many works assess their models mostly through simulation, which presents limitations regarding their real-world applicability. In an attempt to provide full coverage on possible attack scenarios, related works have a slow response time or employ inaccurate attack modeling methods when matching security threats to a variety of applicable environments (e.g., smart homes). In addition, most game-theoretic approaches do not handle the alert matching process, leading to inaccurate modeling of the network state. Although these approaches are developed to provide optimal responses in the long-term, without considering short-term responses, most common networks threats are unsuitably addressed.

Throughout the years, there was motivation to automate the attack mitigation process which led to the development of *intrusion response systems* (IRS). Initial attempts implemented static mapping between detected threats and available countermeasures [7] but lacked flexibility. This work presents the implementation of an IRS which leverages core functionalities of various *graphical network security models* (GNSM) to present a lightweight and efficient template for the application of decision-making processes. Also, the implementation of an effective method for calculating optimal short-term responses, so as to deal with momentary threats and zero-day vulnerabilities in *internet of things* (IoT) environments, will also be presented and evaluated against realistic attack scenarios in a simulated computer network.

The chapter is organized as follows: Section 4.2 presents the necessary background on MTD techniques and other related works; our IRS implementation will be discussed in Section 4.3. Two characteristic attack scenarios for IoT environments (namely, the Mirai botnet and a zero-day scenario) will be discussed in Section 4.4, while the experimental setup will be presented in Section 4.5. Finally, the evaluation results of the IRS will be presented in Section 4.6, while concluding remarks and future work are provided in Section 4.7.

4.2 Background and Related Work

MTD is a broad field encompassing techniques and mechanisms aiming to deceive an attacker by changing the network topology (by implementing shifting mechanisms) and utilizing any available event-based information to monitor malicious activity in the network. Lei *et al.* in [8] explain that MTD can be studied by

elaborating decisive elements that can measure the effectiveness of the implemented mechanisms.

Sengputa *et al.* in [9] indicate that MTD techniques are most advantageous when their implemented mechanisms are not deterministic, for the reason that attackers will ultimately be able to anticipate future shifting actions and calculate their attack strategies accordingly. The authors further discuss the implementation of MTD techniques, focusing on the network and application layers of the *open systems interconnection* (OSI) model. They note that MTD middlebox implementations, which take advantage of existing network devices used to manipulate network traffic (e.g., proxies, firewalls), are problematic due to their static nature and may even disclose information about the network to the attacker [10]. For that reason, they explain how advanced networking technologies, such as *software-defined networking* (SDN) and *network function virtualization* (NFV), can be used to add dynamicity to the MTD techniques. With the former technology, SDN, being the preferred approach in the area of MTD as a more scalable and effective solution, in addition to providing an optimized method for network mapping and multi-stage attack protection.

Cho *et al.* in [11] distinguish three broad MTD approaches: (a) game-theoretic, (b) genetic algorithm based, and (c) machine learning based. While all three are promising, their work focuses on a game-theoretic approach as it provides considerable advantages in terms of implementation flexibility, realistic modeling of the environments, and incorporation of diversified attack scenarios.

Zonouz *et al.* in [12], propose the usage of a *competitive Markov decision process* (CMDP) which is applied on a tree security model as an automated response and recovery engine that preserves availability. This approach presents a holistic solution that models the attacker as a rather intelligent entity, which avoids actions with a low payoff, but is lacking in scaling management and response time.

Shameli-Sendi *et al.* in [6] showcase an automated and interactive IRS which dynamically evaluates response actions with respect to network dependencies and critical processes, by constructing a static but flexible GNSM. The proposed model blindly triggers responses from the received alerts, which are evaluated according to the same security metrics (as defined for assets) to show, upon an attack, the negative impact of a response on different defense points. The limitation is that a response's positive impact computation is static and the security state is not updated when a response is applied. However, an accurate evaluation of responses is provided throughout the response process as their selection takes into account the attack damage cost, confidence level of the attacker and the probability of attack taking place.

Miehling *et al.* in [3], develop an autonomous system for the defense of attacked networks based on a *Bayesian attack graph* (BAG). A probabilistic model is implemented in order to capture the attacker's behavior when progressing through

the network. In their model, the defender is a partial observer, as the attacker's strategy is unknown, who tries to block the attacker's progress through the network by employing mitigation actions concerning network services. The authors describe this problem as a discrete time *partially observable Markov decision process* (POMDP) and consider both network attributes (services, vulnerabilities, etc.) and their representation in the GNSM (attack paths, belief state, etc.) of the decision problem, so as to successfully predict the future actions of the attacker. In [4], the authors present an IRS which takes advantage of dependency attack graphs so as to model a POMDP in a similar manner to [3]. This newer dynamic model is able to handle false alarms and quantify the attacker's progression while calculating long-term effective responses by simulating the effectiveness of decisions with a *partially observable Monte-Carlo planning* (POMCP) algorithm.

4.3 System Modelling

This section presents our proposed modeling for addressing current threats in smart homes, *smart offices/home offices* (SOHO) and IoT networks by taking advantage of graph-based models and their unique characteristics, so as to form a versatile framework for the application of MTD techniques. The IRS implementation is divided into two sub-components, as seen in Figure 4.1, the *attack graph generator* and the *decision-making engine*.

The high-level functionality of the IRS is as follows:

- Initially, the IRS receives information about the network topology from the gateway's *network discovery* module, including: host IP addresses, routing tables, subnetwork definitions, and any discovered vulnerabilities.

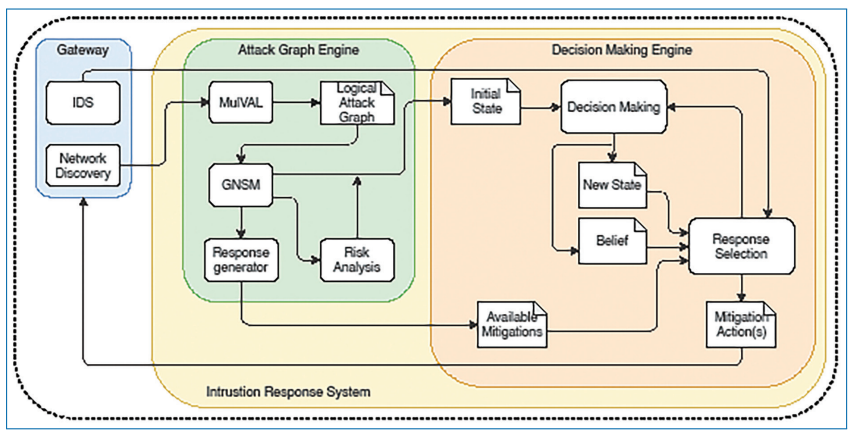


Figure 4.1. IRS high-level architecture.

- Then, the *attack graph engine* processes this information to generate the base GNSM, to perform *risk analysis*, and to pre-calculate all possible mitigation actions (firewall rules) by the *response generator*.
- This information is then forwarded to the *decision-making engine*, with the GNSM forming the initial state of the game-theoretic model and the pre-calculated mitigation actions being the defender's actions.
- Finally, network alerts generated by the gateway's *intrusion detection system* (IDS) are mapped onto the GNSM, which is analyzed by the *response selection* process and the appropriate mitigation action is selected.

4.3.1 Attack Graphs

GNSMs are widely used to model the security state of a network (or a host, depending on their application) using directed graphs, to identify possible *attack paths* (sequences of actions) an attacker may take to reach a desirable state (goal condition), and to perform more complex methods of risk analysis. These paths describe network states with nodes and state transitions with directed edges. These nodes are usually conceptualized to be either preconditions (capabilities an attacker must have to proceed further) or postconditions (capabilities an attacker can obtain, as long as their preconditions are met); capabilities include: acquired privileges, existing vulnerabilities, network attributes, or actions, among others. There are two major categories of GNSMs: *attack trees* and *attack graphs*; with the former [13] describing a single goal condition and every action required to reach it, and the latter describing multi-stage attacks that are not restricted to a single goal condition focusing instead on the attacker's actions rather than on the consequences of those actions.

Various attack graph-based security models have been proposed through the years with the most important being *state attack graphs* (SAG) [14], *logical attack graphs* (LAG) [15] and *Bayesian attack graphs* (BAG) [16]. While SAGs are better in terms of applicability, they scale exponentially in an attempt to cover all possible combinations of the attacker's moves, by not taking into account the generation of duplicate attack paths. LAGs describe logical dependencies among attack goals by employing nodes (facts) as logical statements and are considered a scalable solution for attack graph generation. BAGs are *directed acyclic graphs* where nodes represent random variables and edges depict conditional dependencies between node pairs; they are mainly used to conduct probabilistic risk analysis on networks characterized by rapid changes in their topology or host attributes.

The development of our IRS graphical model is based on the *Multi-host, Multi-stage Vulnerability Analysis Language* (MulVAL), a widely-used framework for producing LAGs in large-scale networks. Its logical dependencies describe how an

attack can be performed by considering logical facts as actions, translated into Datalog derivation sequences. Information about the network and discovered vulnerabilities are translated to Datalog tuples and processed by its internal XSB reasoning engine to produce the LAG. This model contains three node types: OR & LEAF nodes describing states of network devices (security conditions), and AND nodes which describe conjunctive relations between OR & LEAF nodes (exploits). Edges in this model connect preconditions to postconditions through exploit nodes. In the IRS, the MulVAL-generated LAG is converted to a BAG by conducting cycle elimination and by associating *common vulnerability scoring system* (CVSS) metrics with its edges.

4.3.2 Response Generation

Actionable remediation actions, which will be used by the *decision-making engine* and the POMCP model to modify the network topology depicted by the GNSM, are pre-calculated by the *response generation* submodule. These are firewall rules that change the inter-connectivity of hosts, both in and across sub networks, for the purpose of blocking access to vulnerable services or hosts.

The algorithm starts by selecting a node to be blocked (usually all exploits contained in the BAG) and, using *depth-first search* (DFS), explores the corresponding subgraph until LEAF nodes are reached. During this process, nodes are sequentially examined and all that contain enough information and depict access states, are taken into account for the creation of firewall rules and thus they are inserted into a tree structure. Additionally, each visited OR node is added to the tree as an AND operator (as every child must be invalidated to invalidate an OR node) and each visited AND node is added to the tree as an OR operator (as it takes only one child to be invalidated to invalidate an AND node). All paths that at the end do not represent such states, are terminated with a NULL node. MulVAL's Datalog rules are able to accurately describe detected services, as well as service-related information such as ports and IP addresses. Furthermore, all tree paths that are terminated with NULL nodes are removed from the tree to make processing easier, and the remaining paths are then collapsed to remove redundant operator sequences (*see* Figure 4.2). The remaining tree represents the solution in a *disjunctive normal form* (DNF).

$$(R_1 \cap \cdots \cap R_k) \cup (R_1 \cap \cdots \cap R_n) \cup \cdots \cup (R_1 \cap \cdots \cap R_m)$$

To manage the uncertainty that comes with unknown attacks, firewall rules blocking all services of each and every network host (global rules) are also generated. Although this solution is not considered optimal in terms of availability, there are multiple network-level attacks causing rapid changes to the network which

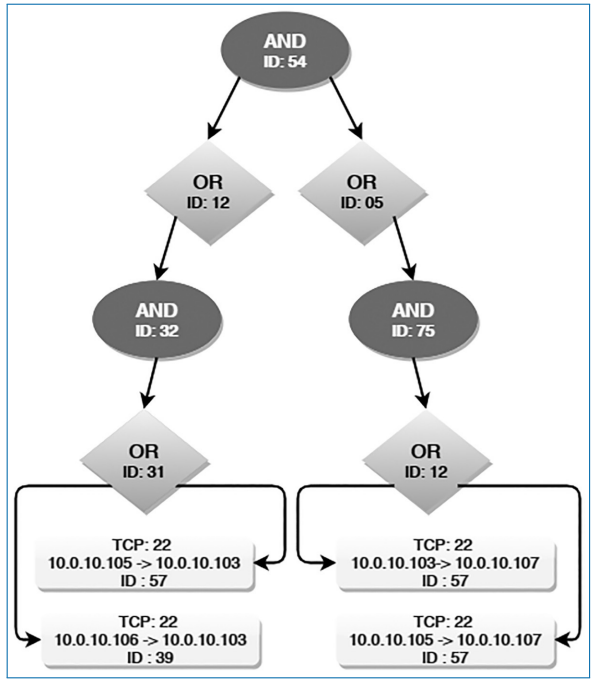


Figure 4.2. Sample tree from the response generation process.

the *attack graph engine* is not able to depict in the BAG in real-time. For example, attacks which communicate through dynamically assigned ports on the targeted network host—a common behavior to malware threats like Zeus which uses the OS-provided API to open a connection to its *command and control* (C&C) server [17], resulting in each communication attempt happening over a different port.

Finally, every solution is associated with a list of affected BAG nodes (that is, the nodes that will be considered invalidated/removed upon deployment of the rule) which is used by the *decision-making engine* to determine the impact of each solution, and chose the solution that optimally covers its belief about which nodes are believed to be exploited by the attacker.

4.3.3 Decision-making Process

The primary aim of IRSs *decision-making engine* is the choice of optimal mitigation actions, from the pre-calculated set received by the *response generation* submodule, in response to sophisticated network attacks. The game-theoretic model implemented is based on the POMDP model presented in [4] executed on top of the BAG generated by the *attack graph engine*.

This model describes a game between an attacker and a defender who is a partial observer, meaning that the attacker's strategy is unknown by the defender. In this game, the attacker aims to exploit vulnerabilities or execute other network attacks to progress through the network and reach a goal condition. The defender aims to block the attacker's progression through the network by selecting the proper mitigation action based on his belief about the network state (belief probabilities on the BAG) and the attacker type (a presumption on the attacker's strategy). Three types of attacker behavior are modeled, representing preset notions about the attacker's true strategy (aggression, knowledge level, and stealthiness).

Probabilistic metrics on exploitation-oriented decisions and actions, such as the probability of exploitation attempt and probability of exploitation success, are assigned by the *risk analysis* process performed by the *attack graph engine* on the base BAG. The execution of the POMDP model is performed in real-time, with each round (discrete time step) leveraging information received by the gateway's IDS to observe the attacker's actions on the network. This observation is the matching of the received alerts on the BAG's nodes (security states) which are considered to be reached by the attacker. Moreover, the decision-making process is based on a belief matrix which is the joint distribution over the security states and attacker types. The belief is updated every round in accordance to the defender's observation and is kept as a metric which bethinks in a recursive manner all previous decisions. All applicable solutions are pre-computed, allowing the optimal and fast execution of the required actions. The cost is the lowest when a firewall rule (or a set of rules, depending the circumstances) covers the widest node area in the BAG.

4.3.4 Further Adjustments

Originally, [4] describes a specific procedure regarding the selection of alerts to be triggered. Alerts are considered valid when exploitation-related preconditions are compromised. At the same time, the original model ignores any alerts that have corresponding postconditions compromised and then samples random alerts, filtered using the binomial distribution, according to their work.

In many occasions, the structure of the underlying GNSM significantly affects the attacker's development in the modeled network when alerts are received that way. Occasionally, the graph's goal conditions may be reached instantly or sometimes never at all, resulting in the absence of a mitigation action. To combat this, the implemented POMDP model employs an alert management policy, so that unknown attacks can be mitigated alongside traditional network attacks and exploitation attempts. This policy operates in two modes: *strict* and *agile*. For both modes, three sets of exploits (AND nodes of the BAG). The first is defined as the

set of *activated* exploits E_{ac}; the second is defined as the set of *available* exploits E_{av}, which includes exploits whose preconditions are compromised and whose belief exceeds a threshold; while the last is defined as the set of *blocked* exploits E_{bl}, which includes exploits that have been blocked by previous mitigation actions. Consequently, we define *strict* policy P_s as:

$$P_S = (E_{bl})^C \cap E_{ac}$$

and *agile* policy P_A as:

$$P_A = E_{av} \cap E_{ac}$$

Depending on the selected policy, alerts are matched to one of the aforementioned exploit sets, as long as there is enough available information from the IDS. The implemented POMDP model is focused on the current state of the network, allowing the IRS to better respond to attacks by providing short-term mitigation action responses when compared with other works, as its applicability does not extend to infinite horizon optimal planning. Attack paths depicted in the BAG are built with less actions in comparison to complex networks, thus it is not necessary to develop a system that attempts to think ahead of the adversary. To that end, the system's complexity is reduced by restricting the POMDP model to only one simulation round.

4.4 Attack Strategies

This work aims to address network-level threats and vulnerabilities relevant to IoT and SOHO environments. The devices of these environments are characterized by their variability of their operating systems and embedded technologies, which, when paired with the current rapidly evolving computing environment, allows for the creation of a multitude of attack vectors. Operations reliability, confidentiality, and availability are among the most important security goals to be considered in the context of securing such systems, especially as even moderate security controls are not implemented neither in host-level or network-level, and as their users are not properly educated on how to properly configure and secure them. Thus, in the current cyber-threat landscape these ecosystems are prime targets of large-scale attacks, including IoT botnets and Trojans. This section presents and analyses two characteristic attack scenarios associated with IoT systems and SOHOs, which will be further examined in the following sections through real-world scenario simulations.

4.4.1 The Mirai Botnet

The first threat of concern is IoT botnets. For evaluation purposes, the Mirai botnet was chosen, as at the peak of its activity became a wake-up call to the security industry [19], with an estimated number of 600,000 systems being infected at the peak of its initial breakout [20]. Infected devices became the source of one of the most severe cases of *distributed denial-of-service* (DDoS) attacks of the recent past, targeting the French web host OVH with a peak traffic size of 1.1 Tbps [21]. The disclosure of its source code, instead of leading to its eradication, significantly increased the number of attacks [22] and became the starting point for the creation of more resilient variants [19].

Mirai is comprised of four components:

- The *bot* executable, which is responsible for the infection through the usage of dictionary attacks, using common pairs of usernames and passwords, against misconfigured IoT devices.
- The *report server* which maintains the database of the botnet, handling incoming reports for infected devices and acts as one of the two intermediary entities between the C&C server and the bot. Bot and report server communication is achieved through the Tor network making its detection a challenging task.
- The *C&C server* is the central unit, providing a botnet management interface to the attacker while allowing the execution of infection and attack commands.
- The *loader* operates as another intermediary entity between the C&C server and infected devices, by sending malicious binaries to victims according to the server's infection commands.

The detection of Mirai is highly dependent on the utilized *network intrusion detection systems* (NIDS) for signature-based detection in the transmitted packets at the IoT environment. The attack can possibly be detected in three distinct actions: (a) during the infection of a new victim, (b) during the DDoS attack, and/or (c) during the transmission of a malicious binary between the loader and the infected victim. Regarding the DDoS attack, it must be mentioned that Mirai is able to use ten attack variations including HTTP flood, SYN flood, UDP flood, ACK packet flood, and so on. However, most of them can be easily detected by a NIDS.

During the execution of the Mirai attack scenario, the IDS at the gateway is expected to generate a number of alerts about the suspicious traffic, the IRS will process them to generate firewall rules to block the suspicious traffic. Depending on the alerts, the most suitable response will be determined by the POMDP model by formulating a strategy that does not only solve the problem but also

considers how every generated action will affect the availability in the SOHO or IoT environment.

4.4.2 Zero-Day Attacks

Zero-day attacks exploit not yet disclosed and unpatched vulnerabilities, who have no available countermeasures or known mitigation actions at the time of exploitation. Especially in IoT environments, the wide variety of communication devices (regardless of their operational technology) and their anticipated integration, constitute a complex and diverse system independent of human intervention—the latter resulting in security patches or mechanisms are not always handled as they should. As noted by [23], important features required in IoT applications, allow access to the entire network when exploited. The same holds with zero-day attacks in SOHO environments where vulnerable devices are present [24].

Similar to the Mirai scenario, detection of zero-day attacks heavily relies on the NIDS and its mode of operation. This type of exploitation is often accompanied with suspicious network packet payloads, thus rendering the detection process feasible to a certain extent. Nonetheless, the zero-day exploitation step does not often reflect the attacker's final goal, but rather the first step of a multi-stage attack (an attack path on the BAG). An attacker in this case, may just take advantage of any available vulnerabilities and pivot from host to host until the desired goal condition is reached. On the other hand, a more sophisticated attacker may take an alternative path with respect to speed and feasibility. Zero-day attacks are investigated by taking into account future weighted transitions for computing the belief metric of the corresponding attack state. Received alerts direct the IRS towards an optimal response that is related to the attacker's state in the graph, in accordance to neighboring exploitation nodes.

4.5 Experimental Setup

The IRS implementation described in previous sections, was evaluated in a realistic simulated SOHO environment in which the devices presented in Table 4.1 were included.

Respectively, a number of external devices are located in the WAN, from where the SOHO's gateway is reachable at *172.16.4.36*. The Mirai external core components (C&C, loader, and report server) are located in *172.16.4.21*, while the

1. https://github.com/budtmo/docker-android
2. https://sourceforge.net/projects/metasploitable

Table 4.1. Overview of the SOHO environment.

Device Name	IP address	Description
Gateway	*192.168.0.1*	In addition to its gateway functionality, it hosts a Suricata IDS instance along with the network discovery tools.
IRS	*192.168.0.3 & .4*	The two halves of the IRS implementation (attack graph engine and decision-making engine respectively).
DHCP	*192.168.0.7*	Dedicated DHCP stand-alone server.
Android Device	*192.168.0.9*	Docker-Android Image[1] running in an Ubuntu virtual machine.
Windows XP	*192.168.0.36*	General purpose Windows XP machine acting as an attack target (with service pack 3 installed).
Windows 7	*192.168.0.17*	General purpose Windows 7 machine acting as an attack target (with service pack 1 installed).
Metasploitable 2	*192.168.0.20*	An intentionally vulnerable Ubuntu device[2] designed for remote and local exploit testing.
BusyBox	*192.168.0.21 & .35*	A software suite implementing a number of basic Unix utilities commonly used on IoT embedded devices. Two instances are deployed in the same Ubuntu virtual machine as the Docker-Android device.

DDoS target located at *172.16.4.26*. In addition, the Zeus C&C server is located at 172.16.4.67—will be further discussed in Section 4.6.

4.5.1 The Mirai Attack Scenario

To further demonstrate the IRS evaluation procedure, the execution of the Mirai attack scenario will be presented in detail, while an overview is given in Figure 4.3. This attack scenario involves a Mirai-infected BusyBox host inside the SOHO network at *192.168.0.21*, communicating with the external Mirai components at *172.16.4.21* to perform a DDoS attack on *172.16.4.26*.

According to [19], the bot normally engages a dictionary attack against TCP ports 23 & 2323 (associated with the TELNET protocol) using a list of common default username/password pairs to establish a connection and gain shell access. This scenario begins with the x86/x64 bot binary being uploaded to the targeted BusyBox SOHO host, with the gateway's IDS and the IRS both operating normally.

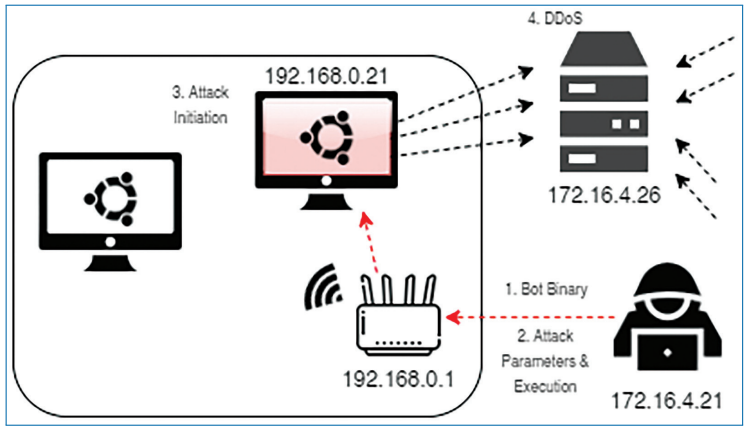


Figure 4.3. Mirai DDoS attack execution.

Generated information is reported back to the Mirai report server. At this point, the attacker may scan potential targets by sending ARP requests, in order to discover the SOHO's topology.

Afterwards, the attack begins with the attacker selecting the attack parameters and confirming the action. As mentioned in Section 4.4.1, there are a wide set of DDoS attacks that the attacker can choose from, in this case the SYN flood attack is used. The infected host will start attacking the external machine by repeatedly sending SYN packets, in an attempt to open as many TCP connections as possible and exhaust the target's resources.

In this demonstration, the SOHO is monitored by Suricata[3] signature-based IDS, thus the mitigation is dependent on the analysis of captured packets that pass through the gateway. However, the Mirai bot communicates with server-side components through Tor, making the detection process a difficult task. During the course of the attack, received alert messages of `event_type = alert` are consumed by the *decision-making engine*.

The IDS generated alerts for the three following actions:

- Target discovery using ARP packets.
- Attempted infections to LAN devices with a dictionary attack (*192.168.0.21*).
- SYN flood attack on the external target device (*192.168.0.21* → *172.16.4.26*).

These alerts initiated the IRS decision-making process which resulted in 120 different security states in the GNSM. In total, one response mitigation action was

3. https://suricata-ids.org

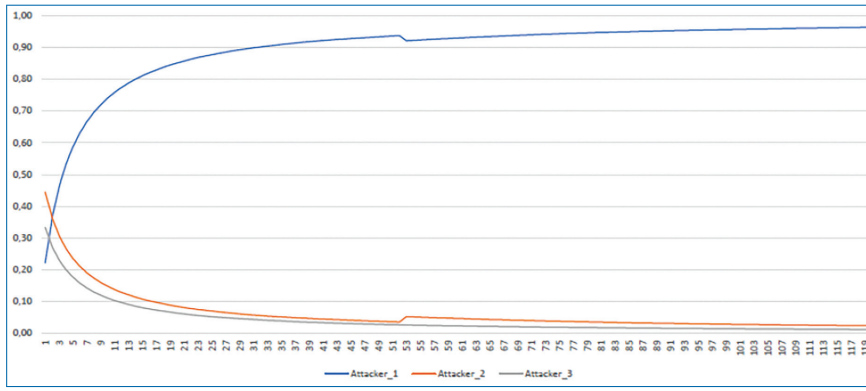


Figure 4.4. Attacker's beliefs for every security state.

selected, a global rule blocking all communications originating from the Mirai-infected host, that was able to prevent the DDoS attack:

```
iptables -A INPUT -s 192.168.0.21 -j DROP
iptables -A OUTPUT -s 192.168.0.21 -j DROP
```

Despite the fact that detection restrictions were encountered, the IRS became more certain of the states' beliefs and the attackers' beliefs over time, that resulted in the persistent block of the infected host located in *192.168.0.21* with the global firewall rule. The attacker type belief throughout the execution of the scenario is presented in Figure 4.4. Initial uncertainty about the attacker type can be seen in the leftmost part of the graph—because the attacker's intentions were not clear for the first few rounds. As the attack progressed, the attacker type belief quickly approached near-certainty, with the POMDP assuming that the attacker follows the behavior assigned to attacker type 1 (the least stealthy of the three). Similarly, the defender's belief on the security states updates with increased certainty. Respectively, the belief computation time is displayed in Figure 4.5.

Upon the application of the firewall rule by the gateway, the bot is restricted from further infecting new prospective victims in the SOHO, let alone take part in any DDoS attack. On top of that, the positive outcome that came with the previous response also prevents the attacker from communicating with the bot. Respectively, the bot is prohibited from sending relative reports, back to the report server.

4.6 IRS Evaluation

The IRS has been evaluated against five attack scenarios in total. The first one, the Mirai scenario, was described in the previous section. The remaining four scenarios

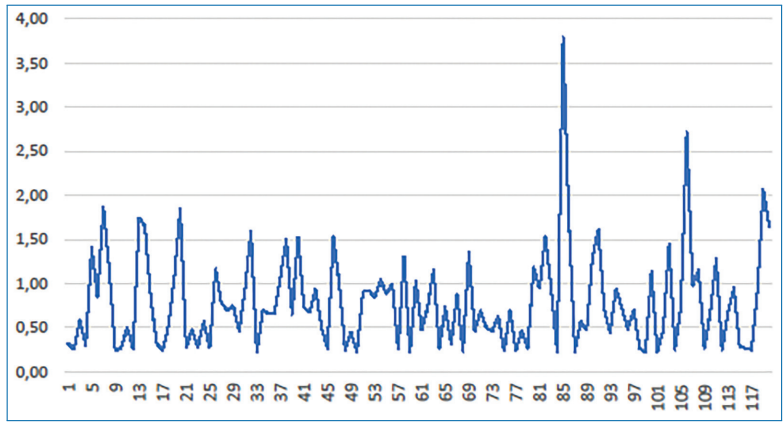


Figure 4.5. Belief computation time per security state.

include: a zero-day attack simulation of the vsftpd exploit[4] in the SOHO testbed, replayed network-level attacks, and the BlackEnergy & Zeus botnets.

Specifically, the zero-day vulnerability which has been exploited is a backdoor existing in vsftpd v2.3.4 binaries which opens a remote shell on TCP port 6200. This specific vulnerability is present in the Metasploitable 2 virtual machine that resides inside the SOHO and is triggered during the FTP login process when a specifically-formed username is entered. In order to effectively simulate a zero-day attack, all information about the vulnerability and its corresponding signatures are removed from all cyber-defence components.

During the testing and evaluation phase, datasets of pcap files have been generated from realistic malware traffic in the SOHO environment, including user enumerations, bruteforce attacks and Metasploit exploits. The complete list includes: (a) a Java-RMI backdoor, (b) a distcc_exec backdoor, (c) an UnrealIRCD backdoor, (d) a Web Tomcat exploit, (e) Ruby DRb code execution, (f) Hydra FTP bruteforce, (g) Hydra SSH bruteforce, (h) a vstfpd exploit, (i) SMTP User Enumeration and (j) a NetBIOS-SSN remote code execution vulnerability. Most of these attacks are carried out in the Metasploitable 2 virtual machine.

The third attack scenario is the Black Energy botnet, whose purpose is to initiate remote DDoS attacks. The malware hides its processes in system drivers and evades detection through obfuscation techniques. The chosen DDoS attack was a SYN flood attack which launched multiple synchronization requests to a SOHO external device. Furthermore, the botnet is managed through an external (to the SOHO) C&C server which is responsible for issuing commands. The target for

4. https://www.exploit-db.com/exploits/49757

the infection step was the Windows XP machine (*192.168.0.36*) of the testbed, while the transmission of the malicious executable was carried over HTTP.

The Zeus botnet is the last attack scenario of the evaluation; a widely known banking trojan whose main purpose was to capture credentials by web injects and keystroke logging, but also had the ability to form botnets. Each Zeus-infected host communicates with an external C&C server for periodic reports and when requested by the botnet operator, all communications are encrypted using the RC4 algorithm and happen over HTTP. The botnet has two distinct steps of detection [17] that lead to the generation of NIDS alerts: (1) the infection step, where the botnet initiates its communication with the C&C, and (2) the establishment of a TCP connection with the C&C server, over which the aforementioned reports are sent to the attacker. Moreover, in the last step, the attacker is able to execute commands on the infected machine (e.g. to capture a screenshot of the desktop, to download and execute other programs, etc.). The Windows 7 machine (*192.168.0.17*) inside the SOHO is the target of this scenario.

4.6.1 Configuration Options

Sixteen configuration options were evaluated on the aforementioned five attack scenarios to determine the effectiveness of a number of IRS's features; the most important ones being:

- The use of CVSS-based or pre-set metrics to calculate the initial host risk and the probabilities of exploitation attempt (from OR ? AND nodes) and success (from AND → OR nodes).
- The belief threshold at which exploit nodes (AND) are considered to be compromised by the attacker. More specifically this threshold controls which nodes will be included in the E_{av} set (*see* Section 4.3.4). Initially, all OR & AND nodes of the BAG are assigned a belief of 0, while LEAF nodes that represent an attacker's ability to execute code on a host are assigned a belief of 0.5 and all remaining LEAF nodes are assigned a belief of 1.
- Whether the response generator will produce both specific (targeting a specific port and protocol) and global firewall rules (blocking all connection attempts of a host), or whether it will be restricted solely to global firewall rules. This option effectively restricts the repertoire of remediation actions available to the defender.
- The alert management policy, strict or agile, which controls the alert matching process and whether the belief state of the IRS will be overridden by the reception of alerts (strict policy) or whether it will be taken into account (agile policy).

4.6.2 Evaluation Results

The evaluation of the IRS was performed against all five scenarios with each scenario repeated sixteen times, one for each configuration option. The results of the evaluation are summarized in the following Table 4.2.

The combination of a high compromised threshold and of the agile alert management policy of configurations #6, #8, #14 and #16 made them consistently

Table 4.2. IRS Evaluation results.

	Configuration				Scenario				
#	Metrics	Comprom. Threshold	Global FW Rules	Alert Manag. Policy	Mirai	Zeus	Zero-day	Replay	BlackEnergy
1	Pre-set (= 0.5)	0.5	True	Strict	✓	✓	×	9/10 ✓	✓
2				Agile	✓	✓	×	9/10 ✓	✓
3			False	Strict	✓	✓	*✓	8/10 ✓	✓
4				Agile	✓	✓	×	8/10 ✓	✓
5		1	True	Strict	✓	✓	×	9/10 ✓	✓
6				Agile	×	×	×	0/10 ×	×
7			False	Strict	✓	✓	×	9/10 ✓	✓
8				Agile	×	×	×	0/10 ×	×
9	CVSS-based	0.5	True	Strict	✓	✓	×	9/10 ✓	✓
10				Agile	✓	✓	×	9/10 ✓	✓
11			False	Strict	✓	✓	*✓	10/10 ✓	✓
12				Agile	✓	✓	×	10/10 ✓	✓
13		1	True	Strict	✓	✓	×	9/10 ✓	✓
14				Agile	×	×	×	0/10 ×	×
15			False	Strict	✓	✓	*×	9/10 ✓	✓
16				Agile	×	×	×	0/10 ×	×

✓ and × indicate that the attack was successfully and unsuccessfully mitigated respectively.

* indicates that a specific rule (targeting a specific port and protocol) was used to mitigate the attack.

unsuccessful. Because the high threshold did not allow for any exploit (AND) nodes being considered compromised, as none of the LEAF node beliefs managed to exceed it, and the agile policy did not override the belief to consider the nodes matched from the IDS alerts.

For the remaining configurations, regarding:

- The *Mirai*, *Zeus*, and *BlackEnergy* scenarios: alerts were matched to the BAG using the broadest criteria available, which forced the decision-making engine to choose global firewall rules no matter the configuration options. That is because these scenarios initiate communications over dynamically assigned ports, as they all use the operating system's API which opens a random port with each call. These changes are rapid enough that the GNSM generation process would have to be repeated several times per minute, which is not optimal nor currently feasible, so as to capture these rapidly changing ports on the resulting GNSM.
- The *zero-day* scenario: (a) for configurations #3, #11, and #15 IDS alerts were correctly matched to TCP port 5000 which resulted in the choice of a specific rule that blocked communications of all hosts with the router over TCP port 5000, and (b) for the remaining configurations alerts were received regarding the exploitation of the zero-day but were incorrectly matched to an entirely different part of the BAG, leading to the choice of incorrect mitigation actions; a result of the lack of information about the exploited vulnerability.
- The unsuccessfully mitigated replay scenarios were: (a) the Java RMI backdoor (failed twice), (b) the Ruby DRb code execution (failed once), (c) the SMTP user enumeration (failed five times), (d) the web Tomcat exploit (failed twice), and (e) the UnrealIRCD backdoor (failed twice). Again, during the execution of these scenarios IDS alerts were received, but as with the zero-day scenario, were incorrectly matched to the BAG.

4.7 Conclusions

Moving target defense is undoubtedly a field that includes many and different implementations addressing the same problem with diverse technologies and mechanisms. The defender-attacker battle is a never-ending game, signifying that foolproof security will never be accomplished in any system and especially in small and often unattended networks. Hereinafter, MTD attempts to provide a security defense framework with sufficient effectiveness.

In this work, a scalable solution that has been tested in a realistic SOHO environment and efficiently addresses the aforementioned situation was presented. The IRS presented in this work is based on a GNSM generated by the MulVAL framework which is converted to a BAG, to perform risk analysis and form the basis for the decision-making process. The decision-making engine implements the POMDP model presented in [4] with heavy modifications to better address unknown and network-level attacks. Among those modifications is the implementation of an alert policy that is able to consider threats throughout all GNSM's possible states.

To evaluate the effectiveness of the IRS implementation against realistic situations, like a Mirai botnet attack, five attack scenarios (Mirai, Zeus, zero-day, 10 malicious traffic replays, and BlackEnergy) were executed in a simulated SOHO environment. Sixteen IRS configurations were tested, so as to determine the optimal configuration, test the effectiveness of the aforementioned modifications, and to identify its limitations.

At the end, the system was highly effective against more traditional threats, such as Mirai, Zeus, and BlackEnergy, however its effectiveness against novel threats (i.e. zero-days), although slightly increased, is somewhat lacking. This work is a starting point for future works, as a number of limitations were identified from this process, including: a) the inability of IRS's GNSM to correctly model the state of a network with rapid changes to its topology (e.g. by including newly connected devices) or to host attributes (e.g. new opened ports); and b) the incorrect matching of IDS alerts to the GNSM observed during the zero-day and some of the replay scenarios—the cause of many effectiveness penalties during the execution of these scenarios.

Acknowledgment


This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] Enisa, "ENISA Threat Landscape 2020: Cyber Attacks Becoming More Sophisticated, Targeted, Widespread and Undetected," 2021. [Online]. Available: https://www.enisa.europa.eu/news/enisa-news/enisa-threat-landscape-2020.

[2] C. Lei, D. Ma and H. Zhang, "Optimal Strategy Selection for Moving Target Defense Based on Markov Game," *IEEE Access*, vol. 5, pp. 156–169, 2017.

[3] E. Miehling, M. Rasouli and D. Teneketzis, "Optimal Defense Policies for Partially Observable Spreading Processes on Bayesian Attack Graphs," in *Proceedings of the Second ACM Workshop on Moving Target Defense*, New York, NY, USA, 2015.

[4] E. Miehling, M. Rasouli and D. Teneketzis, "A POMDP Approach to the Dynamic Defense of Large-Scale Cyber Networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 2490–2505, 2018.

[5] H. Maleki, S. Valizadeh, W. Koch, A. Bestavros and M. van Dijk, "Markov Modeling of Moving Target Defense Games," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, New York, NY, USA, 2016.

[6] A. Shameli-Sendi and M. Dagenais, "ORCEF: Online response cost evaluation framework for intrusion response system," *Journal of Network and Computer Applications*, vol. 55, pp. 89–107, 2015.

[7] T. Toth and C. Kruegel, "Evaluating the impact of automated intrusion response mechanisms," in *18th Annual Computer Security Applications Conference, 2002. Proceedings*, 2002.

