TEST-BASED RISK ASSESSMENT AND SECURITY CERTIFICATION PROPOSAL FOR THE INTERNET OF THINGS

ABSTRACT

This work provides a design of a certification methodology for IoT, paying attention to the test-based risk assessment phase to empower testers with the ability to assess security solutions for large-scale IoT deployments. The resulting approach is an instantiation of the Risk-based Security Assessment presented by ETSI based on the ISO 31000, and it is built on top of different technologies and approaches for security testing and risk assessment adapted to the IoT landscape. The proposed methodology is intended to be used for the different experiments that are proposed in the scope of the ARMOUR project for assessing the fulfilment of several security aspects. It is expected to be used as a baseline to build a new security certification and labelling approach for IoT devices.

Index Terms— Security Certification, Security Risk Assessment, CWSS, Common Criteria, IoT, Security Testing;

1. INTRODUCTION

Nowadays, security aspects represent one of the most significant barriers for the adoption of large-scale Internet of Things (IoT) deployments [1]. Almost every day we can see in the news something related with cyber security and attacks. One of the more named attacks was the Mirai IoT botnet, where several devices were used to perform a DDoS attack against big platforms such as Amazon or Spotify. The vast majority of these devices were IoT devices. In this sense, manufacturers of IoT devices are working together with standardization bodies, to build the next generation of more secure and standardized smart objects, but certification of security aspects remains as an open issue. Security threats are increasing due the ubiquitous nature of the next digital era, transforming these aspects into a major concern for companies, governments and regulatory bodies. A suitable security certification scheme [2] would help to assess and compare different security technologies, in order to provide a more harmonized IoT security view to be leveraged by end consumers. The term certification is used as described in the NIST definition (3), "a comprehensive assessment of the management, operational, and technical security controls in an information system, made in support of security accreditation, to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system". As a result of this process, it is expected a cybersecurity label (labelling process), which contains information that represents or designates the value of one or more security relevant-attributes (NIST definition).

Indeed, the European Cyber Security Organisation Working Group 1 (ECSO WG1) [3] is working on standardisation, certification, labelling and supply chain management, developing a roadmap for the development of security standards and certification. The European Union Agency for Network and Information Security (ENISA) [2] also discusses the main challenges regarding security and privacy of online seals and proposes solutions, such as a label or icon showing the different dimensions of security and verified automatically.

However, a proper risk assessment and certification methodology for security in IoT must overcome different obstacles that are inherent to this paradigm. On the one hand, the high degree of diversity and heterogeneity of devices and products is in conflicting with the need for objective comparisons regarding security aspects. On the other hand, due to the dynamism of typical IoT environments, the certification methodology must take into account these changing conditions, in which the product will be operating. Therefore, agile self-assessment schemes and test automation environments will need to be created and evolved to ensure products have minimum security level appropriate for a context where they are used [4]. Towards this end, a clear identification of threats and vulnerabilities is key to guarantee the success of the approach. In addition, the methodology must cope with the business requirements and needs from the IoT market. It means that security certification approaches should be efficient and cheap, so the product launch in the market is not delayed. Another challenge is how to communicate the result in a way that is understood by the user [2].

To cope with these challenges, this work presents a certification methodology for IoT security that is based on two building blocks, risk assessment and testing, whose last objective is labelling the device's security within a specific protocol and context. This work proposes an instantiation of the risk assessment and testing methodology proposed by the European Telecommunications Standards Institute (ETSI) in [5], focusing on the risk assessment. This proposal is part of the methodology being implemented and developed in the European ARMOUR project[1], whose objective is to automate the security evaluation, in particular the testing, and therefore, to

This work was done in the context of the ARMOUR project Grant id 688237 of the Horizon 2020 Call ICT-12- 2015 Integrating experiments and facilities in FIRE+

[1] http://www.armour-project.eu

make the certification process in IoT faster and easier.

2. RELATED WORK

As this paper is focused on the general certification process and in the security risk assessment part in particular, we present in this section, the current literature related to certification and risk assessment, as well as the efforts to set a common line to create a security certification scheme.

2.1. Security Certification

There is a high number of efforts to establish a general basis for security certification and labelling. For example, DIGITALEUROPE, that represents the digital technology industry in Europe, has published some recommendations for Cybersecurity Certification and Labelling Schemes [4], such as a dynamic label, self certification, global support, test automation and low cost process. ECSO [3] has also done a wide state of the art focusing on standards that can be (potentially) used as the basis for assessing the overall cybersecurity of a product or component, an ICT service, a service provider, organization or a critical infrastructure.

