Proceedings of the Federated Conference on Computer Science and Information Systems pp. 593–602 DOI: 10.15439/2018F226 ISSN 2300-5963 ACSIS, Vol. 15

Secure Cloud Computing: Risk Analysis for Secure Cloud Reference Architecture in Legal Metrology

Alexander Oppermann, Marko Esche, Florian Thiel
Physikalisch-Technische Bundesanstalt (PTB)
Department 8.5 Metrological IT
Abbestr. 2-12, 10587 Berlin, Germany
Email: {alexander.oppermann, marko.esche, florian.thiel}@ptb.de

Jean-Pierre Seifert
Technische Universität Berlin,
Security in Telecommunications,
Ernst-Reuter-Platz 7, 10587 Berlin
Email:jpseifert@sec.t-labs.tu-berlin.de

***Abstract*—In the field of Legal Metrology, a risk assessment is demanded by European directives for certain measuring instruments. In this paper, a previously published reference cloud architecture will be subjected to such an assessment to demonstrate its suitability for providing adequate software protection. A specially tailored and standardized method is used to identify essential threats and common attack vectors for the reference architecture. With the help of calculated probability score and risk factors, the fulfillment of the essential requirements of the applicable European directives are shown. Furthermore, Attack Probability Trees are applied to more complex scenarios to identify suitable countermeasures to increase the resilience level where necessary.**

I. Introduction

LEGAL METROLOGY'S raison d'être is to establish trust between all stakeholders such as customers, manufacturers and users of measuring instrument. Since none of the involved parties alone can guarantee the validity and integrity of measurements, a Notified Body, e.g. the Physikalisch-Technische Bundesanstalt (PTB) in Germany, is obligated to inspect measuring instruments. The essential requirements of the Measuring Instruments Directive (MID) [1], such as reproducibility, repeatability, durability and protection against corruption of measuring instruments and measurements, have to be fulfilled before entering the market. Enhancing public trust in measuring instruments is vital for Legal Metrology, especially in a world with new and increasingly complex technologies in use.

New technologies, like Cloud Computing enable manufacturers and users of measuring instruments to provide improved services to customers that are more flexible and comfortable to, for example, access meters via mobile devices or enable improved service via intelligent data services. However, Legal Metrology faces a radical change through the transformation of well-contained measuring instruments nowadays to future distributed measuring systems. In 2016, the stated transition and security implications for Legal Metrology were described, concluding with a proposition for a Secure Cloud Reference Architecture focusing on these challenges [2]. By fulfilling the essential requirements of the MID and the applicable WELMEC (Western European Legal Metrology Cooperation) guide 7.2 [3] a level of legally adequate security is met. The introduced architecture further tackles threats, such as a malicious insider and data manipulation in the cloud, via fully homomorphic encryption (FHE) [4]. Moreover, exposing FHE to real-world requirements, four application scenarios were developed and applied to Smart Meter Gateway (SMGW) tariffs. These tariff applications were derived from the SMGW's technical guide of the Federal Office for Information Security (BSI) in Germany.

In this paper, a risk analysis is applied to the Secure Cloud Reference Architecture to fulfill the legal requirements (see Section II). This risk analysis is based on software risk assessment for measuring instruments under legal control proposed by WELMEC Working Group 7 [5]. By objectifying the derived probability score for identified threats while following at the same time the guidelines of ISO/IEC 27005, ISO/IEC 15408 and ISO/IEC 18045, this risk assessment method enables comparability and standardizes the otherwise highly subjective assessment process. Furthermore, potential countermeasures are identified and quantified using Attack Probability Trees (AtPT) [6] for derived assets to be suitable protected.

The remainder of this paper is structured as follows. Section II sketches the Secure Cloud Reference Architectures and describes the considered parts for this risk assessment. Section II-A explains the derived assets and applies the risk assessment method and its shortcomings. In order to introduce the AtPT to tackle the further assets in Section III, Section IV gives an overview of the results, conclusions and further work.

II. Secure Cloud Reference Architecture

The distributed measuring instrument and its reference architecture are described in [2] and a summarized overview of its modules can be seen in Figure 1 and 2. The architecture uses virtualization techniques, in order to separate software modules subject to legal control from legally non-relevant ones. The purpose of a reference architecture is to provide a generic software framework which manufacturers can adopt in their products to provide adequate software protection in line with MID requirements.