[8] C. Lei, H.-Q. Zhang, T. Jinglei, Y.-C. Zhang and X.-H. Liu, "Moving Target Defense Techniques: A Survey," *Security and Communication Networks*, vol. 2018, pp. 1–25, 7, 2018.

[9] S. Sengupta, A. Chowdhary, A. Sabur, D. Huang, A. Alshamrani and S. Kambhampati, "A Survey of Moving Target Defenses for Network Security," *CoRR*, vol. abs/1905.00964, 2019.

[10] R. Zhuang, S. A. DeLoach and X. Ou, "Towards a Theory of Moving Target Defense," in *Proceedings of the First ACM Workshop on Moving Target Defense*, New York, NY, USA, 2014.

[11] J.-H. Cho, D. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. Moore, D. S. Kim, H. Lim and F. Nelson, "Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense," *IEEE Communications Surveys & Tutorials*, vol. PP, pp. 1–1, 1, 2020.

[12] S. A. Zonouz, H. Khurana, W. H. Sanders and T. M. Yardley, "RRE: A game-theoretic intrusion Response and Recovery Engine," in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, 2009.

[13] S. Bruce, "Academic: Attack Trees - Schneier on Security," 1999. [Online]. Available: https://www.schneier.com/academic/archives/1999/12/attack_trees.html.

[14] O. Sheyner, J. Haines, S. Jha, R. Lippmann and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proceedings 2002 IEEE Symposium on Security and Privacy*, 2002.

[15] X. Ou, S. Govindavajhala and A. W. Appel, "MulVAL: A Logic-based Network Security Analyzer," in *14th USENIX Security Symposium (USENIX Security 05)*, Baltimore, 2005.

[16] N. Poolsappasit, R. Dewri and I. Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, pp. 61–74, 2012.

[17] H. Binsalleeh, "On the analysis of the Zeus botnet crimeware toolkit," in *PST 2010:2010 8th International Conference on Privacy*, 2010.

[18] Enisa, "Threat Landscape for Smart Home and Media Convergence," 2015. [Online]. Available: https://www.enisa.europa.eu/publications/threat-landscape-for-smart-home-and-media-convergence.

[19] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, pp. 80–84, 1, 2017.

[20] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis and *et al.*, "Understanding the Mirai Botnet," in *26th USENIX Security Symposium (USENIX Security '17)*, 2017.

[21] D. Goodin, "Record-breaking DDoS Reportedly Delivered by >145k Hacked Cameras," Ars Technica, 29 September 2016. [Online]. Available: https://arstechnica.com/information-technology/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/. [Accessed 9 June 2021].

[22] Nexusguard.com, "Distributed Denial of Service (DDoS) Threat Report Q4 2016," 2016. [Online]. Available: https://www.nexusguard.com/hubfs/Nexusguard_DDoS_Threat_Report_Q4_2016_EN.pdf.

[23] V. Sharma, J. Kim, S. Kwon, I. You, K. Lee and K. Yim, "A framework for mitigating zero-day attacks in IoT," *ArXiv*, vol. abs/1804.05549, 2018.

[24] Enisa, "Security and Resilience of Smart Home Environments," 2015. [Online]. Available: https://www.enisa.europa.eu/publications/security-resilience-good-practices/at_download/fullReport.


THREAT

DOI: 10.1561/9781680838350.ch5

Chapter 5

Cyber-Threat Detection in the IoT

By J. Rose[*]*, M. Swann*[†]*, G. Bendiab*[‡] *and S. Shiaeles*[§]

University of Portsmouth
[*]joseph.rose@port.ac.uk
[†]matthew.swann@port.ac.uk
[‡]gueltoum.bendiab@port.ac.uk
[§]sshiaeles@ieee.org

The Internet of Things (IoT) ecosystem is composed largely of heterogeneous internet-based devices, which generate an enormous volume of data every day; this includes sensors, smart devices, and other industrialised modules. However, the complexity of the IoT ecosystem and the quantity of IoT devices available have dramatically increased the volume of both emerging and persistent security vulnerabilities from edge to cloud computing infrastructure, principally due to security problems arising from embedded devices and other legacy hardware. Further, with the emerging IoT technologies, malware campaigns and criminal motivations are increasingly exploiting these underlying services and existing vulnerabilities. In the Cyber-Trust project, we aim to address these security issues to support the growth of the IoT ecosystem while mitigating the resulting complexity and vulnerability when protecting IoT devices. This chapter presents an overview of the IoT devices profiling and threat detection solution proposed by Cyber-Trust to tackle the grand

challenges of securing the IoT devices' ecosystem. In addition, the effectiveness and performance of the proposed solution are in-depth verified, especially against botnets and Zero-day attacks.

5.1 Introduction: Background and Related Work

5.1.1 Major Cyber Threats to IoT

The growing adoption of Internet of things (IoT) technologies results in a more intelligent and connected world. According to the last IoT statistics [1], more than 10 billion active IoT devices in 2022. Further, it is estimated that by 2025, there will be more than 152,200 IoT devices connecting to the Internet per minute. The amount of data generated by these devices is expected to reach 73.1 ZB [1]. However, connecting this large number of IoT devices globally, most of which are readily accessible and easily compromised, allows hackers and malicious actors to use them as the cyber-weapon delivery system of choice in many of today's cyber-attacks, e.g., from botnet-building for launching distributed denial of service attacks, to malware spreading and spamming [2].

On the other hand, IoT devices are essentially resource-constrained in computation, battery power, intermittent connectivity, and network protocols. These limitations hinder the execution of complex security tasks and make them vulnerable to a range of attacks such as malware, data leakage, spoofing, disruption of service (DoS/DDoS), energy bleeding, insecure gateways, injections, ransomware and device hijacking [3]. Leading to significant security and safety concerns that could potentially put human lives at stake [3–5].

IoT security has been an increasingly prevalent topic during the last few years, especially with the increased security incidents involving smart connected devices. In this context, the Open Web Application Security Project (OWASP) IoT project[1]; which is a volunteer community of security professionals, works to investigate the most critical IoT vulnerabilities that hackers can exploit as a basis for all kinds of malicious behaviour, including distributed DDoS attacks, malware distribution, spam campaigns, phishing, fraud, data theft among many others. Furthermore, this project intends to help smart device manufacturers, developers, organisations, and customers better understand the ongoing IoT security risks and take appropriate actions to mitigate them. According to the last report released by the OWASP IoT project [6], the most severe IoT threats for 2018 are:

1. **Weak passwords:** according to the report, weak, guessable, or hardcoded passwords are the Achilles heel of IoT security. If login credentials are not changed from their default setting, a simple brute-force attack can be easily

used to compromise these devices and use them to launch large-scale attacks toward critical cyber-infrastructures.

2. **Insecure network services:** This is another big issue in IoT networks, whereby standard network services running on the devices, such as Telnet, SSH and insecure HTTP protocols, represent significant security issues that manufacturers have not considered. Each open port on a smart device provides a new opportunity for malicious actors to gain access to the device [7].
3. **Insecure interfaces:** Standard interfaces used to communicate with connected devices are not always secured. This includes web interfaces, cloud APIs and mobile interfaces. An insecure interface ecosystem eventually leads to the device compromise through vulnerabilities at this level, such as weak encryption, data filtering and weak authentication methods.
4. **Insecure update mechanism:** IoT security issues are related to the lack of secure update mechanisms, such as missing automatic updates as a feature and missing notifications of security changes. Therefore, IoT device manufacturers should provide periodic security updates/patching to guarantee the security of their devices.
5. **Usage of insecure and outdated components:** Some manufactures use off-brand devices and insecure software components/ libraries to build cheaper IoT devices. However, this practice also brings many vulnerabilities to end-users and creates an entry point for potential cyber-attacks. According to Symantec [8], supply chain attacks are a massive part of the threat landscape, increasing attacks by 78% in 2019.
6. **Privacy issues:** Insecure storage, processing, and disclosure of personal data without express consent can lead to many privacy issues and even compromise the safety of people in the physical world. Moreover, the privacy policy statements of some IoT service providers are unclear about the data collection and does not identify the system capabilities.
7. **Insecure data storage and transfer:** Usually, data collected by smart devices move across a network or retained in a third-party location (e.g., cloud storage). Thus, the potential for it to be compromised increases, especially with the lack of efficient encryption and access control to the device's sensitive data and transfer.
8. **Lack of devices management:** IoT management introduces a host of challenges related to security, where most devices connected to a network are missing efficient security management, such as a lack of system monitoring and update/patching mechanisms, which makes them attractive targets for cyber attackers.
9. **Insecure default setting:** Most IoT devices are shipped with an insecure default configuration and restricted modifications. However, keeping the

default settings such as default passwords will create serious security risks, not only to the device, but also to the whole network.

10. **Lack of physical hardening:** Physical hardening is one of the most critical aspects of IoT security as physical access can be disastrous to devices and allows potential attackers to gain sensitive information (e.g., embedded passwords), insert malicious code and even rewrite the device's firmware.

All these security issues and many others make IoT devices easy targets for hackers and malicious actors, even using them as means for massive cyber-attacks such as Distributed Denial of Service (DDoS) attacks [5]. Thus, there is a crucial need for new techniques specially designed for IoT environments to identify and mitigate potential IoT-related security attacks that exploit some of these security vulnerabilities. In the following sections, we present a comprehensive review of the latest designed techniques for IoT devices profiling and threat detection in IoT.

5.1.2 IoT Threat Detection Methods

Several studies have attempted to design new intrusion detection systems that can identify potential cyber-attacks in IoT networks in recent years. These techniques are classified into two main categories: signature-based and behaviour-based detection techniques. Signature-based methods are the simplest and most effective techniques to detect intrusions and cyber-attacks. They refer to datasets of signatures (or patterns) of known malware. A signature includes information (e.g., cryptographic hash) that can identify the malware (attack) [9] uniquely. The current activity of the network is compared against the signatures to identify potential attacks. If the network traffic signature corresponds to any one of the existing signatures, it is considered malicious, and further defence actions are performed [7]. These techniques provide 100% accuracy rates in detecting known attacks; however, they cannot detect unknown and new attacks (Zero-day attacks) which do not have corresponding signatures [9]. With this limitation, attacks use Obfuscation techniques to change the attack signature and avoid detection.

Anomaly-based detection techniques have been proposed to tackle the limitations of signature-based detection methods. These methods monitor the network activity against a defined set of requirements that refers to a baseline model for the expected behaviour of the network. Any deviation from this average profile will be considered an anomaly and initiate appropriate defensive actions. Anomaly-based detection techniques general start by collecting information that can differentiate the expected behaviour of the network from the abnormal one. Then, this information is used to train a machine learning classifier to detect potential attacks [9]. In this context, the predictive accuracy of many supervised and unsupervised learning

algorithms has been studied in several research works [10–13]. For instance, Verma Abhishek *et al.* [11] studied the performance of different supervised learning algorithms in securing IoT devices against DDoS attacks. The studied algorithms are Random Forest (RF), AdaBoost (AB), Extreme Gradient Boosting (XGB), Gradient Boosted Machine (GBM), and Extremely Randomized Trees (ETC). The experimental results showed that Multilayer perceptron (MLP) classifier using the features selection set derives from the features selection method, outperforms all other classifiers with 83% accuracy rate, 90% True Positive (TP) rate and 23% False Positive (FP) rate.

The effectiveness of deep learning algorithms has additionally been investigated in many research studies. These techniques give a new powerful paradigm that can automatically extract the required features to build the network profile from big data without being particularly programmed [14]. For instance, the Recurrent Neural Network (RNN) has been used in many research studies to model the network activities for intrusion detection in IoT [15], especially their two main variants, Long Short-Term Memory (LSTM) [16] and Gated Recurrent Unit (GRU). Furthermore, Convolutional Neural Network (CNN), which gained great success in images classification, has also been used in many intrusion detection methods for IoT networks [9, 17]. Results from many studies show that Deep learning can significantly improve the accuracy of intrusion detection. For instance, the proposed method in [17] has achieved an average accuracy of 98.9%. Another essential benefit of these techniques is that they can potentially identify Zero-day and unforeseen attacks; however, they have higher false-positive rates. Table 5.1 presents examples of the learning algorithms used in intrusion detection methods for IoT and the achieved results in terms of accuracy, FP and TP.

5.1.3 IoT Devices Profiling Methods

Generally, profiling of IoT devices refers to monitoring and recording data that can be retrieved from different sources (e.g., IoT devices, network assets) to characterise the personal behaviour of IoT devices connected to the network. In this context, the abnormal behaviour of IoT devices can be identified by comparing the current activities of the devices with an existing profile built from historical activities over a set period. If the current behaviour deviates sufficiently from the pre-defined normal one, it will be considered as a potential attack and initiates appropriate defensive actions [19]. Usually, the profiling process could be performed at both the IoT devices and the network level (i.e., network profiling) to retrieve information from the end-user devices and the network assets (e.g., gateways), respectively.

Several research works have presented proposals for profiling IoT devices by using different techniques such as sensor fusion and SDA with Cloud Services

Table 5.1. Populaire learning algorithms used in intrusion detection methods for IoT.

Study	Classification	Test Dataset	Best Results
V. Abhishek *et al.* [11]	Random forest (RF), AdaBoost (AB), Extreme Gradient Boosting (XGB), Gradient boosted machine (GBM), and Extremely Randomized Trees (ETC), Multilayer Perceptron (MLP).	• CIDDS-001, • UNSW-NB15, • NSL-KDD	MLP • Accuracy: 83%, • TP: 90%, • FP: 23%
K. K. Sai *et al.* [12]	SVM, Naïve Bayes, Decision Tree, Adaboost.	Sensor480 with 480 samples	Decision Tree • Accuracy: 100%
Z. Marzia *et al.* [13]	Radial Basis Function (RBF),	Kyoto 2006+	RBF • Precision: 90%
R. Bipraneel *et al.* [15]	Recurrent Neural Network (RNN)	NSL-KDD dataset	Accuracy: 89.00%
K. Jihyun *et al.* [16]	Long Short-Term Memory (LSTM)	KDD Cup 1999	Accuracy: 96.93%
G. Mengmeng *et al.* [18]	Feedforward Neural Network (FNN)	BoT-IoT dataset	Accuracy: 96.82%
V. Huong *et al.* [17]	Convolutional Neural Network (CNN)	IoT intrusion dataset with 357952 samples	Accuracy: 98.90%

to monitor the device usage and retrieve information about critical files, security status, including patching status and firmware integrity [2, 20, 21]. However, in this chapter, we focus on network-level profiling techniques. Network profiling refers to the process of monitoring and logging all network activity by recording information from the packet metadata such as source/destination IP of the packet, start time, duration, sensor identity, the used application-layer protocol [2]. IoT Network profiling can be performed in six principal areas, with open-source and commercial software that provide network operators with the tools necessary to understand, control and manage the networks under their control. The six principal areas, including examples of applications, are summarised in Table 5.1 [22].

As shown in Table 5.2, several open-source and proprietary tools can be used for network profiling and investigating potential cyber threats, such as SiLK (System

Table 5.2. Principal areas of network profiling with examples of tools [22].

Areas	Examples of Tools
Network Spoofing and Redirection	DNSMasq, Ettercap.
Executable Reverse Engineering	Java Decompiler, NET Reflector, IDA Pro, Hopper, ILSpy.
Web App Testing	Mitmproxy, Zed Attack Proxy, Burp Suite.
Active Network Capture and Analysis	Canape, Canape Core, Mallory.
Passive Network Protocol Capture and Analysis	Wireshark, SiLK, LibPCAP, TCPDump, MS Message Analyser.
Fuzzing, Packet Execution and Vulnerability Exploitation Frameworks	American Fuzzy Lop (AFL), Kali Linux, Metasploit, Scapy, Sully.

for Internet Level Knowledge)1, a highly scalable and robust toolset for capturing and analysing network flow data. In addition, proprietary tools such as NetFlow (Sisco), ntopng (ntop) and PRTG Network Monitor offer complete functionality for their respective commercial offerings.

Towards the same direction, the Internet Engineering Task Force (IETF) has introduced the Manufacturer Usage Description (MUD) specification for enhancing the IoT network security by preventing IoT devices from unrestricted access to the network and only allow them to connect to dedicated services [23]. For that, MUD requires that IoT manufacturers provide a behavioural profile of their devices. For instance, an IP camera may need to use DNS and DHCP protocols to communicate with a cloud-based controller and an NTP (Network Time Protocol) server. This information can be used to generate a device-specific access control list (ACL) that set restrictions on this device and, therefore, reduce the potential attack surface on the network. However, the MUD specification is still under development and so not implemented by manufacturers [23].

On the other hand, many research works have proposed different IoT network traffic profiling approaches [22, 24]. For instance, Jonathan Roux *et al.* [24] have proposed an intrusion detection approach for IoT based on radio communication profiling. The proposed solution targets cyber-attacks that may occur through wireless communications by profiling and monitoring the Radio Signal Strength Indication (RSSI) related to the wireless transmissions of the connected devices. This information is collected by the radio probes placed in the smart area (network). Then, a neural network is trained to classify legitimate and illegitimate areas in

which devices usually communicate within the smart place. However, the proposed solution is not fully implemented, and the paper does not provide information about its detection performances (such as accuracy, false positives and false negatives.).

In another work, Andrei Bytes *et al.* [24] have developed new software for automatic feature profiling of IoT devices. The device profile is built based on its technical capabilities such as device firmware, access mode of the device, network operation topology and wireless interfaces. This information is collected from different locations, including direct and indirect sources. The created profile is then used to categorise and compare IoT devices security-sensitive capabilities.

5.2 Cyber-Trust Detection Method

The main goal of the Cyber-Trust project is to propose an innovative cyber-threat intelligence gathering, detection, and mitigation platform to tackle the grand challenges towards securing the ecosystem of IoT devices. The proposed approach captures different phases of the IoT emerging attacks, before and after known or unknown (Zero-day) vulnerabilities. This chapter focus on the detection phase, which involves two main components: network profiling and intrusion detection.

5.2.1 Network Profiling Approach

The network profiling component, also known as the network repository, automatically scan connected devices on the locally available network for potential common vulnerabilities and currently running services. For each device connected to the network, the list of potential vulnerabilities is collated from the public dataset CVE Mitre1 and mapped to the available network services, which are discovered through network port scanning tools such as Nmap. This information is then used to create the device profile and other information about the routing information, the reported hostname, network flow, and topology. Based on the created profiles for each device, the network profiling component computes the out of bound network profile behaviour; this is calculated by the continual monitoring of the network traffic flow from each device across the network. It utilises rate informed heuristic profiling to create an expected throughput pattern for each device on the LAN that it is connected to. This profile is then compared against three different predefined profiles that refer to the network profile that is obtained by a packet capture that is refreshed hourly (HP, Hourly Profile), daily (DP, Daily Profile) and weekly (WP, Weekly Profile). The objective of utilising different profiles separated and refreshed by period is to provide a more accurate map of the network conditions that a device

would experience over time. Increasing profile accuracy and makes the system more adaptable to variable network conditions and varied device usage. The Rate Metric (RM) for these captures is calculated as follow:

$$RM = \frac{n}{t} \tag{5.1}$$

Where n is the total number of bytes transmitted, and t is the time of capture. The component can then take periodic network captures of the LAN traffic from the gateway, this new capture is then run through the same profiling system as the timed profile captures, and a new rate metric is calculated. Finally, a percentage difference (Δ) is calculated, comparing the rate profile of the new capture to each timed profile as follows:

$$\triangle = \frac{CRM}{PRM} \times 100 \tag{5.2}$$

Delta (Δ) is the percentage difference between CRM, the calculated rate metric and PRM, the profile rate metric. Suppose the delta value passes over a threshold value that can be configured per implementation depending on network volatility. In that case, the device's network activity is flagged as out of profile, and a re-scan of the network is initiated to re-scan for any possible actively exploited attack surface on the network. This process is fast but minimal in terms of network impact and will not degrade network performance, even on a small network, as the scan scale will increase or decrease in intensity automatically depending on scan timings and throughput. In addition, this threshold can be raised or lowered depending on if scanning is too frequent; the threshold can be increased on a dynamic, variable load network, for example. The traffic capture, stored in PCAP format that the network profiling component uses to calculate and profile each device, can then be transferred to the machine learning component to check the traffic for patterns that could indicate malicious traffic, including active attacks or ongoing exploitation. This profile can then be used to inform mitigation actions across the affected network.

5.2.2 Intrusion Detection Method

The Cyber-Trust project proposed a hybrid intelligent intrusion detection solution for appropriate and effective detection of malicious cyber threats at the host and network level. The proposed solution combines deep learning and image visualisation techniques to detect sophisticated and newly released cyber-attacks in IoT networks quickly. *Deep learning* is a powerful learning technique that has become progressively dominant in various fields, including intrusion detection. Several researchers have suggested the application of image visualisation to intrusion detection systems.

In this context, the Intel labs and Microsoft threat intelligence team collaborate on a pertinent research project called STAMINA (Static Malware-as-Image Network Analysis) [25], which converts binary input files into grayscale images so that a deep learning algorithm can process and classify them. This research project's primary approach is to convert the content of an input binary file into a simple stream of pixels and convert that into a 2D image that varies depending on aspects like file size. Then, a trained neural network classifier is used to analyse and classify the output image as legitimate or malware. The learning algorithm is trained on a considerable amount of real-world data (2.2 million PE file hashes) that Microsoft has collected from Windows Defenders installations. STAMINA has proven effective, with over 99.00% accuracy in classifying malware and a false positive rate slightly under 2.6%. However, it has its limits. For example, it works well with small files, but it struggles with larger ones.

In the Cyber-Trust project, we have proposed an innovative intrusion detection solution that converts network traffic into RGB images using the visual representation tool Binvis[1]. Then, the produced images are analysed and classified using different learning algorithms, including Residual Neural Network (ResNet50), Self-Organizing Incremental Neural Networks (SOINN) and MobileNet. Our approach was announced on the first of April 2018, which means two years before the announcement of the STAMINA project. The main idea of the proposed solution is presented in our research papers [7, 14] and [26].

Figure 5.1 shows the produced images from the network traffic by using the visualisation tool BinVis. First, the output image is created by assigning specific colours to each byte of the PCAP file and converted into a 2D image by using the clustering algorithm Hilbert space-filling curve. This conversion is performed on each byte depending on its ASCII character reference:

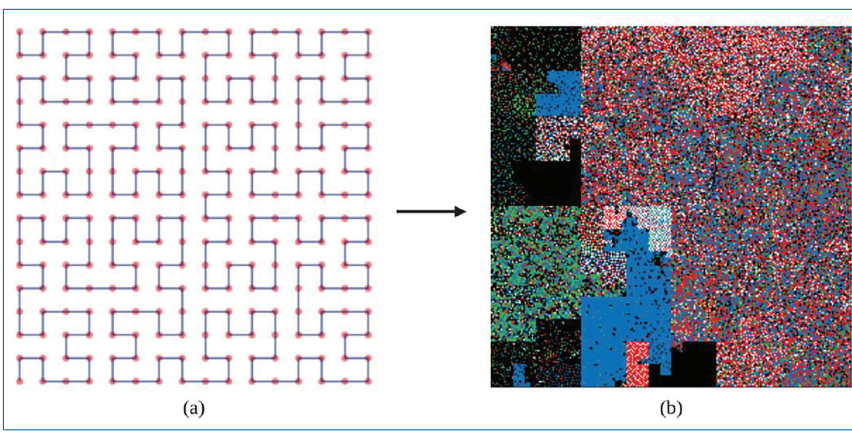


Figure 5.1. The Hilbert space-filling curve mapping and (b) the 2D image.

- Blue for printable characters
- Green for control characters
- Red for extended characters
- Black for the null character, or 0x00
- White for the non-breaking space, or 0xFF

5.3 System Implementation and Testing

5.3.1 Test Bed Setup

In the smart home domain, the experiments were carried out in the Cyber-Trust testbed, which involves a large number (750) of emulated and simulated Small Offices/Homes (SOHOs). Each SOHO includes several virtualized devices and a separate Ubuntu VM acting as a gateway. As shown in Figure 5.1, the network profiling component is deployed in the gateway VM because it needs to communicate with the smart home network (LAN) and collect information about the connected devices. Conceptually, this component may reside on the smart home gateway for data collection and communication or given the additional computational requirements, it may be relocated on a separate hardware device but closely connected to the smart gateway. The network traffic can indeed be collected from the LAN and WAN interfaces of the smart gateway and subsequently processed for storage using NetFlow. The network infrastructure is inferred using a combination of discovery mechanisms (Nmap specifically) and querying the services on the smart gateway (from ARP and DHCP leases to VLAN and routing information).

The intrusion detection component that includes the machine-learning detection module is deployed in another separate VM running Debian GNU/Linux 10.2 at the ISP level (WAN network). This component is deployed in a separate VM due to the computational power required by the machine learning module. For the virtualized devices, different Oss that are used in IoT devices were used in VMs or dockerized form. The smart home network configuration is done via the gateway VM, assigned two Interface Cards; from here, we control the network assignments for both WAN and LAN traffic. The interface card eth0 is referenced as NIC1 (172.16.4.1/24) and has Internet connectivity (WAN). In contrast, the second interface eth1 is referenced as NIC2 (192.168.1.1/26) and acts as a gateway IP for the smart home isolated network (LAN).

5.3.2 Test Dataset

To test the proposed detection approach, we have first created an initial dataset for training the machine learning module. However, the overall process of the machine learning algorithm training is performed incrementally each time new malicious

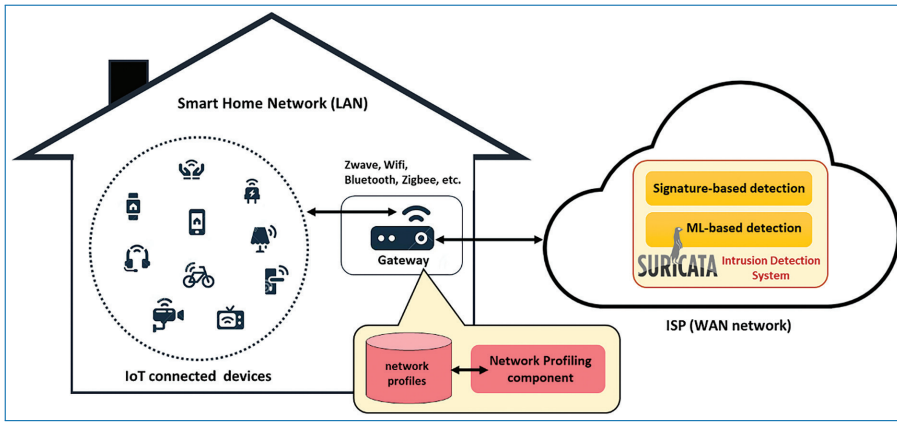


Figure 5.2. Implemented Testbed.

Table 5.3. Malicious traffic percentage according to type of attack.

				Other		
Malware Type	**Trojan**	**DDoS**	**Botnets**	**Zero-day Exploits**	**Backdoors**	**Others**
Percentage	33%	16%	19%	8%	6%	18%

traffic is found, that is, without ignoring the information identified in earlier training phases. This incremental learning significantly improved the detection accuracy of the machine learning module. This dataset consists of more than 900 BinVis images of malicious traffic sourced from multiple malware traffic analysis repositories1. Malicious PCAP files contain real malicious traffic generated by Trojans, Botnets, Keyloggers spyware and Backdoors, to mention a few. While standard PCAP files contain captured regular traffic from the Cyber-Trust project testbed from various clean devices in the network using tcpdump. The dataset of malicious PCAP files and their corresponding BinVis images is publicly available on the open-access IEEE DataPort website2. Table 5.3 shows the percentage of malicious traffic samples in the training dataset.

We have created our collection of PCAP files provided by real malware traffic in the Cyber-Trust testbed for the testing dataset. More precisely, malicious PCAP files were created from different real-world attack scenarios, including the Mirai Botnet, BlackEnergy Botnet, Zeus Botnet, and attack replay scenario, which consisted of several attack types Java-RMI Backdoor, distcc exec backdoor, Web Tomcat Exploit and Hydra Bruteforce attack. The PCAP files were generated by running live demos of each attack scenario and recording inter-device network communication using tcpdump.

Table 5.4. Metris used in the testing.

Metrics	Description	DDoS
Accuracy	Refers to the number of correctly predicted samples out of all the samples	$A = \frac{TP + TN}{TP + TN + FP + FN}$
False Positive Rate	Measures the rate of false alarms produced by the intrusion detection system.	$FPR = \frac{FP}{FP + TN}$
False Negative Rate	Measures the rate of non-captured attacks by the intrusion detection system.	$FNR = \frac{FN}{FP + TP}$
Precision	Measures the percentage of positively classified samples that are truly positive	$P = \frac{TP}{TP + FP}$
Recall	Recall represents the number of normal samples that were correctly classified	$R = \frac{TP}{TP + FN}$
F-Score	F-score is a weighted average between precision and recall	$F\text{-}score = 2 \times \frac{P \times R}{P + R}$

5.3.3 Testing Results

5.3.3.1 Machine learning detection module

Several tests were carried out to evaluate the success of the proposed intrusion detection solution and determine its accuracy. The metrics used to investigate the results of the ML module are Accuracy (A), False Positive Rate or false alarms and False Negative Rate. In these experiments, malicious traffic represents positive instances while normal traffic represents negative instances. True Positive (TP) is the number of malicious instances that have been correctly classified. False Positive (FP) is the number of normal instances that have been incorrectly classified as normal. True Negative (TN) is the number of samples of normal traffic that have been correctly classified. False Negative (FN) is the number of normal PCAP files that have been incorrectly classified as anomalous instances.

By processing these PCAP replays to the A04 component, we can assess these metrics with quantifiable data; the results of this testing resulted in the following overall statistics. Figure 5.3 presents the overall results of the tests, which reached an overall detection accuracy of 98.35%, which is a high rate and meets the required accuracy rate in practical use. By running the tests several times and over 100 runs,

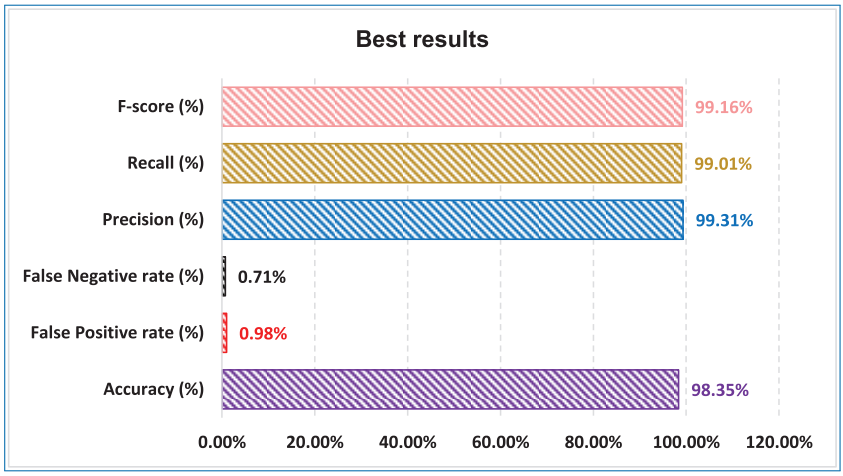


Figure 5.3. Overall testing results.

the best accuracy (A) result was 98.35%, the false-positive rate was 0.98%, and the Negative false rate 0.71%. The precision (P) result was also very high, with a rate of 99.3%, which shows overall solid confidence in the pattern recognition process. In these tests, precision is crucial because getting False Negatives (FN), when malware traffic is considered normal, cost more than False Positives (FP), when normal traffic is considered malicious traffic. The recall percentage (R) had a result of 99.01%. The F1 value (F1) achieved was 99.16%.

5.3.3.2 Network profiling

The proposed network profiling approach is used by the Cyber-Trust IoT platform to dynamically and actively profile and monitor all network-connected devices to detect IoT device tampering attempts and suspicious network transactions. During the tests performed on the proposed solution, the threshold is set to 80% of the percentage difference (PD) in the assigned 60 seconds of capture time. Such a significant difference from the standard transmission rate in any capture was a good baseline for our use cases. However, it is essential to note that the end-user can configure this threshold to match their network use cases if their network activity throughput is markedly more volatile or stable than the SOHO networks, we tested the configuration. As shown in Table 5.5, during the performed tests, the malicious samples were detected as out-of-profile for the devices that have been affected were correctly identified as such, yielding a 100% detection success rate for the attacks tested. Furthermore, by running the tests several times for both malicious and benign network traffic, the best accuracy (A) result was 100% and a false positive rate of 8.3%.

Table 5.5. Results for each kind of attacks.

Malware Type	Out of Profile (Yes/No)	Detected From Profile	Δ%
Zero-day exploits	Yes	D	28.37%
DDoS attack with Mirai Botnet	Yes	H, D, W	98.53%
DDoS attack with Black Energy	Yes	H, D, W,	128.42%
java_rmi	Yes	D, W	96.88%
distcc_exec_backdoor	Yes	D, W	98.64%
Unreallrcd	Yes	D, W	97.69%
Tomcat	Yes	W	395.52%
ruby_drb_code_exec	Yes	D, W	682.16%
hydra_ftp	Yes	D, W	95.15%
hydra_ssh	Yes	D, W	99.14%
Smtp	Yes	D, W	93.50%
netbios_ssn	Yes	D, W	307.39%
Zeus malware	Yes	W	96.70%

5.4 Conclusion

In this chapter, we have introduced the Cyber-Trust approach for detecting network-level attacks in IoT environments. The approach combines network profiling, binary visualisation, and machine learning techniques for detecting advanced and new threat vectors in IoT networks. Testing the proposed solution is performed in the Cyber-Trust testbed, which consists of many simulated and emulated smart home networks. For the training and testing of the proposed solution, we have created a new dataset that includes many 2D images corresponding to malicious and regular network traffic collected from different sources. In comparison, the malicious samples used in the testing phase were created in the Cyber-Trust testbed from real scenarios of attacks that cover a wide range of critical attacks, including DDoS attacks based on Botnets, Zero-day attacks, Malwares, exploits and backdoors. The dataset is now publicly available and ca by researchers in this field, especially with the lack of libelled data for testing machine learning algorithms.

The overall testing results are auspicious, especially when considering the results of the machine learning component, which recorded an accuracy of 98.35% over 100 tests with only a 0.98% FPR and 99.31% precision rating. These results were acquired from testing against device exploitation from unknown and known common vulnerabilities and high impact botnets that have seen extensive infection in

the real world; this speaks to the high efficacy of the solution. However, the overall accuracy of the proposed solution still stands to improve its value with further training. It is worth noting that when it comes to describing future work, tests could be performed to assess whether this model can increase its accuracy with more extensive or alternative forms of binary visualisation training and techniques. The network profiling has achieved good results, where the attacks were identified as out-of-profile for the devices that they have been affected based on the predefined threshold during the testing. The obtained results for this component could be significantly enhanced during the next testing phase by running more samples for an extended period.

Acknowledgment

This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] B. Jovanović, "Internet of Things statistics for 2021 – Taking Things Apart," *DataPort*, 2021. https://dataprot.net/statistics/iot-statistics/ (accessed Jun. 03, 2021).

[2] D. D. Level, "D2 . 1 Threat landscape?: trends and methods," no. 2018, p. 250, 2020.

[3] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," *IEEE Internet Things J.*, 2019, doi: 10.1109/JIOT.2019.2926365.

[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer (Long. Beach. Calif).*, 2017, doi: 10.1109/MC.2017.201.

[5] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Inf.*, 2016, doi: 10.3390/info7030044.

[6] OWASP, "Internet of Things (IoT) Top 10 2018," *OWASP Internet of Things Project*, 2018. https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10 (accessed Sep. 20, 2020).

[7] G. Bendiab, S. Shiaeles, A. Alruban, and N. Kolokotronis, "IoT malware network traffic classification using visual representation and deep learning," in

Proceedings of the 2020 IEEE Conference on Network Softwarization: Bridging the Gap Between AI and Network Softwarization, NetSoft 2020, 2020, doi: 10.1109/NetSoft48620.2020.9165381.

[8] D. B. Davis, "ISTR 2019: Cyber Criminals Ramp Up Attacks on Trusted Software and Supply Chains," *ISTR 24*, 2019. https://symantec-enterprise-blogs.security.com/blogs/expert-perspectives/istr-2019-cyber-criminals-ramp-attacks-trusted-software-and-supply-chains (accessed Jun. 04, 2021).

[9] G. Bendiab, B. Saridou, L. Barlow, N. Savage, S. Shiaeles, "IoT Security Frameworks and Countermeasures," in *IoT Security Frameworks and Countermeasures*, 1st Editio., N. K. Stavros Shiaeles, Ed. Boca Raton: CRC Press, 2021, p. 51.

[10] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[11] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020.

[12] K. S. Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, "Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 171, pp. 2372–2379, 2020.

[13] M. Zaman and C.-H. Lung, "Evaluation of machine learning techniques for network intrusion detection," in *NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–5.

[14] R. Shire, S. Shiaeles, K. Bendiab, B. Ghita, and N. Kolokotronis, "Malware Squid: A Novel IoT Malware Traffic Analysis Framework Using Convolutional Neural Network and Binary Visualisation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, doi: 10.1007/978-3-030-30859-9_6.

[15] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[16] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *2016 International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5.

[17] P. Van Huong, D. V. Hung, and others, "Intrusion detection in IoT systems based on deep learning using convolutional neural network," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, 2019, pp. 448–453.

[18] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep learning-based intrusion detection for iot networks," in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2019, pp. 256–25609.

[19] M. Hildebrandt, "Defining profiling: A new type of knowledge?," *Profiling Eur. Citiz. Cross-Disciplinary Perspect.*, pp. 17–45, 2008, doi: 10.1007/978-1-4020-6914-7_2.

[20] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: A review," in *Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012*, 2012, doi: 10.1109/ICCSEE.2012.373.

[21] S. A. Kumar, T. Vealey, and H. Srivastava, "Security in internet of things: Challenges, solutions and future directions," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016, doi: 10.1109/HICSS.2016.714.

[22] G. Bendiab, S. Shiaeles "D6.1 State-of-the-art on profiling, detection and mitigation," 2019. [Online]. Available: https://cyber-trust.eu/.

[23] D. R. E. Lear, R. Droms, "Manufacturer Usage Description Specification," *2019*. https://datatracker.ietf.org/doc/html/rfc8520 (accessed Jun. 16, 2021).

[24] J. Roux, E. Alata, G. Auriol, V. Nicomette, and M. Kaâniche, "Toward an intrusion detection approach for IoT based on radio communications profiling," in *2017 13th European dependable computing conference (EDCC)*, 2017, pp. 147–150.

[25] L. and J. P. and M. M. Chen, "https://www.intel.com/content/dam/www/public/us/en/ai/documents/stamina-scalable-deep-learning-whitepaper.pdf," 2020. https://www.intel.com/content/dam/www/public/us/en/ai/documents/stamina-scalable-deep-learning-whitepaper.pdf (accessed Jun. 21, 2021).

[26] I. Baptista, S. Shiaeles, and N. Kolokotronis, "A novel malware detection system based on machine learning and binary visualization," in *2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 – Proceedings*, 2019, doi: 10.1109/ICCW.2019.8757060.