The current main security certification standard is the Common Criteria (CC) [6], where the security functional and assurance requirements are specified through Protection Profiles (PPs) for a Target of Evaluation (TOE), which is a set of software, firmware and/or hardware. It uses Evaluation Assurance Levels (EAL) to describe numerically the depth and rigour of an evaluation. CC describes the set of general actions the evaluator has to carry out, but it does not specify procedures to be followed for those actions [3]. In addition, it does not include risk assessment on evaluation results so the final decision in the certification is more binary (i.e. it fulfils the profile or not). Other important schemes are the Commercial Product Assurance (CPA) [7] aiming to evaluate commercial off-the-shelf products, and their developers, the Cybersecurity Assurance Program (UL CAP), which uses the UL 2900 standards [8] and the Certification de Sécurité de Premier Niveau (CSPN), where the security of products is evaluated mainly by means of limited-time black box testing. Similar to CC, it claims to be an alternative less time consuming.

Despite some disadvantages, CC is the main security certification standard, widely recognized and developed, so we base some parts of our certification mechanism for IoT on it, such as the concept of EAL and TOE.

2.2. Risk assessment

As part of this certification process, being able to measure the risk of different IoT security approaches is crucial to quantify their security level, since it allows to compare different configurations and scenarios. There are a high number of risk assessment methods managed by both commercial and non-commercial organizations. Some of them are briefly described below.

The DREAD scheme [9] evaluates threats based on five metrics. However, it is subjective and different people assign different numbers to each of them, obtaining different risks. The OWASP Application Security Verification Standard (ASVS) Project[2] provides a basis for testing web application technical security controls.

Other approaches such as Microsoft's STRIDE [10] model, group threats into categories, whereas the Cenzic HARM [11] (Hailstorm Application Risk Metric) Score does so with the metric. There also exist approaches to scoring the C Secure Coding Rules, as CERT/SEI [12] that uses the FMECA metric, an ISO standard, and proposals that are too large and complex, such as OCTAVE [13].

The Common Vulnerability Scoring System (CVSS) [14] consists of three metric groups that contains multiple metrics, like the Common Weakness Scoring System (CWSS) [15]. Each metric in the base finding metric group is assigned a value and these values are converted to associated weights, and applied to a formula in order to calculate the base finding subscore, that can range between 0 and 100. CWSS users can also use their own quantified methods to derive a subscore. Conceptually, CVSS and CWSS are quite similar. There are some strengths and limitations with CVSS, however. One of CVSS strengths lies in its simplicity but it assumes that a vulnerability has already been discovered and verified whereas CWSS can be applied earlier in the process, before any vulnerability has been proven. In addition, CWSS has the advantage that explicitly supports *unknown* values when there is incomplete information.

CVSS/CWSS are widely used standards, for example in CWE/SANS Top 25[3], in OWASP Top Ten[4] or in the National Vulnerability Database[5] . In addition, they are simple and well defined and its metrics comprises the majority of the metrics of the other risk assessment methods described before. For this reason, we have chosen to base our risk assessment method on CWSS, as it is recommended by the ITU-T in X.1525[6].

2.3. Risk assessment and certification in IoT

The IoT is a growing research field. In fact, there is a big emphasis on the security issues surrounding it. In [16], basic requirements on IoT are discussed, such as testing and certifying security, labelling or standardization. The survey performed in [17] looks at threat mitigation approaches in IoT using an autonomic taxonomy, and [18] identifies common software weaknesses of embedded devices used as part of industrial control systems in power grids. The authors of [19] attempt to classify threat types, besides analyse and characterize intruders and attacks facing IoT devices and services, whereas in [20] and [21] the authors recommend a list of minimum requirements that vendors of IoT devices need to

[2] https://www.owasp.org

[3] http://cwe.mitre.org/top25/

[4] https://www.owasp.org/index.php/Top_10_2017-Top_10

[5] https://nvd.nist.gov/

[6] https://www.itu.int/rec/T-REC-X.1525/en

take into account during development to improve the security and privacy of IoT devices.

However, the risk assessment proposals for IoT are limited, and the majority of them are focused on a specific domain.

In this sense, [22] describe a risk-based adaptive security framework for IoT in eHealth that will estimate and predict risk damages and future benefits using game theory and context-awareness techniques. The authors in [23] focus on Bluetooth technology, extending the calculation formula for Authentication of CVSSv2.

In [24], the authors adapt the DREAD risk model to IoT and finally they recommend aggregating the DREAD score of the vulnerabilities using a weighted average function for which the weights have to be determined based on the system type. However, DREAD is not completely objective, and the results may change with different evaluators.