This approach benefits not only from decreased idle times and an improved cost-efficiency ratio for employed servers, but also facilitates software update processes for manufacturers in the legally non-relevant software part. This improved update

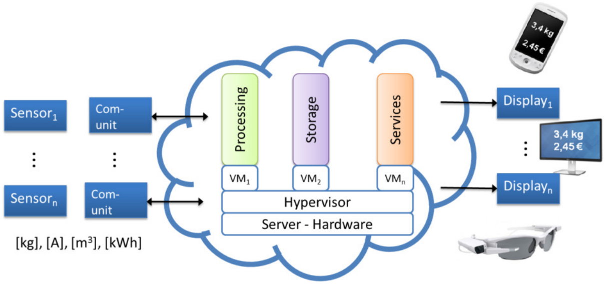

Fig. 1. Overview of the distributed measuring instrument. The measurement device is reduced to only a sensor and communication unit, while processing and storage will be moved to the cloud environment.

process avoids a re-verification of the instrument by the market and user surveillance body and thus decreases downtimes and costs for the manufacturer.

The basis of the Infrastructure as a Service layer (IaaS) is built with the help of the Openstack framework. Core functionalities are mapped to physical devices, such as server, storage and network. Through separation into logical smaller entities via subnetworks, the network and thus the IaaS layer constitute the first low level separation between legally relevant and non-relevant processes.

The Platform as a Service layer (PaaS) consists of a microservice pattern build with Spring Boot and Spring Cloud as well as the Netflix software stack. By reducing services to their core functionality and at the same time minimizing the software lines of code (SLOC), the microservice pattern enables to maintain a clean code base. Furthermore, it offers flexible scaling and efficient resource pooling by cutting idle times of the underlying hardware. Deploying and developing services independently of each other fosters productivity within the software development team and encourages creativity. Nevertheless, stability and downtime will not be a threat to the architecture because of a rigorous separation. A stepwise transition of software versions is encouraged by running different releases side-by-side. The high level separation allows each microservice to be written in the best problem-fitting programming language.

The communication of messages is realized via RESTful API. An active message queue (ActiveMQ) guarantees reliability and pseudo resilience for messages. Messages can be stored in a queue and will be delivered later in time, in case of unavailability of services.

Fully homomorphic encryption (FHE) enables computation of encrypted messages without decrypting them first [7]. The smart meter gateway tariffs application are protected by FHE [4] and hosted at the Software as a Service layer (SaaS). Measurements are encrypted directly in the sensor unit to be processed securely in a centralized cloud structure.

In the next paragraphs, a brief description is given of the most significant legally relevant processes and virtual machines (VM). A summarized overview of the topology is illustrated in Figure 2. Increasing the portability, distributivity and scalability by separating the services via VM another security layer is introduced. The described services represent the common ground of all fields in Legal Metrology to fit a generic reference architecture.

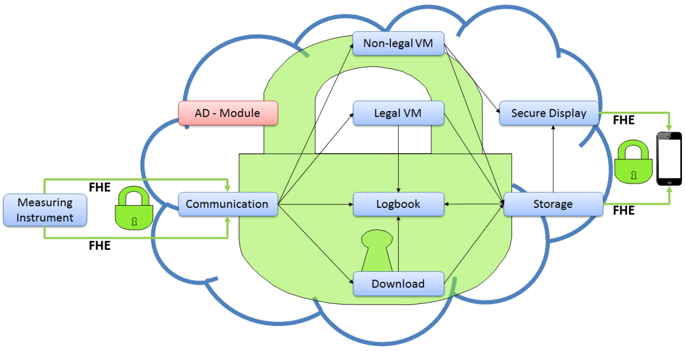

Fig. 2. Overview of the cloud reference architecture. The measurement data is completely secured throughout the whole process via FHE. The legally relevant processes are separated into different virtual machines with well defined communication paths.

a) Logbook: All relevant activities around the measuring system, i.e. arrival, processing, saving of measurement data, user activities, software updates etc are logged via the logbook service hosted in the Logbook VM.

b) Legal Processing: The Legal VM uses the most of all CPU cores available, because it is responsible for processing encrypted measurement data.

c) Download Manager: After an integrity and authenticity verification of the signed software update, the Download Manager will forward the software update, as intended from the manufacturer, to the dedicated machine.

d) Storage Manager: A database stores measurement data for a long period of time. The Storage Manager will make measurement data available via an REST-interface to other authorized services.

e) Monitoring Service: Detecting anomalies within the system, via continuously monitoring the behaviour of all VMs, is an important part of the security mechanisms of the cloud reference architecture. The Monitoring Service provides APIs for real time monitoring.

A. Derivation of Assets to be protected

Esche et al. [5] developed a risk assessment method based on ISO/IEC 27005 [8] and WELEMEC Risk Assessment Guide [9]. The approach consists of three stages and is shortly summarized in the following paragraphs. This algorithm is here applied to the secure cloud reference architecture (see Section III).

