Proposal No. 830929
Call H2020-SU-ICT-03-2018

Project start: 1 February 2019
Project duration: 42 months

Cyber Security for Europe

D3.8

Framework and Toolset for Conformity

Document Identification	
Due date	29 January 2020
Submission date	28 January 2020
Revision	1.0

Related WP	WP3	Dissemination Level	CO
Lead Participant	CYBER	Lead Author	Liina Kamm (CYBER)
Contributing Beneficiaries	CYBER, UMU, BRNO	Related Deliverables	

Abstract: This deliverable describes the demo application that we have set up. The application is meant for gathering and managing information about assets that require cybersecurity and IT security certification. It is possible to define the assets and the different certification needs and events that have been carried out or are planned in the future. This allows us to check the conformity of assets to certificates and take a step towards continuous certification.

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

Executive Summary

The ROCA case[1] became news in 2017. This case involved a serious vulnerability in the eID cards of Slovakia and Estonia, who had to revoke 300 000 and 760 000 certificates, respectively. This vulnerability was found in cards that were certified. Whether this was an issue of an uncertified submodule or a problem of poor recertification, we find that a systemized framework could help prevent incidents like this one in the future.

The aim of task 3.8 *Conformity, Validation and Certification* is to support all other CyberSec4Europe project tasks by analysing technologies, system designs and implementations to determine whether the combination of cybersecurity technologies in use achieves the desired security goals, allowing to compare different systems. This will be supported by the demonstrator that contains the toolset for conformity. This demonstrator deliverable describes a prototype for this toolset.

[1] https://crocs.fi.muni.cz/public/papers/rsa_ccs17

Document information

Contributors

Name	Partner
Liina Kamm	CYBER
Sara Nieves Matheu García	UMU
Petr Svenda	BRNO
Dan Bogdanov	CYBER

Reviewers

Name	Partner
Antonio Skarmeta	UMU (high level review)
Vaclav Matyas	BRNO

History

Version	Date	Authors	Comment
0.01	2020-02-20	Liina Kamm, Sara Nieves Matheu García	v0.1 of the deliverable
0.02	2020-02-27	Liina Kamm, Sara Nieves Matheu García, Dan Bogdanov	v0.2 modifications based on internal reviewer comments
1.0	2020-02-28	Liina Kamm	v1.0 second round of review comments and new deliverable template

Table of Contents

List of Figures

List of Tables

List of Acronyms

C	**CC**	Common Criteria
	CVSS	Common vulnerability score system
E	**EAL**	Evaluation Assurance Level
	ETSI	European Telecommunications Standards Institute
I	**IoT**	Internet of Things
	ISO	International Organization for Standardization
N	**NVD**	NIST National Vulnerability Database
O	**OCL**	Object constraint language
T	**TOE**	Target of evaluation
	TTCN	Testing and test control notation
U	**UML**	Unified modelling language

1 Introduction

1.1 Motivation

Conformity to established standards and best practices is essential for increasing the protection baseline in cybersecurity. Many organisations lack personnel experienced in the domain and, therefore, have a hard time adopting new approaches and techniques. Education is an important component, but in-depth knowledge is hard to transfer. Thus, certification methodologies that distil certain best practices into structure, easy-to-apply guidelines have an important role in the proliferation of cybersecurity innovation.

However, the compacted nature of certification may also bring its downsides. For example, the ROCA case[2] in 2017 involved a serious vulnerability in the national eID cards of Estonia and eID cards of Slovakia, who had to revoke 760 000 and 300 000 certificates, respectively. This vulnerability was found in cards where the chips were certified according to the well-established Common Criteria methodology with an assurance level mandated by European regulation.

While it is currently unclear, what caused the mistake, we see that development in the certification domain is needed for multiple reasons. Firstly, while Common Criteria is flexible, it does not have protection profiles or security targets for everything. The expectation in Common Criteria use is that, once the innovation reaches maturity, the customers and technology vendors assemble to come up with the common points of reference for certifying.

However, this is a limitation for new technologies that may not find adoption due to the lack of certification. This is especially the case for quickly evolving technologies like IoT (Internet of Things). It is not the intention to sidestep due process and reduce security requirements to technologies. Instead, we need to consider new methodologies that are contain considerations for new techniques.

1.2 Our Approach

Inspired by this, in task 3.8 *Conformity, Validation and Certification* we set out to identify frameworks that allow us to describe and compare the security properties of new technologies in the domain of IoT. We have collaborated with the CyberSec4Europe project task 7.3 *Certification – Methodologies, Tools and Infrastructures* and work package 8 *Standardisation* to gather information about what they expect from the toolset. We will continue this collaboration and consider their input.

We have identified the ARMOUR methodology for IoT devices as a suitable approach. It allows us to support other CyberSec4Europe tasks by analysing technologies, system designs and implementations to determine whether the combination of cybersecurity technologies in use achieves the desired security goals, allowing to compare different systems. We give a short description of the methodology in Section 2.

In addition to finding a suitable methodology, we also tested the technology on a hypothetical case of analysing the security of an ID-card based digital signature device. We also saw that the complexity of

[2] https://crocs.fi.muni.cz/public/papers/rsa_ccs17

systematic study can be simplified by technical tools. In this case, we have modified an existing asset from CYBER (the CSA tool) to serve as a prototype for this methodology. In Section 3, we show how the CSA tool can be used to automate and simplify the use of ARMOUR methodology, speeding up its use. Indeed, in Section 3, we showcase our modified CSA tool and its use to apply ARMOUR. Currently the demonstrator shows a prototype of the planned system, allowing the user to define and interlink information in different categories. The demonstrator does not currently offer further automation of processes (e.g. risk score estimation). This will be part of our future efforts, which we describe in more detail in Section 4.