Authors in [25] propose a framework for modelling and assessing the security of the IoT in order to find potential attack scenarios, analyse the security and assess the effectiveness of different defense strategies. Moreover, a security analysis of IoT devices is proposed in [26], performed in a testbed environment using penetration testing methodologies such as port scanning, fingerprinting, process enumeration, and vulnerability scan. However, it does not give a general vision of all the dimensions of the security to advice the user and it does not contemplate labelling.

According to these proposals, current risk assessment approaches for IoT are focused on a specific context, in which certification and labelling aspects are not considered. Indeed, this lack has attracted an increasing attention from several important organizations and entities, such as ENISA, ECSO or DIGITALEUROPE, which are actively working to build a more harmonized and widely accepted security certification ecosystem. In this direction, this work proposes the use of specific technologies to help IoT stakeholders for security risk assessment and testing processes, as the main building blocks to build a security certification and labelling approach for IoT devices. As described below, it represents, in turn, an instantiation of the methodology proposed by ETSI [5], so it aims to design a certification methodology to foster interoperability and acceptance of different stakeholders.

3. GENERAL OVERVIEW

The certification process has been defined in the context of the ARMOUR project, whose objective is to provide duly tested, benchmarked and certified security and trust technological solutions for large-scale IoT using upgraded FIRE large-scale IoT/Cloud testbeds properly-equipped for security and trust experimentations.

Figure 1 shows the overall process of certification and risk assessment, derived from ETSI and ISO 31000 [5], extended to include all the processes of the certification. This approach combines a test-based security risk assessment with a risk-based security testing workstream. We have added a certification phase where the label is generated, because it is not

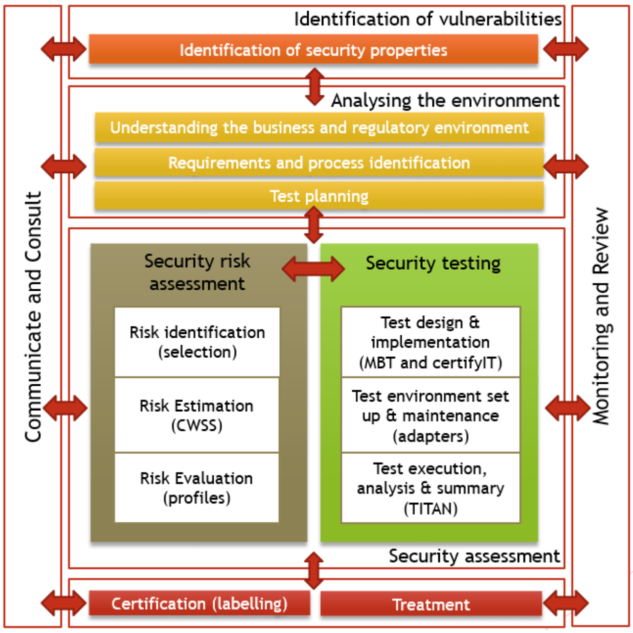
Communicate and Consult
Identification of vulnerabilities
Identification of security properties
Analysing the environment
Understanding the business and regulatory environment
Requirements and process identification
Test planning
Security risk assessment
Risk identification (selection)
Risk Estimation (CWSS)
Risk Evaluation (profiles)
Security testing
Test design & implementation (MBT and certifyIT)
Test environment set up & maintenance (adapters)
Test execution, analysis & summary (TITAN)
Security assessment
Certification (labelling)
Treatment
Monitoring and Review

Fig. 1. General overview of the certification process

considered in the ETSI proposal, and although the original methodology includes a *Treatment* phase (i.e. security controls and other countermeasures), this is not addressed in our instantiation. However, this treatment can be designed from the results of the *Security assessment phase*.

The first phase, *Identification of vulnerabilities*, which is not included in ISO 31000, does an analysis of the IoT environment in order to have a database of threats in IoT. The second phase, which is called *Establishing the context* includes understanding the business, regulatory environment and analyse which security level is required in each of them. For example, in a health context, confidentiality and availability could be considered two very important security properties that could not be as important in home automation. As a result of this phase, several security profiles (e.g. A+, A, B, C, D) related to the context will be defined. Finally, *the test planning* is the activity of developing the test plan (objetive, scope, etc.).

The *Security assessment phase*, which is the most complex, includes the *security risk assessment* (the main subject of this paper) and the *security testing*. Inside *security risk assessment*, three phases are considered:

- *Risk identification*. This phase uses as input the general vulnerabilities identified in the previous phase of *identification of vulnerabilities*. Taking into account TOE, this phase is in charge of selecting which vulnerabilities will be tested.
- *Risk estimation*. This phase assign a risk mark to each vulnerability. For this purpose, default values and test results from the *security testing phase* are provided.
- *Risk evaluation*. This phase compares the result of the risk estimation with the profiles considered in the *Es-*

tablishing the context phase. In this way, the TOE obtains a profile, the highest it fulfils in this specific context.