DATA LEAK
SECURITY BREAC

DOI: 10.1561/9781680838350.ch6

Chapter 6

Utilising Honeypots and Machine Learning to Mitigate Unknown Threats in IoT

By G. Bendiab[*]*, J. Rose*[†]*, M. Swann*[‡] *and S. Shiaeles*[§]

University of Portsmouth
[*]gueltoum.bendiab@port.ac.uk
[†]joseph.rose@port.ac.uk
[‡]matthew.swann@port.ac.uk
[§]sshiaeles@ieee.org

IoT security has now emerged as one of the most important issue in network security. Conventional security techniques, such as firewalls and signature-based intrusion detection systems, have proven ineffective in protecting IoT networks from increasingly sophisticated attack and malware. Due to these constraints, researchers have been compelled to build novel intrusion detection solutions utilising various technologies such as IoT Honeypots and Machine Learning (ML). This chapter describes a novel approach to detect malicious network traffic that employs a honeypot and machine learning. The IoT honeypot system is used to gather intelligence about attacks that target IoT devices. The data gathered are used to understand the attackers' weapons, strategies and new techniques utilised. It is also used to train the machine learning model used on IDS on a continuous basis to improve its detection accuracy. This method is most successful against unknown and zero-day attacks on IoT computers.

6.1 Introduction

IoT devices seem to be almost everywhere these days, they are increasingly being used in vital infrastructure sectors such as healthcare, security, energy, and emergency services. All of these devices add a new entry point to networks, raising a growing security risk [1]. A single compromised device connected to a network can pose a potential threat to the network and serve as a point of entry for a wide range of hacking attempts [1]. According to the most recent threat environment, cyber criminals' techniques have advanced to the point where they are extremely difficult to identify and remediate. According to a recent report by the University of Maryland [2], they are now successfully breaching IoT devices every 39 seconds. Furthermore, security incidents confirm that the larger security problem is that these devices' security flaws can be easily exploited by hackers forming vast botnets (i.e., zombie armies) and in doing so launch significant DDoS attacks [3]. According to A10 Networks' most recent report, nearly 6 million DDoS attacks occurred in the fourth quarter of 2019 [4]. This study confirmed that Mirai remains the malware of choice for botnets, and WD-Discovery has surpassed SNMP (Simple Network Management Protocol) and SSDP (Simple Service Delivery Protocol) as the third most popular source of DDoS [4]. Despite substantial efforts (and budgets) by organisations and the security community to defend connected devices, attackers continue to devise new strategies to obfuscate their operation and avoid detection by cyber defence mechanisms [5]. Current signature-based Intrusion Detection Systems (IDSs) are especially ineffective at detecting unknown and obfuscated malware for which no signatures exist. Furthermore, malware signatures must be updated on a regular basis [6], which requires significant resources and human involvement/expertise to create these signatures [6, 7]. As a result, innovative intrusion detection technologies have become essential for defending against these threats before they cause serious harm.

In this article, we propose a hybrid intrusion detection solution that can enhance the currently deployed IDSs systems for protecting IoT networks from intruders, obfuscated, and zero-day threats using machine learning and established honeypot technology. The honeypot framework deliberately attracts hackers and uses their intrusion attempts to learn more about malicious actors and how they operate. Furthermore, raw data generated by the honeypot system is used for effective and dynamic training of the machine learning model, increasing its detection accuracy. The qualified machine learning model is used to identify possible unknown cyber security threats automatically. The remainder of the chapter is organised as follows: the first section provides context on honeypots and surveys previous work done in this field using machine learning techniques and honeypots software. Section 6.3 then offers a description of the proposed intrusion detection

system. It also addresses the most suitable strategies and algorithms for successfully implementing the proposed system. Finally, the final segment ends the chapter and addresses future work.

6.2 Background and Related Work

An Intrusion Detection System (IDS) is a security mechanism used to protect both the host and the network from potential threats that would normally pass through a typical firewall device [8]. IDSs have traditionally been classified into two types: host-based intrusion detection systems (HIDSs) and network-based intrusion detection systems (NIDSs) [9]. HIDSs are commonly used to monitor and analyse the internal activities on a particular machine as well as the network packets on its network interfaces. On the other hand, NIDSs are used to constantly track network traffic, searching for potentially malicious and unauthorised inputs that could compromise network security and performing automatic precautions to reduce them by sending warnings to the network administrator [8, 10]. NIDSs can be implemented in two ways: signature-based and anomaly-based. Most security defense systems have used a signature-based classification method since the early days of threat detection. This form of NIDS tracks network traffic and compares it to a database of known threats signatures or attributes, where a pattern that defines each particular threat's unique characteristics is generated, so that specific malware can be detected in the future [10]. Signature-based detection techniques are typically very successful at detecting known malware, but they are largely ineffective at detecting unknown and new malware for which no signatures exist [11]. Due to this restriction, modern attackers often mutate their creations in order to maintain malicious functionality by modifying the file's signature, such as polymorphic malware, which can create new variants each time it is executed, resulting in a new signature [9].

Due to the drawbacks of signature-based detection techniques, researchers are now concentrating on anomaly-based detection approaches [9, 10]. This technique classifies network traffic based on trends generated by tracking the characteristics of a typical operation over time. The actual network traffic is then compared to the predefined profile, and any major deviation from the pattern is classified as malicious [9]. This system is particularly effective for detecting unknown and obfuscated threats [10]. With the emergence of new forms of IoT threats on a regular basis, many methods and techniques for anomaly-based detection have been proposed in the literature. Many of these approaches have examined machine learning (ML) [12], with a focus on deep learning (DL) algorithms [13], which provide a powerful paradigm for automatically determining the features needed for

malicious traffic detection [12]. More recent research looked at the use of honeypots to improve NIDSs. Honeypot strategies aim to shift the defense strategy against attacks by allowing organizations to take the initiative [14]. The parts that follow include more information about previous work in malicious network traffic classification using the machine-learning methodology, as well as a history on honeypots and a survey of honeypot-related work.

The use of machine learning to defend against intrusions in IoT networks has recently gained a lot of attention in academia [15]. Usually, these techniques examine usable network traffic information to extract features that can be used to separate malicious traffic from legitimate traffic. The features are then used to train the classifier to detect possible attacks, with each data instance labelled as standard or anomalous. The output results are usually presented in binary format, with two possible values: natural or malware traffic [9]. In this area, supervised learning algorithms such as nearest neighbour classifiers, support vector machines, and rule-based schemes such as decision trees and random forests have shown promising results. In [16], a survey proposed a classification of learning-based intrusion detection systems and addressed the performance of various supervised and unsupervised learning algorithms used in this field in terms of accuracy and false alarm rate. According to the report, the most significant challenge to supervised learning is a lack of accessible datasets with labelled data. According to a study published in [17], current intrusion detection technologies for IoT networks still need to be improved in terms of scalability, detection accuracy, true positive rate, and energy consumption.

In the same vein, the authors of [18] explored the efficacy of various machine learning techniques in protecting IoT devices from DoS attacks. The aim of this research is to propose effective methods for developing IDSs for IoT applications using ensemble learning. Random forest (RF), AdaBoost (AB), Extreme gradient boosting (XGB), Gradient boosted machine (GBM), and Extremely Randomized Trees are the classifiers evaluated (ETC). In more recent work [19], authors have tested five supervised learning algorithms to distinguish normal IoT packets from DoS attack packets. The test classifiers are K-nearest neighbours "KDTree" algorithm (KN), Support vector machine with linear kernel (LSVM), Decision tree using Gini impurity scores (DT), Random Forest using Gini impurity scores (RF) and Neural Network (NN) with 4-layer. The accuracy rates of the classifiers ranged from approximately 91% to 99%.

Deep learning has also received a lot of attention in recent years. Because of its ability to automatically extract powerful features from unlabelled data, these algorithms are recognised as important to intrusion detection in IoT networks. The authors of [20] contrasted deep learning approaches to specific conventional NIDS techniques. The authors discovered that deep-learning-based approaches

outperform convolutional intrusion detection techniques in terms of detection accuracy across a wide range of sample sizes and traffic anomaly types. Many other solutions, such as work in [21, 22], and [23], have used Recurrent Neural Network (RNN) and its variants for network intrusion detection in the same sense. The Convolutional Neural Network (CNN), which has achieved great success in image classification and pattern recognition, has also been used in many intrusion detection systems (IDSs) by analysing images produced by network traffic characteristics [24, 25]. The output of the CNN-based intrusion detection solution was evaluated in [24] using the synthetic datasets KDDCup 99 [26] and NSL-KDD [27]. Auto-encoders and Variational Auto-encoders are two other common deep learning techniques that are currently being used in research. Many recent studies [28, 29] have looked into the robustness of these strategies in intrusion detection. In terms of detection accuracy, the authors of [28] reported that the proposed autoencoder-based IDS outperforms IDSs based on Principle Component Analysis (PCA) by more than 15%. As a result, several recent approaches have investigated the efficacy of using deep learning techniques for intrusion detection. Despite some progress in this area, the subject of using deep learning for intrusion detection is underutilised.

Honeypot technology aims to compensate for weaknesses in intrusion detection systems by collecting information about current threats on a network and detecting the emergence of new threats [30]. A honeypot is a cyber device that impersonates a particular target (e.g., a service, database, or operating system) in order to draw cyberattacks and use their intrusion attempts to collect information about intruders and how they work [30]. The intelligence obtained from a honeypot would significantly aid in the improvement of the security of real-world production systems. Honeypots have historically been rated based on their level of contact, which expresses how much activity an attacker may have with them [31]. There are two types of honeypots in this context: low-interaction honeypots and high-interaction honeypots. A honeypot with a high interaction rate enables attackers to compromise and gain access to the actual vulnerable service or programme [31]. Since they do not emulate any services, High Interaction Honeypots aid in detecting unknown vulnerabilities and gathering detailed information regarding an attacker's procedures. They are, however, more susceptible to infection, and as a result, attackers will gain full control of them in order to compromise and target other actual production systems on the network [14]. Furthermore, they are complex and expensive to deploy and sustain [31]. IoTPOT [32] is one of the first high-interaction honeypots implemented in the field of IoT to impersonate IoT modules. SIPHON [33] is also a scalable, high-interaction honeypot network for Internet of Things applications. Honware [34] is another example of a recently created high-interaction honeypot capable of simulating a wide range of IoT products.

Low Interaction Honeypots, on the other hand, operate as emulators of services and operating systems, allowing the attacker only minimal interaction. As a consequence, these Honeypots are not vulnerable and cannot be corrupted by exploits; however, attackers can easily detect them by executing commands that the emulator does not support [31, 35]. The common tool honeyd [36], which provides a simple method to simulate different services provided by several machines on a single computer, is an example of a low-interaction honeypot. Low-interaction honeypot systems have been used in the field of IoT to capture malicious IoT behaviours. Low interaction honeypots such as Nepentes [37] and Dionaea [38] are also used for large-scale data collection on self-replicating malware in the wild. To simulate the behaviour of IoT computers, Dionaea honeypot [33] employs the MQTT protocol. The developers of [39] used a low interaction honeypot to identify and fix vulnerabilities in IoT devices. The honeypot is designed automatically utilizing machine learning technology to learn the behavioural characteristics of various types of IoT devices.

MIHs (Medium Interaction Honeypots) are a mixture of low and high interaction honeypots. Researchers recognise this type of honeypot system as offering a full honeypot solution for intrusion monitoring and detection [31]. Several MIH IoT honeypot models have thus been proposed in the literature [31, 40, 41], and [42]. For example, the authors of [31] proposed a hybrid honeypot architecture based on low-interaction honeypots (honeyds) that function as service and operating system emulators. Malicious traffic guided to honeyds is then seamlessly routed to high interaction honeypots, where attackers can communicate with real services. In a subsequent paper [41], the authors defined a hybrid IoT honeypot architecture with machine learning for combating zero-day DDoS attacks. In the same vein, the authors of [40] developed a new interconnected and collaborative hybrid honeynet for IoT networks. The authors of [43] defined an IoT-based honeynet network that included both virtual and physical IoT devices. For traffic analysis, the proposed honeypot system made use of supervised machine learning algorithms. Examples of recently formed IoT honeypots are shown in Table 6.1.

6.3 Intrusion Detection Framework

We are primarily interested in detecting and mitigating the unknown malware responsible for Zero-Day attacks in this proposed detection system. The word "zero-day exploit" refers to malicious code written by malicious actors in order to exploit a "zero-day vulnerability." This form of malware can go unnoticed for several years and is extremely dangerous because only the perpetrator is aware of its nature, so no security fixes to address these vulnerabilities and block its subsequent

Table 6.1. Examples of recently developed IoT honeypots.

- **Dionaea** [38]: uses MQTT protocol to simulate the IoT behaviour.
- **U-POT** [44]: for devices that use Universal Plug and Play (UPnP) protocol.
- **ZigBee Honeypot** [35]: simulates a ZigBee gateway .
- **SIPHON** [33]: a high-interaction honeypot platform for IoT devices, with 80 high-interactive devices.
- **Honware** [34]: a high-interaction honeypot framework which can emulate different IoT devices.
- **Thingpot** [45]: Emulates different IoT communication protocol.
- **HIoTPOT** [42]: Emulates Telnet services of various IoT devices.
- **Multiport Honeypots** [40]: a medium-high interaction IoT honeypots that can simulate UPnP services and SOAP service ports.

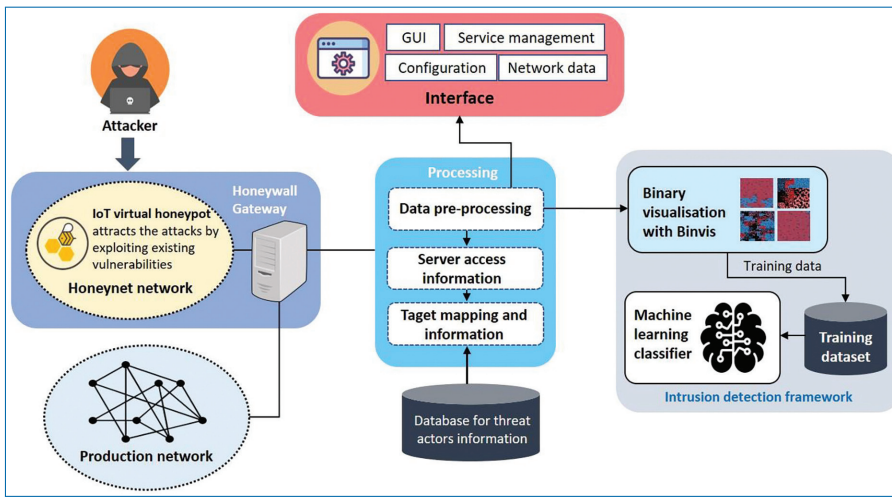


Figure 6.1. Process flow for the proposed solution with the honeypot machine learning and based detection framework.

zero-day exploits are available [9]. We proposed a new detection and mitigation approach based on honeypots and machine learning to address this problem. The honeypot framework attracts hackers by design in order to track, deflect, and analyse hacking attempts to gain unauthorised access to IoT devices. In comparison, the Machine Learning (ML) based detection system, which is an application of machine learning together with binary visualisation techniques, is used to identify possible unknown cyber security threats.

The proposed system's entire mechanism is depicted in Figure 6.1. As shown in Figure 6.1, a Honeywall is built in the honeypot system to isolate the honeynet network from the production infrastructure of the organization. The Honeywall

is also used as the primary point of entry into the honeynet network, providing complete control over all incoming and outgoing traffic to and from the network. Any actions performed with the honeynet system are deemed malicious and are routed to the pre-processing module for further inspection. This data is also converted to a suitable format (2D image) so that it can be used to continuously train the machine learning model and in doing so improve its detection accuracy. Data pre-processing also requires analysing the collected data to better understand the attackers' tools, strategies, techniques, and motives. This can be accomplished by incorporating resources and frameworks into the honeypots to record all system activities.

The total data collected from deep inspection of network and system level interactions with the honeynet devices is logged into a central threat actor database, this includes the low-level information necessary to generate the aforementioned 2D image from deep packet inspection of capture network traffic which is then used in the training of the NIDS system through an ML-based module. This uses these images as the basis of the proposed methods. And provide an authoritative summary of the interactions that have occurred, these are assigned to the database alongside identifiable information such as the originating IP addresses, timestamps, and corresponding service information for the targeted network services.

6.3.1 Honeypot System

In the proposed solution, the honeypot system is mainly used to gather intelligence about attack attempts on IoT devices. It involves two main components: the honeynet and the Honeywall. The honeynet network is used to attract the attackers for intentionally exploiting the vulnerabilities present in IoT devices, where all interactions with this network is considered malicious. The data collection is done at the Honeywall gateway, which is the main point of entry to the honeynet network. Once data is captured, it is securely sent to the preprocessing system for further analysis and for training the classification model. The honeynet network consists of different IoT devices that capture different malicious behaviours. However, building a honeynet of IoT devices is challenging using traditional methods due to the special characteristic of IoT. Thus, many researchers have been tried to design new honeypots for IoT devices [34, 40, 43, 44]. As mentioned previously, Table 6.1 provides some examples of recently developed IoT honeypots.

However, the most appropriates implementation of the IoT-based honeynet system should simulate the whole IoT platform along with all the supported protocols in IoT communications. For example, Thingpot [45] is an IoT virtual

honeypot capable of catching various IoT-based botnets by emulating different IoT communication protocol along with entire IoT platform behavior. In addition, IoT honeypots should be able to provide high-level interactions in order to motivate attackers to perform their malicious activities and therefore, keep track of a dynamic threat landscape.

6.3.2 Machine Learning Detection Framework

The machine learning-based detection framework is a crucial component in the proposed solution. As shown in Figure 6.1, this framework consists of two main steps, first obtaining the corresponding visual representation of the collected network traffic, and second, processing this visual representation by the trained machine learning model. The main idea of this framework is based on the Malware-Squid approach proposed in [46], which represent the cyber-defence service in the Cyber-Trust project [47]. In this approach, we use the Hilbert space-filling curve [48] as its main clustering algorithm, this is achieved by assigning specific colours to each byte as it's converted into a 2D image. This clustering algorithm outperforms other curves in preserving the locality between objects in multi-dimensional spaces, which helps to create much more appropriate RGB images for the classification process [11]. The conversion is performed on each byte depending on its ASCII character reference as follow:

- Blue for printable characters
- Green for control characters
- Red for extended characters
- Black for the null character, or 0x00
- White for the non-breaking space, or 0xFF

These generated byte arrays are then processed using the Hilbert algorithm, transforming them into images that retain optimal locality for pattern recognition, allowing them to be processed by the machine learning image classification models. The size of the output RGB image is 784 (1024*256) bytes. Figure 6.2 shows Bin-Vis images for both normal and malware network traffic, which are created using the Hilbert space-filling curve.

There is a number of learning algorithms available for performing network traffic classification based on the generated 2D images. However, in this work, we are interested in unsupervised learning algorithms that can accurately classify the network traffic as "normal" or "malicious" with a reasonable rate of false alarms. In this context, a variety of unsupervised classifiers such as Autoencoders, Self-Organizing Incremental Neural Network [49], Residual Neural Network (ResNet) [11] and MobileNet [46, 50] have been found to be effective in detecting abnormal network

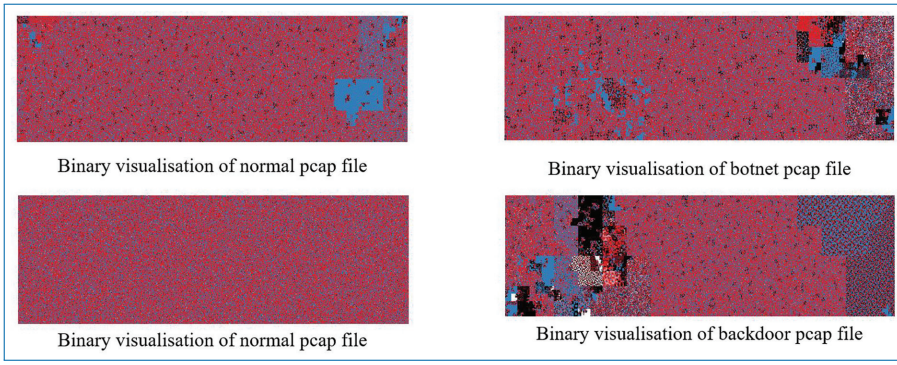


Figure 6.2. Binvis images of both normal and malware network traffic created with the Hilbert space-filling curve [11].

traffic with an overall accuracy value that meets the required values in practical use (from 94% to 96%).

6.4 Conclusion

In this chapter, we introduced a new approach for network intrusion detection based on machine learning and honeypot technology. For the implementation of the proposed intrusion detection framework, we have discussed already developed technologies in the fields of IoT honeypots and machine learning. The use of IoT honeypots that can simulate a whole IoT platform will ensure the logging of a large vector of IoT based security threats characteristics, especially, new threat vectors. Collected malware traffic can be also used to effectively train the ML-based detection system, which will undoubtedly enhance its detection accuracy and therefore, protect the whole production network against the new immerging security threats.

For the future scope, we will implement the proposed IDS framework in a real-world environment and deeply investigate open issues related to IoT honeypots over real-time scenarios. We also intend to compare the performance of the proposed solution in contrast to representative models in this field.

Acknowledgment

This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] M. Safaei Pour, E. Bou-Harb, K. Varma, N. Neshenko, D. A. Pados, and K. K. R. Choo, "Comprehending the IoT cyber threat landscape: A data dimensionality reduction technique to infer and characterize Internet-scale IoT probing campaigns," *Digit. Investig.*, 2019, doi: 10.1016/j.diin.2019.01.014.

[2] M. Cukier, "Study: Hackers Attack Every 39 Seconds," *University of Maryland*, 2021. https://eng.umd.edu/news/story/study-hackers-attack-every-39-seconds (accessed Mar. 10, 2021).

[3] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Inf.*, 2016, doi: 10.3390/info7030044.

[4] H. Taylor, "What Are Cyber Threats and What to Do About Them," *preyproject.com*, 2021. https://preyproject.com/blog/en/what-are-cyber-threats-how-they-affect-you-what-to-do-about-them/ (accessed Mar. 22, 2021).

[5] S. A. Kumar, T. Vealey, and H. Srivastava, "Security in internet of things: Challenges, solutions and future directions," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016, doi: 10.1109/HICSS.2016.714.

[6] R. Samrin and D. Vasumathi, "Review on anomaly based network intrusion detection system," in *International Conference on Electrical, Electronics, Communication Computer Technologies and Optimization Techniques, ICEECCOT 2017*, 2018, doi: 10.1109/ICEECCOT.2017.8284655.

[7] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," *IEEE Internet Things J.*, 2017, doi: 10.1109/JIOT.2017.2694844.

[8] S. B. Ambati and D. Vidyarthi, "A Brief Study and Comparison of, Open Source Intrusion Detection System Tools," *Int. J. Adv. Comput. Eng. Netw.*, no. 110, pp. 2320–2106, 2013, [Online]. Available: http://www.iraj.in/journal/journal_file/journal_pdf/3-27-139087836726-32.pdf.

[9] S. S. G. Bendiab, B. Saridou, L. Barlow, N. Savage, "IoT Security Frameworks and Countermeasures," in *IoT Security Frameworks and Countermeasures*, 1st Editio., N. K. Stavros Shiaeles, Ed. Boca Raton: CRC Press, 2021, p. 51.

[10] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, 2016.

[11] G. Bendiab, S. Shiaeles, A. Alruban, and N. Kolokotronis, "IoT malware network traffic classification using visual representation and deep learning," in *Proceedings of the 2020 IEEE Conference on Network Softwarization: Bridging the Gap Between AI and Network Softwarization, NetSoft 2020*, 2020, doi: 10.1109/NetSoft48620.2020.9165381.

[12] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[13] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, p. 102767, 2020.

[14] S. Yeldi *et al.*, "Enhancing network intrusion detection system with honeypot," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, 2003, vol. 4, pp. 1521–1526.

[15] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Comput. Networks*, vol. 151, pp. 147–157, 2019.

[16] N. F. Haq, A. R. Onik, M. A. K. Hridoy, M. Rafni, F. M. Shah, and D. M. Farid, "Application of machine learning approaches in intrusion detection system: a survey," *IJARAI-International J. Adv. Res. Artif. Intell.*, vol. 4, no. 3, pp. 9–18, 2015.

[17] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the Internet of things: A comprehensive investigation," *Comput. Networks*, vol. 160, pp. 165–191, 2019.

[18] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020.

[19] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 29–35.

[20] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, 2016, pp. 581–585.

[21] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in sdn-based networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 202–206.

[22] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21954–21961, 2017.

[23] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluation of recurrent neural network and its variants for intrusion detection system (IDS)," *Int. J. Inf. Syst. Model. Des.*, vol. 8, no. 3, pp. 43–63, 2017.

[24] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[25] Y. Liu, S. Liu, and X. Zhao, "Intrusion detection algorithm based on convolutional neural network," *DEStech Trans. Eng. Technol. Res.*, no. iceta, 2017.

[26] "KDD Cup 1999 Data," *University of California, Irvine*, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed Mar. 26, 2021).

[27] "NSL-KDD dataset," *University of new brunswock*, 2019. https://www.unb.ca/cic/datasets/nsl.html (accessed Mar. 26, 2021).

[28] P. Madani and N. Vlajic, "Robustness of deep autoencoder in intrusion detection under adversarial contamination," in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security*, 2018, pp. 1–8.

[29] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, 2018.

[30] I. Kuwatly, M. Sraj, Z. Al Masri, and H. Artail, "A dynamic honeypot design for intrusion detection," in *The IEEE/ACS International Conference onPervasive Services, 2004. ICPS 2004. Proceedings.*, Jul. 2004, pp. 95–104, doi: 10.1109/PERSER.2004.1356776.

[31] H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, "A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks," *Comput. Secur.*, 2006, doi: 10.1016/j.cose.2006.02.009.

[32] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the rise of IoT compromises," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[33] J. D. Guarnizo *et al.*, "Siphon: Towards scalable high-interaction physical honeypots," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, 2017, pp. 57–68.

[34] A. Vetterl and R. Clayton, "Honware: A virtual honeypot framework for capturing CPE and IoT zero days," in *2019 APWG Symposium on Electronic Crime Research (eCrime)*, 2019, pp. 1–13.

[35] S. Dowling, M. Schukat, and H. Melvin, "A ZigBee honeypot to assess IoT cyberattack behaviour," in *2017 28th Irish signals and systems conference (ISSC)*, 2017, pp. 1–6.

[36] N. Provos and others, "A Virtual Honeypot Framework.," in *USENIX Security Symposium*, 2004, vol. 173, no. 2004, pp. 1–14.

[37] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, "The nepenthes platform: An efficient approach to collect malware," in *International Workshop on Recent Advances in Intrusion Detection*, 2006, pp. 165–184.

[38] "Dionaea," *Nepenthes Development Team*, 2011. http://dionaea.carnivore.it/ (accessed Mar. 28, 2021).

[39] T. Luo, Z. Xu, X. Jin, Y. Jia, and X. Ouyang, "Iotcandyjar: Towards an intelligent-interaction honeypot for iot devices," *Black Hat*, pp. 1–11, 2017.

[40] W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An IoT Honeynet Based on Multiport Honeypots for Capturing IoT Attacks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3991–3999, 2019.

[41] R. Vishwakarma and A. K. Jain, "A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 1019–1024.

[42] U. D. Gandhi, P. M. Kumar, R. Varatharajan, G. Manogaran, R. Sundarasekar, and S. Kadu, "HIoTPOT: surveillance on IoT devices against recent threats," *Wirel. Pers. Commun.*, vol. 103, no. 2, pp. 1179–1194, 2018.

[43] P. J. Hanson, L. Truax, and D. D. Saranchak, "IOT honeynet for military deception and indications and warnings," in *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, 2018, vol. 10643, p. 106431A.

[44] M. A. Hakim, H. Aksu, A. S. Uluagac, and K. Akkaya, "U-pot: A honeypot framework for upnp-based iot devices," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, 2018, pp. 1–8.

[45] M. Wang, J. Santillan, and F. Kuipers, "ThingPot: an interactive Internet-of-Things honeypot," *arXiv Prepr. arXiv1807.04114*, 2018.

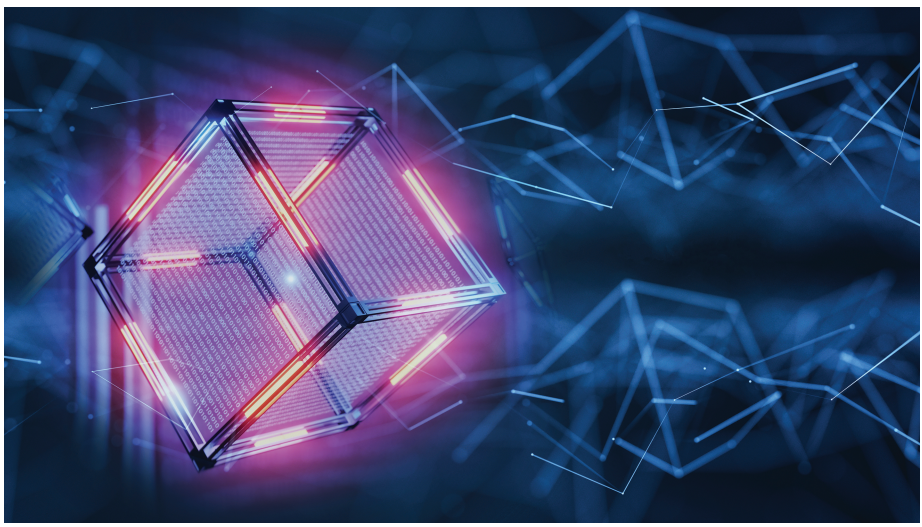
[46] R. Shire, S. Shiaeles, K. Bendiab, B. Ghita, and N. Kolokotronis, "Malware Squid: A Novel IoT Malware Traffic Analysis Framework Using Convolutional Neural Network and Binary Visualisation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, doi: 10.1007/978-3-030-30859-9_6.

[47] "Cyber-Trust," *Cyber-Trust*, 2019. https://cyber-trust.eu/ (accessed Mar. 24, 2021).

[48] H. Sagan, "Hilbert's Space-Filling Curve," 1994.

[49] I. Baptista, S. Shiaeles, and N. Kolokotronis, "A novel malware detection system based on machine learning and binary visualization," in *2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 – Proceedings*, 2019, doi: 10.1109/ICCW.2019.8757060.

[50] L. Barlow, G. Bendiab, S. Shiaeles, and N. Savage, "A Novel Approach to Detect Phishing Attacks using Binary Visualisation and Machine Learning," in *2020 IEEE World Congress on Services (SERVICES)*, 2020, pp. 177–182.



DOI: 10.1561/9781680838350.ch7

Chapter 7

Towards Post-Quantum Blockchain Platforms

By S. Brotsis[1,*], *N. Kolokotronis*[1,†] *and K. Limniotis*[2,‡]

[1]University of the Peloponnese
[2]Hellenic Data Protection Authority
*brotsis@uop.gr
†nkolok@uop.gr
‡klimniotis@dpa.gr

Two of the most significant arising technological advancements currently underway that are showing an ever-increasing spread both in industrial and academic areas, are the blockchains and the advent of quantum computing. Since, blockchains have dramatically advanced in the recent years and have found numerous applications in many fields with the expectation to significantly enhance their security, the conundrum related to the quantum threat and the implementation of post-quantum signatures in blockchains is a trending topic in nowadays scientific community. As any product that is based on cryptographic primitives, this technology is influenced by the advent of quantum computing, since they are not essentially different from other resilient and secure applications in such regard. This chapter provides the theoretical support of the recent developments in the area of post-quantum cryptography (PQC) aiming at the incorporation of secure cryptographic primitives

to the blockchain technology. For this reason, the chapter assesses contemporary PQC algorithms and presents the current situation of the NIST's 3rd round PQC candidates. In addition, it demonstrates the impact of quantum-computing on blockchains and it investigates the incorporation of PQC primitives to the various blockchain platforms. Therefore, this chapter aims to provide guidelines and demonstrate the challenges to both researchers and industry regarding the implementation of post-quantum algorithms in blockchain applications.

7.1 Introduction

Since the evolution of Bitcoin, the blockchain technology has met growing interest in the last years as a novel technology facilitating the degree of decentralisation required by modern applications and services in an efficient and robust way. Blockchain is a distributed database of records, or shared ledger of all the transactions or digital events having been executed and exchanged among a number of parties. Blockchains have already adopted the basic cryptographic primitives, such as the hash functions and the digital signatures, which are used to achieve consensus and authenticate transactions. Most of the most popular blockchain platforms use a linked list of blocks, in which each block pertains a hash pointer of the previous, while the data of each block is organized using Merkle trees. However, such schemes and algorithms cannot guarantee the security requirements that might occur in the future. While, the modern computer society tends to globalization, the goals for security are not only basic requirements, such as tamper resistance and trust, but also compelling security demands for privacy preservation mechanisms and needs for enforcing accountability in many applications [1]. Since, the blockchain technology has been adopted not only to the financial industry, but to many other areas as well [2–4]; its security and business architecture cannot be easily modified. Therefore, the security of blockchains should acknowledge not only the ongoing means of attacks, but also security issues that might surface in the future.

Essentially, for the transaction's authentication, the blockchains are based on the elliptic curve digital signature algorithm (ECDSA), which is not adequate enough to deal with the quantum threat. The Shor algorithm has been proven to demonstrate quantum supremacy over classical computing. If this algorithm is used by an attacker, then the victim's private key can be derived from the public key and the system's security to be compromised. Similarly, if the attacker forges the user's signature, then all the user's assets and privacy will be lost. Therefore, considering the cryptographic underpinnings of blockchains, this chapter underlines the post-quantum security aspects that can be adopted in blockchain technology and enable it to resist quantum attacks based on the Shor's and Grover's algorithms.

More precisely, this chapter presents the impact of quantum-computing attacks on blockchains and it investigates the incorporation of PQC primitives in the various blockchain platforms. Particularly, the most appropriate post-quantum cryptosystems for blockchains are examined along with their main challenges. Therefore, this chapter can be used as a guide for the development of post-quantum blockchains, since it is necessary that both researchers and industry to be aware to the quantum computing area and its advances.

The chapter consists of six sections, including the current introductory section. More precisely, the structure of the document is as follows: Section 7.2 describes the state-of-the-art in post-quantum cryptography (PQC), in which the public key PQC cryptosystems, the PQC signing algorithms and the the current situation of NIST are presented. Section 7.3 deals with the advances of the PQC in the blockchain technology and presents the blockchain platforms that support PQC primitives. Section 7.4 performs a comparison of the performance of PQC primitives that passed to the third round of the NIST call and describes the resistance of PQC algorithms on various cryptographic attacks. Finally, the main conclusions obtained are summarized in Section 7.5.

7.2 State-of-the-Art in PQC

7.2.1 Public-Key Post-Quantum Cryptosystems

Post-quantum cryptography (PQC) refers to cryptographic systems that will provide security even in case that quantum computers become a reality. More precisely, quantum computing makes use of quantum-mechanical phenomena, thus being more powerful than classical computers. In simple words, classical computers operate on bits, which can have one of two values (states), i.e. 0 or 1, whereas quantum computers operate on qubits, which are in a superposition of states, i.e. 0, 1, or (a little bit of) both. Due to this, quantum algorithms can leverage this superposition of states to provide efficient solutions to several mathematical problems in which classical computers practically fail to provide a solution. Although not every problem can be efficiently solved; there exist though several problems which are being considered difficult today, but they are efficiently solvable by a quantum computer. Some of these problems constitute building blocks for contemporary cryptographic algorithms, thus rendering them fully insecure in the post quantum era.

The most famous quantum algorithms, which have direct impact on the security of cryptographic systems, are the Shor's integer factorisation algorithm, which is a quantum algorithm that factors an integer N in polynomial time with respect to the length of N and the Grover's algorithm, which is a quantum algorithm for searching an unstructured database.

Current symmetric ciphers with 256-bit keys such as AES-256, are believed to be quantum-resistant. Similarly, hash functions with proper parameters (i.e., length of the hashed value) are also considered post-quantum secure, in terms of collision resistance. Therefore, post-quantum cryptography research focuses on asymmetric algorithms, so as to replace RSA, (EC)DH and (EC)DSA. These post-quantum secure algorithms are based on mathematical problems that are believed to be difficult in the classical and quantum cases. Moreover, since hash functions are also post-quantum secure, several post-quantum digital signature schemes whose security relie on the security of hash functions also exist.

More precisely, the post-quantum cryptographic algorithms are mainly classified into one of the following categories, whilst each of them rests its security with one specific difficult mathematical problem:

- Code-based cryptography,
- Lattice-based cryptography,
- Multivariate cryptography,
- Hash-based cryptography,
- Supersingular elliptic curve isogeny cryptography.

whereas hybrid approaches are also considered. In addition, a few algorithms are based on the security of zero-knowledge proofs, which are described next.

Code-based cryptography

The security of the cryptographic algorithms included in this class is based on coding theory – i.e., with the inherently different problem of decoding an erroneous codeword which has been produced through an unknown error correcting code. The most classical such system is the McEliece's cryptosystem, whose security is based on the syndrome decoding problem. McEliece's cryptosystem provides fast encryption and relatively fast decryption, which is an advantage for performing rapid blockchain transactions. However, McEliece's cryptosystem requires large matrices that act as public and private keys, which may be a restriction in constrained environments.

Lattice-based cryptography

This class includes cryptographic algorithms whose construction is based on lattices, which are sets of points in n-dimensional spaces with a periodic structure. These algorithms rest their security on the known difficulty of specific mathematical problems in the field of lattices, like the Shortest Vector Problem (SVP), being NP-hard, which is related with the finding of the shortest non-zero vector within a lattice. Other similar lattice-based difficult problems also exist, such as the Closest

Vector Problem (CVP), the Shortest Integer Solution (SIS) or the Shortest Independent Vectors Problem (SIVP). An important lattice-based problem, which is being "present" in several lattice-based cryptographic system, is the "learning with errors" (LWE) problem, which has security reductions to variants of SVP.

Multivariate cryptography

Multivariate cryptography relies on the complexity of solving systems of multivariate equations, which have been demonstrated to be either NP-hard or NP-complete. In general, it is known that such cryptographic schemes have some limitations into their decryption speeds (due to the involved "guess work". Currently, some of the most promising multivariate-based schemes are based on Hidden Field Equations (HFE) for a generic survey of mathematical problems in the field of multivariate cryptography.

Hash-based cryptography

This scheme includes cryptographic digital signatures schemes whose security relies on the security of the underlying hash function instead of on the hardness of a mathematical problem. This kind of schemes was initiated since the late 70s, when Lamport proposed a signature scheme based on a one-way function.