Every measuring instrument that undergoes conformity assessment has to fulfill the essential software requirements listed in Annex I of the MID before being put on the market. From these requirements three relevant assets are selected here that are noteworthy to be protected for all kind of measuring instruments, i.e. *measurement data*, *software that is critical for measurement characteristics* , and *metrologically relevant parameters* stored or transmitted. For each, the MID requires integrity and authenticity protection. Consequently, these assets

TABLE I
FORMAL DEFINITION OF THREATS

ID	Threat Intention	Description
B1	Integrity of transmitted measurement data	An attacker alters measurement data during transmission.
B2	Authenticity of transmitted measurement data	An attacker creates tampered measurement data, that will be assigned wrongly to a verified measuring instrument.
B3	Evidence of an intervention	An attacker prevents legally relevant events from being registered in the logbook.
B4	Integrity of Parameters	An attacker alters persistence saved parameters, e.g. connection parameters.
B5	Availability of the Logbook Service	An attacker prevents a legally relevant service from answering requests.

must be secured against intentional or unintentional changes. By fulfilling this demand, integrity and authenticity of these assets are guaranteed. In addition, the MID requires evidence of an intervention, i.e. events registered in a logbook, to be available during verification.

a) Threat definition: A threat is any invalidation of a security property of a given asset. To define a threat, aside from the asset definition, several attacker models should be taken into account, for example, inside attacker and external attacker. Usually the market participant with the highest skill level can be used as a reference model. Additionally, different access levels and their associated roles within a measuring instrument take an important part in the risk assessment. In Table I these five assets are linked to a threat intention and short description of what an attacker wants to achieve. The assets itself will be further described in separate tables, where the attack vectors (technical steps needed to implement a threat) are broken down into atomic attacks with a time, expertise, knowledge, window of opportunity and equipment column that are individually scored (see Section III), according to [10]. This procedure has the advantage of objectifying the risk assessment procedure based on scores for well-defined features of any attack. This enables manufacturers and Notified Bodies alike, to be able to compare the same threats for different measuring instruments.

b) Identification of Attack Vectors: The second risk assessment phase is the least formalized stage. It starts with the examination of the manufacturer's documentation of the measuring instrument. Followed by creating a collection of possible attack vectors, needed to realize the prior identified threats from stage one. The collection comprises attack vectors reaching from simple to very complex structured attacks.

c) Calculating Probability Score and Risk Score: In phase three, the interim results from stage one and two are combined, i.e. an adverse action with at least one associated attack vector. Thereafter, the likelihood of implementing such an attack has to be calculated. The evaluation is based on the following five features [11] that lay the foundation to score and identify the resources that all attacks have in common:

- Elapsed Time (0-19 points)
- Expertise (0-8 points)
- Knowledge of the TOE (0-11 points)
- Window of Opportunity (0-10 points)
- Equipment (0-9 points)

The amount of *elapsed time* represents the time needed to implement a specific attack by any chosen attacker. The score ranges from 0 (equals 1 day) to 19 (more than half a year). *Expertise* represents the skill set of an attacker, where 0 is a layman and 8 is given when an attacker has to have competence in more than one field. *Knowledge of the Target of Evaluation (TOE)* scores the needed information on an attacked measuring instrument. It starts with publicly available knowledge (0) and ends with critical insider knowledge (11), that usually resides with the manufacturer. The *window of opportunity* evaluates the possibility available to an attacker, where 0 represents unlimited access, which would be common for measuring instruments connected to the Internet. If the access is difficult, a value of 10 should be given. In case it is impossible to obtain access, no rating is done and the attack vector would be removed from the list. The last category scores the *equipment* needed to carry out the attack. Standard available hardware or software is described by 0, where 9 represents multiple bespoke devices or software.

After successfully calculating the sum yielded by the five categories for the chosen attack, a probability score is matched to the different ranges of the total sum. In Table II the Common Criteria evaluation is also included in the final probability score calculation, so that a basic resistance results in a total sum of 10-13 points while 24 or more points represent a high resilience against the rated attack. Finally, the resistance evaluation is associated with the probability score, where 1 represents an unlikely occurrence while 5 stands for high probability to occur.