The *security testing* phase is related to the creation of tests for testing security. However, in the ETSI proposal, the automation of this phase is not contemplated. In this sense, the proposed instantiation is intended to use specific technologies to help for automating this process, easing the update of the label to cope with changing conditions in which the device operates. The integration of such approaches is being done in the scope of the ARMOUR project. It also comprises three phases:

- *Test design and implementation*. This phase aims to design a test suite to obtain metrics and use it in the risk estimation, testing therefore, the risk's grade of each vulnerability.
- *Test environment set up and maintenance*. The execution of the tests suites is ensured through test adaptors, which are needed to adapt the generated test code to each IoT device.
- *Test execution, analysis and summary*. The tests designed in the previous phase are executed. From the execution we gather information related with the result of the tests and related with some metrics, for example time.

Finally, with the data collected from the test execution, taking into account the profile obtained, the context and the certification execution (explained in the following section), the certification process generates a label, helping the user to know the security level of the TOE.

Moreover, the figure shows additional support activities like *Communication and Consult* and *Monitoring and Review* that are meant to set up the management perspective, continuously controlling, reacting, and improving all relevant information and results of the process.

In the following sections, we explain how we have instantiated the ETSI architecture through the use of concrete approaches and technologies. We detail each phase and how they are intended to be executed, paying more attention to the security risk assessment part.

3.1. Identification of vulnerabilities

In first place, we perform a identification of vulnerabilities in the IoT environment. We relied on the analysis done in the context of oneM2M standardisation activities, which covers the whole IoT/oneM2M domain. From them, we did a mapping between these vulnerabilities and eight general vulnerabilities in order to simplify and group them into more general threats adapted to IoT. These vulnerabilities has been extracted from some of the most referenced security aspects that can be found in current IoT literature [20] [19], The assignment has been done following the Table 1, where the relation between them is specified in the second column. The vulnerability 12 (context awareness) is intended to be in the profiles defined in the next section. The purpose of this aggregation is to have a more compacted security dimensions and simplify the label, since it will show the security marks for each of them. In this way, this simplification from 21 vulnerabilities to only 8 will help the user to have a more understandable and easy view of the global security.

Table 1. Relation between OneM2M vulnerabilities and the general ones

General vulnerability	Relation	Vulnerabilities
Lack of Authentication	Protection against a device with non Valid ID	V10,V14
	Protection against a device with a Valid ID but non valid authentication key or certificate	V3,V13
	Cryptographic suite	V1,V4,V19
	Protection against a server with non Valid ID	V10,V6,V14
	Protection against a server with a Valid ID but non valid authentication key or certificate	V4, V13
Lack of confidentiality	Percentage ciphered (general)	V7,V13
	Cryptographic suite	V19
	Percentage ciphered (related with keys)	V6
Lack of authorization	Different profiles per device	V18,V10
	Protection against a replacement with a more privileges key	V3
Dos attack	Protection against attacks performed by a legitimate device	V2,V14
	Protection against attacks performed by the server	V5
	Protection against attacks changing the key of the SO	V3
Lack of integrity	Percentage of integrity protection	V8,V13
	Cryptographic suite	V19
Replay attack	General protection	V9
	Protection of the authorization mechanisms	V17
Insecure cryptography	Dictionary attacks and related	V19
	Cryptographic suite and key length	V19
Lack of fault tolerance	Low cascade impact	V11
	Exception control against buffer overflow	V15
	Protection against injection attacks	V16
	Control the input data	V20
	Control scripts	V21

- Lack of authentication. The endpoints should be legitimate.
- Replay attack. Intermediate entity can store a data packet and replay it at a later stage.
- Insecure cryptography. The cryptographic suite and key length must be enough to avoid certain type of attacks, such as dictionary attack or force brute.
- DoS attacks. Several endpoints can access to the server at the same time in order to collapse it.
- Lack of integrity. Received data are not tampered with during transmission; if this does not happen, then 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 fault tolerance. Exceptions should be controlled to avoid faults that affects the endpoints.

3.2. Establishing the context

In this phase, we perform a risk analysis in order to determinate which level of security is needed in a specific domain in a similar way to EALs in CC. In this way, we add the context variable to the label, since for example, it is more critical

Table 2. Example of profile definition

		Profiles			
Vulnerability	Risk	A	B	C	D
Lack of confidentiality	Low	x	x	x	x
	Medium		x	x	x
	High			x	x
	Critical				x
Replay attack	Low	x	x	x	x
	Medium		x	x	x
	High			x	x
	Critical				x
...					