This page has been intentionally left blank.

2 ARMOUR Methodology

The ARMOUR methodology for security certification of IoT devices [1] is based on the ETSI proposal described in [2], which combines an extended security assessment derived from ISO 31000 and typical security testing activities following the standard ISO 29119. This methodology was initially developed and evaluated in the RASEN[3] research project. The ETSI methodology describes two different perspectives to combine risk assessment with security testing: a test-based risk assessment perspective, in which risk assessment is improved by additional input from testing to calculate the associated risk, and a risk-based testing perspective, in which risk assessment is used to improve the security testing process.

The ARMOUR methodology combines the two perspectives of the ETSI proposal and adds additional activities inherent to the certification process, such as labelling (included in the Communication and Consult phase, Figure 1). Labelling has also been considered by regulatory and security organizations and it is explicitly mentioned in the Cybersecurity Act [3] [4]. The label should visually represent the security level obtained in a way a non-expert consumer could understand it.

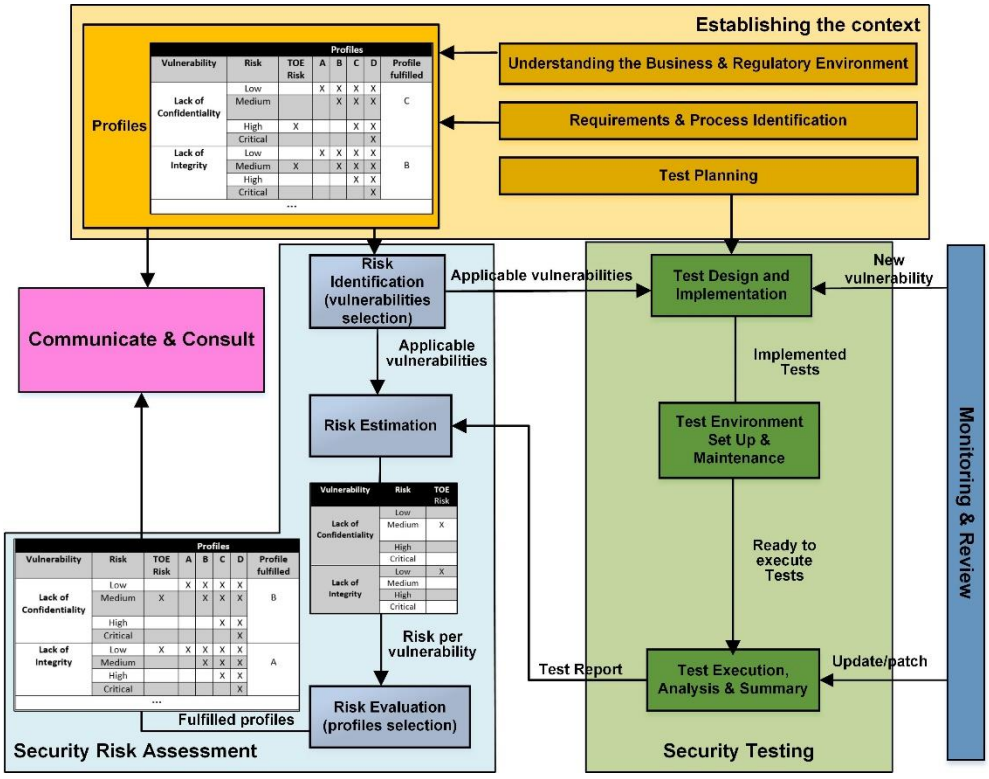

Figure 1: ARMOUR methodology

[3] http://www.rasenproject.eu

Although ARMOUR proposes a specific instantiation of the methodology, different tools and techniques could be used to perform the certification process [5]–[8].

Before describing the process, it should be noted that the certification approach takes a set of vulnerabilities applicable to a certain Target of Evaluation (TOE) as a starting point. In particular, NVD (the NIST National Vulnerability Database) is used to find vulnerabilities previously discovered in the TOE. Furthermore, the set of generic oneM2M [9] vulnerabilities is used for testing purposes, in order to discover zero-day vulnerabilities for the TOE. Then, the resulting set of vulnerabilities is mapped to five general security properties, which have been extracted from some of the most referenced security aspects that can be found in current IoT literature[10]–[12].

- Lack of authentication. The endpoints should be legitimate.
- Lack of integrity. Received data are not tampered with during transmission; if this is not met, any change can be detected.
- Lack of confidentiality. Transmitted data can be read only by the communication endpoints.
- Lack of authorization. Endpoint services should be accessible to endpoints who have the right to access them.
- Lack of availability. Exceptions should be controlled to avoid faults that affect the endpoints.

It should be noted that the proposed mapping provides a more simplified view of the security aspects, allowing to measure the risk associated with the lack of each property. In addition, the resulting security label can be specified by using such properties to provide a multidimensional security description.

It should be pointed out that the ARMOUR methodology adapts the ETSI concepts and processes. In this sense, the first process, **establishing the context,** is related with understanding the business, regulatory environment, and the laws and analysing which security level is required in each of them. For example, in a medical context, confidentiality and availability could be considered two very important security properties that could not be so important in a domestic context. The set of regulations (including GDPR for data) and stakeholder's requirements are processed to create sets (for different levels) of baseline requirements. As a result, different profiles are created for each context, describing the risk tolerated for each general vulnerability in order to obtain it. Table 1 shows an example of security profiles for a specific domain. Here, if the TOE has a critical risk in lack of confidentiality, it will be only able to obtain the profile D. It should be noted that the profiles are incremental, meaning that if the TOE fulfils the profile A, it also fulfils B, C and D.