Supersingular elliptic curve isogeny cryptography

This scheme includes cryptographic algorithms whose security relies on the isogeny protocol for ordinary elliptic curves but enhanced to withstand the quantum attack. Such cryptosystems usually employ key sizes in the order of a few thousand bits.

Other approaches

Post-quantum cryptography based on zero-knowledge proofs: Based on the classical concept of zero-knowledge proofs, these cryptographic algorithms are generalizations of hash-based cryptographic schemes, enriched by nice cryptographic properties of symmetric ciphers towards constructing zero-knowledge proofs.

Hybrid approaches: The hybrid schemes seem to be the immediate next step towards post-quantum security, since they appropriately merge pre-quantum and post-quantum cryptosystems, aiming to protect the exchanged data both from quantum attacks and from attacks against the used post-quantum schemes. However, such schemes involve implementing two complex cryptosystems, which require significant computational resources and more energy consumption. Therefore, future developers of hybrid post-quantum cryptosystems for blockchains will have to look for a trade-off between security, computational complexity and resource consumption.

7.2.2 Post-Quantum Signing Algorithms

In real-world applications today, the most widely used cryptographic schemes for digital signatures are RSA, Digital Signature Algorithm (DSA), and Elliptic Curve Digital Signature Algorithm (ECDSA. However, as it is already mentioned, such digital signature schemes are not post-quantum secure. Therefore, it is essential, for blockchain applications to provide a long-term security and ensure that the digital signatures are secure against post-quantum computers. To this end, we subsequently focus explicitly on post-quantum signing algorithms.

Hash-based digital signatures

The hash-based signature (HBS) algorithms are schemes with minimal security requirements, reasonably fast, providing small size signatures and having strong security guarantees (their security proofs are relative to plausible properties of the cryptographic hash functions).

HBS schemes can be classified as stateless and stateful schemes which can be further categorized as One-Time Signature (OTS), Few-Time Signature (FTS), Multi-Time Signature (MTS), and Hierarchical Signature (HS), depending on key and signature generation. A nice taxonomy of these schemes can be seen in Figure 7.1.

Stateful one-time signature (OTS) schemes: The Lamport scheme, the Winternitz scheme, and its variants WOTS+, WOTS^{PRF} are characteristic algorithms lying in in this class. To sign a message with OTS schemes, the private key is uniformly generated at random, whereas the public key is derived by the private key, by appropriately involving a hash function; the irreversibility of the hash function, as well its collision resistance, ensure that knowledge of the public key does not allow the computation of the private key. The Lamport scheme, even if it possesses great security properties, it is actually practically inappropriate due to several limitations; first is the one-time signature scheme (i.e., each signature can be used only once), whereas it requires extremely large sizes of keys; the derived signatures are also large (see Table 7.1). The fact that it is an OTS scheme implies that each secret key is being used only once for signing; otherwise, an attacker may be capable to derive useful information for imitating the user via setting valid signatures (since the attacker will be able to learn part of the secret key). The drawbacks that are related with the efficiency of the Lamport scheme are being alleviated by the Winternitz One Time signature (WOTS) scheme, which utilizes a so-called Winternitz parameter that controls a time/memory trade-off. Therefore, in principle, reducing the space required for keys and signatures makes WOTS a good choice for memory-constrained embedded devices, but at the cost of slower signing and verifying process.

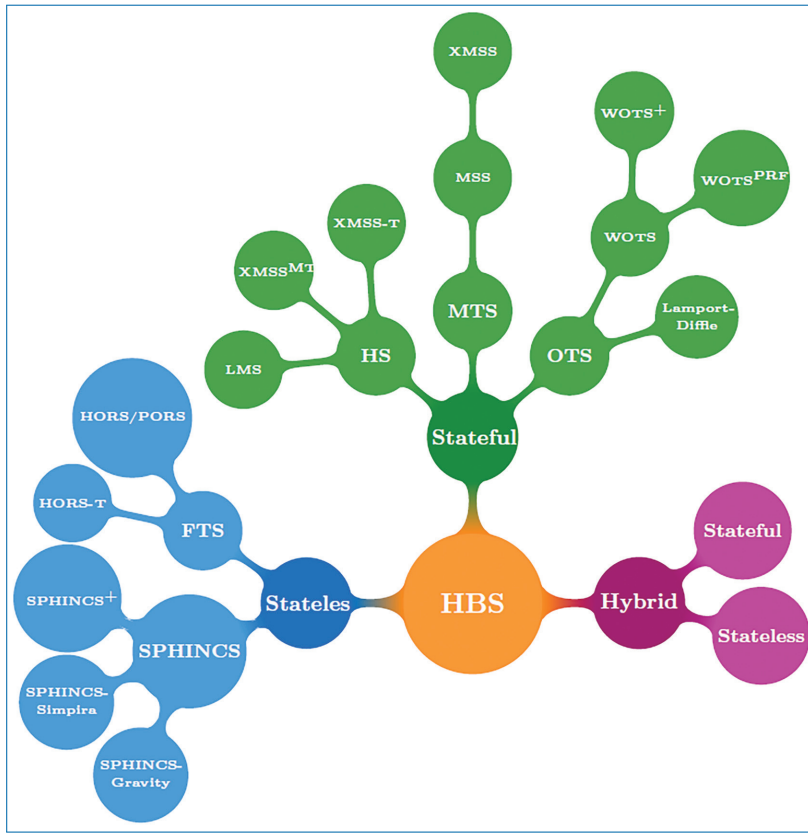


Figure 7.1. A taxonomy of HBS cryptographic scheme [9].

Table 7.1. OTS and FTS schemes for 384-bit message length and about 128-bit post-quantum security level.

Signature Scheme	Type	Signature Size (Kb)	Key Size (Kb)
Lamport	OTS	18.4	36.9
WOTS	OTS	4.8	4.8
WOTS+	OTS	3.2	3.2
$WOTS^{PRF}$	OTS	3.2	3.7
HORS-T	FTS	17.3	0.05

Stateful Multi-time Signature Schemes (MTS): To tackle with the inherent limitations of OTS schemes, MTS schemes are proposed to construct many-time signatures by using OTS as an underlying primitive. The first such scheme has been proposed by Merkle, being called Merkle Signature Scheme (MSS) [5]. This scheme utilizes a so-called Merkle tree, which suffices to combine a large number of OTS

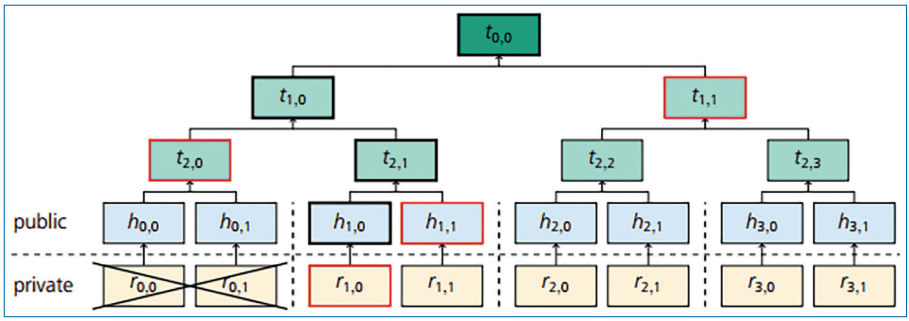


Figure 7.2. A Merkle tree with a verification path for the OTS public key h1,0 [5].

key pairs into a single binary hash tree structure (as shown in Figure 7.2). The root of the tree constitutes a global public key. Due to the properties of the underlying hash functions that are being used to build a Merkle tree, the signer (and nobody else) can easily prove that an one-time public key (e.g. a WOTS+ public key) is associated with a global public key, by revealing appropriate nodes of the tree, determining the authentication path, which allow the validator to reconstruct the path from the relevant one-time public key to the tree's root upon signature verification.

Moreover, there are several other efficient ways to handle Merkle trees, especially the authentication (i.e. appropriately caching the authentication path from the previous signature). Such clever techniques give rise to more efficient signature schemes based on Merkle trees – with the Extended Merkle Signature Scheme (XMSS) being a prominent example [6]. The XMSS scheme is an appropriately modified Merkle hypertree, where the inherent leaves of the tree are based on a WOTS+ scheme. More precisely, the XMSS scheme utilizes a Merkle tree with a major difference being the use of bitmask XOR of the child nodes prior to concatenation of the hashes into the parent node. The use of the bitmask XOR allows the collision resistant hash function family to be replaced. Each leaf of the tree is the root of child trees (also XMSS trees) being called L-trees, which hold the OTS public keys.

Stateful Hierarchical Signature Schemes (HS): Stateless hash-based signature schemes are generally considered slow, since it is necessary to construct a new tree to generate a new key pair. Therefore, hierarchical signature schemes (HS) constitute the next step towards improving efficiency. HS schemes are actually MTS schemes that use other hash-based signatures in its construction. The idea of HS is based on the formation of a hyper-tree that involves tree chaining by using multiple layers of MSS tree. By these means, the upper layers are used to sign the roots of the layers below while only the lowest layer is used to sign messages. Notable examples of HS

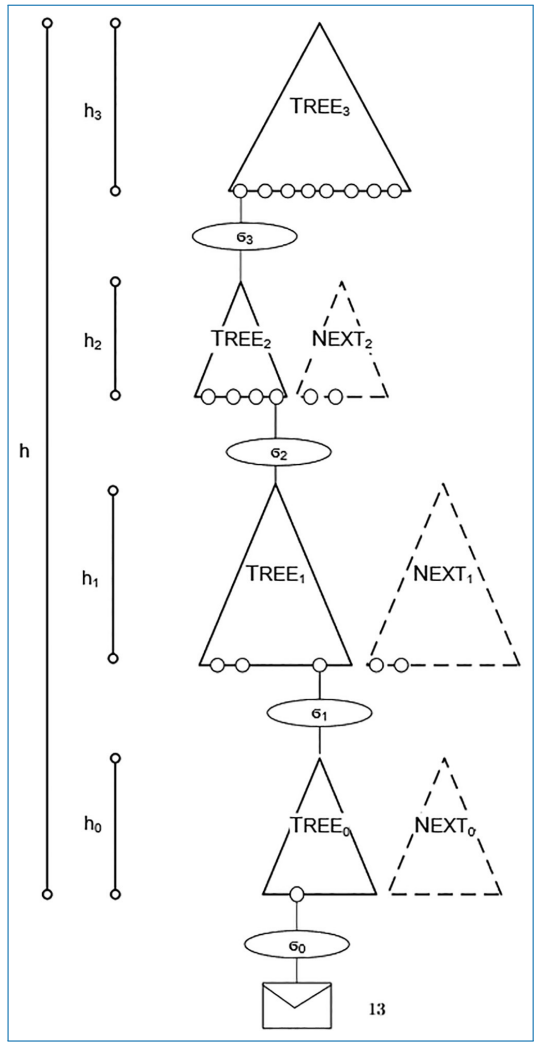


Figure 7.3. XMMS^{MT} with 4 layer [42].

are XMSS-MultiTree (XMSS^{MT}) (see also Figure 7.3), XMSS with tightened security (XMSS-T) and Leighton Micali Scheme (LMS). A XMSS^{MT} is a nice option for applications that require many messages to be signed, provided that the techniques mentioned above for optimization (use of PNRG, caching of authentication path etc.) are still present.

Another, more recent, stateful HBS scheme, which utilizes a blockchain for storing "authentication paths" is the so-called BPQS scheme [7]. BPQS is actually a modified XMSS scheme, using a single authentication path (i.e. a chain and not a tree). The researchers in [7] suggest thar BPQS fits well with blockchains.

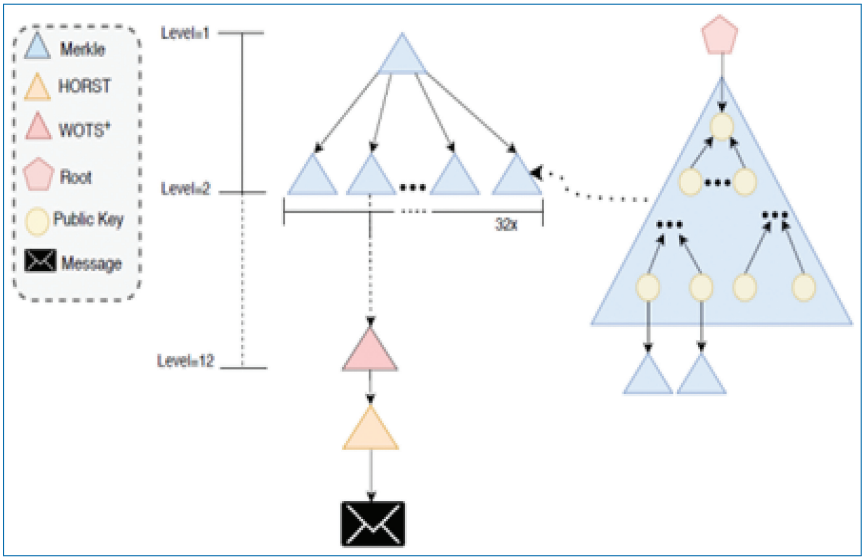


Figure 7.4. Hypertree structure used in SPHINCS [9].

Stateless Hierarchical Signature Schemes (HS): The mail property of stateful hierarchical signature schemes is that the signing process requires the renewal of the secret key. In other words, for stateful signature schemes, signing requires keeping state of the used one-time keys and making sure they are never reused. However, there are also stateless hierarchical signature schemes, with the most prominent example being the SPHINCS [8] and its variants SPHINCS-Simpira, Gravity-SPHINCS and SPHINCS+. Similar to XMSS^{MT}, SPHINCS uses a hypertree such that the upper layers use XMSS with WOTS+ to sign roots of their ancestors, while the lowest layer uses a Merkle tree construction with HORS-T for signing messages (as shown in Figure 7.4). Since the stateless schemes do not keep a record of used key pairs, hence to ensure the correct few-time usage of key pairs, SPHINCS deploys multiple HORS-T key pairs and selects a random one for each signature generation (HORS-T are few times – instead of one time – signature primitives (FTS)). Hence, no path-state tracking is required.

In stateless schemes such as the SPHINCS, generating all private (HORS-T and WOTS+) keys with a PRNG and computing one tree in each layer for signature generation results in an efficient computation. Nevertheless, stateless schemes pose the following performance issues. First, the signature generation is more expensive because the key pairs are used in random order rather than successive order; hence, several optimization algorithms that are being used in stateful schemes are not applicable. Moreover, in contrast to WOTS+, HORS-T signatures are relatively much larger [9]. Note that Table 7.1 also provides relevant information on HORS-T, as an FTS primitive, compared to OTS primitives. A summary between the discussed

Table 7.2. Comparison between stateful and stateless signature schemes in [9].

Signature Scheme	Instantiation	Base Scheme	Key Re-use Capability	Signature Size (Kb)	Key Size (Kb)
MSS	SHA-384	WOTS	2^{60}	7.7	0.05
XMSS	SHA-256	$WOTS^{PRF}$	2^{60}	4.7	0.03
$XMSS^{MT}$	AES-128	$WOTS^{PRF}$	2^{80}	10.7	Private key = 26.1 Public key = 1.8
SPHINCS	SHA-256	HORS-T & WOTS+	Unlimited	41.0	1.0

stateless (SPHINCS) and stateful (MSS, XMSS, $XMSS^{MT}$) HBS schemes is given in Table 7.2, whereas an overall evaluation, is given in Table 7.3.

Even though post-quantum security is considered to be present in HBS schemes, all the potential attack surface should be also examined, mainly stemming from implementation attacks – i.e., side channel attacks and fault attacks. In a side-channel attack, the attacker gains extra critical information (i.e., relative to a secret key) by monitoring and/or measuring quantities such as power consumption, electromagnetic leaks, timing for performing an execution etc. In a fault attack, a fault, which can be either natural or malicious, is misbehavior of a device that causes the computation to deviate from its specification, which could also yield some information on the secret key. HBS schemes are vulnerable to hardware fault attacks both in the presence of natural and malicious faults, so special attention should be given on appropriately implementing such schemes. Moreover, another problem in the stateful signature schemes is the so-called cloning. Such a threat occurs whenever a private key is copied and then used without coordination with execution units (known as non-volatile cloning) or without coordination with storage units, known as volatile cloning.

Some researchers consider XMSS and SPHINCS to be impractical for blockchain applications due to their performance (relatively slow signing speed, whereas the size of the signature in SPHINCS is 41kb), so alternatives have been suggested.

Code-based digital signatures

Several post-quantum code-based signing algorithms have been proposed; probably the most known are the schemes from Niederreiter and CFS (Courtois, Finiasz, Sendrier), which are similar to the McEliece's cryptosystem. The signatures of such schemes are short in length and can be verified really fast, but similarly to

Table 7.3. An overall generic evaluation of stateful and stateless HBS schemes [9].

Type	Pros	Cons	Use Case
Stateful	– Shorter signature size – Faster signature generation time	– State synchronization problem synchronization failure – Face cloning problem	Performance-constrained environment
Stateless	– No state synchronization. problem – No cloning problem	– Longer signature size – Slower signature generation time	Resource-constrained environment

the McEliece's cryptosystems, the use of large key sizes requires significant computational resources and, as a consequence, signature generation may become inefficient [10].

Multivariate digital signature schemes

This class of post-quantum signatures typically yields large public keys, but very small signatures. Some of the most popular multivariate-based schemes rely on Matsumoto-Imai's algorithm or on variants of HFE, which can generate signatures with a size comparable to the currently used RSA or ECC-based signatures. Other relevant multivariate-based digital signature schemes have been proposed, like the Rainbow. In general, it is widely assumed that such cryptosystems need to be further improved in terms of key size.

Lattice-based digital signature schemes

Among the several lattice-based signature schemes described in the literature, the ones based on Short Integer Solution (SIS) seem to be promising due to their reduced key size. For several years, it was assumed that BLISS-B (Bimodal Lattice Signatures B), whose security rests with the hardness of the SIS problem, could be a very nice option due to its good performance. However, it is found out that BLISS is vulnerable to side-channel attacks [10]. Besides BLISS, there are in the literature other lattice-based signature schemes that rely on the SIS problem but that were devised specifically for blockchains [11]. Moreover, lattice-based blind signature schemes have been used to provide anonymity and untraceability in distributed blockchain-based applications for the IoT.

Isogenies digital signature schemes

Although supersingular elliptic curve isogenies can be used for creating post-quantum digital signature schemes, there are not many such schemes known, whereas they also are not efficient. Some schemes of this class indicate though that "it is necessary to address key size issues when implementing isogeny-based cryptosystems and Supersingular Isogeny Diffie-Hellman (SIDH), especially in the case of resource constrained devices".

Zero-knowledge proofs for digital signatures

There is one important post-quantum digital signature scheme, called Picnic, which has a significantly different design principle compared to all the previous. Picnic, which is submitted to the NIST competition, is based on non-interactive zero-knowledge proofs, where the proof of knowledge is instantiated using the MPC-in-the-head approach. The signature is a proof of knowledge of a secret key for a block cipher that encrypts a public plaintext block to a public ciphertext block, which together form the public key of the signature scheme. All the cryptographic building blocks can be instantiated using symmetric-key primitives (block ciphers and hash functions), whereas the MPC (Multi-Party Computation) protocol can be instantiated with information-theoretic security.

7.3 Blockchain and Post Quantum Cryptography

To tackle the quantum threat in the blockchain technology, several researchers have proposed post-quantum-enabled blockchain solutions or even some adjustments to popular distributed leaders. Commercial blockchains have also analyzed and addressed the impact of quantum computers. These include the Quantum Resistant Ledger (QRL) which uses XMSS, the IOTA which uses WOTS and Corda which uses BPQS.

7.3.1 Bitcoin

The platform Bitcoin uses the ECDSA with the Koblitz curve secp256k1 algorithm and the hash function SHA-256 to authorize the transferring of coins and assets. Defined by the Standards for Efficient Cryptography Group (SECG), the Koblitz curve provides several advantages, such as efficiency, reduction of the key size and security, but the main drawback is its weakness against the quantum attack. Therefore, to secure the digital signatures that are included in Bitcoin transactions against the Shor's algorithm, the authors in [13], implemented a signature scheme based on the TESLA# algorithm, which uses the BLAKE2 and the SHA-3 functions,

hence yielding a fast signing and verifying signing scheme. However, qTESLA is not present in the third round of evaluation in the NIST competition.

The research of lattice-based cryptography, which lays the foundation for the design of anti-quantum attack signature scheme, is not only fruitful to resist the quantum threat, but it is also suitable for blockchains. Therefore, the authors in [14] proposed a transparent e-voting blockchain system, which could be applied in Bitcoin. In this scheme the voters that operate maliciously are audited, while code-based cryptography is used to resist quantum threats. More precisely, a certificate-less traceable ring signature algorithm is introduced in the proposed blockchain-enabled e-voting system to solve the problem of verifying public key certificates and the Niederreiter's code-based cryptosystem is adopted to address the quantum threat in the e-voting protocol.

7.3.2 Ethereum

The authors in [15] proposed a framework that encrypts and sensitive industrial data, while the uploader decides with whom this data can be shared with. The architecture is modeled to operate with the popular Ethereum platform and the Inter Planetary File System (IPFS). However, similar and traditional platforms are also able to provide the necessary requirements for the framework's operation. The framework uses the Elliptical-Curve Diffie-Hellman Key Exchange (ECDH) and the SIDH algorithms. Thus, the advantages and drawbacks of each algorithm is discussed in that paper, concluding that SIDH is the most suitable approach because it is post-quantum secure and it ensures security against attackers with quantum computing capabilities. The Ethereum platform is also modified in [16], in which paper, the authors applied a multivariate-based cryptosystem (the Rainbow signature scheme) and compared its efficiency with the current version of Ethereum, which is based on the ECDSA.

7.3.3 IOTA

IOTA is a popular distributed ledger designed for the IoT ecosystem. The platform is considered as a quantum resistant, rather than as a quantum-proof ledger. In particularly, is does not use conventional public key cryptography, but the IOTA Signature Scheme (ISS) that is based on WOTS. In this platform, the users in IOTA sign the message's hash, which means that the security of ISS is based on the cryptographic strength of the hash function. Therefore, IOTA transactions are quantum resistant, but require a new private/public key to be generated each time that a transaction is being signed with the private key, because a part of the private key is revealed in the signature process.

7.3.4 QRL

While designing the QRL, great emphasis has been given to the cryptographic security of its signature scheme, in order to be secure against both classical and quantum attacks, not only at the present day, but also in the future decades. QRL replaces secp256k1 with XMSS, using the hash function SHA-256 and offers 196-bit security with expected security against the brute force attack until the year of 2164. The asymmetrical hypertree signature scheme that is being used in QRL is consisted by chained XMSS trees and provides the dual advantage of using a validated signature scheme and the permission of generating ledger addresses with the capability of signing transactions without a pre-computation delay that is observed in XMSS constructions.

7.3.5 Corda

Corda typically supports conventional public key signature algorithms, such as ECDSA and RSA (the default signature is ECDSA with NIST P-256 curve – i.e., secp256p1). However, at an experimental level, SPHINCS has been employed towards providing post-quantum security. Moreover, very recently, researchers from R3 (i.e. the company supporting Corda) proposed the aforementioned BPQS signature scheme, forming an improvement of the XMSS (and, actually, the blockchain by itself plays such a role, thus comprising a blockchained signature scheme).

7.3.6 Hyperledger Fabric

The Hyperledger Fabric does not provide (by default) post-quantum security. However, it has been announced that achieving post-quantum security is one of the priorities with respect to further advancements of the ledger. To this end, such an approach has been very recently suggested in a research paper [17]. The researchers present the so-called PQFabric, which is the first version of the Hyperledger Fabric enterprise permissioned blockchain whose signatures are secure against both classical and quantum computing threats. In this paper, the researchers implement and analyze hybrid signatures that are configurable with any post-quantum signature algorithm.

The authors redesign the credential-management procedures and specifications of the Fabric network and they created hybrid signatures that are a combination of the classical and quantum-safe digital signatures. The comparative benchmarks of PQ-Fabric are performed with some of the NIST candidates and alternates, namely Falcon-512, Falcon-1024, Dilithium-2, Dilithium-3, Dilithium-4 and qTesla-p-I.

The proposed system is built on-top of Fabric v.1.4 and the LIBOQS v0.4, which is used for the implementation of the post-quantum cryptographic algorithms.

The integration presented in [17], was not straightforward, and therefore three core modules of the Fabric's codebase were modified to allow the incorporation of hybrid quantum signatures, (1) the Blockchain Cryptographic Service Provider (BCCSP) that offers the implementation of a uniform interface. This interface calls the relevant signature scheme based on the key type that is being used; (2) the local Membership Service Provider (MSP) that extracts the cryptographic keys, both public and private – since the hybrid quantum-classical cryptography needs two keys – from the X.509 certificate; and (3) the cryptogen, which is a template used to create the cryptographic material needed to run the Fabric platform from its configuration files. Therefore, the modified MSP obtains the private and public keys from the X.509 certificate, stores them for each node in an internal structure and then provides them to the BCCSP module every time that a message is signed. The signature scheme simple allows the LibOQS to re-hash the already hashed message, but this action has a cost for the platform's performance. Particularly, the speed of the signature algorithm is the key factor that impacts the performance of schemes with larger signature sizes and keys.

7.4 Performance and Resistance of Potential Blockchain Post-Quantum Cryptosystems

7.4.1 Performance Assessment

The performance of post-quantum digital signatures has been extensively studied in the literature. Such a performance evaluation has been considered with respect to several underlying hardware platforms, as well as, in several networking protocols with several assumptions on the underlying communication channel. In the case of FALCON, the authors measured its performance in terms of spent time instead of cycles. For Rainbow, the values indicate the performance of the key-compressed version that require much more computational effort than the regular version due to the involved decompression process. However, most cryptosystems have been evaluated after optimizing them for AVX2, a 256-bit instruction set provided by Intel. The only exception is the performance of SPHINCS for the HARAKA version, whose optimized version was implemented to take advantage of the AES-NI instruction set.

It is interesting to point out that this performance evaluation presented in Table 7.4 is based on appropriate hardware that can be used for running both a regular blockchain node (i.e., a node that only interacts with the blockchain) or a

Table 7.4. An overall performance evaluation on post-quantum signatures being present in the 3rd round of NIST evaluation [19].

Scheme	Algorithm	Execution Time (ms)	Size (Bits)
Dilithium	Dilithium II	$KeyGen = 0.18$ $Sign = 0.82$ $Ver = 0.16$	$K_s = 22,400$ $K_p = 9,472$ $\sigma = 16,352$
Falcon	Falcon-512	$KeyGen = 16.77$ $Sign = 5.22$ $Ver = 0.05$	$K_s = 10,248$ $K_p = 7,176$ $\sigma = 5,52$
Rainbow	Rainbow-Ia-Cyclic	$KeyGen = 0.48$ $Sign = 0.34$ $Ver = 0.83$	$K_s = 743,680$ $K_p = 465,152$ $\sigma = 512$
GeMSS	GeMSS128	$KeyGen = 13.1$ $Sign = 188$ $Ver = 0.03$	$K_s = 107,502$ $K_p = 2,817,504$ $\sigma = 258$
Picnic	Picnic-L1-FS	$KeyGen = 0.005$ $Sign = 4.09$ $Ver = 3.25$	$K_s = 128$ $K_p = 256$ $\sigma = 272,256$
SPHINCS+	SPHINCS+ – SHA256 – 128f – simple	$KeyGen = 2.95$ $Sign = 93.37$ $Ver = 3.92$	$K_s = 512$ $K_p = 256$ $\sigma = 135,808$

full blockchain node (i.e., a node that stores and updates periodically a copy of the blockchain and that is able to validate blockchain transactions).

The conclusions derived can be summarized as follows: first, with respect to multivariate-based cryptosystems, MQDSS provides small keys, its lightest version is quite fast, but the sizes of its signatures are among the largest in the comparison (whereas other multivariate schemes have large sizes. In contrast, the rest of the compared multivariate-based schemes have keys with large sizes, but they generate short signatures; note also that MQDSS does not continue in the third round.

Next, with respect to lattice-based signatures, they generally require smaller keys than the multivariate schemes, but they produce larger signatures. Amongst all of them, FALCON – which continues to the third round of the NIST competition – makes use of the smallest key sizes and signature lengths. qTESLA is also fast, but its major drawback is the large key sizes; qTESLA is not present in the third round of evaluation in the NIST competition. The fastest scheme is Dilithium (amongst all the types of post-quantum signatures – not only amongst lattice-based). DILITHIUM obtains, in terms of performance, very similar results

Table 7.5. Time (ms) of key-pair generation, signing and verification [7].

Scheme	KeyGen	Sign	Verify
BPQS ($w = 4$, *SHA256)*	0.569	0.08	0.10
BPQS ($w = 4$, *SHA384)*	1.107	0.16	0.19
BPQS ($w = 16$, *SHA256)*	0.872	0.19	0.20
BPQS ($w = 16$, *SHA384)*	1.719	0.39	0.38
ECDSA SECP256K1 (SHA256)	0.10	0.34	0.25
Pure EdDSA Ed25519 (SHA512)	0.18	0.08	0.16
RSA3072 (SHA256)	561.1	5.39	0.17
SPHINCS-256 (SHA512)	0.69	144.5	1.76

to ECDSA-256. Unfortunately, DILITHIUM key sizes are much larger than the ones used by ECDSA-256.

However, apart from Dilithium, another option that achieves good performance is the lightest version of the Rainbow. This is also verified, apart from the aforementioned results in [10], in the evaluation over the TLS protocol [18]. Note also that Rainbow necessitates smaller parameters than Dilithium, thus rendering the algorithm a very strong candidate for future (including blockchain) applications. Falcon provides the best verification time, but it is slow in signing. The slowest digital signature algorithms are Picnic, GeMSS and SPHINCS (all of them are alternate algorithms in the NIST competition).

In order to summarise the results (in terms of performance), we illustrate the performance results of the candidates (and the alternates) in the third round of NIST (see Table 7.4). This table is based on the results from [18], which are in fully compliance with the survey presented in [10].

As stated above, SPHINCS is generally a very slow signing algorithm. It is interesting to point out though that the BPQS, being also hash-based (and outside of the NIST competition) suffices to achieve better performance than SPHINCS, whereas it is blockchain oriented. This is illustrated in Table 7.5. It can be seen that, despite the relevant parameters of BPQS, it is much faster than SPHINCS in terms of signing and verifying (with performance actually comparable to traditional public key digital signature schemes). The main drawback is the key generation time, which however is comparable, in some cases, with the SPHINCS. Regarding the signature size, all BPQS modes outperform XMSS for the first number of signatures. However, BPQS signatures grow linearly with the number of times a key is reused and, thus the length of the signature output is dynamic (it starts small and increases per additional signature).

Table 7.6. Time for generating XMSS trees for a QRL wallet [20].

XMSS Tree Height	No. of OTS Signatures	Hash Function/Algorithm	Gen. Time
18	262.144	SHA2_256 / SHA2	1h 10min 49sec
10	1.024	SHAKE_128 / SHA3	11sec
12	4.096	SHA2_256/ SHA2	1h 20sec
12	4.096	SHAKE_128/ SHA3	48sec
12	4.096	SHAKE_256/ SHA3	46sec

Table 7.7. Information on transactions in QRL [20].

Transaction Size (Bytes)	Signing Time	Signature Size (Bytes)	Verification Time	Block #	Block Size (Bytes)
2662	1sec	2500	4min 36sec	81188	2915
2662	1sec	2500	9sec	81168	2915
2662	1sec	2500	3min 0sec	80944	2915
2704	–	2500	–	80939	2958
2662	1sec	2500	1min 2sec	80205	2915
2662	1sec	2500	24sec	66804	2915
2705	–	2500	–	66739	2959

It is also interesting to focus more carefully on XMSS, and especially on the QRL – which is a ledger supporting XMSS for achieving, by default, post-quantum security. It is known that XMSS has several limitations (and that's why SPHINCS and BPQS are considered as improvements of XMSS); however, XMSS is indeed one cryptographic primitive that is currently used in a post-quantum secure commercial blockchain.

We next present recent experimental results on QRL, aiming to see in practice the performance of QRL (implementing XMSS) in a conventional workstation [20]. The experiments have been conducted in an Intel Core2Duo E6750 @ 2,66GHz processor, with 6 Gb RAM (DDR2 @ 400MHz) and Windows 10 Pro, 64 bit, as an operating system. To perform several measurements, the researcher produced several different wallets with different parameters for the XMSS. The results are shown in Table 7.6.

Moreover, the researcher in [20] proceed in performing several transactions in a testing environment (provided by the QRL), with the ultimate goal to see in practice the corresponding signing and verification times. This is shown in Table 7.7, for

the second wallet. As it is shown in this table, the size of the signature is constant, which is expected since the size of the signature is related with the height of the XMSS tree (or, equivalently, with the number of the OTS signatures). More precisely, in QRL the size of the signature is given by the relation $2180 + (height * 32)$ bytes. The variations in verification time are probably due to the load of the miner in the tested blockchain and the experiments tool placed.

7.4.2 Attacks on PQC Primitives

As NIST has stated the importance of side channel attacks (SCA) and countermeasures. More precisely, in the original NIST PQC call for proposals in 2016, it was stated that *"the Schemes that can be resistant to SCA at lower cost are more preferable than those whose performance is severely hampered by any attempt to resist side-channel attacks."* NIST also hopes to see implementations that will have protective mechanisms against side-channel attacks, such as timing attacks, fault attacks, power monitoring attacks, etc. Therefore, in this section, it is presented a number of SCA and ISD attacks against the NIST PQC 3rd round candidates.

These attacks on the NIST's 3rd round candidates are categorized as:

- Classical Cryptanalysis (CC), which mathematically analyses the corresponding cryptosystem.
- Static Timing Analysis (STA), which manipulates variable runtime of an algorithm.
- Fault Attacks (FA), which are semi-invasive techniques to deliberately induce faults and disclose cryptographic internal states.
- Simple Power Analysis (SPA) and Advanced (differential/correlation) Power Analysis (APA), which non-invasively exploits the variations in the cryptographic algorithm's power consumption.
- Electromagnetic attacks (EMA), which exploit the radiation from a cryptographic algorithm.
- Template attacks (TA) that use a sensitive device to obtain access to the secret.
- Cold-boot attacks (CBA), which exploit the memory remanence to read data out of a computer's memory when the computer has been turned off.
- Countermeasures (CM) that protect/hinder attacks through masking or hiding techniques.

Therefore, the next table (Table 7.8) presents which schemes are directly susceptible on the aforementioned attacks.

Table 7.8. A summary of attacks on NIST PQC 3rd round candidates.

			SCA								
		Algorithm	**CC**	**STA**	**FA**	**SPA**	**APA**	**EMA**	**TA**	**CBA**	**CM**
Finalists	KEMs	Classic McEliece,			✓			✓		✓	
		Kyber			✓	✓		✓	✓	✓	
		NTRU				✓				✓	
		Saber						✓			✓
	Signs	Dilithium			✓			✓			✓
		Falcon			✓						
		Rainbow	✓				✓			✓	
Alternatives	KEMs	BIKE		✓	✓						
		FrodoKEM		✓		✓	✓	✓	✓	✓	✓
		HQC		✓			✓				
		NTRU Prime					✓		✓		✓
		SIKE	✓	✓							
	Signs	GeMSS	✓				✓				
		Picnic	✓				✓				
		SPHINC+			✓						

7.5 Conclusions and Future Directions in PQC Blockchains

This chapter considered the post-quantum security aspects in blockchain technology. More precisely, it has assessed contemporary PQC algorithms and the current situation of the NIST's 3rd round PQC candidates. In addition, it has presented the impact of quantum-computing attacks on blockchains and it has investigated the incorporation of PQC primitives in blockchains.

Currently, quantum computing is an area that has gained a lot of interest from both the academia and the industry. Sequentially, new attacks might be developed against the post-quantum cryptosystems. Therefore, it is necessary that both researchers and industry to be aware to the quantum computing area and its advances and for this reason, we present the challenges and the future directions in PQC blockchains.

7.5.1 Transitioning to Post-quantum Blockchains

The transition to post-quantum blockchains necessitates the involved steps to be considered carefully. Therefore, several researchers have discovered new methods for the implementation of post-quantum security to the blockchain technology. For example, in [21] the authors introduced a scheme that extends the validity of the blockchain, if the security of the digital signatures or of the hash functions is imperiled. However, hard forks or smooth-forks might occur and for this case, the authors proposed a soft-fork mechanism [22]. In another work [23], a commit–delay–reveal protocol is proposed that enables the Bitcoin users to move funds from the non-quantum-resistant protocol to a version that adhere to a quantum-resistant signature scheme. This transition protocol can work well even if the ECDSA has been formerly compromised.

7.5.2 Keys – Signature Sizes and Performance Challenges

The key's sizes in post-quantum cryptosystems are among 128 and 4,096 bits, meaning that the post-quantum cryptosystems demand key's sizes much larger than the public key cryptosystems. Some signature cryptosystems, which are based on supersingular isogenies, appear to be auspicious to solve the key size issue, but such schemes generate large signatures and provide pour performance compared to the public key cryptosystems. As one issue is seemingly solved several others are created, since the blockchains store a vast number of signatures. In a similar way, the hashed-based cryptosystems have comparatively small key sizes, which comes to contradiction with the size of their signatures, which is often more than 40 KB. On the other hand, the majority of the multivariate-based cryptosystems generate short signatures, but the keys used for their generation and verification might need several kilobytes. The lattice cryptosystems, which are based on DILITHIUM are very fast, but their signature length is 2701 bytes and their key size is approximately 1500 bytes.

The post-quantum cryptosystems need a considerable amount of (a) execution time, (b) computational and (c) storage resources. To some extent, some schemes reduce the number of the signed messages with the same key. This practice results to the generation of new keys repeatedly and to the dedication of the computational resources for this purpose that could be otherwise used for certain blockchain processes. Nevertheless, the current research in post-quantum cryptosystems is not adequate for having a good trade-off among the size of the keys and the scheme's performance for the blockchains. Therefore, novel approaches are required, which will minimize the cryptosystems' energy consumption and therefore, the performance of the blockchain network.

7.5.3 General Directions

A large distributed network, such as the blockchain, necessitates exceptional consideration when migrating to a post-quantum cryptography, due to the limitations of the downtime and the synchronous update. Such transitions require not only performance assurance and backwards compatibility, but also slow rollouts and rollbacks. Therefore, a post-quantum implementation of a blockchain network requires the following steps:

I. Software rollout: A slow rollout of the software to all the network's peers. This migration should be backwards compatible, with the nodes to be able to continuously sign and verify signatures, as well as, to validate X.509 certificates classically until they change to a post-quantum mode.
II. Key rollover: While the certificate authority will be modified with a post-quantum key, the node certificates should be re-issued following a key rollover method.
III. Slow rollout of the PQC keys: When the key-pairs of post-quantum keys will be generated, the configuration files of each node that belongs to the network should be updated.
IV. The final step will be the rollout of post quantum keys to the client peers.

Therefore, all the above steps should be taken into consideration when implementing post-quantum digital signatures or encryption algorithms to a blockchain platform.