The final risk will be calculated by multiplying the impact score for the threat with the probability score, that is issued in Table II, of the most likely realized attack vector:

$$\text{risk score} = \frac{\text{impact score}}{5} \cdot \text{probability score} \tag{1}$$

TABLE II
CALCULATION OF A TOE AND ASSOCIATION OF A PROBABILITY SCORE ACCORDING TO [5]

Sum of Points	TOE Resistance	Probability Score
0-9	No rating	5
10-13	Basic	4
14-19	Enhanced Basic	3
20-24	Moderate	2
>24	High	1

TABLE III
ATTACK VECTORS FOR THREAT B1

Attack-ID	Attack Vector	Time	Expertise	Knowledge	Window of Opportunity	Equipment	Sum	Damage
A3	Manipulate data in transit	19	8	11	10	0	48	1
A4	Exchange processing unit	7	6	11	4	0	29	1

TABLE IV
PREREQUISITES FOR ATTACK VECTOR A3

Attack-ID	Attack Vector	Time	Expertise	Knowledge	Window of Opportunity	Equipment	Sum	Damage
A3.1	MITM-attack	1	6	11	10*	0	28	1
A3.2	decrypt-encrypt data	19	8	11	0	0	38	1

III. METHODOLOGY OF ASSETS

In this section, the risk assessment algorithm will be applied to the secure cloud reference architecture, that were both briefly introduced in the previous section. The threats listed in Table I will be treated sequentially and will pass the three stages of risk assessment. Afterwards, in Subsection III-C the Attack Probability Tree (AtPT) is introduced to describe more complex attack scenarios, by introducing a prescribed way to construct attack vectors in a standardized and compact way. At the end, suitable countermeasures for attack vectors will be discussed briefly.

A. Integrity of transmitted measurement data

The threat intention of the attacker is to undermine the integrity of transmitted measurement data by manipulating measurement data during transmission. The sensor unit will be considered, that collects the data and encrypts them with a protected public key via FHE before sending them to the cloud reference architecture. The transmission is secured by Transport Layer Security (TLS) and additionally by a x.509 certificate at the cloud service endpoint, so that the sensor unit usually knows the receiver. An insider attack is assumed with the attacker having the access rights of an administrator. For this threat, two attack vectors are taken into consideration, namely A3 and A4 (see Table III). A3 needs two prerequisites A3.1 and A3.2 (see Table IV), in order to be feasible.

To manipulate the data in transit, the attacker has to carry out an active Man-In-The-Middle attack (MITM) (see Table IV A3.1), that means the connection has to be rerouted via the attacker's interception device and the TLS-connection has to be captured during key exchange. Furthermore, the certificate has to be forged by, for example, getting the private key of the server and the client to establish active sessions at both ends with the impersonated certificates needed for authentication. The client's improper validation of the certificate would be a big advantage for the attacker.

The time needed to execute such an attack would be less than a day (1), if the attacker is an expert (6) and has critical knowledge of the system (11). While the window of opportunity is difficult (10), since the manipulation has to be carried out during transmission within the boundaries of transmission delay. There is no special equipment needed (1), that exceeds standard hardware. So the total sum of points for this attack (48) leads to high TOE resistance (see Table II).

Even if A3.1 (MITM) is successfully established, the data itself is still encrypted by FHE. Lattice based cryptography is provable secure and provides worst-case security that is still not broken by quantum algorithms. Therefore, the maximum time of more than half a year (19) assumed for A3.2. The attacker has to have expertise on several fields (8) to decrypt and/or break cryptography as well as having critical system knowledge (11) at disposal. Once, the cryptography is broken, the window of opportunity is unnecessary (0). From the authors' point of view standard hardware (0) is sufficient. This yields a total sum of 38 points and again implies high resilience against the attack vector.

The two attack vectors A3.1 and A3.2 both need to be executed to form A3. The result is shown in Table III and implies a high resilience (48) for this attack vector. According to Table II, the sum score translates to a probability score of 1. Since this threat has potential influence on all future measurement values, the impact score is 5 and the subsequent risk ($\frac{\text{impact score}}{5} \cdot \text{probability}$) also takes on a value of 1. PTB does not accept technical solutions with a risk greater than 3. This solution qualifies for PTB certification.

Another attack vector is to exchange the FHE-processing unit (A4) in the cloud, in order to manipulate the data during processing. First, the attacker needs to have access to the software repository, to manipulate the FHE-processing unit and then deploy the manipulated software into the cloud service. Furthermore, the hash of the manipulated software has to match the comparative hash, that the market surveillance monitor evaluates. Given the bonus of an insider attacker with the access level of an administrator, it should be feasible, yet the time frame for execution is less than two months (7). The attacker needs to be at least an expert (6) in IT and the window of opportunity is moderate (4), since a lot of security mechanisms have to be worked around. No special hardware (0) is needed. This yields a total sum of 29 and means a high TOE resistance and a probability score of 1. The threat influences all future measurements, the impact score is 5 and the resulting risk has a value of 1.