Vulnerability	Risk	A	B	C	D
Lack of confidentiality	Low	x	x	x	x
	Medium		x	x	x
	High			x	x
	Critical			x	x
Lack of integrity	Low	x	x	x	x
	Medium		x	x	x
	High			x	x

	Critical				x
…					

Table 1: Example of Profiles

The last activity of the first process, the test planning, is the activity of developing the test plan (objective, scope, order, testing technique etc.) to assess the TOE. In this activity, the techniques of testing are chosen regarding each vulnerability, as well as the order of the tests and their scope.

At the beginning of the security assessment phase, the **risk identification** activity identifies the potential vulnerabilities that can be applicable to the scenario and context are identified. The rest of the general vulnerabilities will be labelled by default with a low risk if the vulnerability cannot be exploited or with critical risk if the TOE does not have protection against it. Here, a simpler security testing could help to identify critical areas in which risk identification should deepen, for example using vulnerability scanners. Finally, the selected vulnerabilities are used as input for the testing process, to define the test cases.

Test Design and Implementation generates the test cases associated to the vulnerabilities and threats considered. In this phase, the tests are also implemented and assembled to test procedures.

In the instantiation proposed in the ARMOUR project, model based testing (MBT) is used to automate this process. Nevertheless, other techniques and tools can be used to implement the tests. MBT has shown its benefits and usefulness for systematic compliance testing of systems [13]. In this approach, the structure of the system is modelled by unified modelling language (UML) class diagrams, while the system behaviour is expressed in object constraint language (OCL)[4], using the CertifyIt tool [14]. Functional tests are obtained by applying a structural coverage of the OCL code describing the operations of the TOE. The tests are exported in testing and test control notation (TTCN) v.3 language or JUnit using the tool CertifyIT. To cope with the particularities of each IoT device, CertifyIt also generates a set of interfaces called adapters that must be implemented to link the high level tests with the code of the real device.

Test Environment Set Up & Maintenance involves establishing and maintaining the environment in which tests are executed. The environment can be local (e.g., a device) or remote (e.g., a large-scale infrastructure such as FIT IoT Lab), but in any case, typical actions are reserving resources and uploading the code to the devices.

Test Execution, Analysis & Summary deals with the test execution as well as with the systematic analysis and summary of test results. The tests can be executed on a local or external large-scale testbed such as FIT IoT Lab, and tools to automate the execution of the testing process can be used, such as TITAN or JUnit. Here, a previous risk assessment process can help to prioritize the execution of the tests, leading with the time and likelihood of discovering new vulnerabilities.

In the instantiation proposed in ARMOUR project, the high level test commands are relayed by the adapter to the device so that it can execute them. With the automation of this process, if a new vulnerability is discovered, the recertification process can be done in a cheap, fast and easy way, which is key to addressing

[4] http://www.omg.org/spec/OCL/2.4

the dynamic nature of cybersecurity in IoT. The results of the tests help to establish the security level (cybersecurity label) in a more refined way, since they are used as input in the risk estimation activity to measure the risk and decide later if the risk is tolerable or not.

Risk estimation calculates the risk level, understanding the origin of the risk and its consequences. Although this is usually a hard activity, due to the subjectivity and imprecision of the information, this methodology tries to solve this issue by providing additional input from the security testing process. This activity uses as input the results from the security-testing phase (e.g. test pass or fail or more granular information such as the ciphersuite) to provide objective metrics to perform the risk estimation.

The instantiation proposed in ARMOUR uses the common vulnerability score system (CVSS) [15] to calculate the risk associated to each general vulnerability. CVSS is well defined, its metrics comprises the majority of the metrics of the other risk assessment methods and it is widely used in several vulnerability databases. CVSS consists of three metric groups, the base metrics produce a score from 0.0 to 1.0, modified by the optional temporal and environmental metrics. Each metric in this group is assigned a value, and these values are converted to associated weights, and applied to a formula in order to calculate the base subscore. The test results are mapped into CVSS metrics to obtain the risk in an objective and empirical way (e.g., the percentage of non-ciphered data is assigned to the attack vector metric, and the complexity of the attack is based on the ciphersuite used, algorithm and key length). After that, the numerical CVSS values are mapped to risk intervals (low, medium, high and critical) in order to compare the risk obtained in the next activity.

Risk Evaluation compares the results of risk estimation with the level of risk analysed at the beginning of the process, at the establishing the context phase. In this sense, the evaluator can decide if the risk level is tolerable for the domain being considered, or there should be changes in the system definition. In case it is not tolerable, the process can be repeated.

The profile is determined by comparing the results obtained in the risk assessment with the profiles available for the specific context, choosing always the highest profile fulfilled for each vulnerability. For example, in Table 2 a TOE has obtained a Medium risk level in lack of confidentiality, which allows it to obtain B, C and D profiles. However, it will obtain the highest one, in this case the B profile. This process is repeated for all the vulnerabilities.

		CVSS	Profiles				
Vulnerability	**Risk**	**TOE**	**A**	**B**	**C**	**D**	**Profile fulfilled**
Lack of confidentiality	Low		x	x	x	x	B
	Medium	x		x	x	x	
	High				x	x	
	Critical					x	
Lack of integrity	Low	x	x	x	x	x	A
	Medium			x	x	x	
	High				x	x	

	Critical					x	
Lack of authentication	Low	x	x	x	x	x	A
	Medium			x	x	x	
	High				x	x	
	Critical					x	
…							

Table 2: Evaluation of a TOE

As an output of the general certification process, a cybersecurity label associated to the risk of the scenario tested is obtained. The process of labelling is included in the communicate and consult process. It should be noted that labelling has to take into account the context of the scenario that is being tested and the certification execution. For this reason, and based on Common Criteria approach for trying to homogenize the terms, three mains aspects are considered to be included in the cybersecurity label.