Acknowledgment


This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] S. Brotsis, K. Limniotis, G. Bendiab, N. Kolokotronis and S. Shiaeles, "On the suitability of blockchain platforms for IoT applications: Architectures, security, privacy, and performance," *Computer Networks*, p. 108005, March 2021.
[2] B. Sotirios, K. Nicholas, L. Konstantinos and S. Stavros, "On the Security of Permissioned Blockchain Solutions for IoT Applications," *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pp. 465–472, 2020.

[3] S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiaeles, D. Kavallieros and E. Bellini, "Blockchain Solutions for Forensic Evidence Preservation in IoT Environments," *IEEE Conference on Network Softwarization (NetSoft)*, pp. 110–114, 2019.

[4] N. Kolokotronis, S. Brotsis, G. Germanos, C. Vassilakis and S. Shiaeles, "On Blockchain Architectures for Trust-Based Collaborative Intrusion Detection," *2019 IEEE World Congress on Services (SERVICES)*, pp. 21–28, 2019.

[5] R. C. Merkle, "A certified digital signature," *Conference on the Theory and Application of Cryptology*, pp. 218–238, 1989.

[6] J. Buchmann and E. A. H. A. Dahmen, "XMSS-a practical forward secure signature scheme based on minimal security assumptions," *International Workshop on Post-Quantum Cryptography*, pp. 117–129, 2011.

[7] K. Chalkias, "Blockchained post-quantum signatures," *Cryptology ePrint Archive: Report 2018/658*, 2018.

[8] Bernstein, M. Schneider, P. Schwabe and Z. Wilcox-O'Hearn, "SPHINCS: practical stateless hash-based signatures," *Annual international conference on the theory and applications of cryptographic techniques*, pp. 368–397, 2015.

[9] S. Suhail, R. Hussain, A. Khan and C. S. Hong, "On the role of hash-based signatures in quantum-safe internet of things: Current solutions and future directions," *IEEE Internet of Things Journal*, 2020.

[10] T. M. Fernández-Caramès and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," *IEEE Access*, vol. 8, pp. 21091–21116, 2020.

[11] Y.-L. Gao, X.-B. Chen, Y.-L. Chen, Y. Sun, X.-X. Niu and Y.-X. Yang, "A secure cryptocurrency scheme based on post-quantum blockchain," *IEEE Access*, vol. 6, pp. 27205–27213, 2018.

[12] A. Coladangelo, "Smart contracts meet quantum cryptography," *arXiv preprint arXiv:1902.05214*, 2019.

[13] M. C. Semmouni, A. Nitaj and M. Belkasmi, "Bitcoin Security with Post Quantum Cryptography," *Networked Systems*, pp. 281–288, 2019.

[14] W. Yin, Q. Wen, W. Li, H. Zhang and Z. Jin, "An anti-quantum transaction authentication approach in blockchain," *IEEE Access*, vol. 6, pp. 5393–5401, 2018.

[15] J. Preece and J. Easton, "Towards encrypting industrial data on public distributed networks," *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4540–4544, 2018.

[16] R. Shen, H. Xiang, X. Zhang, B. Cai and T. Xiang, "Application and Implementation of Multivariate Public Key Cryptosystem in Blockchain (Short Paper)," *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 419–428, 2019.

[17] B. Das, A. Holcomb, M. Mosca and G. C. Pereira, "PQ-Fabric: A Permissioned Blockchain Secure from Both Classical and Quantum Attacks," *arXiv preprint arXiv:2010.06571*, 2020.

[18] D. Sikeridis, P. Kampanakis and M. Devetsikiotis, "Post-Quantum Authentication in TLS 1.3: A Performance Study.," *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 71, 2020.

[19] T. G. Tan, P. Szalachowski and J. Zhou, "SoK: Challenges of Post-Quantum Digital Signing in Real-world Applications".*eprint.iacr.*

[20] N. Sakellion, "Post-quantum cryptography in blockhain technologies," Cyprus, 2020.

[21] M. Sato and S. Matsuo, "Long-term public blockchain: Resilience against compromise of underlying cryptography," *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, 2017.

[22] F. Chen, Z. Liu, Y. Long, Z. Liu and N. Ding, "Secure scheme against compromised hash in proof-of-work blockchain," *International Conference on Network and System Security*, pp. 1–15, 2018.

[23] I. A. I. D. Stewart, A. Zamyatin, S. Werner, M. Torshizi and W. J. Knottenbelt, "Committing to quantum resistance: a slow defence for Bitcoin against a fast quantum computing attack," *Royal Society open science*, vol. 5, no. 6, p. 180410, 2018.


IoT

DOI: 10.1561/9781680838350.ch8

Chapter 8

Trust Management System Architecture for the Internet of Things

By C.-M. Mathas, C. Vassilakis†, N. Kolokotronis‡ and K.-P. Grammatikakis§*

University of the Peloponnese
*mathas.ch.m@uop.gr
†costas@uop.gr
‡nkolok@uop.gr
§kpgram@uop.gr

The Internet of Things has enabled the interconnection of billions of devices, which cooperate to support a large number of applications and application features. In this context, the number of the devices that need to interact to realize the desired functionalities has substantially grown, and this has rendered traditional access control methods hard to manage and ineffective. To respond to this challenge, trust-based access control has emerged, where each device is assigned a level of trust, and this level is consulted to determine whether data and operation accesses should be permitted or declined. In this chapter, we propose an approach to trust computation in the Internet of things, which synthesizes behavioral, device status and associated risk aspects into a comprehensive trust score, that can be consulted to realize trust-based access control. The proposed approach also considers device ownership

relationships and owner-to-owner trust relationships, which are utilized in the trust computation process.

8.1 Introduction: Background and Driving Forces

In the context of computing, parties interact with each other to access services and information. Traditionally, access control mechanisms are employed to safeguard such accesses: authentication mechanisms provide the necessary guarantees about the identities of the interacting parties (i.e., that either the service/information requestor or the server are indeed who they claim they are), whereas authorization mechanisms enforce information/service access policies, ensuring that only authorized clients can access the information/service resources provided by the servers. While this approach is adequate for a number of information system use cases, and predominantly in client-server systems where a closed set of clients or client groups interact with a limited set of servers that are known *a priori*, modern internet-scale computing necessitates the interaction between unknown parties, with each party being able both to request and offer services and/or information. In such an environment, traditional access control systems are deemed insufficient, since interacting parties are highly likely to be unknown to each other before the beginning of the interaction. In this respect, a different approach is needed to allow interacting parties to decide:

1. Whether the requestor is entitled to access the service/information requested and
2. Whether the provider is trusted as a source of the particular service/information.

To address the issues listed above, the concept Trust management has been introduced. The authors in [1] define trust management as an underpinning that facilitates the enforcement of security policies by verifying actions against these policies, in an automated fashion. Following this definition, the execution of an action is permitted if the interacting party has provided credentials that are assessed to be sufficient; if this holds, the interacting party's actual identity need not be known or verified. In other words, the checks made need only to process and verify some symbolic representation of the requesting party's trust level, which is now clearly distinguished from the requesting party itself (a person or an agent acting on behalf of the person). To further promote the benefits of the trust-based approach, the presentation and validation of credentials can be replaced by the inspection

and assessment of a *set of properties*, which are testified for and validated by some interacting party, while digital certificates are used to represent the aforementioned properties and safeguard their validity [2–4].

Following this rationale, the initial collection of trust management system elements listed in [4] is revised as described below:

1. *Security policies*, which comprise a group of trust assertions that are regarded as "ground truth" and are therefore trusted in all cases.
2. *Trust-related properties*, which represent characteristics of communicating parties that are pertinent to the enforcement of security policies; typically, such properties are examined as antecedents of rules that comprise a security policy. Trust-related policies are safeguarded through digital signatures or other prominent means.
3. *Trust relationships*, which are a special kind security policy.

While the scheme presented above explicitly lists two interacting parties, i.e., the service/information requestor and server, trust establishment may involve more parties, resulting in a highly decentralized model: firstly, trust-related properties may be (and typically are) provided and testified for by third parties. Secondly, *trust relationships* may designate other trust management system entities with which a trust management system instance liaises to exchange any of the system elements listed above (security policies, trust-related properties or trust relationships), including also trust assessments that can be taken into account when a trust management system instance assesses the trust level of an interacting party.

The trust level of an interaction peer may be computed by taking into account all its observable characteristics: this includes (a) *the security characteristics of the interaction peer*, along with the current evaluation of the peer's integrity assessment (possible compromise of firmware, operating system, system files; security patch version; etc.) and security defenses employed by the device (firewalls; IDS/IPS; etc. [5]) and (b) *behavioral characteristics of the interaction peer*, relating to whether the interaction peer (i) functions in compliance to its predefined usage description and (ii) exhibits abnormal behavior.

Services, information and resources are actually *assets* which hold a *value* for their respective owners and thus necessitate protection through trust management or other pertinent means. Protection aims to safeguard assets from a number of *threats*, which manifest risks against them, and may ultimately lead to the demotion of their value [5]. As a result, the process of protecting the assets must incorporate a risk assessment of each interaction, and the choice and application of the appropriate defensive measures as dictated by the assessment's results. This is in line with the

procedure described in the ISO/IEC 27001 standard [6] for addressing risks, which encompasses the following two steps:

1. *information security risk assessment*, which is further refined in (i) establishment and maintenance of information security risk criteria that include the risk acceptance criteria (ii) identification of information risks and (iii) analysis of information security risks and (iv) evaluation of information security risks and
2. *information security risk treatment*, where (i) suitable options for mitigating information security risks are chosen, after considering the outcomes of risk assessment, (ii) appropriate controls for the realization of the chosen security risk treatment options are chosen, taking also into account the cost/benefit ratio of applying the chosen security risk treatment options and (iii) the information security risk treatment approach is validated, after reviewing any residual information security risks and knowledgeably accepting their presence (or returning to the step of choosing appropriate controls).

Trust and risk assessment are two closely associated concepts, following the rationale that the evaluation of information security risks involves a calculation of the probability that the risks in question will occur [6], and the result of this calculation is dependent on the trust level that is assigned to systems that could prove to be threat agents. This rationale is reflected on definitions of trust found in the literature: according to [7] "Trust is the willingness of a party to be vulnerable to the action of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective to the ability to monitor or control that other party"; on the same note, [8] defines trust as "An attitude of confident expectation in an online situation of risk that one's vulnerabilities will not be exploited". These lead us to the conclusion that trust reduces the level of risk, based on the conviction that a trusted system will not ultimately operate as a threat agent. Overall, a system's trust assessment must be incorporated as a critical parameter of a risk assessment.

Finally, attackers are increasingly employing more complex attack methods which include multi-stage, multi-host attack paths, with each path representing a series of exploits utilized by the attacker to compromise a network [10]. To this end, attack graphs can be employed to perform a comprehensive risk analysis of a network, by taking into account the cause-consequence relationships involved in a network's shifting states. Furthermore, the probability of the exploitation of such relationships can also be considered [9].

In this chapter, we firstly overview existing trust- and risk-based approaches to security, and identify areas of improvement, with a special focus on the domain of

the internet of things. Subsequently, we present an approach for trust computation, which synthesizes different aspects into a single, comprehensive trust score that can be used for applying trust-based access control. We also describe an architecture for realizing the proposed approach.

8.2 Fundamentals of Trust Management

In this section we will overview the three main foundations of trust and risk management namely (a) *behavioral-based methods*, focusing on the observed interactions of the devices, (b) *status-based methods*, focusing on the devices' security aspects and (c) *risk assessment-oriented methods*, focusing on the quantification of the risk associated with the devices and operations. For each of the three foundations, we present methods, tools and information sources that can be employed for realizing trust and risk management in the relevant context.

8.2.1 Behavioral Aspects

The behavior of a device can be monitored and used in the process of trust and risk assessment. The term "behavior" in this context refers to the observable activities performed by the device, and this predominantly includes network traffic directed towards other nodes. This network traffic can be:

- *Compared against a predefined static model of behavior* that has been specified for the device and prescribes the operation of a benign instance of the device. Deviations from the prescribed behavior are then treated as indications of malicious behavior and demote the trust level, increasing correspondingly the risk level. Manufacturer Usage Description Specification files [11] are the main tool in this area.
- *Compared against a dynamically built model of behavior for the device*; under this approach, the behavior of the device instance is profiled at a state that is known to be benign, and further behavior is compared against the baseline within the profile. Deviations from the baseline are flagged as anomalies, reducing the trust level and increasing the associated risk. Provisions for dynamic evolution of the profile can be made.
- *Matched against a known set of malicious requests*. Under this approach, the network traffic emanating from the device is matched against a malicious requests signature database, to identify whether the device is the source of attacks to other devices; if so, it can be concluded that the device has been compromised, and consequently trust and risk assessments are adjusted accordingly.

Another aspect that can be taken into account at this point concerns the observable consequences of information flows, rather than the information flows themselves. Under this viewpoint, information that has leaked from a device (e.g., user passwords or personal data) constitutes evidence that the device does not provide an adequate level of security (including the case that it discloses information to entities that should not be trusted), and on these grounds the trust level to this device is reduced.

8.2.2 Status-based Approaches

Status-based approaches to trust and risk assessment examine the current state of the interacting device, regarding its security aspects. The goal is to determine whether (a) a breach has already been made to the device, having resulted in tampering of either software or its configuration and (b) how prone the device is to breaches, in the sense that known vulnerabilities have not been appropriately and timely handled through installation of patches. The security controls that apply to the device, are also taken into account since they moderate the device's vulnerability levels. In more detail, the following aspects are considered in status-based approaches:

- Have critical files been tampered with? Relevant validations span across:
  - the device's firmware;
  - the operating system and other software;
  - the system/network config files;
  - the audit and event logs.
- Have the latest patches been installed? Missing patches increase the vulnerability level of the device and therefore demote the trust level.
- Which security controls are in effect to protect the device?

8.2.3 Risk Assessment

Nowadays, the security of, and trust placed on, digital systems have become an ever-growing concern as technology plays an increasingly important role in our societies. An important manifestation of this aspect is the abundance of attacks deployed against organizations, governmental bodies and the society [12]. The mitigation of such attacks traditionally entails cybersecurity risk assessments which aid in the identification of critical assets, the threats they are exposed to, the probability of a successful attack, and the potential consequences. This approach, along with the prioritization of the identified risks, is the only way to identify the appropriate measures to be applied [12].

Risk assessment encompasses the identification, estimation and prioritization of the risks linked to an organization's assets and operations. This activity plays a critical role in the context of risk management, by providing the basis for the treatment of identified risks. The possible treatment approaches are: risk acceptance – when the risk level is deemed acceptable after consideration of the organization's risk management policy; risk mitigation – through security controls; risk transfer – by delegating accountability to an insurance company; or risk avoidance – through the removal of the corresponding asset. Some of the core concepts of risk assessment include but are not limited to: assets, vulnerabilities, threats, attack likelihood, and impact [13].

An asset can be any item that holds value for an organization, and is characterized by several properties. Assets can be classified as tangible (e.g., hardware) or intangible (e.g. public image of a business); additionally, assets can be a constituent part of a system or be the entire system. Vulnerabilities are properties of the assets that can be exploited, and can be defined as weaknesses of the assets themselves or weaknesses of the controls that protect them. A threat is an action that could compromise an asset, and is usually associated with the exploitation of a vulnerability. A threat can occur deliberately (e.g., applying a brute force attack to find the administrator's password) or unintentionally (e.g., erase a file through an erroneous action). These concepts are combined in the term *cyber-risk* which defines the probability of a successful threat (attack) emerging and the consequences for the assets involved.[1]

8.3 Trust Management Systems

Trust management models target at enabling nodes that participate in the trust management system to determine a trust metric value for other nodes within the system. Approaches to how trust models approach trust computation vary regarding numerous aspects, including the input used to compute trust, the way that trust values are updated, the consensus sought for trust value computation, the scale at which trust is measured, their resilience against attacks and so forth. Furthermore, trust management models vary with respect to architectural paradigm they follow, i.e., the way that the components participating in the trust management system are deployed in the target network, the relationships between the components and the information flows.

In the following subsections we survey existing trust models and their architectures, commenting on their merits and demerits.

1. https://www.thebalancesmb.com/assets-definition-2947887

8.3.1 Review of Existing Trust Models

This section overviews the trust models that have been proposed by the literature trying to find an effective and efficient trust computation method. In service-oriented networks, an IoT device acting as a service requester needs a way of evaluating which of its peers can be trusted to provide it with the requested service, while taking into consideration the energy demands of carrying out such evaluation. This is the challenge that trust management models are aiming to solve. We present trust management models as seen in the literature and we categorize each model by trust dimensions, resiliency against certain attacks and qualitative characteristics.

8.3.1.1 Trust dimensions

Trust models are composed of several trust dimensions which can vary between them depending on the approach followed. In this section we present the five most essential trust dimensions, namely, trust composition, trust propagation, trust aggregation, trust update and trust formation [14].

Trust composition. Refers to the components the model in question takes into account. The components are Quality of Service (QoS) and Social trust.

- QoS trust refers to the trust level assigned to a node based on the evaluation of its competence in delivering the requested service. It is considered as the "objective" evaluation of trust. In order to compute QoS trust, models use various trust properties including competence, cooperativeness, reliability, task completion etc.
- Social trust refers to the social relationship between owners of IoT devices. Social trust is used in systems where IoT devices must not be evaluated only on a QoS basis but also on a social basis, which is the device's commitment and willingness to cooperate. It can also be derived from similarity of devices. Social trust properties include connectivity, honesty, unselfishness etc.

Trust propagation. Refers to the way trust values are disseminated between entities. In general, there are two approaches, namely distributed and centralized.

- In distributed trust propagation each device acts autonomously by storing trust values and disseminating them as recommendations to other devices as needed.
- In centralized trust propagation a central entity exists, which is responsible for storing trust values of the monitored network and disseminating them as needed.

Trust aggregation. Refers to the computation techniques used by a model to combine trust obtained from direct observation with indirect trust coming from recommendations. Main aggregation techniques include weighted sum, Bayesian inference, and fuzzy logic.

- Weighted sum is a technique where weights are assigned on the participating values either statically either dynamically. For example, one model could use a trust property, e.g., competence, in order to assign higher or lower weights.
- Bayesian inference considers trust to be a random variable which follows a probability distribution. It is a simple and statistically sound model.
- Fuzzy logic uses approximate reasoning meaning that it doesn't use a binary evaluation variable but rather a variable whose values range between 0 and 1 for example, or even linguistic limits like High and Low which are translated using a membership function.

Trust update. Describes when trust values are updated. There are two approaches: event-driven and time-driven.

- Event-driven is the approach in which trust values are updated when an event occurs.
- Time-driven is the approach in which trust values are update periodically.

Trust formation. Refers to how the overall trust is formed out of the trust properties considered. Trust can be formed by considering only one trust property (Single-trust) or many properties (Multi-trust).

- Single-trust is when only one property is taken into consideration when computing trust and it is usually a property of QoS. It is considered as a narrow approach because trust is multi-dimensional, but it is useful in cases with limited resources.
- Multi-trust is the multi-dimensional approach in computing trust, because it uses more than one trust properties to form the overall trust evaluation of a device.

8.3.2 Trust Management Models

In this section we survey the different trust models proposed in the literature. For each model, the approach adopted for trust computation is presented, with an overview given in Table 8.1 while salient features of the models presented in detail in [31] (Table 3.5).

Bao, 2012 [17]. This model is proposed for social IoT(SIoT) systems based on Community of Interest (CoI). A device has a single owner and an owner can

Table 8.1. Overview of different trust models.

	Composition		Propagation		Aggregation			Update	Formation	
Model	**QoS**	**Social**	**Distrib**	**Central**	**Weigh**	**Fuzzy**	**Bayes**	**E/T**	**Sin**	**Mul**
[15–18]	X	X	X		X			E/T		X
[19, 20]	X	X	X		X		X	E/T	X	
[21]	X		X		X	X		T	X	
[22]	X		X		X	X		T	X	
[23]	X		X		X			E	X	
[24]	X			X	X			T	X	
[25]	X	X	X	X	X			E		X
[26]	X			X	X			E/T	X	
[26]	X			X	X			E	X	
[27]	X		X		X			T	X	
[28]	X		X	X				T	X	
[29]	X		X	X	X			E	X	
[30]		X	X	X	X	X		E		X

have multiple devices. The owners reserve a list with friends. Nodes that are part of similar communities have a better chance of having similar interests and capabilities. The authors consider both QoS and Social trust composition and define three trust properties: community-interest (Social), cooperativeness (QoS), and honesty (QoS);the interested reader is referred to [31] (Table 3.5) for more details. The trust value is a real number in the range [0,1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. The trust values are calculated by taking into account direct observations; in case such direct observations aren't any available, trust values can ve sourced from recommendations. Trust aggregation is performed using weighted sums, while the model follows a distributed architecture. It is worth mentioning that the weights that were used for past experiences can be dynamically adjusted when new evidence occurs to rebalance the trust convergence rate and trust fluctuation rate. In the simulation results, the effect that changing weights have is observed, but a way to dynamically adjust them is not mentioned.

Chen, 2016a [18]. This model is very similar to Bao, 2012. Main differences include: 1. A general approach for the computation of overall trust is not discussed. Instead, overall trust computation for specific scenarios is discussed. 2. The friends (nodes) lists exchanged between nodes upon interaction are encrypted with a one-way function in a way that nodes can identify only common friends. Hashing is

cost-efficient. 3. The model is tested in two real-world scenarios, namely, "Smart City Air Pollution Detection" and "Augmented Map Travel Assistance".

Bao, 2013 [19]. This model is proposed for social IoT(SIoT) systems based on the Community of Interest (CoI) concept. A device can have only one owner and an owner can have multiple devices. Owners maintain personal friend lists. Nodes that are part of similar communities have a higher probability of sharing similar interests and capabilities. The authors consider both QoS and Social trust composition. The trust value is a real number in the range [0,1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. The trust properties considered are honesty, cooperativeness and community-interest; please refer to [31] (Table 3.5) for more details. The trust propagation is distributed. The models' trust aggregation scheme uses Bayesian inference for the calculation of direct trust, and weighted sums are used for the aggregation of recommendation into indirect trust. An important aspect of this model is the introduction of a novel strategy for storage management which can be efficiently applied to large-scale IoT systems.

Chen, 2016b [20]. This model is an extension of Bao, 2013 [19]. Extensions include: 1. In the evaluation of recommenders, it introduces two additional properties, namely, friendship and social contact, which are further analyzed in [31] (Table 3.5). In trust aggregation it combines the direct with the indirect trust to form the overall trust. 3. Its simulations outperform EigenTrust [32] and PeerTrust [33] in trust convergence, accuracy, and attacks resiliency.

Chen, 2011 [21]. This model considers only QoS metrics for evaluating trust, namely, end-to-end packet forwarding ratio (EPFR), energy consumption (EC), and package delivery ratio (PDR). Each node maintains a data forwarding transaction table which includes the values: (1) Source: the trust and evaluation evaluating nodes, (2) Destination: the evaluated destination nodes, (3) $RF_{i,j}$: the times of successful transactions made between nodes i and j, and (4) $F_{i,j}$: positive transactions. It follows a distributed scheme in terms of trust propagation. In trust aggregation, a fuzzy trust model is used, and the overall trust is formed using a weighted sum of direct and indirect trust based on recommendations. The direct trust is computed by first aggregating the aforementioned QoS metrics, then labeling the results as a positive or negative experience based on a threshold and then a fuzzy membership function computes the direct trust based on the number of positive and negative experiences. Additionally, the model was tested on simulations and achieved better performance from BTRM-WSN [34] and DRBTS [35] in both packet delivery ratio and detection probability of malicious nodes.

Mahalle, 2013 [22]. This model considers three QoS metrics: Experience (EX), Knowledge (KN) and Recommendation (RC) ratings. It follows a distributed scheme, as every device considers the ratings of its neighbors for the calculation of the trust score. Trust is calculated periodically using Mamdani-type fuzzy rules

(representing If-Then relationships between their input variables) from the linguistic values of the three aforementioned metrics. Trust scores (as linguistic values) are then mapped to a set of access control permissions. Experience (EX) is the weighted sum of a number of previous interaction ratings between two devices (+1 for a successful interaction and −1 for an unsuccessful interaction), Knowledge (KN) is the weighted sum of direct and indirect knowledge ratings, and Recommendation (RC) is the weighted sum of RC ratings from a number of devices about the device to be trusted. The three metrics are mapped to their linguistic variables using predefined numeric (crisp) ranges. The model was tested in a simulated environment of wireless sensors with communication between sensors being controlled by trust ratings, resulting in more energy efficient communications, and proving to be scalable.

Prajapati, 2013 [27]. This model proposes the forming of trust values based on how satisfactory was a node's response to requests for specific services that were made to it: these satisfaction quantifications are combined to form the *Direct Trust* value. If a Direct Trust value is available, then this value is used; in the absence of a Direct Trust value, the Recommended Trust value is computed by sourcing and aggregating trust assessments from other peer nodes. In case the target node is joining the cloud for the first time, and therefore neither Direct Trust nor Recommended Trust values for it are available, a predefined Ignorance Value is used. Direct Trust is defined as the weighted sum of the rated service satisfaction ratings over time (with the weights decreasing over time, thus favoring newer ratings). Recommended Trust is defined as the weighted sum of the Direct Trust values of the other nodes. The weights used in the calculation of each Direct Trust value are based on two factors. The first one is the number of positive interactions between the two nodes (trustor and trustee). The second one is the Satisfaction Level which depends on factors such as recovery time, maximum-load performance, connectivity and availability as provisioned by the service agreement.

All nodes maintain a Direct Trust Table and a Recommended Trust Table containing the respective trust values with both tables being updated periodically. This model follows a distributed model as in the case of Recommended Trust, the trust values of all network nodes are considered.

Saied, 2013 [26]. This model considers ratings given to a specific node and service at a given time while also taking into consideration its state (e.g., age, resource capacity, etc.) It follows a centralized scheme with a Trust Manager (TM) node receiving reports from the network and calculating the trust values on demand. This leads to reduced communication overheads – since trust values are calculated and transmitted on demand, less memory usage for each node – since the trust values can be requested again from TM, and thus being energy efficient. The model operates in five phases: (1) TM receives reports from the network nodes, (2) TM calculates the trust values of a number of candidate nodes and sends a list of trustworthy

nodes to the requesting node, (3) the requesting node receives the list and interacts with a chosen trustworthy node, (4) the requesting node rates the service provided by the chosen trustworthy node and sends the rating to the TM, and finally (5) TM updates its trust values accordingly. Trust is calculated as the weighted average of the scores given to a node while taking into consideration the reputation of the node providing the score, the contextual similarity of all the reports concerning the same node, and the age of the report – favoring the most recent reports. Contextual similarity is calculated from the node capabilities between two nodes – to locate similar nodes, and/or from the difference of required resources between two services – to locate nodes able to run a similar service. Initially all nodes of the network are deemed trustworthy.

Mendoza, 2015 [23]. This model is a distributed version of the model proposed by Saied *et al.* [26]. It is noted that centralized schemes may not be suitable for IoT systems as server installation and server costs may be prohibitive. The rating scheme of this model defines ratings for a specific node and service. The model's operation comprises three phases: (1) nodes announce their presence to their neighbors and maintain a list of neighbors, (2) nodes request services from their neighbors and rate the interaction positively or negatively, and (3) nodes calculate and save trust values for their neighbors, based on these interactions. The response rating is defined as the fixed value of the provided service weighted by an adjusting factor, with the negative response rating being equal to two times the positive response rating. The provided service value is proportional to the processing requirements of the service, as more processing power or energy is required to run a service the higher the service value will be. The trust value of a node is calculated as the sum of all interaction ratings. The model was tested against On-Off Attacks (OOA) and it is noted that a large number of neighbors can cause delays in the assignment of the maximum distrust score to the malicious nodes.

Namal, 2015 [24]. This model considers four parameters: availability of resources to its users, reliability of produced information, response time irregularities, and capacity. It follows a centralized scheme with a Trust Manager (TM) module, hosted on the cloud, receiving filtered data from Trust Agents (TA) distributed on the network which in turn receive raw data and monitor the state of the network nodes. The TM implements a Monitor, Analyze, Plan, Execute, Knowledge (MAPE-K) feedback control loop and calculates the trust using the weighted sum of the trust parameters for all parameters considered. The trust parameter is also a weighted sum of the current value and the previous value calculated. This model shows advantages in: availability and accessibility – as the TMS is hosted on the cloud and is accessible from the internet, scalability – as the TMS utilizes TAs filtering the raw data, and flexibility – as the TAs can be deployed in a flexible manner.

Khan, 2017 [26]. This model considers ratings given to a node by its neighbors, these ratings are the combination of three variables: belief, disbelief and uncertainty – as defined in Jøsang's Subjective Logic. This model is proposed as part of an extension of the RPL routing protocol utilizing the proposed model to isolate malicious nodes. It follows a centralized scheme with a central node (e.g., RPL border router or cluster-head) calculating trust values for all network nodes and deciding to isolate malicious nodes. Each node of the network is assumed to be able to detect and therefore rate the performance of its neighboring nodes; each of the three aforementioned variables is defined as follows: belief is the number of positive interactions divided by the total number of interactions & a constant k, disbelief is defined similarly but instead of the positive interactions the number of negative interactions is used, and uncertainty is also defined similarity but with the constant k used instead of the number of positive/negative interactions. The central node calculates the trust value of each network node by combination of the trust values regarding the node to be trusted and using a threshold the central node isolates malicious nodes from the network.

Djedjig, 2017b [36]. This model considers two QoS parameters: selfishness and energy, and one social parameter: honesty as ratings given about a node from its neighbors. This model is a proposed extension of the RPL routing protocol, as in Khan *et al.* [21], to isolate malicious nodes. It follows a distributed scheme with each node calculating the trust values of its one-hop neighbors while also considering the trust values of its one-hop neighbors. Trust calculation is performed as follows: (1) each node calculates the direct trust values of its one-hop neighbors as a weighted sum of the honesty, energy and unselfishness metrics (definitions of which are not discussed in detail) with each metric being the weighted sum of the current value of the metric and the previous value of the metric, (2) each node receives the direct trust values calculated by its one-hop neighbors concerning the node to be rated, and (3) the indirect trust is then calculated by each node as the average of the direct trust calculated by the node itself and its neighbors. All nodes are assumed to be equipped with Trusted Platform Module (TPM) chips.

Medjek, 2017 [14]. This model is based on the one proposed by Djedjig *et al.* [36] with the difference in the metrics considered: honesty, energy and mobility. The main difference is the network architecture as this model applies to RPL networks consisting of a Backbone Router (BR) that federates multiple 6LoWPAN networks, each consisting of a 6LoWPAN Border Router (6BR) connected to the BR and the rest of the network nodes. This model follows a distributed scheme with each network node calculating the trust of its one-hop neighbors, as in [36], with the added steps of notifying its 6BR if a node is found to be untrustworthy and with the 6BR in turn notifying the BR of the malicious node. All nodes are assumed to be equipped with a Trusted Platform Module (TPM) and all nodes are registered

with the BR at installation time, with every node having a unique ID assigned by the BR. Several lists are maintained by the various network nodes; the BR maintains two lists: one of potential malicious nodes and one of all nodes and their states; the 6BR maintains three lists: one of all 6BR area nodes, one of all the mobile nodes, and one of the potential malicious nodes; finally the remaining nodes also maintain three lists: one of potential malicious nodes, one of suspicious nodes and a copy of the mobile node list from the 6BR. Three modules operate on the various network nodes: IdentityMod controls access to the network and ensures that every node has a unique ID, MobilityMod ensures that both the BR and the 6BRs are aware of mobile nodes and of their status, and IDSMod is responsible for attack detection and mitigation. Trust is calculated in a similar fashion to [36] with the values of the honesty metric supplied by the IDSMod and the values of the mobility metric supplied by the MobilityMod; the three metrics are not discussed in detail.

Nitti, 2014 [25]. This work proposes two models, namely the "subjective" model and the "objective" one. These models consider the following parameters: (i) node credibility, (ii) service ratings, (iii) transaction factor – identifying which transactions are important to avoid trust levels increasing only by many small transactions, (iv) number of transactions per node – to detect abnormalities in the number of transactions for a given node, (v) computation capacity – nodes with higher computational capabilities can inflict more damage if they are malicious, (vi) the notion of centrality – a node plays a more central role if involved in many connections or transactions in the network, and (vii) the relationship factor – considering the type of two nodes' relationship.

The subjective model follows a distributed scheme where each node stores the necessary information to calculate the trust values locally. Two situations are covered relating to the social relationship between nodes: when the rating node has a social relationship with the rated node and when the two nodes have no direct social relationship. In the first situation trust depends: on the centrality of the rated node in relation to the rating node – by count of the common friends out of all the neighboring nodes, the direct experience of the rating node – further defined as the weighted sum of both short-term and long-term opinions, and the indirect experience of the rating node's friends – defined as the weighted average of the trust values assigned to the rated node by the rating node's friends, weighted by their credibility. In the second situation trust depends: on the opinions of the chain of common friends connecting the two nodes, again weighted by their credibility. Generally, after each transaction a rating (positive/negative) is given to the node providing the service and to the nodes whose opinion was considered in calculating the trust value. Negative recommendation ratings are given to both malicious nodes and to nodes in their neighborhood, thus isolating the malicious nodes and their influence further.

The objective model follows a more centralized scheme where each node reports its feedback to special nodes, referred to as Pre-Trusted Objects (PTO), responsible solely for maintaining the distributed storage system, in this case a Distributed Hash Table (DHT) and more specifically one following the Chord architecture. Trust is calculated in a similar fashion as in the subjective model; node centrality is defined as the total number of transactions performed by the node to provide a service divided by the total number of transactions performed to either provide or request a service, and both short-term and long-term opinions consider the ratings of every network node weighted by their credibility. Nodes with few social relations, high computation capabilities and nodes involved in a large number of transactions between them are assigned low credibility, as they are more likely to become malicious.

Wu, 2017 [28]. The system model consists of four entities with three trust relationships among them. The four entities are defined: RFID tags, RFID readers, authentication centers and one administration center, with the first three being grouped in domains. A domain has multiple RFID readers connected with the domain authentication center which authorizes the readers to interact with the RFID tags, and the domain authentication centers are connected with the administration center. The trust relationships of this system model are defined as: intra-domain trust – trust relationship between RFID tags and readers of the same domain, inter-domain trust – trust relationship between authentication centers, and cross-domain trust – trust relationship between RFID tags and readers belonging to different domains.

The trust management model consists of two layers: the authentication center trust layer – a centralized trust management system managing the trustworthiness of authentication centers, and the reader trust layer – two proposed trust management schemes managing the trustworthiness of RFID readers. The RFID tags are always assumed to be trusted.

The first reader trust management layer scheme proposed uses the Dempster-Shafer evidence theory and consists of four steps: (1) the interaction of an RFID reader is recorded by its neighbors, (2) the neighbors calculate the local trust values which are then transmitted to the authentication center, (3) the authentication center calculates the global trust of the RFID reader by using the Dempster knowledge rule, and finally (4) it the RFID reader is malicious or malfunctioning the administration center is notified. Possible RFID reader interaction events are identified and marked as: malicious behavior, malfunctioning behavior and normal behavior by the neighboring RFID readers, each counting the number of events within a specified time frame. Using the number of recorded events the neighboring RFID readers can calculate the local trust value for each type of interaction events as: the number of events marked as malicious/malfunctioning/normal divided by the total

number of recorded events. The final value of the local trust value is then chosen from the event-specific local trust values using a threshold. The authentication center calculates the global trust of the RFID reader by aggregating the event-specific local trust scores calculated by the neighboring RFID readers and then choosing the final integrated event-specific score using a threshold.

The second reader trust management layer scheme proposed considers the fact that events may not be detected by neighbors of the RFID reader and thus the first reader trust management layer scheme may not be applicable to certain situations. Each RFID tag keeps record of the last interaction with an RFID reader, more specifically the RFID reader ID, a timestamp and the rating assigned to the RFID reader by the tag. This record is sent at the next time the RFID tag interacts with any RFID reader (and is then deleted from the RFID tag), with the RFID reader forwarding the record to its authentication center which checks for abnormalities and if any problem arises, it notifies the administration center as well as the authentication center the previous RFID reader belongs.

The proposed authentication center trust layer scheme considers abnormal event reports by RFID readers and affects the trust value of the domain authentication center the readers are part of. Calculation of trust in this case can be performed by either of the two methods proposed for the reader trust management schemes.

Mahmud, 2018 [30]. This model considers three social trust metrics for a pair of nodes, namely: relative frequency of interaction, intimacy and honesty, and the deviations of generated data from the historical data of the node that generated the trust metric and its neighbors. Two trust dimensions are defined: node behavioral trust and data trust; both calculated by combination of direct (from the rating node) and indirect (from the rating node's neighbors) interactions, with indirect interactions being weighted by the distance of the neighbor to the rated node. Node behavioral trust is calculated using an Adaptive Neuro-Fuzzy Inference System (ANFIS), a fuzzy system using back propagation to tune itself. The three inputs to ANFIS are defined as: relative frequency of interaction is defined as the ratio of interactions with the rating node out of all interactions of the rated node in a given time period, intimacy is defined as the ratio of time amount spent interacting with the rating node out of the total time spent interacting with all nodes except the rating node, and honesty is defined as the ratio of successful interactions out of the total number of interactions of the rated node with its rating node. Three linguistic terms are used in ANFIS for each of the three inputs: Low, Medium and High. Deviations of generated data, used to calculate the data trust, are defined as follows: direct data trust is defined as the deviation of instantaneous data from the historical data generated by the rated node, and indirect data trust is defined as the deviation of instantaneous data from the historical data from the historical data generated by the rated node's neighbors.

Arabsorkhi, 2016 [37]. The work of Arabsorkhi *et al.* presents the general principle behind many proposed trust management models considering ratings given to network nodes for the quality of the services provided over a specific time period. If the rating node has enough information to determine the trust value from its own ratings over the specified time period (by direct observation) it can proceed to calculate the trust value of the node to be rated. If not, then the rating node can query the rest of the network and aggregate the trust values assigned by the other network nodes to the rated node.

Yuan, 2018 [29]. This model considers ratings given after node interaction for the quality of provided services. The network model consists of IoT edge nodes being part of a domain federated by an edge broker node, which in turn contact a central cloud server responsible for the final calculation of trust values. Three trust values are calculated: the direct trust about a device to another device (D2D direct trust), the feedback trust about a node by an edge broker (B-to-D feedback trust), and the overall trust (the final trust value) about a device. D-to-D direct trust is updated and based on the history of direct interaction between nodes, it is defined as the ratio of positive interactions and the number of total interactions between the two nodes. B-to-D feedback trust is updated by the edge broker periodically and is based on all the D-to-D direct trust values concerning an edge node (except self-ratings); the edge broker aggregates the D-to-D direct trust values using weights derived by use of object information entropy theory, overcoming the limitations of assigning the weights manually. The overall trust value is calculated as the weighted sum of the D-to-D direct trust and the B-to-D feedback trust, thus considering the opinion of the rating node as well as the opinion of the whole network about the rated node.