Table 3. CWSS Metrics resume

Group	Metric	Summary
Base Finding	Technical Impact (TI)	The potential result that can be produced by the weakness, assuming that the weakness can be successfully reached and exploited.
	Acquired Privilege (AP)	Thetype of privileges that are obtained by an attacker who can successfully exploit the weakness.
	Acquired Privilege Layer (AL)	The operational layer to which the attacker gains privileges by successfully exploiting the weakness.
	Internal Control Effectiveness (IC)	The ability of the control to render the weakness unable to be exploited by an attacker.
	Finding Confidence (FC)	The confidence that the reported issue is a weakness that can be utilized by an attacker.
Attack Surface	Required Privilege (RP)	The type of privileges that an attacker must already have in order to reach the code/functionality that contains the weakness.
	Required Privilege Layer (RL)	The operational layer to which the attacker must have privileges in order to attempt to attack the weakness.
	Access Vector (AV)	The channel through which an attacker must communicate to reach the code or functionality that contains the weakness.
	Authentication Strength (AS)	The strength of the authentication routine that protects the code/functionality that contains the weakness.
	Level of Interaction (IN)	The actions that are required by the human victim(s) to enable a successful attack to take place.
	Deployment Scope (SC)	Whether the weakness is present in all deployable instantiations of the software, or if it is limited to a subset of platforms and/or configurations.
Environmental	Business Impact (BI)	The potential impact to the business or mission if the weakness can be successfully exploited.
	Likelihood of Discovery (DI)	The likelihood that an attacker can discover the weakness.
	Likelihood of Exploit (EX)	The likelihood that, if the weakness is discovered, an attacker with the required privileges/authentication/access would be able to successfully exploit it.
	External Control Effectiveness (EC)	The capability of controls or mitigations outside of the software that may render the weakness more difficult for an attacker to reach and/or trigger.
	Prevalence (P)	How frequently this type of weakness appears in software.

the lack of confidentiality in health than in home automation, which will be reflected in the profiles associated to each of them.

From this analysis, we define several profiles that indicates which level of security must be achieved by the TOE in the context considered to obtain each profile, following the notation of Table 2. In this case, the device needs a low risk level in confidentiality if it wants to obtain the A profile. This will be clarified in Section 4. It is worth noting that if a device fulfils one specific profile, it also fulfils the lower ones, so if A profile is obtained, it also fulfils B, C and D profiles.

3.3. Security testing

From the vulnerabilities considered in the first phase, we produce security tests that are used during the security risk assessment phase, allowing us to refine the risk associated to each vulnerability. In ARMOUR project, this process is intended to be automatized [27] to derive on a agiler and easier certification process.

In the first phase, *Test design and implementation*, we design a test suite to test the risk's grade of each vulnerability. To automatize this process, a Model-based testing (MBT) approach is used to specify the tests and their behaviour [28]. MBT has shown its benefits and usefulness for systematic compliance testing of systems [29]. In this approach, the structure of the system is modelled by Unified Modelling Language (UML) class diagrams, while the systems behaviour is expressed in Object Constraint Language (OCL)[7], using the CertifyIt tool [28]. Functional tests are obtained by applying a structural coverage of the OCL code describing the operations of the IoT system under test.

Secondly, in the *Test environment set up and maintenance phase*, adapters are used to cope with the particularities of each IoT device with its own specific interfaces.

Finally, in the *Test execution, analysis and summary phase*, the tests defined in MBT are exported in Testing and Test Control Notation (TTCN) v.3 language using the tool CertifyIT. The main goal of the use of TTCN-3 in the proposed methodology is the systematic and automatic testing of security properties in IoT devices for improving efficiency and scalability. The tests will be executed on a local or external large-scale testbed such as FIT IoT Lab[8] , where the scenario tested will be implemented by means of TITAN[9]. TITAN is a TTCN-3 compilation and execution environment for different platforms that in combination with CertifyIt create executable tests, whereas FIT IoT-LAB offers the large-scale testbed on which the test cases are executed.

3.4. Security risk assessment

From the vulnerabilities considered in the first phase, we identify the potential ones that can be applicable to the scenario and context. 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. Risk assessment phase is intended to provide different results from testing to serve as the baseline for the certification scheme, providing a risk mark associated to the vulnerabilities considered in order to be able to compare different scenarios and to be used by the certification to obtain the final label. Towards this end this methodology is based on the identification of different metrics per functional block (e.g. lack of authentication), that is, the factor we take into account in the risk mark. Test results help to increase the trust level on the risk associated to each vulnerability in IoT products and solutions. In order to be able to obtain the profile that is used for the labelling, we do an intermediate step obtaining the risk mark of each vulnerability considered. We propose using the CWSS mechanism, that takes into account the metrics we can see in Table 3. We modify some of the metrics of CWSS, since CWSS in intended to be used in software environments and our purpose is a risk assessment for IoT. These are the considerations:

- Internal Control Effectiveness is set to Non applicable, since we consider this is a software property not applicable to general IoT environments.