- Target of evaluation (TOE): In CC, a TOE is defined as a set of software, firmware and/or hardware possibly accompanied by guidance. In this case, the TOE also includes the protocol tested (and library), the configuration (e.g., key length, hash function) and the context where it has been tested. For example, if the IoT device must be used in an industrial context or a consumer market context (e.g., a smart house).
- Profiles (level of protection): A, B, C and D. The level of protection is related to the risk associated to the tested scenario.
- Certification execution: The proposed certification execution follows the same levels of Evaluation Assurance Levels (EALs) as Common Criteria.

In the cybersecurity labels, there is a tradeoff between the simplicity of understanding by a non-expert consumer and the information presented. Following the recommendations of ENISA [16], as security requirements are in fact multi-dimensional, the result of the evaluation needs to be communicated appropriately to the user. For this reason, the cybersecurity label includes the profile of each general vulnerability in a visual way, using a spider chart (pentagon) like in [17], where the vertices are the five general vulnerabilities and the internal lines, the profiles. At the same time, the visual concept of more area more risk and the usage of degraded colours (red and green) helps a non-expert consumer to understand the cybersecurity label.

Finally, as security is a dynamic concept, the usage of a digital QR as cybersecurity label is proposed to be updated in case of a new vulnerability is discovered in the product. In this sense, the label (and certificate) is valid unless a recertification is needed or the conditions defined in the security certification process are still valid. For example, a cryptographic algorithm specified in the security profile may become obsolete. Therefore, the communication to the user could be instantaneous through the monitoring process, where the state of the IoT device is periodically or continuously assessed from a security point of view to initiate the recertification process. The proposed design for the cybersecurity label is shown in Figure 2.

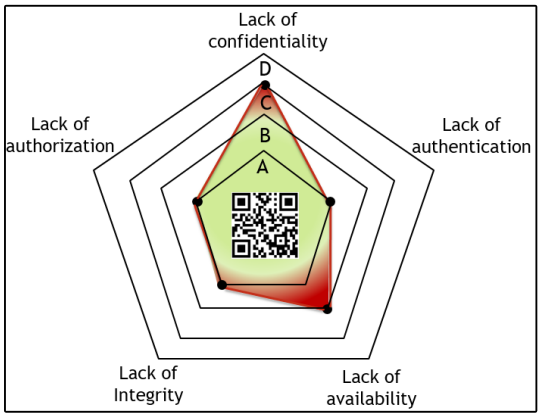

Figure 2: Example of security label

Figure 1 shows additional support activities like **communicate and consult** and **monitoring and review**. Although monitoring is not addressed in ARMOUR, this activity is intended to continuously control and react to the changes on the device security. If there is a security change or a new vulnerability is detected, it can update or re-execute the tests. On the other hand, communicate and consult, apart from the labelling, is meant to gather information from inside (e.g., risk, context) and outside (e.g., experts, databases, laws) the process, and to communicate it in an appropriated way.

3 Demonstrator Structure and User Interface

In this section we describe the demonstrator tool. First we discuss the goal and scope of the demonstrator, then we explore the components of the demonstrator and include screenshots to illustrate how the tool handles these components, and finally, we talk about the technical system parameters.

3.1 Demonstrator Goal and Scope

As is the case with most general frameworks, the ARMOUR methodology can be very difficult to get started with, let alone apply for a person acquainting themselves with it for the first time. Moreover, a complex systematic study can be simplified by technical tools even for those who are familiar with the methodology.

We have modified an existing asset from CYBER (the CSA tool) to serve as a prototype for this methodology. The CSA tool is originally intended for incident management, but has been altered so that a user can enter and interlink information concerning a target of evaluation. At the current stage, the demonstrator is a prototype of the planned system, and as such does not offer further automation of processes (e.g. risk score estimation, automatic scheduling of tests for recertification). This will be part of our future efforts.

The demonstrator is divided into three main categories concerning assets, risks and tests. The user can add information about all of these categories and interlink them based on their connections. For an object in any category it is possible to define a level of importance, which will help with prioritising. All these categories are further described in Subsection 3.2.

3.2 Demonstrator Objects

3.2.1 Assets

The first thing a user does during the assessment process is define the target of evaluation (TOE) and the assets that are connected to it. Assets can be hardware, software, data or employees of an organisation. The TOE itself is an asset and can contain other assets as subcomponents. The attributes of an assets can be seen on Figure 3. In our example, we describe the eID card as the TOE. A key generation algorithm can be added as a subcomponent asset of the TOE.