8.4 Trust Management System

The objective of the trust management system is to serve an authority within the protected Internet of Things infrastructure perimeter, which undertakes the following tasks:

- Consolidates observations on the status, behaviour and associated risk of devices into a comprehensive trust score, which indicates the degree to which each device is deemed to be trustworthy.
- Can be queried by other entities within the protected Internet of Things infrastructure perimeter, to provide the abovementioned assessments, for the perusal of the entities. Indicatively, trust assessments can be used for the visualization of trust within the network, for making decisions whether actions

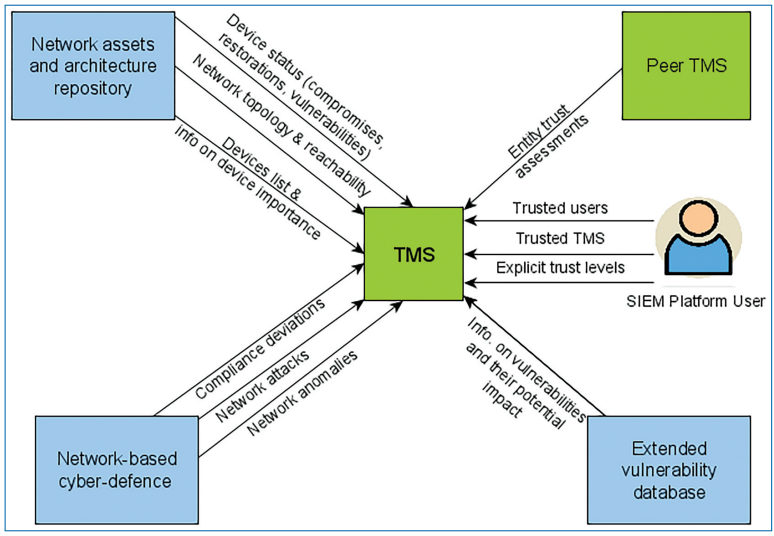


Figure 8.1. SIEM platform elements providing information to the TMS.

originating from or being directed to some device should be allowed or not, for raising alerts to security officers and so forth.

- Provides timely notifications to other entities within the protected Internet of Things infrastructure perimeter, to alert them of noteworthy events related to the level of trust associated with devices. In particular, demotions of device trust level below some threshold and the restoration of previously demoted trust of devices are emitted, allowing relevant components of the protected Internet of Things infrastructure perimeter, to take appropriate actions, such as enabling or disabling defence mechanisms.

8.4.1 TMS Context

The TMS is envisioned to operate in the broad context of a platform following the Security Information and Event Management System (SIEM) principles [38], sourcing information required for its operation from other platform modules, as depicted in Figure 8.1.

In more detail, the information sourced from other platform elements, which act as *security information and event management* (SIEM) providers is as follows:

- *platform users* provide information regarding the peer users they trust, the peer TMSs that are trusted and explicit device trust specifications. Naturally, user interaction with the TMS is mediated through an appropriate application.
- *The CyberDefense module* provides data regarding the network anomalies detected (deviations from the nominal device and network behaviour), the

non-compliant traffic (traffic flows that have not been whitelisted as "acceptable behaviour" for the device) and network attacks (primarily in the context of signature-based detection), either originating from some device or targeted against it.

- *The iIRS (intelligent Intrusion Response System) module* provides information regarding the devices that are in the scope of the TMS, their importance, the vulnerabilities existing on devices, events of device compromises, as well as network topology and reachability information.
- *The eVDB (extended Vulnerability DataBase) module* provides information on the detected vulnerabilities, including their impact, underpinning the assessment of the impact that vulnerabilities may have on the trust level of the affected device.
- *The Device profile repository* provides information on the cases that a device is removed from the system and when the device health is restored after a compromise (i.e. the malware is removed or "clean" versions of the operating system/firmware are installed).
- *The TMS*, acting as a trusted peer entity, provides trust assessments which are combined by the receiving TMS instance with the own device trust estimations, to synthesize a comprehensive trust score.

The TMS, in turn, publishes information regarding changes in the trust level of the devices through the SIEM platform information bus (a pub/sub component that delivers specific types of information published to it to entities that have registered their interest in receiving these types of information), as depicted in Figure 8.2. This information can be exploited as follows:

- SIEM platform operator and end-user interfaces may use this information to generate alerts, especially in the cases of noteworthy trust demotion.

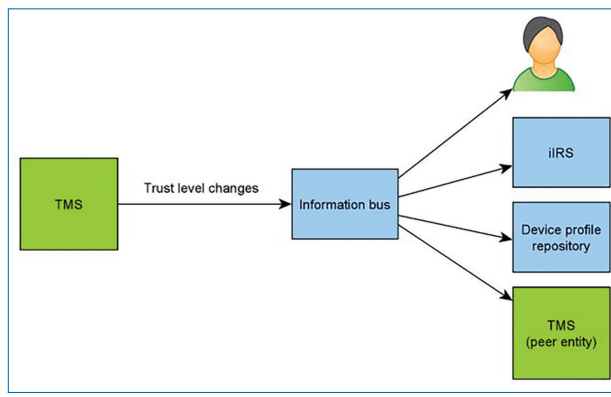


Figure 8.2. TMS outgoing information flows.

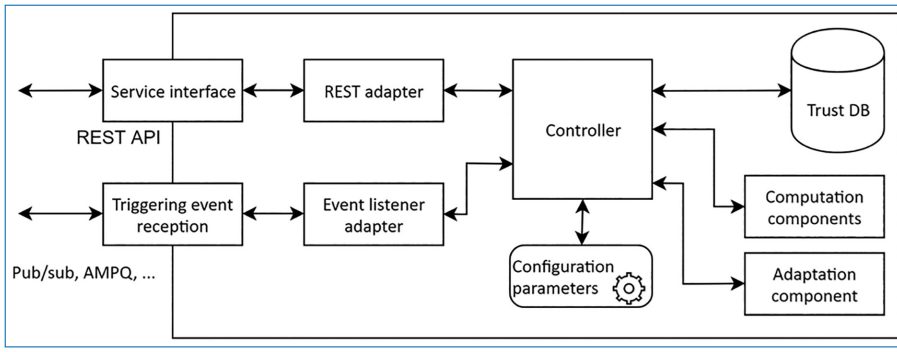


Figure 8.3. TMS high-level design.

- Defence mechanisms, and in particular the iIRS can exploit this information to apply or disable restrictions in network traffic.
- The Device repository updates its own database, guaranteeing information consistency and dissemination of the trust level to any other interested component.
- Peer TMSs can use this information to update their trust assessments.

8.4.2 TMS Application Architecture

Figure 8.3 illustrates the conceptual view of the Trust Management System. Its architecture is designed to allow for exposing a coherent API, enabling any adaptation aspects to be implemented internally considering all the appropriate contexts (network & resource availability, situation criticality etc.). Reception of information needed to recompute the trust and risk scores – including device status, behaviour and associated risk aspects are mainly intercepted through asynchronous messaging, through a dedicated communication channel, following the pub/sub paradigm. In this way, the TMS is decoupled from event producers and their timings; however, content consumption via APIs can be also used. Reciprocally, the TMS publishes events regarding notable changes of trust and risk levels, while also offering the same information under REST APIs. Adaptation, where needed, will be supported by an adaptation component to be developed and maintained separately from the computational aspects, promoting separation of concerns.

Figure 8.4 depicts the data view of the TMS, indicating:

(a) the data maintained internally in the TMS database;
(b) the messages that the TMS subscribes to in order to obtain the necessary information to compute trust and risk levels, as well as the sources of these messages, according to the overall SIEM system architecture;

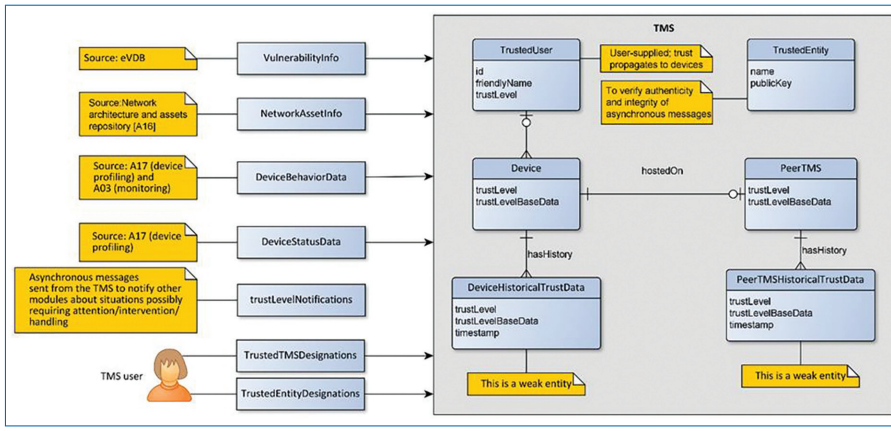


Figure 8.4. TMS data view.

(c) the information that the TMS receives directly from the users (typically, through a UI);
(d) the messages that the TMS makes available to the asynchronous communication infrastructure, for the perusal of other Cyber-Trust components.

Trusted Peer TMS are curated directly by users. Users additionally provide information regarding other trusted entities in the platform: this pertains to modules that generate asynchronous messages to the information bus, and are expected to be consumed by the TMS. Each trusted entity specification provides the data needed by the TMS to verify the authenticity and integrity of received messages, i.e. the name of the peer and its certificate. While users are not commonly expected to be proficient with such data, automated procedures upon the setup of the platform are expected to relieve the user of the task of manually setting up this information. Should updates to this information be needed, automations, configuration assistants and wizards may also ease the task of the users.

8.4.3 TMS Design

In Figure 8.5, the entities involved in proposed trust model and the relationships between them are illustrated. The elements may appear in the context of the IoT, Smart Home, or SOHO environments and include:

- *Devices*, which function within the considered environment.
- *Users*, that own devices. A single user can have many devices. Users can establish trust relationships between them, with these relationships having the following properties (a) they are *weighted*, (b) they are *directed*, (c) they are *not*

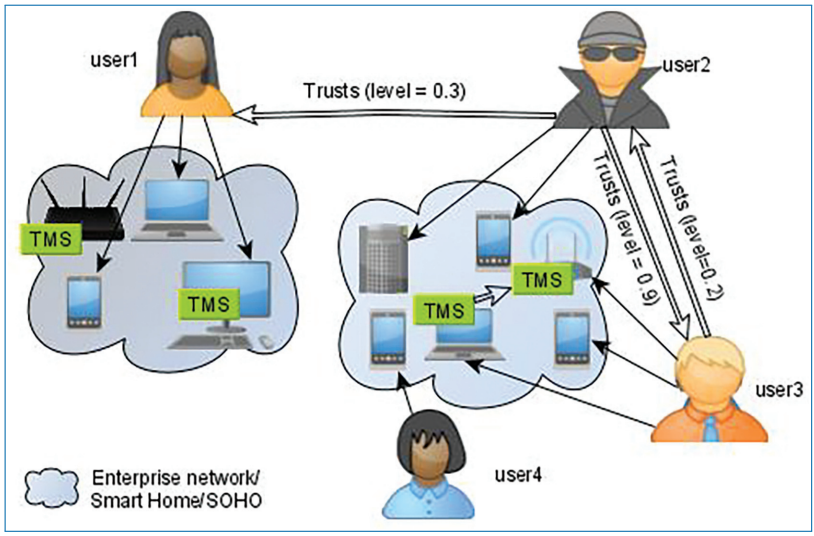


Figure 8.5. The entities in the proposed trust model and their relationships.

transitive and (d) they are *not necessarily symmetrical*. The following example illustrates these properties:

- User u_1 states that s/he trusts another user u_2. This is done by providing a *trust level*, which expresses u_1*'s* confidence that u_2 will not perform malicious actions against u_1 -or even take activities that have positive effects on u_1.
- The declaration of trust of u_1 towards u_2 does not necessarily mean that u_2 *also trusts* u_1, expressing the fact that trust may not be reciprocated [23]. It is still however possible that u_2 makes a separate, independent assertion that s/he trusts u_1; such an assertion may express a different trust level than the respective assertion made by u_1.
- Trust is not transitive: if u_1 trusts u_2 and u_2 trusts u_3, no assumption is made that u_1 trusts u_3. An explicit assertion by u_1 is required to establish any trust relationship to any other user in the domain of discourse.

- *Trust Management System instances (TMS): TMSs are effectively software agents which perform trust level computations towards devices within the considered environment. The trust value computation for a device is performed by considering multiple factors which are either collected through monitoring the activities within the environment or explicitly provided. The factors taken into consideration are:*
  - *the device status*: this includes (1) information about the integrity of the device, i.e. information attesting the legitimacy of the software/firmware/operating system and its configuration, as opposed to the

aforementioned components being compromised; and (2) information on the device's *resilience*, i.e. if the device's software/firmware/operating system/configuration have any known vulnerabilities, as opposed to the case that no known vulnerabilities are present.

- *the device behaviour*: this encompasses the following information:
  1. if the device has been reported to perform attacks or has been identified to be the target of attacks.
  2. if the device's resource utilization metrics comply with a predefined specification which defines what constitutes normal behaviour or if they diverge from it. Some examples of these metrics include, but are not limited to, network usage, CPU load, and disk activity. Practically, any class of system metrics that can be quantified, and for which baseline metrics can be created so as to allow computation of deviations from the baselines is eligible for incorporation within this dimension. Similar practices are widely employed in monitoring infrastructures, such as Nagios [39] and may include metrics such as number of connected users, amount of free disk, total number of processes, number of processes corresponding to some specific service instance, etc.
  3. If the device's behaviour conforms to some predefined reference behaviour that is whitelisted as "normal". MUD specification files [5] can provide such information, nevertheless they have not been widely adopted and manufacturer support is lacking.
- *the risk associated with the device:* IoT devices may become targets of attacks and some attacks may succeed. A probability indicating that a device will eventually be compromised can be calculated by considering technical information such as its vulnerabilities and its reachability inside the network. Attack graphs can be utilized to this end [36]. The level of impact of a successful attack on an organization/person owning a device is not always the same and can vary depending on the perceived value of the device. The perceived value of the device is directly linked with the assets it encompasses (e.g. the value of a device hosting a database is dependent on the value of the data in the database) or with the value/criticality processes it is responsible for (e.g., a vital signs monitor on a smart watch vs. a vital signs monitor used in remote surgery).

  Another aspect that must be considered when calculating the risk associated with a device d is the set of devices that are accessible through it, and whether is would be possible for attackers to use it as a bastion from where they assault other devices, attempting to compromise devices of high value in the context of more advanced, multi-staged attacks. In this

respect, the risk associated with *d* is dependent on (a) the probability that *d* is compromised itself, (b) the probability that devices reachable from *d* are compromised in the context of a multi-stage attack and (c) the perceived value of devices reachable from *d*.

Taking the above into account, the associated risk dimension combines the above-mentioned aspects i.e. (i) the technical probability that the device is compromised with the perceived value of the device, and (ii) the probability that the device is used as a stepping stone to attack other devices, in conjunction with the business values of the assets associated to these devices, to synthesize a single, comprehensive metric expressing the business risk applicable to a device.

- The trust relationship between the user that owns a device running a TMS instance and the user whose device is under trust evaluation. This aspect moderates the weight of trust level assessments, so that trust level assessments sourced from trusted TMSs (i.e. TMSs running on devices belonging to trusted users) are taken more strongly into account, while the importance of trust assessments sourced from non-trusted TMSs (i.e. TMSs running on devices belonging to users of unknown or low trust) is attenuated.

An overall trust assessment is formed by the TMS instances by synthesizing the three trust dimensions: (i) status-based, (ii) behaviour-based, and (iii) associated risk-based trust.

Furthermore, trust relationships can be established between TMS instances, in the same fashion that trust relationships are established between users. Similarly to the case of user-to-user trust relationships, TMS-to-TMS trust relationships are (a) weighted, (b) directed, (c) non-transitive and (d) *not necessarily symmetrical*. The trust relationships between TMS instances are explicitly provided by the users owning the devices on which TMS instances are run. Once a trust relationship stating that TMS instance T_1 trusts TMS instance T_2 is established, T_1 will source trust assessments for devices from TMS T_2, and take them into account when computing the respective devices' trust levels.

Finally, users are allowed to set explicitly the trust level of the devices they own, overriding the computations made by the TMS. This provision is accommodated to handle false positives mainly related to network attacks (an attack is flagged by relevant modules but was not actually performed), network anomalies (e.g. excessive traffic was detected but this was due to a user-initiated backup or a software/firmware update) and compromises (e.g. some software on the device was misclassified as malware). The TMS will be able to provide both the automatically computed and the explicit trust level of the device, so that relevant applications

will be able to detect devices where major discrepancies exist and keep the users informed about such deviations, promoting awareness and facilitating intervention, as needed.

According to the description listed above, the TMS composes the trust score in a hierarchical fashion, as depicted in Figure 8.6, undertaking a holistic view towards trust assessment. To perform this composition, the TMS necessitates different types of information for each device. The TMS operates in the broad context of c and sources the required information from other SIEM platform modules, as

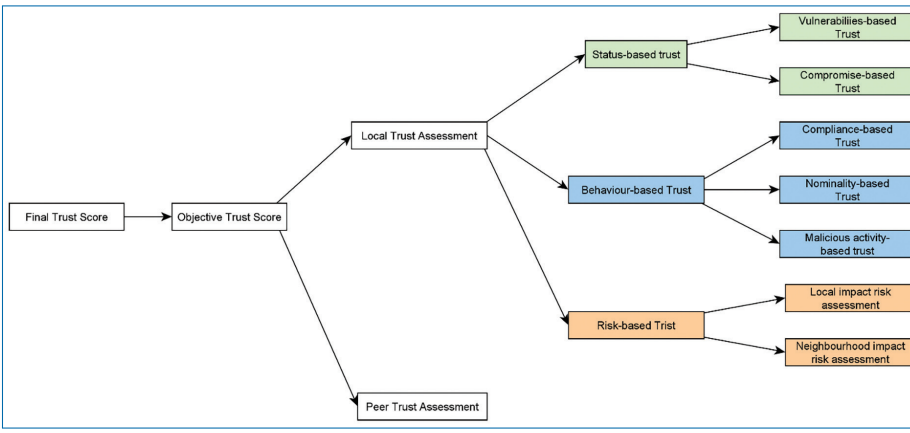


Figure 8.6. Trust score composition dimensions and aspects.

8.5 Conclusions

In this chapter, we presented an approach to trust computation in the Internet of things, which synthesizes behavioral, device status and associated risk aspects into a comprehensive trust score, that can be consulted to realize trust-based access control. The proposed approach also considers device ownership relationships and owner-to-owner trust relationships, which are utilized in the trust computation process.

Different parameters of the trust management computation process may be configured and tuned; notably, varying approaches may be used to compute the overall trust score based on the partial, dimension-specific scores; trust demotions may be subject to aging, i.e. their effects may decay over time, or may remain in effect until their root causes are known to be resolved; SIEM data may be associated with confidence levels, and these levels could be considered in the overall trust score computation. All these parameters are dependent on the particular context in which the

TMS operates. Our future work includes an in-depth study and analysis of these aspects; additionally the proposed TMS architecture will be evaluated, to quantify its overall performance, as well as its resilience against specific attacks that are launched against IoT networks.

Acknowledgment


This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] M. Blaze, J. Ioannidis, and A. D. Keromytis, "Experience with the KeyNote Trust Management System: Applications and Future Directions," 2003, pp. 284–300.

[2] P. A. Bonatti and P. Samarati, "A uniform framework for regulating service access and information release on the Web," *J. Comput. Secur.*, vol. 10, no. 3, pp. 241–271, Jul. 2002, doi: 10.3233/JCS-2002-10303.

[3] K. Irwin and T. Yu, "Preventing attribute information leakage in automated trust negotiation," in *Proceedings of the 12th ACM conference on Computer and communications security – CCS '05*, 2005, p. 36, doi: 10.1145/1102120.1102128.

[4] C. A. Ardagna, E. Damiani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "Trust Management," in *Security, Privacy, and Trust in Modern Data Management*, Springer, 2007, pp. 103–117.

[5] C. Vassilakis *et al.*, "Cyber-Trust Project D2.1: Threat landscape: trends and methods," 2018.

[6] Joint Technical Committee ISO/IEC JTC 1, "International standard NEN-ISO/IEC 27001: Information technology – Security techniques – Information security management systems – Requirements," 2013.

[7] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An Integrative Model of Organizational Trust," *Acad. Manag. Rev.*, vol. 20, no. 3, p. 709, Jul. 1995, doi: 10.2307/258792.

[8] C. L. Corritore, B. Kracher, and S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model," *Int. J. Hum. Comput. Stud.*, vol. 58, no. 6, pp. 737–758, Jun. 2003, doi: 10.1016/S1071-5819(03)00041-7.

[9] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs," *IEEE Trans. Dependable Secur. Comput.*, vol. 9, no. 1, pp. 61–74, Jan. 2012, doi: 10.1109/TDSC.2011.34.

[10] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM conference on Computer and communications security – CCS '06*, 2006, pp. 336–345, doi: 10.1145/1180405.1180446.

[11] D. R. E. Lear, R. Droms, "Manufacturer Usage Description Specification," 2018. https://tools.ietf.org/html/draft-ietf-opsawg-mud-25 (accessed Apr. 13, 2020).

[12] J. R. C. Nurse, S. Creese, and D. De Roure, "Security Risk Assessment in Internet of Things Systems," *IT Prof.*, vol. 19, no. 5, pp. 20–26, 2017, doi: 10.1109/MITP.2017.3680959.

[13] M. Theoharidou, A. Mylonas, and D. Gritzalis, "A Risk Assessment Method for Smartphones," 2012, pp. 443–456.

[14] J. Guo, I.-R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in internet of things systems," *Comput. Commun.*, vol. 97, pp. 1–14, Jan. 2017, doi: 10.1016/j.comcom.2016.10.012.

[15] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "New trust metric for the RPL routing protocol," in *2017 8th International Conference on Information and Communication Systems (ICICS)*, Apr. 2017, pp. 328–335, doi: 10.1109/IACS.2017.7921993.

[16] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, "A Trust-Based Intrusion Detection System for Mobile RPL Based Networks," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Jun. 2017, pp. 735–742, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.113.

[17] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proceedings of the 2012 international workshop on Self-aware internet of things – Self-IoT '12*, 2012, p. 1, doi: 10.1145/2378023.2378025.

[18] I.-R. Chen, F. Bao, and J. Guo, "Trust-Based Service Management for Social Internet of Things Systems," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 6, pp. 684–696, Nov. 2016, doi: 10.1109/TDSC.2015.2420552.

[19] F. Bao, I.-R. Chen, and J. Guo, "Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems," in *2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*, Mar. 2013, pp. 1–7, doi: 10.1109/ISADS.2013.6513398.

[20] I.-R. Chen, J. Guo, and F. Bao, "Trust Management for SOA-Based IoT and Its Application to Service Composition," *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 482–495, May 2016, doi: 10.1109/TSC.2014.2365797.

[21] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," *Comput. Sci. Inf. Syst.*, vol. 8, no. 4, pp. 1207–1228, 2011, doi: 10.2298/CSIS110303056C.

[22] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad, "A fuzzy approach to trust based access control in internet of things," in *Wireless VITAE 2013*, Jun. 2013, pp. 1–5, doi: 10.1109/VITAE.2013.6617083.

[23] C. V. L. Mendoza and J. H. Kleinschmidt, "Mitigating On-Off Attacks in the Internet of Things Using a Distributed Trust Management Scheme," *Int. J. Distrib. Sens. Networks*, vol. 11, no. 11, p. 859731, Nov. 2015, doi: 10.1155/2015/859731.

[24] S. Namal, H. Gamaarachchi, G. MyoungLee, and T.-W. Um, "Autonomic trust management in cloud-based and highly dynamic IoT applications," in *2015 ITU Kaleidoscope: Trust in the Information Society (K-2015)*, Dec. 2015, pp. 1–8, doi: 10.1109/Kaleidoscope.2015.7383635.

[25] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness Management in the Social Internet of Things," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1253–1266, May 2014, doi: 10.1109/TKDE.2013.105.

[26] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, and P. Herrmann, "A Trust-based Resilient Routing Mechanism for the Internet of Things," in *Proceedings of the 12th International Conference on Availability, Reliability and Security – ARES '17*, 2017, pp. 1–6, doi: 10.1145/3098954.3098963.

[27] Y. Ben Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Trust management system design for the Internet of Things: A context-aware and multi-service approach," *Comput. Secur.*, vol. 39, pp. 351–365, Nov. 2013, doi: 10.1016/j.cose.2013.09.001.

[28] S. K. Prajapati, S. Changder, and A. Sarkar, "Trust Management Model for Cloud Computing Environment," *arXiv.org*, Apr. 2013, [Online]. Available: https://arxiv.org/abs/1304.5313.

[29] X. Wu and F. Li, "A multi-domain trust management model for supporting RFID applications of IoT," *PLoS One*, vol. 12, no. 7, p. e0181124, Jul. 2017, doi: 10.1371/journal.pone.0181124.

[30] J. Yuan and X. Li, "A Reliable and Lightweight Trust Computing Mechanism for IoT Edge Devices Based on Multi-Source Feedback Information Fusion," *IEEE Access*, vol. 6, pp. 23626–23638, 2018, doi: 10.1109/ACCESS.2018.2831898.

[31] N. Kolokotronis *et al.*, "Cyber-Trust Project D5.1 State-of-the-art on proactive technologies," 2019.

[32] K. Govindan and P. Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 2, pp. 279–298, 2012, doi: 10.1109/SURV.2011.042711.00083.

[33] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the twelfth international conference on World Wide Web – WWW '03*, 2003, p. 640, doi: 10.1145/775152.775242.

[34] Li Xiong and Ling Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 07, pp. 843–857, Jul. 2004, doi: 10.1109/TKDE.2004.1318566.

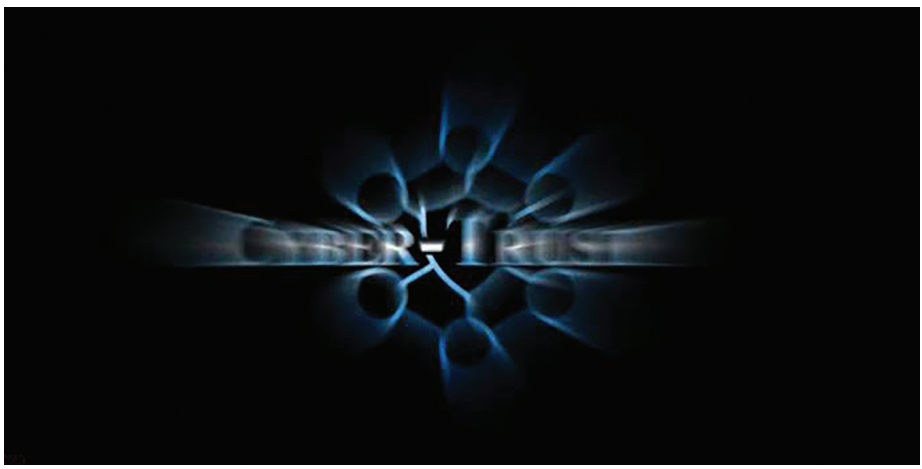
[35] F. Gómez Mármol and G. Martínez Pérez, "Providing trust in wireless sensor networks using a bio-inspired technique," *Telecommun. Syst.*, vol. 46, no. 2, pp. 163–180, Feb. 2011, doi: 10.1007/s11235-010-9281-7.

[36] R. Ismail and A. Josang, "The Beta Reputation System," in *Proceedings of the BLED 2002 Conference*, 2002, [Online]. Available: https://aisel.aisnet.org/bled2002/41.

[37] A. Arabsorkhi, M. Sayad Haghighi, and R. Ghorbanloo, "A conceptual trust model for the Internet of Things interactions," in *2016 8th International Symposium on Telecommunications (IST)*, Sep. 2016, pp. 89–93, doi: 10.1109/ISTEL.2016.7881789.

[38] S. Bhatt, P. K. Manadhata, and L. Zomlot, "The Operational Role of Security Information and Event Management Systems," *IEEE Secur. Priv.*, vol. 12, no. 5, pp. 35–41, Sep. 2014, doi: 10.1109/MSP.2014.103.

[39] A. J. H. Witwit and A. K. Idrees, "A Comprehensive Review for RPL Routing Protocol in Low Power and Lossy Networks," 2018, pp. 50–66.



DOI: 10.1561/9781680838350.ch9

Chapter 9

Cyber-Trust Evaluation Process

By V.-G. Bilali[1,*], *A. Kardara*[2,†], *D. Kavallieros*[3,‡] *and G. Kokkinis*[2,§]

[1]Institute of Communication and Computer Systems (ICCS)
[2]Centre of Security Studies (KEMEA)
[3]University of the Peloponnese
[*]giovana.bilali@iccs.gr
[†]a.kardara@kemea-research.gr
[‡]d.kavallieros@uop.gr
[§]g.kokkinis@kemea-research.gr

The Internet of Things (IoT) environment is constantly changing, shaped by both technical and social needs. The rapid IoT advancements and therefore the increase in the number of the interconnected data between services and infrastructure that potentially may pose threat into cyberspace, was the commencement of the Cyber-Trust project conceptualization [1]. Cyber-Trust conducts extensive research excellence in areas where IoT is widely applied. The structure of the project has been relied among others to taking into consideration stakeholders needs, so that the results that the project will produce are realistic based on final user needs.

In this context, an evaluation plan is designed, to assess platform's operations. In this chapter is presented the validation, verification and evaluation methodology that Cyber-Trust followed during the first pilot phase of the project's lifecycle. Cyber-Trust Evaluation Process contains information on how technical partners are going to validate technical components based on system's specifications and the appropriate methods in which end-users will evaluate all the functions of the platform. Validation, verification and evaluation goals are in line with project's objectives. This chapter is also guided by the project deliverables related to (a) use case scenarios, (b) Cyber-Trust architecture, (c) end-user requirements, and d) the integration of the overall system.

9.1 Introduction

Validation, verification, and evaluation are methods that exist under the same umbrella of the entire Evaluation Process. As the evaluation is the final stage in which the total "product" is assessed by actual or potential users, we refer to the whole process by this name. In many cases each of these methods are embedded in each other, but in different stages. From now and on, for the sake of simplicity, when we want to indicate the overall assessment procedure, we will refer to the Evaluation Process which includes the three aforementioned methods as three different stages contained in it.

In a whole, the validation methodology assesses whether the product constructed based the criteria (requirements) given by end-users answering to the question "Does this developed system do what is intended?", the verification whether the system executes specific functions based on the system's specifications answering to the question "Did we build the right product?", and the evaluation is referred whether the developed platform as a total has met their desired needs.

Validation, verification and evaluation methods have been formulated and implemented by various companies, enterprises as well as projects. Many multilevel frameworks have been developed to assess different products, including both objects and methodologies. Their scope among others is to ensure quality, enhance performance of the product and based on the acquired results (if the evaluation is continuous) to define the next steps.

The state of the art of evaluation process frameworks have been identified below, proving that the framework utilized for building Cyber-Trust assessment methodology is an extensible and customizable methodology.

9.2 State of Knowledge

9.2.1 General Evaluation Process

In this subsection is introduced a general evaluation process upon which Cyber-Trust's methodology based on. This frame is broadly used in order to evaluate the final product and is consisted by specific step-by-step procedures.

1. Beginning with setting the frame (e.g., context, objectives, use cases, requirements etc.)
2. Design the system.
3. Defining the evaluation groups, evaluation objectives, evaluation strategy etc.
4. Setting up and executing pilot trials in order to evaluate the "product".
5. Evaluation results and assessment

Based on the Step 5 the evaluation is considered as successful or not. For improvement purposes, when the first evaluation iteration is completed, Step 5 can provide feedback on Step 3 that continues the process until the end of the second iteration phase and goes on.

Almost the same steps are used in Section 9.2.2 where the assessment took place in different type of "products". Thus, the conclusion drawn is that the evaluation methodology is used regardless of the type of the evaluation object. The Cyber-Trust Evaluation Framework is explained in Section 9.3.

9.2.2 Implemented Evaluation Framework

Innovate Uk [2] is a national funding agency investing in science and research in the UK that has implemented an evaluation framework to objectively understand how a policy or other actions was enforced and what the consequences were. It evaluates their investment activities towards three (3) areas performing (a) process evaluation, (b) impact evaluation and (c) economic evaluation. The framework follows a circular flow that enables the evaluation of the first circle to have a total impact by giving feedback on second circle and modify the rationale of the new circle that will begin (second circle).

The Evaluation Framework for National Cyber Security Strategies (NCSS) [3], targets to improve the cyber-security policy guidelines, by assessing the system and providing improvements to the defined strategy. It is consisted of 4 Phases, beginning with the initial one (a) developing the strategy, (b) executing the strategy, (c) evaluating the strategy, and end-up to (d) maintaining the strategy. For evaluation

purposes, a set of evaluation objectives has been set related to each evaluation phase.

The National Institute of Standards and Technology (NIST) [4] has distributed the Cyber Security Framework (CSF) to develop a standardized approach to cyber security assessments for all sectors of the state's critical infrastructure. The CSF can be tailored to a variety of technologies, life-cycle stages, enterprises. The stages in the general work process are (a) defining the scope and priorities (b) orientation (c) creating a current profile (d) risk assessment (e) creating of a target profile (f) identifying, evaluating, and prioritizing gaps, (g) implementing the action plan.

PDCA (Plan-Do-Check-Act) [5] is an iterative, four-stage approach for continually improving processes, products or services, and for resolving problems. It involves systematically testing possible solutions, assessing the results, and implementing the ones that have shown to work. The PDCA/PDSA framework is effective in a wide range of organizations. It can be used to improve any process or product by dividing it into smaller steps or stages and working to improve each one.

9.3 Evaluation Framework of Cyber-Trust

Cyber-Trust from the beginning of the project sets the basis of the evaluation framework by introducing deliverables related to use case scenarios, end-user requirements, platform's architecture, and tools specifications which entailed core elements to feed evaluation process. However, the actual evaluation process began after the 1st integration phase, reaching the point where a concrete platform has been created, and can be used as a pilot during the evaluation phase.

Before the evaluation through pilot starts, the evaluation material synthesized and distributed to the end-users. The evaluation elements will be analysed below in Section 9.3.5. Also, the 7 steps described in the Figure 9.1 are analysed inside the chapter.

9.3.1 Context

The context of evaluating Cyber-Trust constructed to reach two (2) goals. The former is to reassure users about the platform's features and offerings, and the latter to quantify the solution's impact to establish it on the end user community (see Section 9.4). The consortium should first validate if the developed solution meets the end user acceptance criteria, reaching the proper thresholds for each component (e.g., cyber-attacks detection rate both at device and network level etc.).

Cyber Trust aims to advance environments with a secure setting in which European citizens feel guarded, have a sense of autonomy, and feel secure in the context

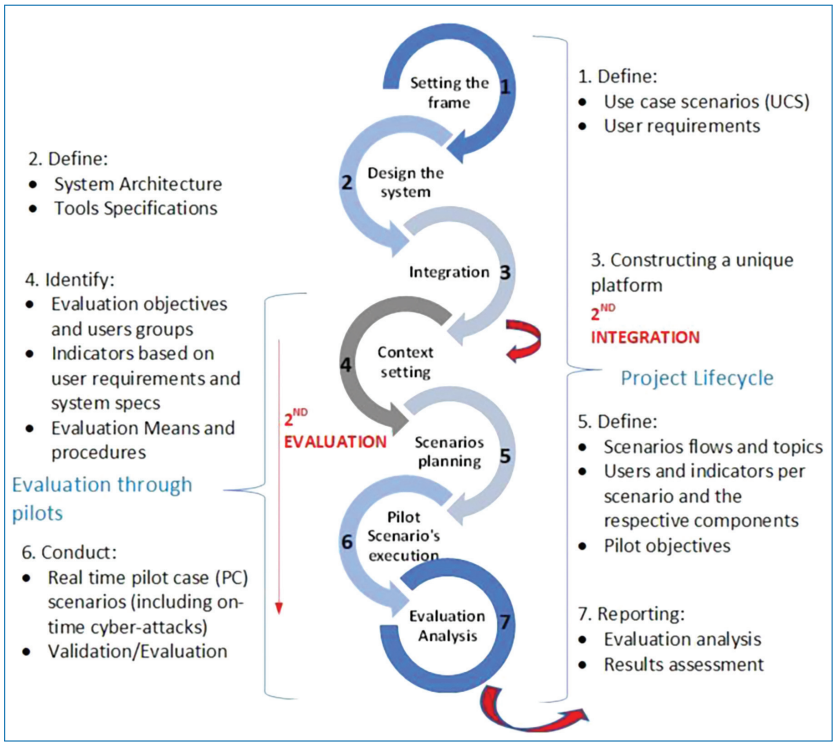


Figure 9.1. Cyber-Trust Evaluation process inside the structure of the project (Image used from D8.5).

of digital framework security. Therefore, the project not only aims to strengthen the current state-of-the-art in a variety of cyber security domains. Cyber-Trust will use advanced cyber-threat intelligence operations, identification, and mitigation mechanisms to resolve the challenges of securing the environment of IoT devices.

9.3.2 Objectives

A number of strategic objectives were established to ensure a successful pilot implementation, testing, and evaluation process.

Starting with the implementation of the first Proof of Concept (PoC) of pilot test, accompanied by the analysis of the gathered results, an operational system that provides all the expected services was developed. Continuing with pilot testing process, which is an essential part of the priorities. The platform will be thoroughly tested for achieving specific goals, such as detecting specific cyber-attacks at the device and network level (e.g., zero-day vulnerability), monitoring and developing a framework for efficient continuous vulnerability assessment and remediation, improving IoT network resistance to specific types of attacks (e.g., DDoS), and finally, providing advanced threat intelligence.

Also, security, reliability, efficiency, interoperability, and scalability are all critical evaluation goals that contribute to a successful evaluation and thus pilot testing process. As a result, the cyber-security platform will have advanced far beyond the current state of cyber security, ushering in a new era for the next generation of cyber-security architectures.

9.3.3 Assessor Teams

The end-user groups involved in the evaluation process were Smart Home Owners (SHO), Internet Service providers (ISPs), and Law Enforcement Agencies (LEAs). The three (3) groups evaluated the Cyber-Trust platform via three (3) different customized User Interfaces (UIs). There is an additional UI dedicated to ICT Administrators for ISPs users too. Each stakeholder group will access different components' functionalities, as each UI was designed solely to meet the daily needs of stakeholders, as depicted in Table 9.1.

9.3.3.1 End-Users high level needs

Table 9.1. Main purposes behind the demands of the stakeholders.