- Business Impact is set to Non applicable, since we

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

[8] https://www.iot-lab.info/

[9] http://www.ttcn-3.org/

consider the context after, in the profiles.

- Finding confidence is set to Non applicable, since we are evaluating a scenario before being attacked.
- Some of the CWSS metric values are obtained from the tests execution. The reason of doing it, is to perform a better adaptation of CWSS to the IoT environment, gathering the value of the metrics directly from the practice, from a security test.
- Some of the CWSS metric values are set by default taking into account the vulnerability.

Finally, we calculate the subscores and the general score for each vulnerability by means of the CWSS formula:

$$S_v = BF_s * AS_s * E_s$$

where BF_s, AS_s and E_s are the subscore metrics of CWSS (Base finding, Attack surface and Environment) that are calculated (using the notation of the Figure 3) as:

$$BF_s = [(10 * TI + 5 * (AP + AL) + 5) * f(TI)] * 4$$

$$AS_s = [20*(RP+RL+AV)+20*SC+15*IN+5*AS]/100$$

$$E_s = [(10 + 3 * DI + 4 * EX + 3 * P) * EC]/20$$

where $f(TI) = 0$ if $TI = 0$; otherwise $f(TI) = 1$.
In order to be able to obtain the profile fulfilled, we associate CWSS score intervals with risk levels (low, medium, high and critical) and we determine it 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.

Table 4. Mapping between CWSS scores and risk levels of the profiles

CWSS score	Risk level
0-30	Low
31-62	Medium
63-84	High
85-100	Critical

3.5. Certification and labelling

As an output of the general certification process, we obtain a label associated to the risk of the scenario tested. For this approach, 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 CC approach, three mains aspects are considered to be included in the label:

- TOE (Target of Evaluation): 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 the context where it has been tested.
- 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 EALs than CC.

Following the recommendations of ENISA [2], as security requirements are in fact multi-dimensional, the result of the evaluation need to be communicated appropriately to the user. For this reason the label includes the profile of each general vulnerability considered, in addition to the certification execution and the TOE. In this way, we provide the user with more information and we do not show a false sense of security, since for example, a bad mark in confidentiality could be compensated with a good mark in authentication if we combine the marks by means of an arithmetic function. For making this more visual, we propose the usage of an octagon like the triangle in [30] or the pentagon in [3], where the vertices are the eight general vulnerabilities and the internal lines, the profiles. In addition, as security is a dynamic concept, we propose the usage of a QR as label, that can be updated in case of a new vulnerability discovered in the product. In this way, if the product security has to be updated, the procedure will be automatic, since we already have the tests. Therefore, the communication to the user will be instantaneous.

4. EXAMPLE

As an example of the risk assessment methodology, we present the results of two experiments using CoAP and CoAP-DTLS (CoAPs) [31], which are widely used in IoT scenarios. The CoAPs experiment is focused on testing of the establishment of the DTLS secure channel and on a CoAP request, whereas the CoAP experiment only does a CoAP request, without the establishment of a security channel. These experiments are developed in a home automation context. Note that depending on the context and scenario (CoAP or CoAPs), the vulnerabilities applicable are different.

4.1. CoAP-DTLS

We present as an example of subscore calculation, the Lack of Confidentiality, one of the vulnerabilities defined in section 3.1. First, we define the test whose result gives us the value of the *technical impact*. The test starts with the establishment of the communication between a smart object and a server. The sniffer observes all the conversation through a packet sniffer such as Wireshark. We calculate the percentage of non-encrypted communication, which is the value of the *technical impact*, and the algorithm and key length, which are in clear in DTLS. The test also gives us the value for the *external control* (low/high if the communication is in clear or completely ciphered, best available if it is partially ciphered with enough key length and moderate in other case).