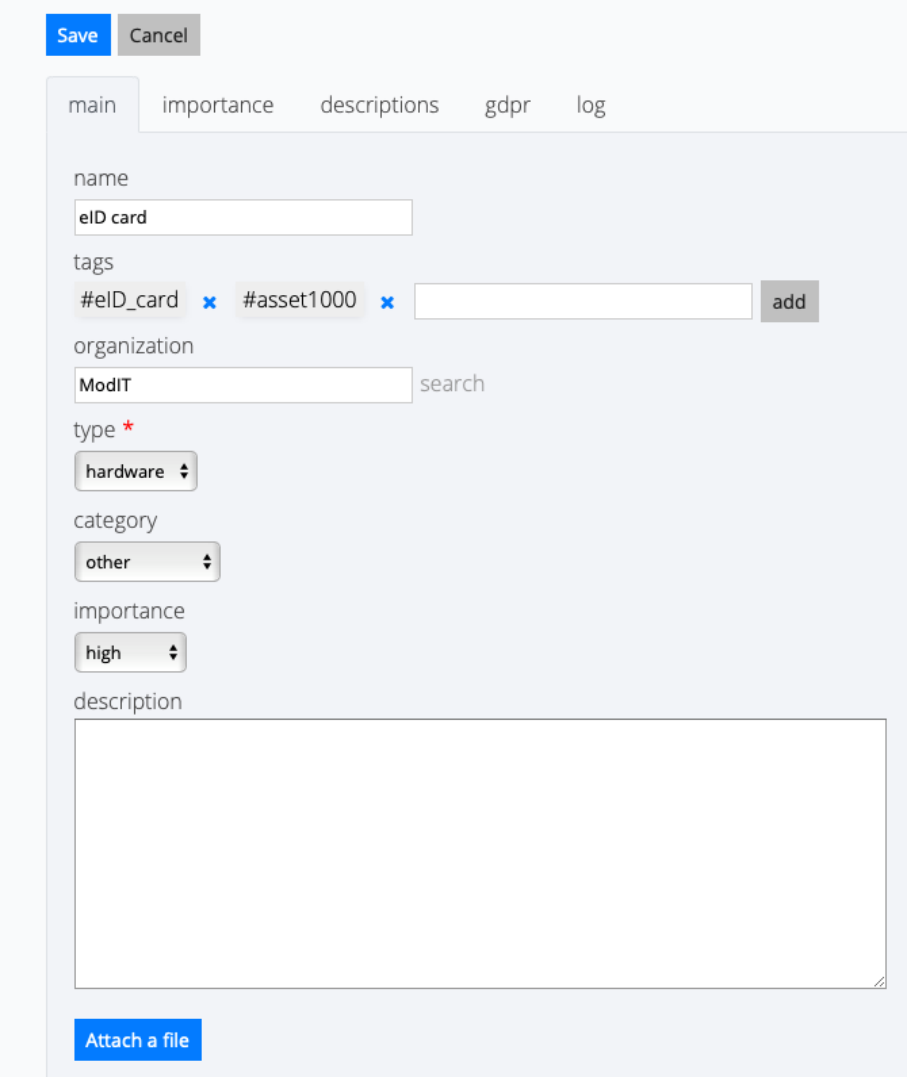

Figure 3: Asset attribute definition

3.2.2 Risks

Risks include vulnerabilities, threats and scenarios. The ARMOUR methodology classification of threats into the five general vulnerabilities (lack of authorization, lack of confidentiality, lack of authentication, lack of integrity, lack of availability) can be added to each individual threat that is discovered. Figure 4 shows vulnerability view of the demonstrator.

Figure 5 shows the list of threats and Figure 6 the list of scenarios. Specific threats are classified into the five vulnerabilities and can be added to asset descriptions and tests. Scenarios enable the user to attach vulnerabilities and tests to a TOE and define their risk level. This will, in future work, be used to issue a security label.