End-Users	Targets
Smart Home Owners (SHOs)	• Safeguarding Smart Home Devices and Infrastructure o Monitoring smart homes assets health status, risks levels. o Detecting abnormal traffic behavior and notifying for minor or critical vulnerabilities or possible attacks. o Alerting SHO for cyber-attacks at device and network level. o Updating devices, infrastructure security settings.
Internet Service Providers (ISPs)	• Safeguarding Customers o Monitoring customers' network infrastructure o Providing crucial information to LEAs when it is requested by their customers.
Administrators (Admins)	• High-level orchestration of ISP UI account.
Law Enforcement Agencies (LEAs)	• Improving Chain of Custody • Reduce the time needed to exchange information, which might contain forensic evidence, regarding cyber-attacks between LEAs and Internet Service Providers

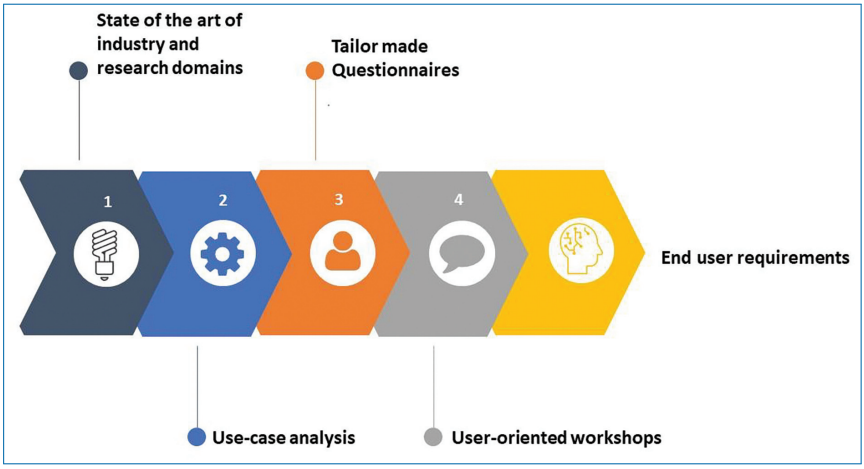


Figure 9.2. End-user extraction methodology.

9.3.3.2 End-User requirements methodology

The extraction of the end-user requirements came from both the research analysis and the aggregation of user demands. Four sources were used to determine the platform's requirements. The actions taken toward these sources are:

- The analysis of existing industry solutions and research activities-domain knowledge
- The analysis of Cyber-Trust use cases
- Conduction of dedicated workshops with the end-user groups
- Creation of targeted Questionnaires (5 Questionnaires in total)

The methodology is outlined in the Figure 9.2.

The end-user requirements were divided into functional and non-functional categories based on the content of each requirement, and then prioritised using the MoSCoW methodology.

9.3.3.3 Cyber-Trust components

Cyber-Trust contains a variety of components designed to achieve the scope of the project. The roles and the responsibilities of the components initially described through the architecture documentation and then redefined to technical deliverables, tailored to the architectural and operational needs of the tools.

Some of the components of the Cyber-Trust, presented in Figure 9.3, are used in the backend system and are not available to the users. Thus, these components are not evaluated by the stakeholders. The only components that are assessed are those with graphical user interface.

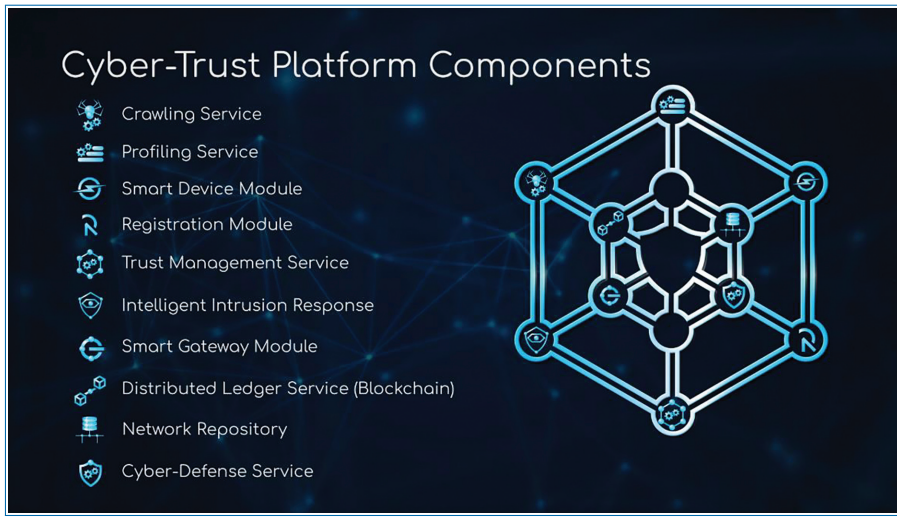


Figure 9.3. Components of Cyber-Trust (Image used from the Cyber-Trust dissemination video).

Table 9.2. Capabilities distribution among components.

C-T Components & Services	Detection	Protection	Mitigation	Storage and Sharing
Crawling Service	x			
Profiling Service				x
Smart Device Module	x	x		
Registration Module				x
Trust Management Service	x	x	x	
Intelligent intrusion response	x		x	
Smart Gateway module	x		x	
Distributed Ledger Technology				x
Network repository	x			
Cyber-Defense service	x		x	

In Table 9.2 the components are classified based on their capabilities. The following is a descriptive analysis of the tools based on their capabilities:

- Crawling Service is responsible for detecting web pages and security-related websites regarding cyber-threat intelligent in order to identify emerging threats, exploitation kits and zero-day vulnerabilities.

- Profiling Service's stores centrally information and profiles connected to Cyber-Trust devices and detects the correlation of devices' existing information with newly acquired data from other secure repositories and sources.
- Smart Device Module is running on the device and inform users for their device's health status (such as vulnerabilities detection, firmware updates, etc.). The users will be informed via alerting channels, such as mobile-app-messages.
- Registration Module provides registration capabilities to various actors, such as users and organizations including Smart Home Owners (SHOs), Internet Service Providers (ISPs), Law Enforcement Agencies (LEAs).
- Trust Management Service gathers the actions/behaviours and the vulnerabilities of the IoT devices and responds accordingly by increasing or decreasing trust.
- Intelligent Intrusion Response running on a network gateway at the user premises providing continuous monitoring of the Smart Home's security status and the computation of possible mitigation actions to sophisticated cyber-attacks.
- Smart Gateway Module is a component which is running on network gateway and is using Machine Learning techniques in order to identify network anomalies.
- Distributed Ledger Service (Blockchain) is basically related to integrity storage and enhanced sharing capabilities through the blockchain. Some principal operations are storage of data related to forensic evidence, validation of the transactions, consensus, etc.
- Network Repository is a set of tools that are used to collect, manage, and store information on a network's architecture including the topology and the security defences.
- Cyber-Defense Service deals with the cyber-attack's detection and mitigation on networks

9.3.4 Integration Phase

Cyber-Trust entails two (2) integration phases within its lifecycle, at present, the first phase has been successfully achieved. Its importance stems from the fact that the Cyber-Trust components through this phase (a) became functional and (b) were interconnected as a unified system. Three consecutive tests were incorporated into a completed Integration methodology. These tests are (a) the Functional Testing, (b) a Stress Test Plan (including Load and Stress Test, Dimensioning of Resource Utilization) and (c) a Penetration Testing Plan. The system integration and overall functional testing were focused on workflows (Use Cases), and the aim was to

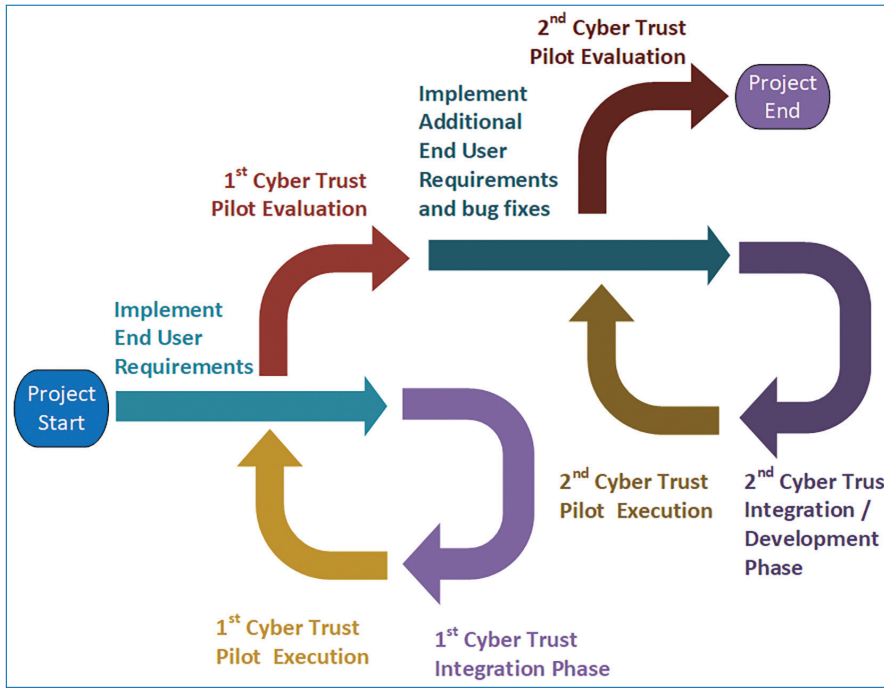


Figure 9.4. Cyber-Trust development and evaluation overall plan (Image used from D8.1).

ensure that components' messages are transmitted correctly and that the Cyber-Trust components communicated properly. Each workflow has been analysed with communication links among various components identified. The first version of the integrated platform is used in the first pilot phase.

9.3.5 Pilot and Evaluation Process

As shown in Figure 9.4, the evaluation process implemented in two (2) repeating development Cyber-Trust cycles or "sprints" in total. Cyber-Trust captured and implemented end-user requirements in the first sprint, then proceeded with system implementation before the first pilot phase. Prior the start of the second "spring" and during its duration comments collected during the first pilot apply. The goal of this structure is for end-users to receive the product that they want and benefit from using it.

9.3.5.1 Pilot trials

Pilot tests realized both synchronously an asynchronously. Synchronous tests were performed in real-time in pilot trials over a series of system evaluation sessions utilizing a dedicated six (6) hours slot. Asynchronous tests were happened remotely,

at evaluators' pace, throughout the first (1st) pilot period. Both testing methods enabled end-users to run the platform and gain experience from it while also providing valuable feedback and comments for the second evaluation phase. During both testing methods, human rights, GDPR (679/2016) compliance and e-privacy regulations were applied to all pilot cases for all testing performed.

9.3.5.1.1 Pilot scenarios

As the platform executed synchronously and asynchronously, scripts were also created for both testing purposes. For the former type of procedure one (1) consolidated pilot scenario with numerous test cases were created. In that scenario all the evaluators were able to participate. During the live trials real attack scenarios were made, making the end users familiar with dealing with cyber-attacks. For the latter type of procedure four (4) user-oriented pilot scenarios with multiple test cases were constructed and distributed to the end-users giving them the opportunity to execute all the test cases before or after the live tests. Through those scenarios the user was able to retrospect features and rules that were implemented to the platform (visualised through the UI).

9.3.5.2 Functionality verification

Cyber-Trust has created a Functionality Verification plan which includes all the appropriate actions that verify the project's functions, as it was specified by the end-user groups of the project. Functional, and non-functional requirements were included in the Functionality List. During the pilot scenarios mentioned in Section 9.3.5.1.1 the Functionality List was able to be revised and completed with the verification status (Achieved, Not Achieved, Partially and Modified). Since, the end-user requirements were converted to system specifications by the initial year of the project, the end-user requirements verification status provides an answer to the question "Does this developed system do what is intended?" [6].

9.3.5.3 Components validation (KPIs)

Key Performance Indicators (KPIs) validated the system and the components based on numerical metrics. The technical partners and the end-users enabled to validate the platform and validating the platforms components and pilot oriented KPIs. In a more simplified point of view, validation is the procedure enabling to answer the question "Did we build the right product? [7]". The KPIs of the Cyber-Trust product is constantly measured during the pilot and integration phases. Recently and compared to the last integration phase the measurements have shown that the KPI values are increasing sharply (with a minority of constant values), indicating that the quality of the product performance continues to rise.

9.3.5.4 Usability questionnaire

A single main questionnaire was developed in the sense of Cyber-Trust. The questionnaire contained closed-ended Likert scale questions, and the layout is focused on two key areas: platform satisfaction and efficiency and effectiveness of platform operations. In both questionnaire zones, the questionnaire framework and questions were tailored to each of the target end-user audiences.

The Cyber Trust questionnaire is based on the System Usability Scale [8] (SUS) and Technology Acceptance Model [9] (TAM) methodologies. SUS is a reliable tool for calculating usability. The answers are consisted by five and three Likert scale options for each respondent ranging from strongly agree to strongly disagree. In TAM, two major factors influence a user's decision about how and when to use the technology. These two factors are (a) perceived usefulness and (b) perceived ease of use. The decision of an end user to use a designed approach is influenced by the individual's personality toward using a particular method. A person's attitude toward using a tool is influenced by its perceived utility and ease of use. The two methodologies mentioned above display in Cyber Trust the Measured Perceived Ease of Use and perceived usefulness to provide a consistent and coherent analysis.

9.4 Evaluation Impact

Aside from the efficacy and performance criteria, the accessibility of web-based systems has recently become more important due to user satisfaction – being one of the powerful determinants. The academic literature has investigated usability issues of web-based platforms. Prior studies have offered valuable insights into the performance of web-based platforms. A systematic analysis is necessary to analyze the work performed in a cybersecurity environment, compare the gathered findings, identify the targeted topics and challenges that remain unresolved, and discuss future research topics that may be pursued.

In addition to the above, impact assessment is frequently used to determine whether a platform has been fully incorporated. It is also be used to address product design issues, such as determining which solution among the alternatives a platform considers to be the most promising. The second pilot phase was completed by the Cyber-Trust end-user groups (Table 9.3), and the questionnaire results analyzing the impact in the end user community are given below.

The diagram above depicts the distribution of end users. The figure also illustrates the normal distribution. What is given is fair considering the distribution of Cyber Trust end users and available network interfaces (Figure 9.5).

The objective of the Cyber Trust consortium is to see if the responses were consistent and trustworthy. Cronbach's alpha (or coefficient alpha) was devised by Lee

Table 9.3. End users distribution statistics.

End_users_distribution					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	LEAs	5	15,2	15,2	15,2
	ISPs	8	24,2	24,2	39,4
	ISPs in 3D workshop	4	12,1	12,1	51,5
	ADMINs	3	9,1	9,1	60,6
	SOHOs	13	39,4	39,4	100,0
	Total	33	100,0	100,0	

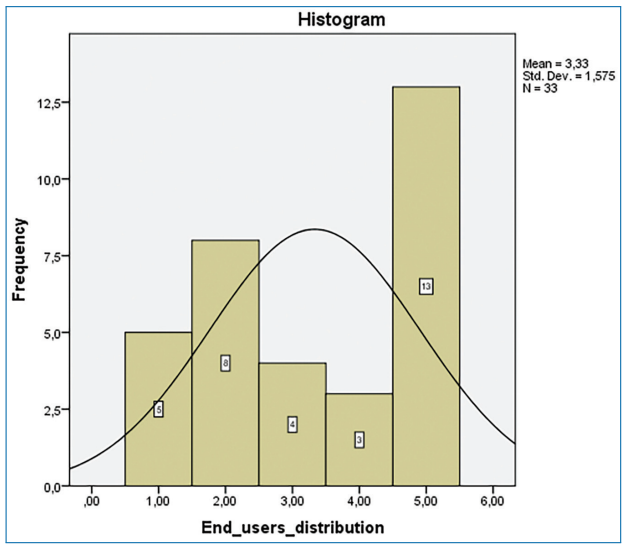


Figure 9.5. End users distribution graph.

Cronbach's alpha	Internal consistency
$\alpha \geq 0.9$	Excellent
$0.9 > \alpha \geq 0.8$	Good
$0.8 > \alpha \geq 0.7$	Acceptable
$0.7 > \alpha \geq 0.6$	Questionable
$0.6 > \alpha \geq 0.5$	Poor
$0.5 > \alpha$	Unacceptable

Figure 9.6. Cronbach's alpha interpretation.

LEAs		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
,963	,973	22

ISPs		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
,972	,946	23

ADMINs		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
,991	1,000	15

SOHOs		
Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
,949	,951	25

Figure 9.7. Cronbach's alpha results as indicated by the four different End user groups.

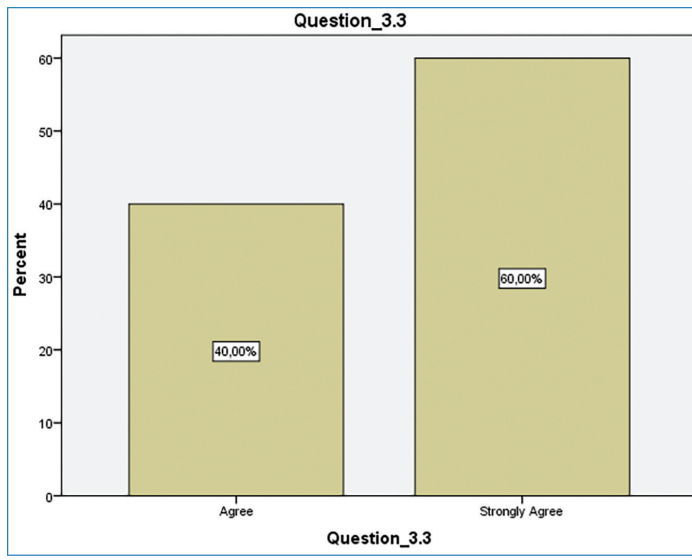


Figure 9.8. The percentage of evaluators answered to "I found easy to learn how to navigate within the Cyber-Trust platform" question.

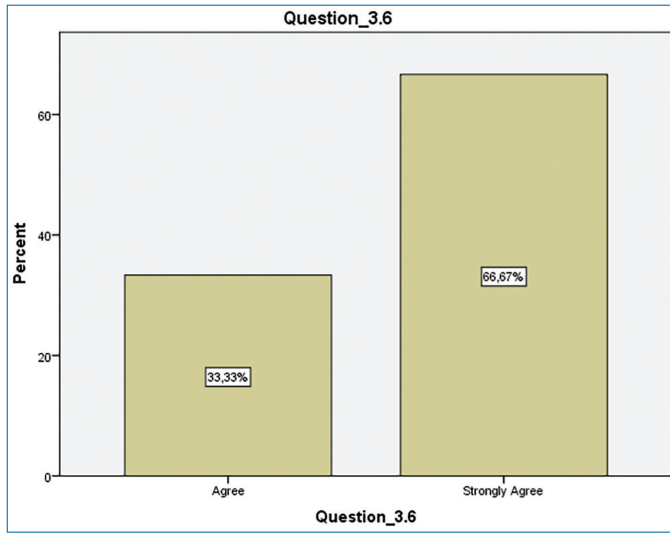


Figure 9.9. The percentage of users answered to "I felt very confident in completing all of my work using the Cyber-Trust platform" question.

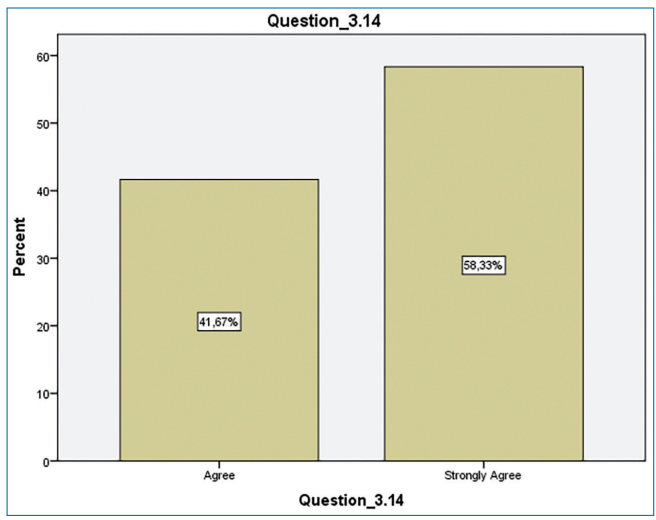


Figure 9.10. I managed to access and retrieve all information needed.

Cronbach to assess the consistency of multiple-question Likert scale surveys. The total consistency rating of a measure is determined by the coefficient of reliability, which ranges from 0 to 1. With an average internal consistency of 0.968, end users rated 96.8% reliability with the CT platform.

The majority of replies to the evaluation questionnaire indicate that the platform's user friendliness is rapidly improving. With a user-friendliness score of 62%,

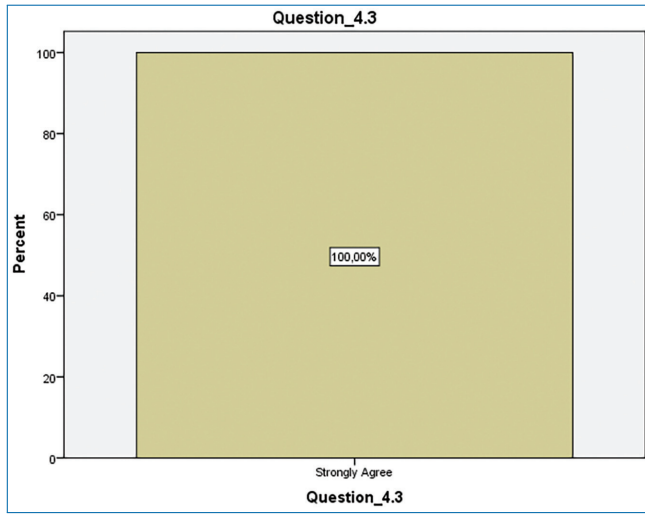


Figure 9.11. The percentage of users answered to "In the 2D-UI: By the time I logged into the system, I used at least 3 clicks rule for accessing information related to cyber-attacks" question.

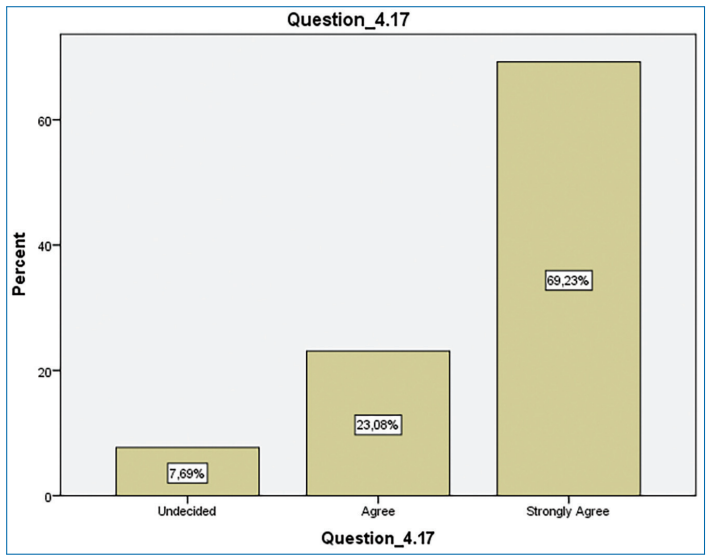


Figure 9.12. The percentage of users answered to "I would imagine that most end-users would agree that Cyber-Trust is necessary to safeguard their IoT devices against malicious cyber-attacks" question.

the CT appears to be a user-friendly platform. Furthermore, in terms of navigation and time-consuming issues, end users find it convenient and in line with their requirements. The CT looks to be an adaptable platform to varied end user needs, with an average score of 60%. "I felt quite confident in completing all of my work

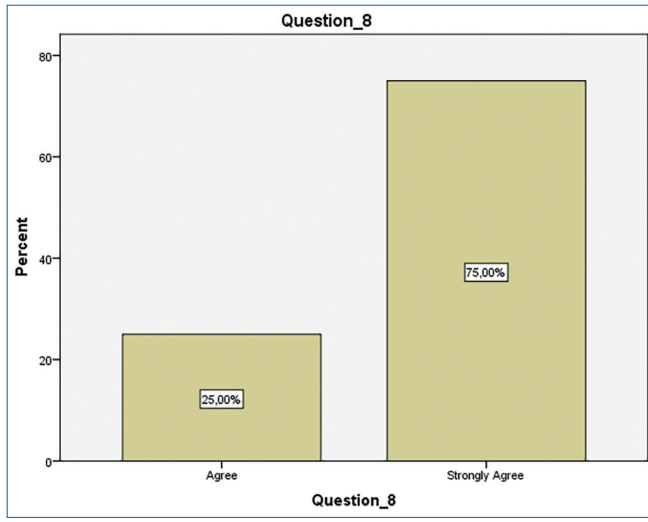


Figure 9.13. I did not experience any disorder (e.g. sickness) during the 3D interaction.

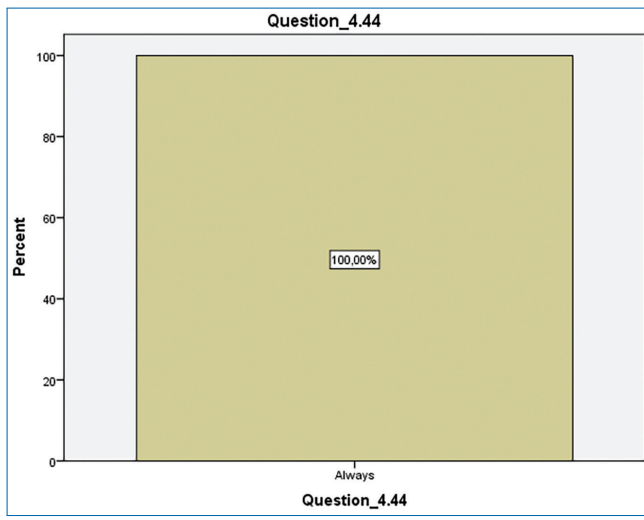


Figure 9.14. I managed simply to navigate and view all the information of a specific CVE ID.

utilizing the Cyber-Trust platform," said 67 percent of those polled in response to the question. These data demonstrate how relevant end users rate the "easy of use" of the Cyber Trust platform. Finally, 58 percent indicated they were able to access and obtain all of the information they required, indicating the platform's perceived utility.

In terms of platform efficacy and efficiency, 100 percent of the End user audience believes the Cyber-Trust user interface (UI) follows the three-click rule for acquiring information concerning cyber-attacks. Furthermore, 46% of Cyber-Trust

community end users strongly agree that Cyber-Trust is vital to protect their IoT devices from malicious cyber-attacks. Finally, when it refers to the Cyber Trust's 3D component, 100 percent of users said they did not feel any disorder (such as illness) throughout the 3D interaction as well as to simply navigate and view all the information of a specific CVE ID.

9.5 Conclusions

In a nutshell, collecting and analyzing data from pilot activities reveals the satisfaction rate of the stakeholders and the level of system's performance. More specifically, the intercorrelation of the project's tasks (containing use cases, user requirements, state of the art deliverables and the description of tools) from the beginning, enabled Cyber-Trust to record the needs of the stakeholders as well as the areas of application of the platform. The design of the evaluation methodology created based on known standards (SUS, TAM), was adapted to the scope of the project, and the evaluation material was designed to assess the technological advancements of the Cyber-Trust. Also, comments during the pilot phase eventually led to the drastic modification or enhancement of an evaluation element. Consequently, the Cyber-Trust Evaluation Process is vital not only for gathering information and evaluating pilots but also for providing feedback on what features in the graphical user interfaces (GUIs) and procedures need to be improved. The results obtained through the Cyber Trust platform will lead to the advancement of revolutionary emerging solutions that improve commercial visibility and feasibility of a high technical readiness level product that offers a comprehensive solution to cyber security issues.

Acknowledgment

This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] Cyber-Trust, "The ever-evolving IoT landscape: Blessing or Curse?," October 2020. [Online]. Available: https://cyber-trust.eu/2020/10/12/the-ever-evolving-iot-landscape-blessing-or-curse/.

[2] Innovate UK, "Innovate Uk_ Evaluation Framework," 2018.
[3] ENISA, "An evaluation framework for Cyber Security Strategies," ENISA, 2014.
[4] NIST Information Technology/Software and Systems Division, "Methodology Overview," NIST, May 2017. [Online]. Available: https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/cftt-general-0.
[5] Investopedia, "PDCA Cycle," August 2020. [Online]. Available: https://www.investopedia.com/terms/p/pdca-cycle.asp.
[6] J. Bach, "Good Enough Quality: Beyond the Buzzwords," IEEE Computer, 1997.
[7] S. Rakitin, "Software Veriication and Validation for Practitioners and Managers," Artech House, 2001.
[8] J. R. Lewis, "Computer System Usability Questionnaire," [Online]. Available: https://garyperlman.com/quest/quest.cgi.
[9] UserSense, "Technology Acceptance Model (TAM model)," [Online]. Available: https://www.usersense.at/analysing-usability-testing/technology-acceptance-model.

DOI: 10.1561/9781680838350.ch10

Chapter 10

Smart Home Testbeds for Business

By P. Douris[1,*], *A. Salis*[1,†], *E. Sfakianakis*[2,‡], *M. Rantopoulos*[2,§], *D. Kavallieros*[3,¶] *and G. Sargsyan*[4,‖]

[1]Centre of Security Studies (KEMEA)
[2]OTE group Technology & Operations
[3]University of the Peloponnese
[4]CGI
[*]p.douris@kemea-research.gr
[†]a.salis@kemea-research.gr
[‡]esfak@oteresearch.gr
[§]mrantopoul@cosmote.gr
[¶]d.kavallieros@uop.gr
[‖]gohar.sargsyan@cgi.com

We present a testbed, which hosts and interconnects ten (10) simulated, 750 emulated and one cyber-physical Smart Home (SoHos). The SoHos are digitally organised in three testbeds. Our chapter is structured in five Sections.

This chapter provides information regarding the design, architecture and implementation of these large number of SoHos, deployed for running multiple cyber attacks (more than 20 different attacks) for testing and validating the capabilities of the Cyber-Trust platform developed during the European Commission co-funded research and innovation Horizon 2020 Cyber-Trust project [1].

In Section 10.1, the significance of these testbeds, both from a marketing and exploitation perspective for different types of organisations; these shall benefit from the exploitation of the results and relevant information, arising from the use and utilisation of the aforementioned platform. In Section 10.2, we present the

requirements, both technical and non-technical as well as the interconnectivity of different, heterogeneous technologies present. In the Section 10.3, the main results are presented and some discussion on them follows. Section 10.4 is dedicated to the exploitation of the results, their impact on potential business and possible extensions.

10.1 Introduction

Nowadays, the massive production of affordable, easy-to-access and easy-to-use smart devices, in combination with the increasing and improving telecommunications network coverage has led to the advent of the so-called SoHos. Moreover, the extreme complexity, associated with the fact that data, coexisting networks (often multiple types of networks), pass from multiple networks, which reside at the same place, the coexistence of different protocols, such as 4G, 5G, Wi-Fi, etc. as well as the need for continuous machine-to-machine communication and the associated protocols (e.g. Bluetooth) are indicative of the level of complexity present. To this end, security and privacy issues, arising as a result of the presence of different protocols and the fact that the same data travel via different protocols and are at the same time exposed to the internet lead to a further increase of complexity.

The popularity of SoHos and their adoption from an increasing number of people all around the globe is increasing more and more, both in non-commercial as well as in commercial environments. Evidently, there are entities, such as organisations, companies and bodies, belonging to the latter category, which can greatly benefit from the results produced, the conclusions drawn and lessons learnt, after conducting research on SoHos. These entities include, but are by no means limited to the following main categories:

- Information and Communication Technologies (ICT)
- Research organisations, carrying out and/or interested in pilot testing
- Security organisations/companies
- Any technology organisations/companies with a focus on or active in the field of SoHo technologies, services and/or the associated smart devices

Therefore, having actively contributed to the field of security technologies, testbed set-up, in general; and after conducting research in the field under consideration through KEMEA's active participation in the testbeds and execution of pilots of the project "Cyber-Trust (CT)", OTE's/Cosmote's contribution in testbed, and CGI's contribution in exploitation and market uptake, we are in a very good position to present and share valuable results and specifications, based on the successful experiments carried out within the context of "Cyber-Trust" [1]. These

exploitable results of "Cyber-Trust" mainly fall within the scope of potential business and exploitation-wise solutions, strategies and associated business, financial, technological and research areas.

Now, we proceed with the specifications, both technical and non-technical, entailed in the process of the testbeds set-up, their interconnectivity, the different-heterogeneous technologies present as well as some noteworthy tools, requirements and details.

10.2 Cyber-Trust Testbed Specifications

First of all, the testbeds have been set up with the aid of different, intrinsically heterogeneous virtualization technologies, be they:

- Microsoft Hyper-V: which is installed on KEMEA's premises and has been used to set up KEMEA's testbed and the associated Virtual Machines (VMs); this is not cloud-based.
- OpenStack: which constitutes open-source cloud software; it is installed on OTE's premises and has been used to set OTE's SoHo VMs
- Variety of Operating Systems used for the cyber-physical

The testbed did not only include the SoHos but also the Cyber-Trust platform, Command and Control Server for the Mirai, Black Energy, ZEUS and ZitMo attacks.

So, the individual technologies mentioned above needed to be:

- interconnected
- made to work continuously, in real time (or at least continually sometimes), and
- synchronised and capable of interacting with the user(s) as realistically as possible, so as to be able to imitate the real-world functionalities and characteristics of smart homes and/or smart home networks

Undeniably, this testbed, a graphical representation of which is shown in Figure 10.1, is truly complex from an infrastructure point of view as well as from a connectivity and functionality one. Nevertheless, not only is the high level of complexity justified, but it is necessary, as well. The significance of its complexity lies in the fact that the real-world system is really complex and it involves a wide variety of different coexistent technologies, so the underlying complexity in the testbed is deemed as necessary, should the simulation be as realistic as possible. Therefore,

considering the different technologies present in the real-world situation (such as Bluetooth, 4G/5G, Wifi ones, infrared, let alone the different architectures, versions and implementations of them), the resources needed, and the cost of a real-world testbed, the large amount of time spent for setting our complex testbed up and the difficulty involved can be justified.

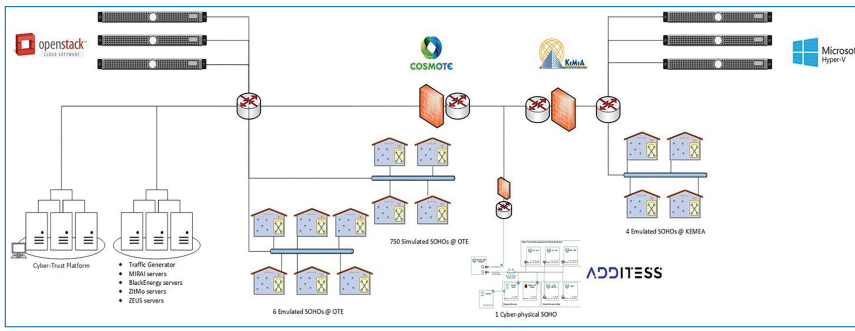


Figure 10.1. Graphical representation of cyber-trust testbed.

The structure, components and basic characteristics and resources of each simulated SoHo of our testbed are shown in Figure 10.2, below:

SOHOs	A05/TMS	A13/IRE	A13/IRG	A04/A16	MTSPL	Ubuntu	Win XP	Win7	Win7 SP2	A12/SDA	Android	Bbox #1	Bbox #2	TOTAL	vCPU	VRAM (GB)	VHDD (GB)
SOHO 1	1	1	1	1	1		1	1		1	1	1	1	11/7	20	66	352
SOHO 2	1	1	1	1	1				1	1	1	1	1	11/6	18	58	320
SOHO 3	1	1	1	1		1		1		1	1	1	1	11/6	18	58	320
SOHO 4	1	1	1	1		1			1	1	1	1	1	11/6	18	58	320
SOHO 5	1	1	1	1	1	1			1	1		1	1	11/5	20	64	336
SOHO 6	1	1	1	1	1	1				1	1	1	1	11/6	18	56	304
SOHO 7	1	1	1	1	1	1		1		1	1	1		11/6	24	70	390
SOHO 8	1	1	1	1	1	1		1		1		1	1	11/6	22	72	416
SOHO 9	1	1	1	1		1	1		1	1	1	1		11/6	24	72	406
SOHO 10	1	1	1	1		1	1		1	1		1	1	11/6	22	74	432
TOTAL resources allocated for the Smart Homes															202	652	3708

Figure 10.2. Virtual machine components of a typical testbed.

10.3 Interconnectivity via an ad-hoc Routering Process

The interconnectivity among the heterogeneous networks has been achieved by means of a customised routering process, with user-defined IP routing tables, simulating the exposure of IP addresses from the provider as well as the routing process within the domestic, business, industrial networks, which host the SoHos. The router emulator is an Ubuntu VM which implements the routing process. Together with the traffic generator, which is another Ubuntu VM, responsible for the network traffic generation. Furthermore, to ensure the connections are established securely, the necessary, dedicated certificates have been issued and installed into each SoHo; the open-source software OpenVPN [2] has been used for the

establishment of secure connections via the Secure Sockets Layer (SSL) and the standard ssh service in Ubuntu, too. In the case of different Operating Systems, OpenSSH tools have been used to the same purpose.

10.3.1 Cyber-Trust SoHo Components

Therefore, a graphical representation of our deployed SoHo together with its connections and interactivity with the **Internet Service Provider (ISP)** as well as any external or internal networks (e.g. WAN, LAN, etc.) is presented in Figure 10.3 below.

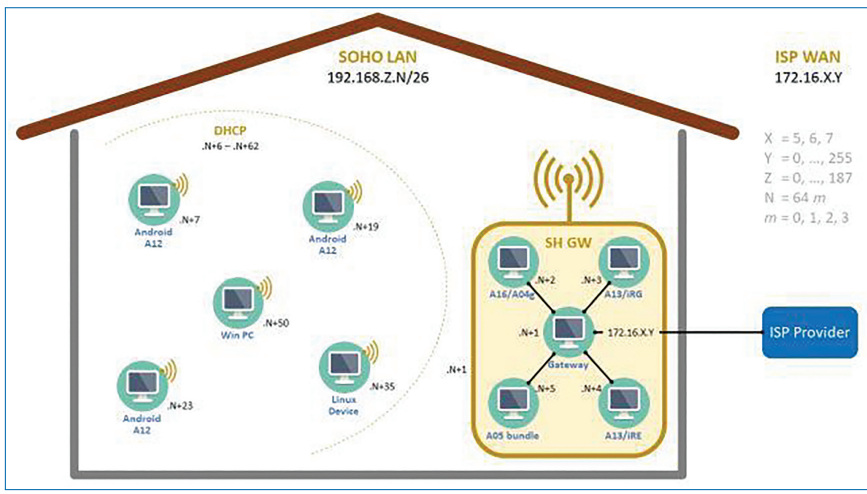


Figure 10.3. Deployed smart home (SoHo) ecosystem.