The rest of the values are filled by default for this vulnerability. For the *technical impact*, the percentage of non-encrypted data is employed since this value represents the potential result that can be produced by the weakness. By sniffing the data, the attacker cannot acquire any privilege, so the values related to this (*acquired privilege* and *acquired privilege layer*) are none and quantified set to zero. Moreover, *External control* is given by the test. For the attack surface metrics, we set the *required privilege* to none, since the attacker does not need any privileges to sniff, only access to the Internet, so the *required privilege layer* is network and the *access vector*, Internet. The attacker does not require authentication and user interaction, so *authentication strength* is set to none and *level of interaction* is set to automated. This vulnerability is present in all the devices of the experiment, so the *deployment scope* is all. Finally, in environmental metrics, the *likelihood of discovery* and *exploit* is high, since it is easy to use this vulnerability without knowledge, and the *prevalence* is high, taking into account that this type of attack is frequently performed to discover sensitive data or ways to exploit another vulnerability. We obtain the mark for this vulnerability, obtaining a low risk level:

$$BF_s = [(10 * 0.94 + 5 * (0.1 + 0) + 5) * 1] * 4 = 59.6$$

$$AS_s = [20*(1+0.7+1)+20*1+15*1+5*1]/100 = 0.94$$

$$E_s = [(10 + 3 * 1 + 4 * 1 + 3 * 1) * 0.3]/20 = 0.3$$

$$S_v = 59.6 * 0.94 * 0.3 = 16.81$$

We apply this method to all the threats, calculating the CWSS scores and doing the mapping between the scores and the risk levels in order to compare them with the available profiles. As we can see in Table 5, CoAPs fulfils for *lack of confidentiality* all the profiles, so as we choose always the highest profile, the label assigned to this vulnerability will be A.

The obtained profile in each vulnerability in addition to the specification of the certification execution, conform the final label for CoAPs. The result of the label will be presented to the user in a multidimensional label like in Figure 2 (left).

4.2. CoAP

In first place, we have to select which vulnerabilities can apply to this TOE. As CoAP does not have integrity protection, ciphering, DoS protection, authentication and authorization mechanisms, these vulnerabilities are going to be labelled by default with critical risk. Dictionary attack will be labelled with low risk, since it is not applicable to CoAP, it does not use a key. Following the example presented before, we calculate the scores for the rest of the vulnerabilities and we compare the obtained CWSS scores with the available profiles for home automation (Table 5), obtaining that CoAP fulfils completely only D profile. As shown, CoAPs obtains a higher profile (or equal) in all the vulnerabilities than CoAP. For example, CoAPs obtains a low risk mark in *lack of authentication* (the A profile), whereas CoAP obtains a D profile, which reflects the lost of security of not using a secure channel, in this case established by DTLS. The multidimensional label for CoAP is shown in Figure 2 (right).

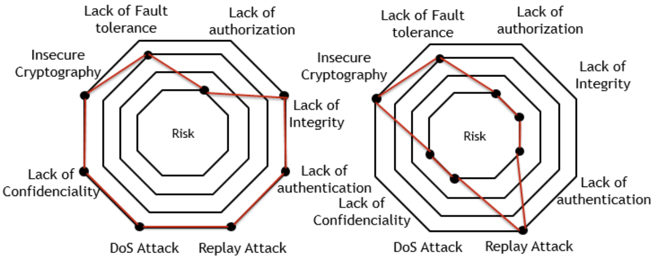

Fig. 2. Multidimensional label obtained by CoAP and CoAPs, respectively, in home automation

5. CONCLUSION

Nowadays, the IoT ecosystem demands for large-scale deployments, where devices can provide a high level of security, in order to cover typical threats and vulnerabilities. One of the main ambitions of the ARMOUR project stems from the specification of a certification approach, to give experimenters the ability to assess and compare different IoT security technologies. Towards this end, there is a real need to consider a systematic and automated methodology that enables scalable testing approaches for security aspects in IoT. This document aimed to provide an initial description of the proposed methodology for this purpose based on the ETSI proposal, focusing on the risk assessment. The automatic generation of security tests as well as the automation of the results by means of MBT, CertifyIT and TITAN will be our future line of work.

6. REFERENCES

[1] Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, and D. Qiu, “Security of the internet of things: perspectives and challenges,” *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[2] ENISA, “On the security, privacy and usability of online seals. an overview.,” Dec. 2013.

[3] ECSO WG 1, “Standardization certification labeling and supply chain management,” .

[4] DIGITALEUROPE, “Digitaleuropes views on cybersecurity certification and labelling schemes,” Mar. 2017.

[5] ETSI, “Methods for testing & specification; risk-based security assessment and testing methodologies,” 2015.

[6] CCRA, “Common criteria,” http://www.commoncriteriaportal.org.

[7] “Commercial product assurance (CPA),” https://www.ncsc.gov.uk/scheme/commercial-productassurance- cpa.

[8] “UL-2900,” https://standardscatalog.ul.com/standards/en/outline 2900-1 2.