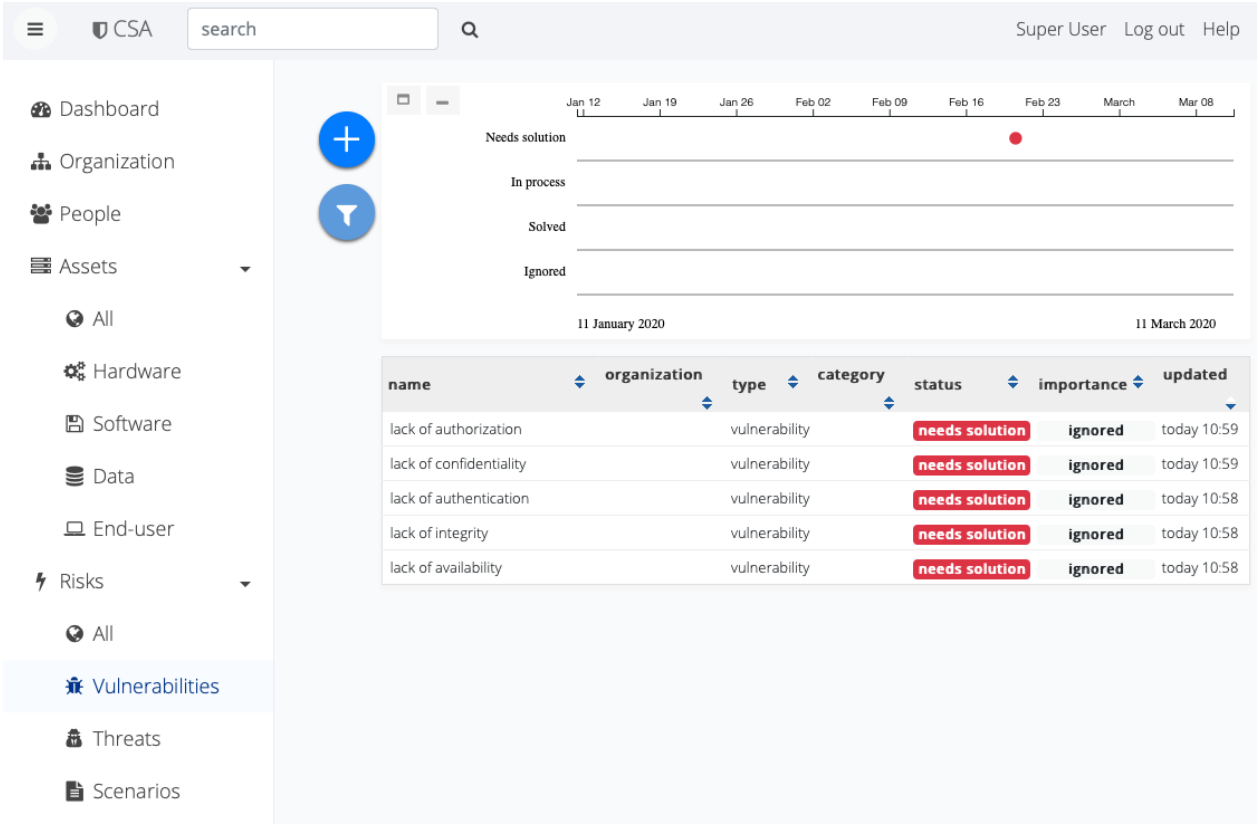

Figure 4: Vulnerabilities

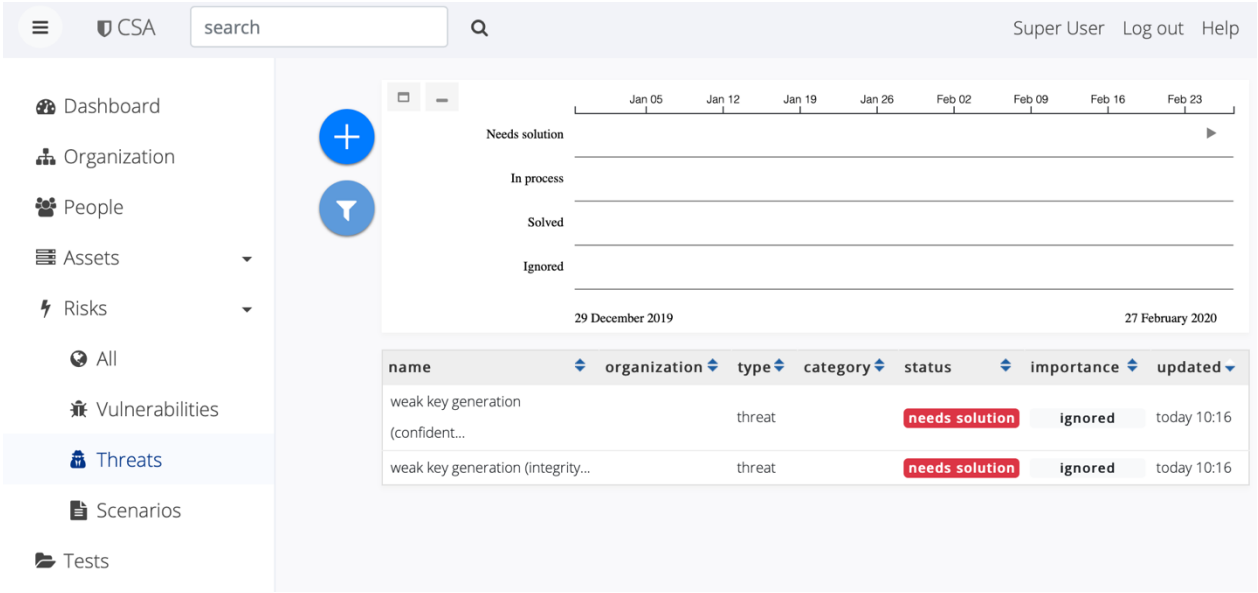

Figure 5: Threats

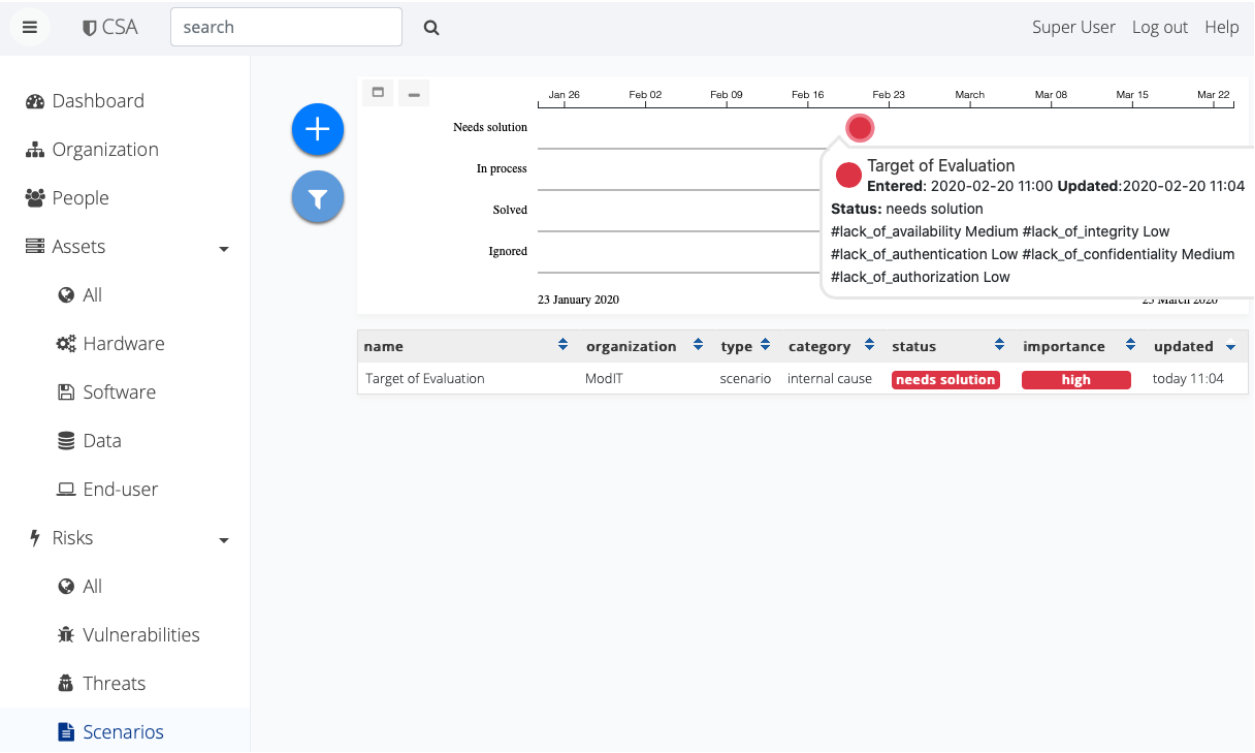

Figure 6: Scenarios

3.2.3 Tests

Tests (Figures 7, 8) contain the different levels of testing that can be done to verify the conformity of different parts of an asset. A test can be connected to one or more assets. If a test is connected to an asset that is a subasset or part of a TOE, the test will automatically be connected to those as well. A test has dates connected to it (start and end). Using this their execution can be logged. These dates allow the tests to also be scheduled for the future. When an asset is modified and a test is connected to it, the test must be run again. This scheduling will be done automatically in future work.