TABLE V
ATTACK VECTORS FOR THREAT B2

Attack-ID	Attack Vector	Time	Expertise	Knowledge	Window of Opportunity	Equipment	Sum	Damage
A1	Manipulate sensor unit	4	8	11	0	7	30	1
A2	Replace sensor unit	4	8	11	0	7	30	1
A3	Spoof identity	19	6	11	0	0	36	1

TABLE VI
PREREQUISITES FOR ATTACK VECTOR A3

Attack-ID	Attack Vector	Time	Expertise	Knowledge	Window of Opportunity	Equipment	Sum	Damage
A3.1	Steal key from vault	1	6	11	0	0	18	1
A3.2	Obtain certificate	19	6	0	0	0	25	1
A3.3	Generate false data	19	6	11	0	0	36	1

B. Authenticity of transmitted measurement data

The threat intention of B2 is to attack the authenticity of transmitted measurement data. In Table V three attack vectors A1-A3 are summed up, while the third is composed of three sub attack vectors displayed in Table VI.

The easiest way of attacking the authenticity is to manipulate the origin of the measurement data: the sensor unit itself (A1). The idea behind this attack vector is just to compromise the authenticity, thus it is enough to break the seal and replace the physical sensor with a tampered one, that calculates, for example, a smaller measurement value. Breaking the seal implicates forging a new seal, so that the instrument does not seem to be manipulated to market surveillance.

The time needed for this invalidation of authenticity (A1) is less than a month (4) and the attacker needs to be expert on several fields (8), since forging an official calibration seal needs knowledge and special equipment (7). Furthermore, replacing the physical sensor requires critical knowledge (8). The window of opportunity is unlimited (0) for this attack vector, because the instrument in the field is not subject to constant surveillance. In total, the attack vector reaches 30 points and represents a TOE with high resistance with an associated probability score of 1, which translate to a risk level of 1 because of its influence of all future measurement values (impact score of 5). However, it is noteworthy that in Legal Metrology there is no higher protection level achievable than a sealed hardware solution.

The second attack vector A2 deals with obtaining security features from the original sensor unit (physical sensor + communication unit) and replacing this unit with a tampered one that is identically constructed. Hereby, the attacker extracts, for example, the protected key (public key) needed for encryption from the original sealed instrument and then stores this security feature in an identical, but tampered unit. A2 differs from A1 since it does not involve tampering original hardware, but buying malfunctioning hardware on purpose and putting it into use. The scores are the same as for the previous attack vector. It is again considered very hard to forge an official verification seal, which is reflected in the total sum of 30 points and offers high resilience.

With the last attack vector A3 the identity of the sensor unit will be spoofed by masquerading the IP address of the attacker's sensor unit, for example, by faking the source address field in the TCP header. In order to be successful at the cloud service endpoint, the attacker has to first obtain the protected key from the software vault in the cloud service, in order to be able to encrypt its fake measurement data (A3.1). Given the fact that an insider attacker with the privileges of an administrator is considered, the access to the cloud architecture is self-evident. The attacker will retrieve the information in less than a week (1). The postulated skill set of an expert (6) is needed in an IT related area and critical knowledge (11) of the system is demanded. A3.1 yields in total 18 points, which is considered as an enhanced basic resistance level.

As a next step (A3.2), the attacker has to get his hands on the private key of the x.509 certificate. It is assumed that this is very time consuming (>6 month) (19) but feasible for an expert (6), in order to forge the x.509 certificate and overcome the authentication barrier. The attack vector A3.2 has a total sum of 25 points and achieves high resilience against this threat.

As a last action, the attacker has to generate false measurement data with the stolen key from A3.1 and authenticates himself against the cloud service endpoint with a forged certificate, in order to achieve the objective to compromise the authenticity of the measurement data. Because of the logical AND operation of A3.1 and A3.2 the highest value will run into A3. That leads to the time frame of more than 6 months (19), an expert level (6) and the requirement of critical system knowledge (11), which totals into 36 points and reaches a high resistance level. The probability score evaluates to 1 with an associated risk level of 1 because of the influence of all future measurements (impact score 5).