Table 5. Evaluation of CoAPs and CoAP

		CWSS		Profiles				Obtained Profile	
Vulnerability	Risk	CoAPs	CoAP	A	B	C	D	CoAPs	CoAP
Lack of confidentiality	Low	x		x	x	x	x	A	D
	Medium				x	x	x		
	High					x	x		
	Critical		x				x		
Replay attack	Low	x	x	x	x	x	x	A	A
	Medium				x	x	x		
	High					x	x		
	Critical						x		
Insecure cryptography	Low	x	x	x	x	x	x	A	A
	Medium				x	x	x		
	High					x	x		
	Critical						x		
Lack of authentication	Low	x		x	x	x	x	A	D
	Medium				x	x	x		
	High					x	x		
	Critical		x				x		
DoS attacks	Low	x		x	x	x	x	A	D
	Medium				x	x	x		
	High					x	x		
	Critical		x				x		
Lack of integrity	Low	x		x	x	x	x	A	D
	Medium				x	x	x		
	High					x	x		
	Critical		x				x		
Lack of fault tolerance	Low			x	x	x	x	B	B
	Medium	x	x		x	x	x		
	High					x	x		
	Critical						x		
Lack of authorization	Low			x	x	x	x	D	D
	Medium				x	x	x		
	High					x	x		
	Critical	x	x				x		

[9] Microsoft, "DREAD scheme," https://msdn.microsoft.com/enus/ library/ff648644.aspx.

[10] MICROSOFT, "The STRIDE threat model," https://msdn.microsoft.com/enus/ library/ee823878(v=cs.20).aspx.

[11] CENZIC, "HARM score," http://doc.cenzic.com/sadoc9x14ba847/harm.htm.

[12] CERT/SEI, "C programming language secure coding standard," 2007.

[13] Richard A. Caralli, James F. Stevens, Lisa R. Young, and William R. Wilson, "Introducing OCTAVE allegro: Improving the information security risk assessment process," Tech. Rep., CERT, 2007.

[14] FIRST, "Common vulnerabilities scoring system (CVSS)," 2014, https://www.first.org/cvss.

[15] MITRE, "Common weakness scoring system (CWSS)," 2014, https://cwe.mitre.org/cwss/cwss v1.0.1.html.

[16] AIOTI, "Report on workshop on security and privacy in the hyper-connected world," 2016.

[17] Qazi Mamoon Ashraf and Mohamed Hadi Habaebi, "Autonomics chemes for threat mitigation in internet of things," *Journal of Network and Computer Applications*, , no. 49, pp. 112–127, 2015.

[18] Margus Vlja, Matus Korman, and Robert Lagerstrm, "A study on software vulnerabilities and weaknesses of embedded systems in power networks," in *2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, 2017.

[19] Mohamed Abomhara and Geir M. Kien, "Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security*, vol. 4, pp. 65–88, 2015.

[20] K. Moore, R. Barnes, and H. Tschofenig, "Best current practices for securing internet of things (IoT) devices," 2016.

[21] BITAG, "Internet of things (IoT) security and privacy recommendations," 2016.

[22] Habtamu Abie and Ilangko Balasingham, "Risk-based adaptive security for smart iot in ehealth," in *7th International Conference on Body Area Networks*, 2012.

[23] Yanzhen Qu and Philip Chan, "Assessing vulnerabilities in bluetooth low energy (BLE) wireless network based iot systems," in *IEEE International Conference on Intelligent Data and Security*, 2016.

[24] Hunor Sndor and Gheorghe Sebestyn-Pl, "Optimal security design in the internet of things," in *Digital Forensic and Security (ISDFS), 2017 5th International Symposium*, 2017.

[25] Mengmeng Ge, Jin B. Hong, Walter Guttmann, and Dong Seong Kim, "A framework for automating security analysis of the internet of things," *Journal of Network and Computer Applications*, 2017.

[26] Vinay Sachidananda, Shachar Siboni, Asaf Shabtai, and Yuval Elovici, "Let the cat out of the bag: A holistic approach towards security analysis of the internet of things," in *3rd ACM International Workshop*, 2017.

[27] Gianmarco Baldini, Antonio Skarmeta, Elizabeta Fourneret, Ricardo Neisse, Bruno Legeard, and Franck Le Gall, "Security certification and labelling in internet of things," in *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec. 2016.

[28] F. Bouquet, C. Grandpierre, B. Legeard, F. Peureux, N. Vacelet, and M. Utting, "A subset of precise UML for model-based testing," in *3rd int. Workshop on Advances in Model Based Testing*, 2007, pp. 95–104.

[29] G. Bernabeu, E. Jaffuel, B. Legeard, and F. Peureux, "MBT for global platform compliance testing: Experience report and lessons learned," in *25th IEEE International Symposium on Software Reliability Engineering Workshops*, 2014.

[30] Smart-Grid Task Force Stakeholder Forum, "Best available techniques reference document for the cyber-security and privacy of the 10 minimum functional requirements of the smart metering systems," 2016.

[31] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," June 2014.