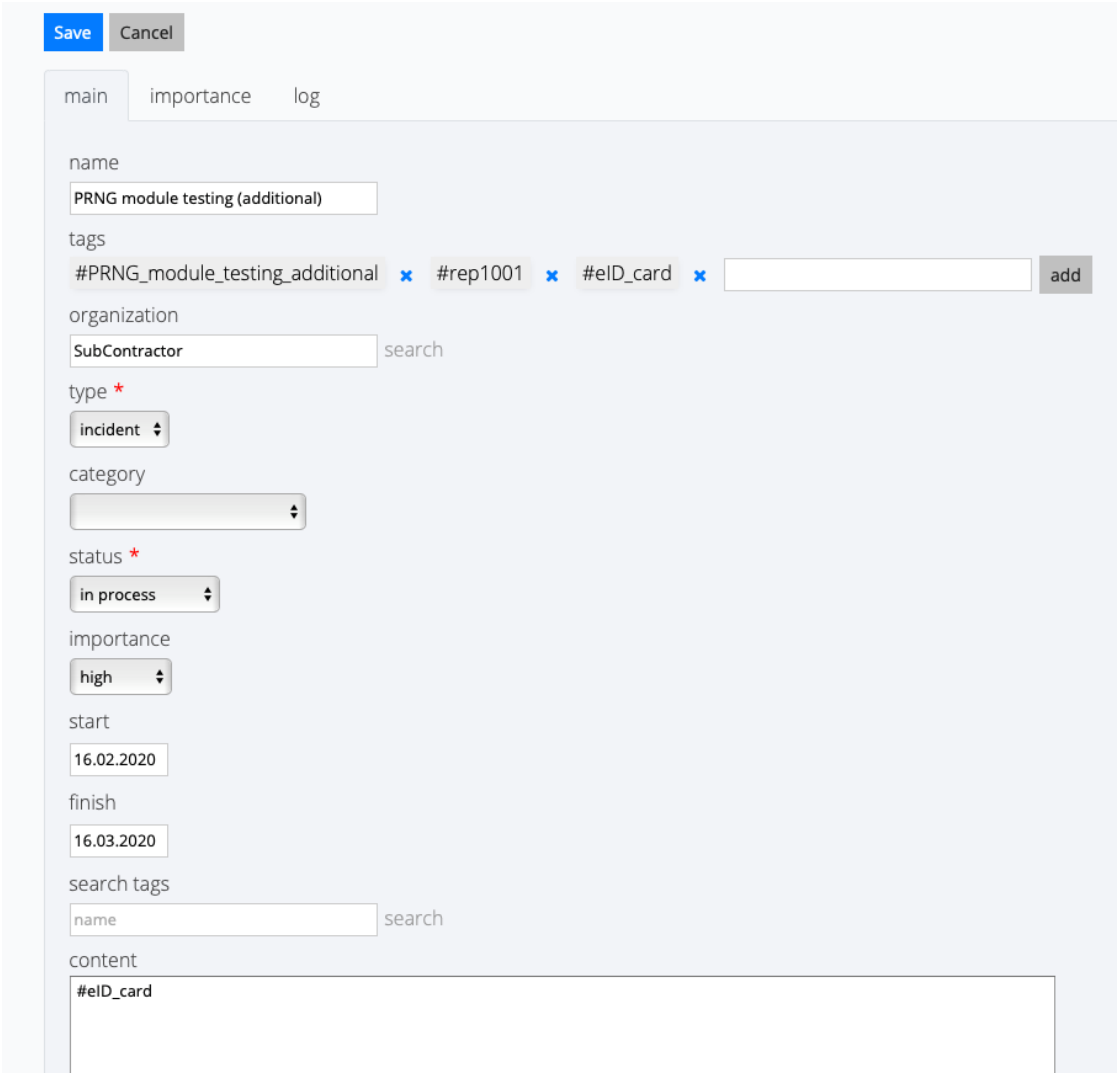

Figure 7: Test view

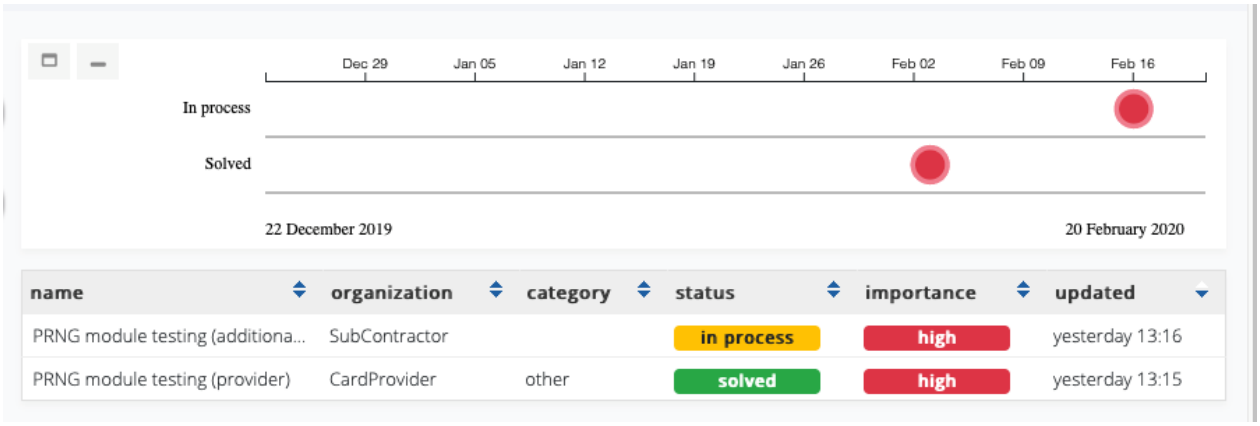

Figure 8: Tests dashboard

3.3 System Parameters

The system parameters are the following:

- The demonstrator backend is implemented in the Python programming language;
- The frontend is implemented using React, a JavaScript library for user interfaces;
- The database is implemented using MySQL.

4 Further Work

For the M36 deliverable *Validation and Certification Methodology*, we will develop an updated version of the framework for validation based in addition to the ARMOUR project methodology also on the meta-schema for certification by ECSO and the NIST CPS.

As mentioned, we have collaborated with task 7.3 *Certification – Methodologies, Tools and Infrastructures* and work package 8 *Standardisation* to gather information about what they expect from the toolset. We will continue this collaboration and consider their input.

Currently the demonstrator is a prototype. It allows the user to enter and interlink data of different categories. During the project we aim to add features that enable the automation of processes. For example, for recertification, users would simply have to declare that a component has been modified and the relevant defined tests will automatically be scheduled for them. We will also add the generation of a security label to the system.

5 References

[1] ARMOUR project, *D1.1: ARMOUR Experiments and Requirements*. 2016.

[2] ETSI, «ETSI EG 203 251: Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies». 2015.

[3] Euopean Parliament, «REGULATION (EU) 2019/881 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification (Cybersecurity Act)». 2019.

[4] ECSO, «A Meta-Scheme Approach v1.0». 2017.

[5] S. N. Matheu, J. L. Hernandez-Ramos, A. F. Skarmeta, «Toward a Cybersecurity Certification Framework for the Internet of Things», *IEEE Security Privacy*, vol. 17, n.º 3, pp. 66-76, 2019, doi: 10/gf256z.

[6] S. N. Matheu, S. Perez, Hernandez-Ramos, A. F. Skarmeta, «On the automation of security testing for IoT constrained scenarios», *20th World Conference on Information Security Applications (WISA)*, Jeju, Korea, 2019.