For threat intention B3 the same risk assessment procedure is carried out and noted in tables. Yet, this methodology is limited and it quickly becomes extremely difficult to map all requirements and dependencies for all possible attack vectors. As a solution, Esche et al. introduced the attack probability tree that visualizes in a very compact manner the attack vectors and make it easy to deduce a probable attack path. Furthermore, it enables to derive the attacker motivation. In the next section a short theoretical introduction of the AtPT will be given and

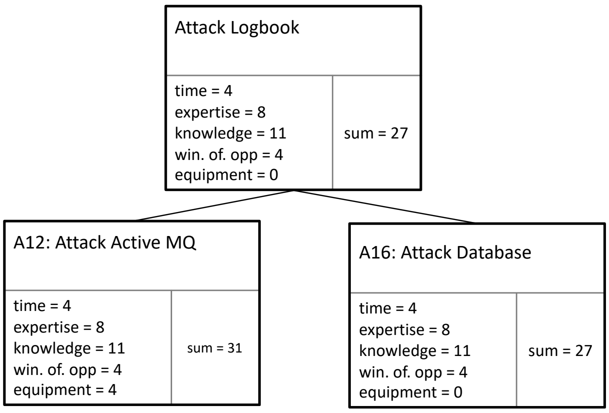

Fig. 3. AtPT for threat intention B3. View: root node and two attack vectors.

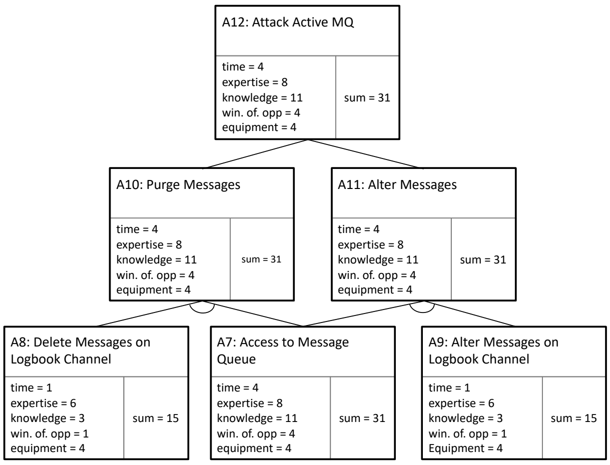

Fig. 4. AtPT for threat intention B3. View: Subtree of attack vector Active Message Queue.

subsequently applied to B3 until B5.

C. Attack Probability Tree

Esche et al. introduced attack probability trees (AtPT) as an extension of attack trees by Mauw and Oostdijk [12] to tackle two main objectives: developing a method to standardize the deduction of attack vectors and to efficiently visualize the interdependencies of attack vectors in order to easily derive attacker motivation and as a result the most likely attacker path. [6]. Additionally, each node embodies features with its own score, such as time, expertise, knowledge, window of opportunity and equipment, that have been previously collected in tables. Furthermore, the logical relationship between parent and child attacks are visualized and attack nodes are linked either by an AND- or OR-statement.

Information enter the tree via the leaves, so that parent nodes' and finally the root's attributes can be calculated from the bottom to the top. The rules for both statements and each attribute/point score are extensively described in [6]. Briefly summarized: for AND-statements, the *maximum* for each attribute chosen; for OR-statements, the *smaller* sum score indicates the threat to select. A great side-effect of AtPTs is the reduction of required time for revaluation of individual attacks, because of the possibility of reusing attack nodes, that are common among different attacks without recalculating attributes.

The following subsections use the AtPT approach for risk assessment of the cloud reference architecture. Nevertheless, the corresponding tables were generated, as introduced in the previous sections. However, due to space constrains, they are not published here.

D. Evidence of an Intervention

In this scenario an attacker prevents legally relevant events from being registered in the logbook. The threat intention is to attack the availability of the evidence of an intervention. In case of a successful manipulation, the user cannot present all relevant logbook entries that market surveillance demands.

In this paper, only the AtPT for a logbook attack is presented. Another attack scenario with the same attack attributes is evaluated for the storage service of the instrument, with a similar-looking AtPT. Because of the complexity of the attack, the AtPT is divided into four subtrees (see Figures 3-6), that will be described in the next paragraphs.

An AtPT is read from the root to the leaves. For attacking the logbook, two possibilities are available. Either the attacker aims for the active message queue (Active MQ) or for the database of the logbook service (see Figure 3). Since these two attack vectors are alternatives, they are linked by an OR-connection. If the two vectors would be needed to be executed together, they would be linked by an AND-connection graphically expressed by an arc.

When attacking the Active MQ (A12), an attacker could either purge messages (A10) or alter message (A11) on the logbook channel. For both actions, access to the message queue is required (A7) with the combination of deleting a message (A8) or changing a message (A9) on the logbook channel represented by an arc below the linked nodes (see Figure 4).

The actual scores in Figure 4 are calculated from the bottom to the top, for example, attack vector A10 consists of nodes A8 and A7. Since the latter two nodes are linked by an AND-statement the greater value is put across to A10. The time to purge a message takes less than a month (4) and stems from A7 accessing the message queue. Furthermore, it is required to be an expert in several areas (8), to have critical knowledge of the system (8) and the window opportunity is moderate (4). These attributes stem also from A7. However, the equipment to purge messages on the active MQ is specialized (4), since the software is an expert tool written in python without a graphical user interface. Yet it is indeed publicly available.