10.4 Tools Used & Utilised – Methodologies Adopted

Given the testbeds under consideration lie in different infrastructures, conversions of virtual hard disks to the formats of interest, cross compilation and building tasks play a significant role in enforcing and maintaining compatibility across the different environments. Additionally, the establishment of a continual testing/verification procedure, ensuring the viability of the interacting testbeds has been of paramount importance.

Regarding the conversion among different virtual disk formats, such as VDI (Oracle Virtualbox, openstack), VMDK (Oracle Virtualbox, VMWare products, QEMU, Parallels Desktop for Mac, openstack), VHD (Hyper-V, Oracle Virtualbox, openstack), VHDX (Hyper-V, openstack), the image file format of Parallels version2 HDD (Oracle Virtualbox), qcow2 (openstack, QEMU), raw

(just to mention a few widely used ones), open-source tools have been used. These tools include qemu-img [3], the VBoxManage command-line tool as well as StarWind V2V Converter.

Moreover, several types of experiments have been carried out, including cyber-attacks, the logging of the attacks, and severity alerts have been generated and graphically presented through a Graphical User Interface (GUI), built and set up in terms of the same Project to bridge the Human-Computer Interaction (HCI); i.e. the platform interface.

10.5 Results & Discussion

Now, we present the generated results. The testbeds functionality has been proved to be excellent. More specifically, the phases of Cyber-Trust assessment, evaluation, integration, testing, pilot execution and results analysis together with the corresponding results (i.e. functionality verification results, evaluation results, performance measurements results etc.) have been made available to date. These include:

(i) System integration and overall functional testing results: the Cyber-Trust Platform is based on event−driven, loosely coupled service-oriented architecture that implements a publish/subscribe approach, supported by direct component communication via RESTful interfaces.

(ii) Performance Testing Results: Load and stress testing have been conducted and appropriate **Key Performance Indicators** (**KPIs**) have been defined and evaluated. The tests include regression testing, connectivity and accessibility of services testing, load and stress testing, etc.

(iii) End-user evaluation results: The evaluation process covers the different methods used to assess the evaluation material. It also presents the evaluation material (e.g., manuals, questionnaires, test case scenarios, requirements, KPIs etc. In more details, synchronous and asynchronous types of tests were used to evaluate the platform as a whole and its services. Synchronous tests were carried out concurrently in a series of system evaluation sessions using a dedicated three-hour (3) slot, with the involvement of various stakeholders. Asynchronous tests were performed at the pace of evaluators, remotely. A training demo towards the end-users has been carried out as well. In both testing methods, functional requirements were verified by the related end-user group and the non-functional requirements were verified both from technical partners and evaluators. Moreover, a usability test examining efficiency, effectiveness, satisfaction, ease of use and usefulness was shared.

(iv) Penetration testing and results (to be extracted/published): These will

- determine the minimum level of security;
- include penetration testing at application level;
- be associated with session management, authentication, access control;
- take into account password complexity, user management, edit/recover password, open ports, reverse proxy, etc;
- encompass best practices mentioned, based on: OWASP, ASVF, ISO27001
- incorporate code review of components utilising automated means.

All the aforementioned results extend far beyond the present context; for instance, to scientific areas including e-privacy, GDPR, ethics, etc. An overview of the Evaluation and Assessment procedure is presented in Figure 10.4 below.

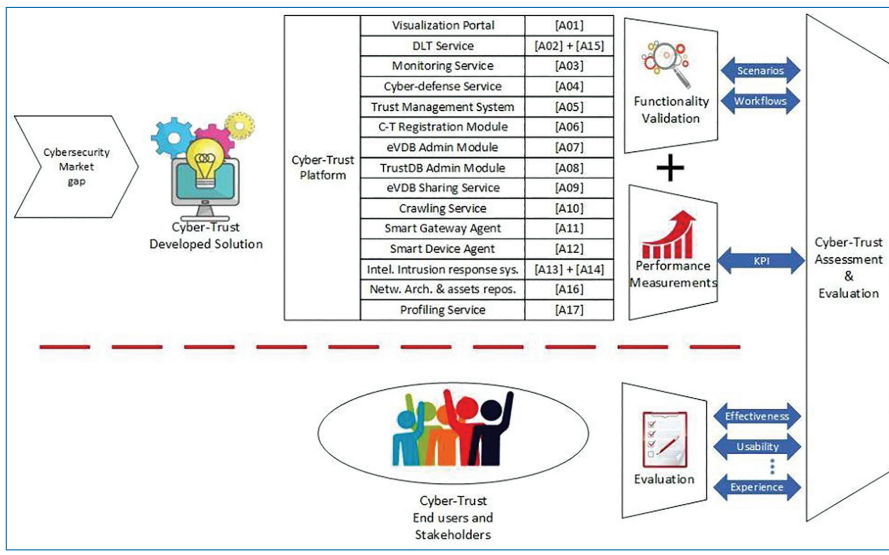


Figure 10.4. Evaluation and assessment procedure.

10.6 Exploitation of Results & Impact on Business

The aforementioned results can be greatly exploited first of all by the consortium members of the Cyber-Trust project, by the organisations, companies and authorities, engaging in the following fields or similar [4]:

- ICT
- Research with piloting

- (Information) Security
- Cyber Crime
- Smart home solutions/devices/electronics/equipment
- Smart devices

Additionally, the exploitation of the results extends far beyond the categories mentioned above. More specifically, setting out from the results collected, processed and post-processed after carrying out the cyber-attack simulations [5, 6] and the associated pilots as well as the related [7, 8], the interested entities can take full advantage of them within the following contexts:

1. Simulation of cyber-attack and prediction of impact on business. Multiple scenario-based subsequent simulations with and without (possibly) affected components and evaluation of impact on business together with disaster recovery scenario and optimisation (optimal scenario/scenario adoption. Dynamic optimisation possible.
2. A step closer to (co-)simulation-in-the-loop side by side with the real-world business activities.
3. Testing and hardening of processes, components, upgrades of self-defence and cybersecurity components, improvement of failover strategies.
4. Improvement of existing smart home devices, equipment, software
5. New smart-home devices, equipment, software
6. Improvement of interconnectivity among (technologically) heterogeneous smart homes and smart home devices
7. Develop strategies for bridging and tackling different, currently incompatible smart components/and or devices, including but not limited to those bearing agnostic components and/or closed-source code.

10.7 Conclusion

In the present article, we have presented the results from our simulated, tested SoHo platform, their exploitation potential in several fields, mainly from a business perspective as well as their impact on business and extensions. We have also analysed the challenges faced, as far as their complexity is concerned, both in terms of interconnectivity of inherently different, though compulsorily interacting and cooperating technologies and protocols. To this end, we have also presented our successfully adopted methodology, custom routing process to ensure interconnectivity as well as smooth, uninterrupted cooperation among components. Last, but not least, we have discussed the broad applicability of our implementations and justified the

need for setting up such complex, heterogeneous and resource demanding testbeds towards the realisation of a realistic, nearly real-world simulation environment.

Competing Interests

The authors declare there is no conflict of interest regarding the publication of this chapter.

Acknowledgment


This work has received co-funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

References

[1] Cyber-Trust Project – https://cyber-trust.eu/ – Advanced Cyber-Threat Intelligence, Detection and Mitigation in Trusted IoT, EU H2020 project grant agreement no. 78669.
[2] OpenVPN https://openvpn.net/
[3] https://linux.die.net/man/1/qemu-img
[4] G. Sargsyan *et al.*, "Final Exploitation and Technology Implementation Plan" Deliverable (D9.10) of Cyber-Trust 2021.
[5] G. Boulougaris *et al.*, "Device-level attacks: proposed solutions" Deliverable (D6.6) of Cyber-trust 2021.
[6] G. Bendiab *et al.*, "Network-level attacks: methods and results" Deliverable (D6.7) of Cyber-Trust 2021.
[7] V. G. Bilali *et al.*, "Platform's 1st evaluation report" Deliverable (D8.3) of Cyber-Trust 2021.
[8] V. G. Bilali *et al.*, "Platfrom's 2nd evaluation report" Deliverable (D8.5) of Cyber-Trust 2021.


SECURITY
CONFIRM
click here for more information

DOI: 10.1561/9781680838350.ch11

Chapter 11

Securing Today's Complex Digital Realities

By A. Rajkumari and C. Wallace†*

CGI
*a.rajkumari@cgi.com
†craig.wallace@cgi.com

Today's organizations require agility and innovation to deliver seamless digital experiences—anytime, anywhere. In response, customer, employee and supplier ecosystems have become more complex, connected and open. At the same time, cyber risks and threats are growing in velocity and complexity.

To address these challenges, enterprises need a balanced and proactive cybersecurity approach. This includes managing human and non-human digital identities and access, protecting both information and operational technologies, securing multi-cloud environments, safeguarding automation and artificial intelligence workloads, and complying with increasing regulations.

Our cybersecurity approach for today's modern work environments has been tested and proven. We bring accelerators in the form of maturity models, reference architectures, technical know-how, cross-domain expertise, risk management methods, and client lessons learned to accelerate and empower your business. With CGI's Cybersecurity Advisory Services and Accelerators, you can increase agility and innovation while ensuring holistic management of cyber risks.

In this chapter, we cover the new digital reality and what it means for cybersecurity, and how CGI is helping its clients secure their connected operations.

11.1 Today's Digital Reality and What it means for Cybersecurity

Enterprises are continuously evolving to deliver value to customers, citizens, employees and shareholders at pace in response to fast-changing needs.

New technologies, data sources and connections are enabling this evolution, including multi-cloud environments, edge computing, automation, artificial intelligence (AI), Internet of Things, 5G, micro-services, devices, and application programming interfaces (APIs). However, cyber threat actors are harnessing these same advances to create an increasingly sophisticated and dynamic risk landscape. The cybersecurity arms race is escalating.

Enterprises also are expanding their supplier ecosystems and customer bases. Many are involved in mergers, acquisitions, divestitures and reorganizations, and have increasingly hybrid workforces (human and non-human) operating from almost anywhere.

GOOD TO KNOW

Growing importance of cybersecurity [1]:

- Cybersecurity is the most frequently mentioned business priority
- 64% say securing cloud platforms is a key cybersecurity priority for their organization
- 25% say they do not know whether they have mechanisms in place to locate where key data assets are processed and stored

Preventable identity-related breaches [2]:

- 79% of organizations have experienced an identity-related security breach in the last two years, and …
- 99% believe their identity-related breaches were preventable.

The future is a hybrid world [3]:

- By 2025, there will be 55.7 billion connected devices worldwide, 75% of which will be connected to an IoT platform.
- By 2023, 75% of the G2000 commit to providing technical parity to a workforce that is hybrid by design, rather than by circumstance, enabling them to work together separately and in real-time.

Mergers and acquisitions are increasing [4]:

- Since 2000, more than 790,000 M&A transactions have been announced worldwide with a known value of over US$ 57 trillion.

11.2 Protecting the Business Without Inhibiting Innovation and Pace

In this digital reality, executives have top priorities:

- **Enable innovation and collaboration at pace:** Today's organizations extend beyond traditional enterprise boundaries to external ecosystems—so does security. A modern approach across the continuum of security operations enables the safe creation, operation and evolution of flexible, efficient and collaborative ecosystems, and ensures seamless experiences.
- **Reduce risk exposure and effectively manage risk:** An insights-led approach to risk management uses rich data to identify and manage risks holistically across the enterprise in near real time, allowing for proactive and comprehensive risk mitigation and fast response to threats. It includes managing human and non-human digital identities and their secure access, advanced threat monitoring and response and so on.
- **Improve regulatory compliance:** Data is everywhere and is fueling innovation, new revenue opportunities, better user experiences, and optimized operations. Ensuring the right access to this data is critical to complying with increasingly strict regulations.
- **Adopt a proactive stance through real-time situational awareness:** Critical to modern security operations is having the right processes, skills and technologies. These technologies include advanced analytics, artificial intelligence, machine learning, automation and orchestration of cybersecurity workflows, as well as real-time visualizations of your vulnerability and threat landscape.

- **Be prepared and respond effectively when a crisis occurs:** As cyber threats and risks grow in volume and complexity, modern organizations are prepared for crisis situations and are ready to respond effectively, while capturing lessons learned.

What does the success look like? In Figure 11.1 the answer is given with details and explanation.

Figure 11.1. What does the success look like?

11.3 CGI Cybersecurity Advisory Services

We highlight eight key advisory services to help clients achieve an insights-led balanced approach to cybersecurity in this new complex and connected digital reality [5].

11.3.1 Digital IAM Services

The variety, volume and velocity of both human and non-human (or silicon) identities (e.g., Internet of Things sensors, devices, software, artificial intelligence, micro services, and application programming interfaces) and their access needs are increasing dramatically. With our Digital IAM Advisory Services, you can achieve agility and innovation while keeping digital identities and their access to critical systems and data both secure and frictionless. Our services range from identity governance and administration (IGA) strategy and roadmap development, to specific IGA advisory services for the new classes of silicon and external identities, to IAM operating model design, to IAM federation and integration across your enterprise and ecosystems.

11.3.2 Secure Multi-Clouds Operations Advisory

Hybrid, multi-cloud environments are becoming the new normal, creating complex security environments. Our experts can advise you on how to integrate cloud services securely into your IT landscape. We start with a maturity and risk exposure assessment and then design a blueprint for building an operating model that secures your operations in a hybrid world. We accelerate this process by bringing pre-defined controls for Amazon Web Services, Microsoft Azure and Google Cloud, as well as a maturity model and reference architecture.

11.3.3 Secure Automation Advisory

We know that automation is a key enabler of cost and operational efficiencies, as well as an improved customer experience. Many enterprises seek to automate tasks and use artificial intelligence to drive that automation, and we can help you do this securely. Through this advisory service, we assess your automation maturity, including security aspects, using our maturity model. We also assess pain points, identify data privacy issues in processes (e.g., security calls in HR processes), and catalog your target systems.

11.3.4 Digital Risk Management Advicory

The digital world comes with new risks—from evolved threats, to interconnected systems and technologies, to hybrid and perimeter-less work and IT environments, to complex data and privacy regulations. This requires much more dynamic, fluid and continuous risk management, crisis preparedness and rapid response. Our experts can help you manage your risks effectively, while ensuring you continue to deliver business outcomes at pace. Our services include integrated risk management programs, dynamic visualization of enterprise risks, privacy and compliance assessments, supply chain resiliency and risk management, cybersecurity crisis preparedness, and crisis response support.

11.3.5 Digital Security Operations Modernization Advisory

Security operations approaches of the last decade or even the last five years (e.g., pre-cloud, pre-smartphone, pre-artificial intelligence (AI), pre-bots, pre-Internet of Things/operational technology) no longer are viable. Today's digital demands require a fundamental change in security operations, whether evolutionary or transformative. Through our advisory services, we assess your current state of capabilities across tooling, processes and talent. This includes evaluating your environment scope, data sources, connectivity, logging and event streams, deep analytics and

AI, incident processing, threat intelligence, orchestration and automation, hunt capabilities, and designs. We report our findings and jointly develop a modernization strategy and roadmap with prioritized practical initiatives (e.g., re-platforming, mentoring and upskilling/training). We also offer a hybrid "own vs. buy" advisory service and assist you in developing the supporting strategic business case.

11.3.6 Cybersecurity Privacy by Design Framework

Easier access to development platforms means that more development is happening outside of the IT department (e.g., citizen developers and shadow IT). Enterprises increasingly seek greater connectivity and interoperability of the systems and services within their supply chains to improve efficiency, collaboration and the user experience. Data and privacy regulations and breaches have increasingly expensive consequences. All of these factors reinforce the fact that embedding cybersecurity and privacy into every project is much more efficient and effective than managing it as an afterthought. Security and privacy teams should establish standard, ready-to-use solutions for all IT and business projects. Our experts can assist you in building frameworks to achieve this level of readiness and reuse. After a thorough analysis of your current landscape, we recommend specific measures to fill gaps, including tooling advice and support.

11.3.7 Security Service Center Design

Increasingly, digital organizations require flexible access to new skills, retention of critical knowledge, and automation to ensure business continuity and resilience. We can work with you to design a security service center that meets modern needs, standardizes practices, and delivers the right level of expertise. We start by gaining an understanding of the services required, and then build a service catalog, design how to engage the service center, and establish a continuous improvement process.

11.3.8 Security Operating Model Design

When you initiate digital initiatives, new organizational structures, or mergers, acquisitions or divestitures, or carve-outs, your security target operating model (TOM) must be adapted to ensure all processes and infrastructures reflect these changes. Our experts work with you to design and implement your TOM by assessing your as-is state, identifying weaknesses and gaps, designing a new model (including processes and governance), and gaining approval and acceptance. We use proven templates and best practices to accelerate the process.

11.4 Cases in Point

Serving as access control broker for 10+ million industrial IoT digital assets for an industry-wide service

For a large nationwide program involving the rollout of millions of industrial IoT digital assets, CGI designed, built, implemented, hosted, ran and supported the data services that lie at the heart of this program. Our IAM advisory services, along with security services enable companies to access information to improve their services and customer experiences. These IAM services are crucial to the maintenance of consumer confidence which underpins the nationwide program and rollout.

Our solution provides a high-availability, high resilience communication service in accordance with specifications and provides an access control function that cryptographically validates all access requests and verifies right of access against IoT registration data. It also includes an industry-wide federated identity provider (IDP) service, enforcing federated two-factor authentication for employees of industry parties, roles and privilege assertion using SAML, and self- service management by industry party administrators. In addition, the IDP service also includes effective management of privileged staff, management of risk in accordance with ISO 27005 and delivery of associated security services.

Moving to the cloud securely and reliably

When a large aerospace and defense company sought to implement its public cloud migration strategy, data security and service reliability were of critical importance. Based on our significant experience in third-party vendor management, as well as managing cloud environments and their related risks, the client engaged us to assist in negotiating the security management aspects of its public cloud contracts.

This included developing a standard security annex and contract clauses, analyzing cloud provider security practices, conducting negotiation workshops, and providing a residual risk assessment. For the tailored security annex, we defined criteria for selecting applicable security requirements based on service type and identified process improvements.

In addition to completing negotiations, the client now has a standard set of requirements and documented process to support future procurements that includes early involvement of the security team.

Innovation, collaboration, co-creation, experimenting and prototyping with partners and clients

We invest in collaboration, innovation, knowledge exchange with internationally recognized experts in the cybersecurity field with the aim of enhancing our cybersecurity knowledge, skills, services and approaches. An example is the Horizon 2020 European partnership research and innovation project Cyber-Trust, where

CGI together with 8 other partners from 7 European countries joint forces to develop innovation advanced cyber-threat intelligence, detection, mitigation ecosystem [6]. We are also regular contributor to different innovation fora on cybersecurity topic engaging our clients and partners.

11.5 Achieving a Balanced, Proactive, Insights-led Cybersecurity Approach

We know that without the right cybersecurity and privacy protections, you face evolving risks and obstacles to innovating and collaborating effectively. Therefore, our goal is simple. We want to help you operate and transform at pace and with confidence—today and into the future.

With 45 years of experience in securing critical business systems across a range of industries globally, our cybersecurity approach for today's modern work environments has been tested and proven. Thanks to this experience, we bring accelerators in the form of maturity models, reference architectures, technical know-how, cross-domain expertise, risk management methods, and client lessons learned to accelerate and empower your business.

By staying abreast of rapidly changing technologies, ecosystems and threats, our consultants work closely with you to understand your environment and needs. We help you to achieve the right balance between business agility and effective deterrence, defense, detection and response capabilities.

We stand ready to help you to secure your digital operations.

References

[1] CGI, Voice of our Clients 2021 – https://www.cgi.com/en/voice-of-our-clients
[2] IDSA, Identity Security: A Work in Progress – Identity Defined Security Alliance (idsalliance.org)
[3] IDC, The Future Enterprise: The Next Normal Priorities Driving Technology Investments, October 2020, and FutureScape_2021_Cloud
[4] Institute for Mergers, Acquisitions and Alliances (IMAA), https://imaa-institute.org/mergers-and-acquisitions-statistics/
[5] CGI Cybersecurity Advisory Services, https://www.cgi.com/en/cybersecurity/cyber-advisory-services
[6] Cyber-Trust Project – https://cyber-trust.eu/ – Advanced Cyber-Threat Intelligence, Detection and Mitigation in Trusted IoT, EU H2020 project grant agreement no. 78669

DOI: 10.1561/9781680838350.ch12

Chapter 12

Security and Privacy in Digital Twins

By G. Sargsyan

CGI
gohar.sargsyan@cgi.com

Digital Twins term is one of the important topics in the digitalization world which is becoming increasingly important in different areas of industries. There are many debates which explore the growing importance of digital twins, including, possibility that they will take the control over humans, or the difficulties to interact with digital twins, the end-users of it, their impact on society and sustainability and making this world a better place, and last but not least, security and privacy aspects in Digital Twins. This chapter will explore the security and privacy in Digital Twins based on the author's – G. Sargsyan's presentation given during the event "Digital Twin a Promising Thing?" on Oct 29, 2020 in Amsterdam, which was broadcasted globally and organized and hosted by Amsterdam University of Applied Sciences in collaboration with the Digital Society School [1]. In this event the author shared her views on digital twins for different industries, risks, privacy, security and ethical considerations introducing practical examples, which is introduced in this chapter. Recommendations how to manage risks, security and privacy concerns are also offered and demonstrated in this chapter.

12.1 Today's Digital Reality and What it means for Cybersecurity

Digital twins are virtual replicas of physical devices that combing data science and IT can be used to run simulations before actual devices are built and deployed. They are also changing how technologies such as IoT, AI and analytics are optimized. Digital twins are becoming a business imperative, covering the entire lifecycle of an asset and forming the foundation for connected products and services. Although the term "digital twin" was first coined in 2002 [2], the concept itself goes back further. In 1970 NASA pioneered this idea of working with digital models of real-world systems during its Apollo missions. Being able to create accurate simulations, based on real-world data, played a significant role in helping NASA bring its astronauts safely back to Earth following equipment failure on Apollo 13 [3].

Nowadays, digital twins are becoming a business imperative, covering the entire lifecycle of an asset and forming the foundation for connected products and services. Companies that fail to respond will be left behind.

There is tremendous amount of market research conducted on the Digital Twin topic. To name a few, selected facts and figures are introduced from market research. According to MarketsAndMarkets report, the digital twin market is expected to grow from $3.1 billion in 2010 to $48.2 billion by 2026 at a CAGR of 58% from 2020 to 2026 with some of the largest adopters being healthcare and defense [4]. Gartner argues that by 2021, half of large industrial companies will use digital twins, resulting in those organizations gaining a 10% improvement in effectives [5]. With the number of connected devices forecast to grow to 42bn by 2025, according to research group IDC, we are rapidly entering the era of "hyper-data". Each of those devices emits a constant stream of data, enabling us to build a digital cloud that will metaphorically encircle our planet. We can, to use the jargon, create "digital twins" of the real world [6]. To reflect the reports, it is evident that digital twins will transform the world and business need to stay relevant not to miss their opportunities.

12.2 Cases in Point

I highlight three cases in point from business and real world about security and privacy in digital twins.

12.2.1 Smart City

Imagine the challenges associated with moving a city. This is the reality faced by Sweden's northernmost city, the mining town of Kiruna. To continue the safe

growth of mining – an industry central to the city's economy and culture – Kiruna and its 18,000 residents are moving 3 kilometers east. While new homes and a new city center are built, some of Kiruna's most historic buildings, such as the Kiruna Church, recognized as one of Sweden's most popular and beautiful wooden buildings, will be physically moved to the new city center.

To enable the world's largest municipal relocation, Kiruna needed an innovative approach, and city managers established the Kiruna Sustainability Center (KSC) to develop and test new ideas for sustainable solutions. The KSC brings together an ecosystem of municipalities, industry experts, researchers, universities and citizens in an effort to drive greater innovation and new business opportunities.

During the initial phases of the Kiruna relocation, CGI helped city Kiruna to devised an innovative concept called Hidden City that uses Microsoft HoloLens augmented reality in combination with geographic information system (GIS) equipment and data to digitally map and visualize the underground infrastructure. The project is pioneering the outdoor use of HoloLens, which by design, is made to be used indoors. For Kiruna, Hidden City provides an accurate underground image before starting infrastructure repairs [7].

Hidden City was a finalist in the "innovative idea" category of the World Smart City Awards 2018 and finalist for the "best innovator" award at the Kiruna City business awards. Kiruna and CGI also have been featured by Microsoft in its customer story: "Moving a city with the help of Microsoft HoloLens [8].

12.2.2 Transport: Rail

Despite substantial investments in the Betuweroute and the port railway, the volume of goods on the Dutch railways has been virtually stable for about 15 years, while the other hinterland transport modalities for goods (truck & barge) continue to grow (source: CBS). Surprising in times when sustainability is rightly becoming increasingly important. The Ministry of Infrastructure and Water Management in The Netherlands has therefore expressed the ambition that rail freight transport should have doubled by 2030.

For better and more efficient business process management of management, CGI helped ProRail to develop test and introduce innovations including creating a Digital Twin [9, 10]. This digital twin is the basis for the information systems with which we control ProRail's core processes. The world of grid operator ProRail is outside and a lot is happening there. The network performs all kinds of tasks in different places at the same time. Measurements provide digital information that is collected, purified, modeled and combined. A Digital Twin is generated from that information, which is in fact a digital representative of the real world. But it's more than that. The Digital Twin also represents planned/designed and already vanished

objects. It thus covers the entire life cycle of the object structure and associated information management. Moreover, use of the network is part of the 5D world of ProRail. 5D is a combination of 3D location-bound information with a time registration and the level of detail of the product. This in turn serves as the basis for the information systems with which ProRail manages its core processes.

12.2.3 Aerospace and Defense

Aerodynamics of a fighter jet are insanely complex that computer simulations quickly reach their limits. As a result BAE (Brithish Airpospace) is creating 3-D printed models for supersonic wind tunnel tests to refine the shape of the aircraft. The digital twin concept will be used to design test and support every single system and structure for Tempest, which is scheduled to enter into an active service in 2035. Still in the concept phase, the Tempest will be one of the first sixth-generation (6G) fighters and is designed to complement current combat craft. It will have configurable, Artificial Intelligence and cyber-hardened communications that allow the aircraft to act flying command and control center, with the pilot acting more as an executive officer than as a dogfighter. By taking entirely digital approach they also transform the way the organisation works. The BAE systems achieved what traditionally would have taken a number of months in a number of days. As a result they are working faster for the future triggering open mind and innovation [11].

12.3 Risks, Security, Privacy and Ethics

With all discussed above, there's an element of risk involved. Now let's look at the potential risks and challenges a digital twin can expose you to from a security and privacy perspective. The obvious concerns are security, privacy, surveillance and ethics that need to be addressed before these systems are deployed. Wider application of the digital twin concept creates ethical challenges as well as technical ones. Companies usually own the assets they use in their factories. Once you have sold a physical product to a customer, who owns the rights to its digital twin? Concerns over privacy and the potential misuse of data are already widespread in the worlds of e-commerce and social media. Now consumers are raising the same questions about the growing number of connected products in their lives. Consumer rights advocates are already raising questions about the use of connected toys that collect data on the behavior and preferences of their users, for example.

Ethical, privacy and societal implications of Digital Twins are another dimensions which are vital and need attention. So far speculations about the ethical, privacy and legal provisions for regulating the development and usage of a Digital

Twins have been based on the concept of the physical and the digital twin remaining separate entities, as tighe term "twin" itself suggests. Responsibility, ethics, decency, morality will not only experience a renaissance, they will need to be imbued with immense significance, for this is a matter of data and transparency.

12.4 Digital Twins Security Drivers, Concerns and How to Manage

Security and privacy by design approach is becoming a norm in the current complex digital environments. By operationalising security and privacy by design approach, security can become a vital enabler of trust in the operation of products and assets using digital twins. The digital twin can become the full driver of communication and collaboration across the organisation's entire digital thread, in other words it can become a framework to unify and orchestrate data across a product's life cycle. This can happen only if the selected and just right security policies and technologies are applied and maintained to preserve and maintain digital trust. The participants can collaborate and safety operate products, assets and processes though digital twins, solely in an authenticated and trusted ecosystem.

As with any digital security strategy, consistent updating of technologies and policy is critical so the organisation can stay one step ahead of cyber criminals, and securing the multiple endpoints of products, assets and processes will require a complex, multi-layered, distributed approach to security.

For organisations that want to create or improve their digital twin initiatives, projects or progarmmes, and to ensure the success of their digital transformation in general, they can count on the security team. Now the security team has the opportunity to position itself as a business enabler that drives innovation and business outcomes. Thus, the security team can become the guarantor of digital trust, implementing security by design into the digital twin initiatives, but also throughout the organisation's culture, practices, processes and platforms.

The safe inclusion of the whole ecosystem and supply chain into the digital twin will be crucial, as all partners will need to be part of the model for it to function properly. While all stakeholders engagement and collaboration has their own challenges, it is critical that parties need to collaborate effectively to be able to manage the security and privacy risks and be able to succeed.

References

[1] Digital Society School events – Webinar – https://digitalsocietyschool.org/event/digital-twin-a-promising-thing-webinar/ Amsterdam Oct 29, 2020.

[2] M. Grieves, Florida Institute of Technology, Digital Twins Presentation, Conference 2002 University of Michigan, Society of Manufacturing Engineers conference in Troy, Michigan.
[3] Apollo 13 mission report, Manned Spacecraft Center, Sept 1970 https://sma.nasa.gov/SignificantIncidents/assets/apollo-13-mission-report.pdf.
[4] MarketsAndMarkets Report 2019 (176 pages) "Digital Twin Market by Technology, Type (Product, Process, and System), Application (predictive maintenance), Industry (Aerospace & Defense, Automotive & Transportation, Healthcare), and Geography – Global Forecast to 2026".
[5] Gartner, Prepare for the Impact of Digital Twins, 2017 https://www.gartner.com/smarterwithgartner/prepare-for-the-impact-of-digital-twins.
[6] IDC research, "The birth of 'digital twins' will transform our world", https://www.ft.com/content/22158d06-3b5e-11ea-b232-000f4477fbca January 2020.
[7] CGI Case Study – "Using augmented reality and precision data to enable the future smart city" 2017–2018", https://www.cgi.com/en/case-studies/kiruna-sweden-augmented-reality-smart-future-city.
[8] Microsoft, CGI – "Moving a city with the help of Microsoft HoloLens" https://www.youtube.com/watch?v=1wq7ZQMUy-k.
[9] H. Moonen – "Digital Twins zijn cruciaal bouwblok voor toekomst railgoederenvervoer" https://www.cgi.com/sites/default/files/2021-03/clm_ed._5_-_digital_twins_zijn_cruciaal_bouwblok_voor_toekomst_railgoederenvervoer.pdf CLM No1 Vervoer March 2021.
[10] R. Voute – "CGI helpt ProRail bij ontwikkelen, testen en invoeren van innovaties", Grond, Weg, waterbouw, September 3, 2020 https://www.gww-bouw.nl/artikel/cgi-helpt-prorail-bij-ontwikkelen-testen-en-invoeren-van-innovaties/.
[11] GDC – Global Defense Corp – "BEA Systems Developed Digital Twin and 3D Printing Techniques for Tempest fighter" August 21, 2020.

About the Editors

Dr. Gohar Sargsyan is a Director at CGI Inc. with a role of ICT Innovation and Partnership collaboration lead. Last years she was focusing on the Security domain covering Cyber Security, Space Security, Border Security, Digital Policing, Public Safety, Security and Privacy and Security, Security in Digital Twins, Financial Crime and Fraud Detection, Blockchain and Sustainable Finance. She has successful track record in different industries and has extensive experience in Strategic Advisory, stakeholder management and business/IT alignment.

Started from 2002 she is an appointed EU expert for the European Commission (EC) IT Innovation Programmes where she assesses new digital and technological solutions for all industries offered by EU and international consortia. She partners with leading business and science organisations in selected EU programmes for creation, implementation and delivery of complex platforms and services, among them Cyber-Trust, where she was the exploitation and innovation leader. She is a stakeholder member on EU panels and high level think tanks on Future Internet, IPv6, EU Cybersecurity, Open Innovation 2.0, EU Green Deal and Europe Digital among others. She is a founding member of Open Innovation Strategy and Policy Group under support of EC DG CONNECT since 2009 and Innovation Luminary Academy and Awards since 2013.

Dr. Sargsyan holds MS (Master of Science) of Applied Mathematics at Yerevan State University with Master Thesis on Cryptography, MS CIS (Master and Silences Degree of Computer and Information Sciences) and MBA at UCLA, then she also accomplished her PhD on Informatics and Automation Problems Optimization and Mathematical Modeling with a project of Video Conferencing for Poor Internet Bandwidth. She achieved a number of qualification certificates on Security, Data, Business and Sustainability Leadership.

She is an inventor with IP ownership. She is a regular (keynote) speaker and chairwoman of international events and sessions. To name a few, she is a steering committee member of IEEE CSR, IEEE SecSoft, Open Innovation 2.0, World Smart Capital initiative. She publishes regularly in international prominent business and scientific venues, so far being an (co)author and co(editor) of more than 80 publications including books.

Mr. Dimitrios Kavallieros co-supervises the cybersecurity research team of the MultiMoDal Data Fusion and Analytics Group (M4D) of the Multimedia Knowledge and Social Media Analytics Laboratory (MKLab) which is part of the Informatics and Telematics Institute (ITI) of the Center of Research and Technology, Hellas (CERTH, Greece). He is also a senior research associate at M4D focusing on cybersecurity research and innovation technologies.

Prior to this role, he was a programme coordinator and a research associate at the Center for Security Studies – KEMEA of the Hellenic Ministry of Citizen Protection. He holds an MSc in Ethical Hacking and Computer Security from the University of Abertay. He is currently PhD candidate at University of Peloponnese. He has co-edited a book on technologies strengthening security (Technology Development for Security Practitioners) and he has more than 25 articles in conference proceedings, journals and book chapters covering mainly cybersecurity, cyberterrorism and digital forensics topics.

Mr. Kavallieros has participated in several European and National funded research projects assuming different roles such as senior researcher, coordinator, project manager, technical manager, scientific leader and mainly focused on Cybersecurity, Cybercrime and Cyberterrorism, IoT & Cloud Security/Forensics, Digital Forensics and Blockchain technology topics. He is an invited expert to different selected stakeholders' groups and initiatives and a speaker in cybersecurity events to share his experience in all the security programmes, testbeds, results and way ahead towards better and more secure world.

He was the project coordinator of the EU H2020-project Cyber-Trust on IoT cybersecurity and of the EU H2020-project FORESIGHT on cybersecurity preparedness, cyber range and simulation and the innovation manager of the EU H2020-project PROPHETS on human factors and new methods to prevent, investigate and mitigate cybercriminal behaviors. In addition, he was the project manager of several H2020 projects in KEMEA, including but not limited to AIDA, LOCARD and SPARTA. Now in CERTH he is a member of the Scientific and Technical management team of SECANT and CTC projects.

Dr. Nicholas E. Kolokotronis is an Associate Professor and the Head of the Cryptography and Security Group at the Department of Informatics and Telecommunications, University of the Peloponnese. He received his B.Sc. in mathematics from the Aristotle University of Thessaloniki, Greece, in 1995, an M.Sc. in highly efficient algorithms (highest honors) in 1998 and a Ph.D. in cryptography in 2003, both from the National and Kapodistrian University of Athens. Since 2004, he has held visiting positions at the University of Piraeus, University of the Peloponnese, the National and Kapodistrian University of Athens, and the Open University of Cyprus. During 2002–2004, he was with the European Dynamics S.A., Greece, as a security consultant. He has been a member of working groups for the provisioning of professional cyber–security training to large organizations, including the Hellenic Telecommunications and Posts Commission (EETT).

Dr. Kolokotronis has published more than 100 papers in international scientific journals, conferences, and books and has participated in more than 25 EU–funded and national research and innovation projects. In Cyber-Trust, he was the technical coordinator. He has been a co–chair of conferences (IEEE CSR 2021, 2022), workshops (IEEE SecSoft 2019, IEEE CSRIoT 2019, 2020, and ACM EPESec 2020), and special sessions focusing on IoT security. Moreover, he has been a TPC member in many international conferences, incl. IEEE ISIT, IEEE GLOBECOM, IEEE ICC, ARES, and ISC. He is currently a Guest Editor in "Engineering – cyber security, digital forensics and resilience" area of Springer's Applied Sciences Journal (since 2019) and in the Reviewer Board of MDPI's Cryptography journal (since 2020), whereas he has been an Associate Editor of the EURASIP Journal on Wireless Communications and Networking (2009–17) and a regular reviewer for a number of prestigious journals, incl. IEEE TIFS, IEEE TIT, Springer's DCC, etc. His research interests span the broad areas of cryptography, security, and coding theory.

Contributing Authors

G. Bendiab
University of Portsmouth
gueltoum.bendiab@port.ac.uk

V.-G. Bilali
Institute of Communication and Computer Systems (ICCS)
giovana.bilali@iccs.gr

R. Binnendijk
CGI
raymond.binnedijk@cgi.com

S. Brotsis
University of the Peloponnese
brotsis@uop.gr

T. Chantzios
University of the Peloponnese
tchantzios@uop.gr

P. Douris
Centre of Security Studies (KEMEA)
p.douris@kemea-research.gr

O. Gkotsopoulou
Vrije Universiteit Brussel
olga.gkotsopoulou@vub.be

K.-P. Grammatikakis
University of the Peloponnese
kpgram@uop.gr

A. Kardara
Centre of Security Studies (KEMEA)
a.kardara@kemea-research.gr

D. Kavallieros
University of the Peloponnese
d.kavallieros@uop.gr

G. Kokkinis
Centre of Security Studies (KEMEA)
g.kokkinis@kemea-research.gr

N. Kolokotronis
University of the Peloponnese
nkolok@uop.gr

P. Koloveas
University of the Peloponnese
pkoloveas@uop.gr

I. Koufos
University of the Peloponnese
ikoufos@uop.gr

K. Limniotis
Hellenic Data Protection Authority
klimniotis@dpa.gr

C.-M. Mathas
University of the Peloponnese
mathas.ch.m@uop.gr

A. Rajkumari
CGI
a.rajkumari@cgi.com

M. Rantopoulos
OTE group Technology & Operations
mrantopoul@cosmote.gr

J. Rose
University of Portsmouth
joseph.rose@port.ac.uk

A. Salis
Centre of Security Studies (KEMEA)
a.salis@kemea-research.gr

G. Sargsyan
CGI
gohar.sargsyan@cgi.com

E. Sfakianakis
OTE group Technology & Operations
esfak@oteresearch.gr

S. Shiaeles
University of Portsmouth
sshiaeles@ieee.org

S. Skiadopoulos
University of the Peloponnese
spiros@uop.gr

M. Swann
University of Portsmouth
matthew.swann@port.ac.uk

C. Tryfonopoulos
University of the Peloponnese
trifon@uop.gr

C. Vassilakis
University of the Peloponnese
costas@uop.gr

C. Wallace
CGI
craig.wallace@cgi.com