[7] S. N. Matheu-Garcia, J. L. Hernandez-Ramos, A. F. Skarmeta, «Test-based risk assessment and security certification proposal for the Internet of Things», *IEEE 4th World Forum on Internet of Things (WF-IoT)*, Singapore, 2018, pp. 641-646, doi: 10.1109/WF-IoT.2018.8355193.

[8] S. N. Matheu-Garcia, J. L. Hernandez-Ramos, A. F. Skarmeta, G. Baldini, «Risk-based automated assessment and testing for the cybersecurity certification and labelling of IoT devices», *Computer Standards & Interfaces*, vol. 62, pp. 64-83, 2019, doi: 10.1016/j.csi.2018.08.003.

[9] oneM2M, «Technical report TR-0008», 2018. Available online: http://www.onem2m.org/images/files/deliverables/Release2A/TR-0008-Security-v_2_0_1.pdf. [Accessed: 26-02-2020].

[10] K. Moore, R. Barnes, H. Tschofenig, «Best Current Practices for Securing Internet of Things (IoT) Devices». 2016.

[11] M. Abomhara, G. M. Koien, «Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks», *Journal of Cyber Security and Mobility*, vol. 4, n.º 1, pp. 65-88, 2015, doi: 10.13052/jcsm2245-1439.414.

[12] F. A. Alaba, M. Othman, I. A. T. Hashem, F. Alotaibi, «Internet of Things security: A survey», *Journal of Network and Computer Applications*, vol. 88, pp. 10-28, 2017, doi: https://doi.org/10.1016/j.jnca.2017.04.002.

[13] G. Bernabeu, E. Jaffuel, B. Legeard, F. Peureux, «MBT for global platform compliance testing: Experience report and lessons learned», *25th IEEE International Symposium on Software Reliability Engineering Workshops*, Naples, Italy, 2014, doi: 10.1109/ISSREW.2014.91.

[14] F. Bouquet, C. Grandpierre, B. Legeard, F. Peureux, N. Vacelet, M. Utting, «A subset of precise UML for model-based testing», *Proceedings of the 3rd International Workshop on Advances in Model-Based Testing - A-MOST '07*, London, United Kingdom, 2007, pp. 95-104, doi: 10.1145/1291535.1291545.

[15] FIRST, «Common Vulnerability Score System (CVSS) v3.1», 2015. Available online: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf. [Accessed: 26-02-2020]

[16] ENISA, *On the security, privacy and usability of online seals. An overview*. 2013.

[17] H. Baars, R. Lassche, R. Massink, H. Pille, «Smart grid security certification in Europe. Challenges and recommendations». ENISA, 2014.