Now one could argue, that using a specialized software and obtaining access to the message queue needs less time than proposed here. However, the whole AtPT does not end with obtaining access to the message queue (A7), but rather continuous and becomes more detailed in how the access could be obtained in a malicious way.

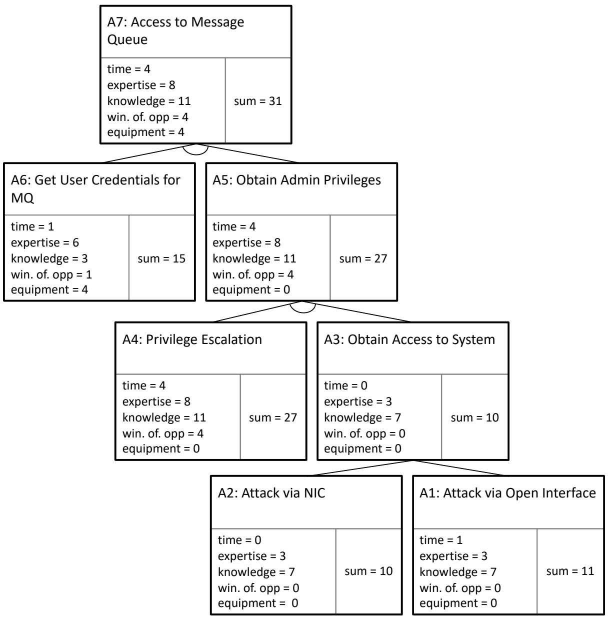

Fig. 5. AtPT for threat intention B3. View: Subtree of attack vector Access to Message Queue.

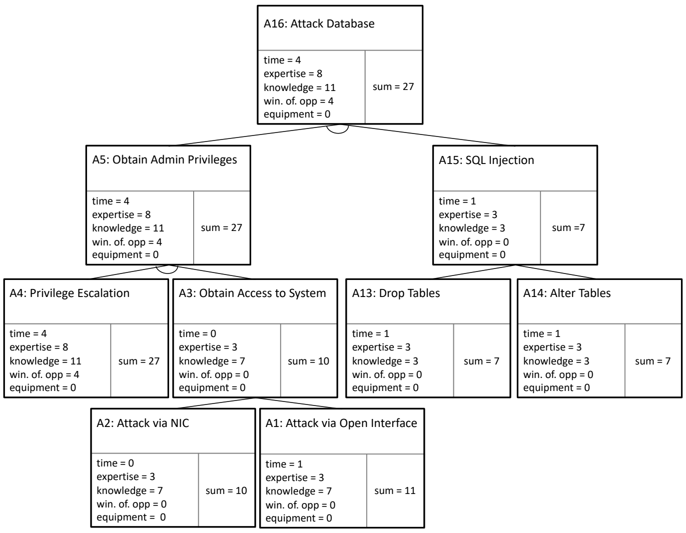

Fig. 6. AtPT for threat intention B3. View: Subtree of attack vector Attack Database.

In Figure 5 an exemplary attacking path is detailed. Node A7 consists of obtaining administrator privileges in the virtual machine (A5), that runs the active MQ or is at least in the same subnet. With these new privileges the specialized software can be executed, which triggers node A6 to get the credentials for the message queue.

To get hold of the user credentials, less than a week (1) is estimated. An expert level (6) and restricted knowledge of the measuring system is required. The window of opportunity for an inside attacker is easy (1) even so specialized software (4) is needed. Node A6 holds a total sum of 15 points which would be considered as an enhanced basic resistance level. However, A6 is to be evaluated in conjunction with A5 through the AND-connection.

The attack vector A5 depends again on a privilege escalation through exploiting Common Vulnerabilities and Exposures (CVE) of the underlying system (A4) and obtaining access to the virtual machine (A3). To accomplish a privilege escalation, the attack is assessed with less than a month (4), expertise on more than one field (8), critical system knowledge and moderate window of opportunity (4). Further, no special equipment (0) is expected. A privilege escalation is considered as a difficult endeavor with 27 points in total that translate to a high resilience. This corresponds again to a probability score of 1 with an impact score of 5 and results into a risk of 1.

Obtaining access to a virtual machine and therewith to the distributed measuring system (A3) is possible in two ways that are alternatives (OR-connection). Either the system is penetrated through a network interface card (NIC) (A2) or via an open physical interface (A1), such as a USB port. Considering the fact that an inside attacker with administrator privileges is assumed, that logs remotely into the measuring system for maintenance reasons, this attack is achievable in less than a day (0). To clarify, it is assumed that the inside attacker does not automatically have administrator privileges on the remote machine, but as an employee of the manufacturer. Furthermore, to login remotely requires only a proficient expertise and sensitive system knowledge (7). The window of opportunity is negligible (0), since this can belong to the attacker's daily routine. No special equipment is needed (0). The TOE resistance is basic (10 points in total).

The attack via an open interface (A1) differs from A2 only in the time attribute. It is assumed that the attacker has to physically approach the hardware to carry out the attack. That takes additional time (less than a week (1)) and is more inconvenient than opening a SSH-shell from the desktop pc in the office.

To sum up, the attack path just described consists of A2, A3, A4, A5, A6, A7 then a decision has to be made if the messages should be altered or deleted. However, in terms of likelihood the nodes do not differ, but practically spoken deletion is often easier. The path would continue via A8, A10.

To completely describe the AtPT for compromising the evidence of an intervention via a logbook attack, the alternative path via the database attack vector (A16) has to be described, as shown in Figure 6. For attacking the database, administrator privileges (A5) are needed combined with an attack against the database such as SQL injection (A15) or via command line interface (CLI). The path down to the leaves for A5 is already described in the previous paragraphs. Its TOE resistance depends on leaf A4, that describes the privilege escalation via a CVE. Attack Vector A15 is divided into dropping tables (A13) or modifying tables (A14).

The scores for A13, A14 are equal and subsequently A15 is identical as well. For both attacks, less than a day is assumed, only a proficient expertise level (3) is needed, no special equipment (0) is required and the window of opportunity

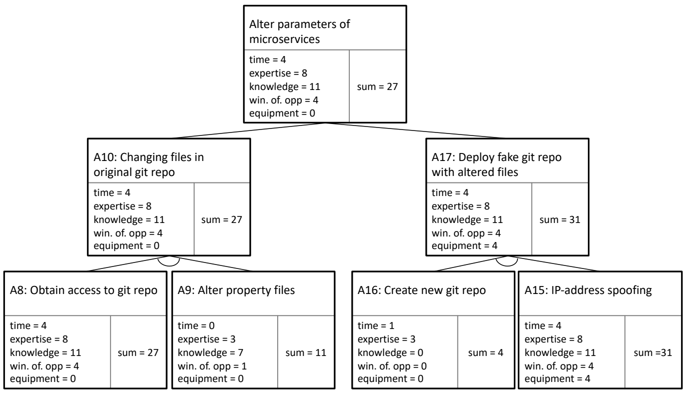

Fig. 7. AtPT for threat intention B4. View: Root, Alter parameters of microservices.

is unlimited (0). In total, the database attacks combine to 7 points, which translate to no resistance at all (no rating). However, since A5 and A15 are connected via an AND-statement the parent node A16 receives the TOE resistance high, since the attacks depend on the privilege escalation to be carried out.

The most likely attack path would be via the database, since no special software is needed, thus less time is required for learning and incorporating the software. To compromise a database, no new software has to be deployed so that the effort on the attacker side is less than attacking the message queue, especially if the intention is to just compromise the integrity of the measuring instrument.

E. *Integrity of Parameters*

Threat intention B4 aims for harming legally relevant software parameters to violate the security properties integrity and authenticity. In the following paragraphs the presented scenario offers an attacker to alter persistent saved parameters of the logbook service by attacking the configuration service. Two possible attack scenarios are presented via an AtPT. The tree is compartmentalized into several subtrees, because of its size (see Figures 7-9). As already pointed out, the subtree consisting of the node A1-A5 could be reused for several attack scenarios without revaluation. Due to space constraints it was renounced to map the whole subtree of A5 downwards in Figure 8. A complete subtree can be seen in Figure 5.

It is proposed that the attacker changes microservice property files in the original git repository to attack the microservice architecture (A10) and provides, for example, false message queue groups. That could lead to loss of messages in the legal relevant logbook. Aiming for the configuration basis can cause fundamental harm and chaos to the whole system.

To be able to carry out attack vector A10, it is assumed that the attacker has to obtain access to the original git repository (A8) and is able to alter the property files (A9). Nodes A8 and A9 are linked via an AND-connection to A10 (see Figure 7).

In order to obtain access to the git repository (A8), a SSH-key has to be created (A7) and placed into the specific folder for the git repository to be evaluated (A6). A7 and A6 are linked via an AND-statement to A8 (see Figure 8). Attack vector A6 is linked to subtree A5, that describes accomplishment of obtaining administrator privileges (a complete subtree is shown in Figure 5).

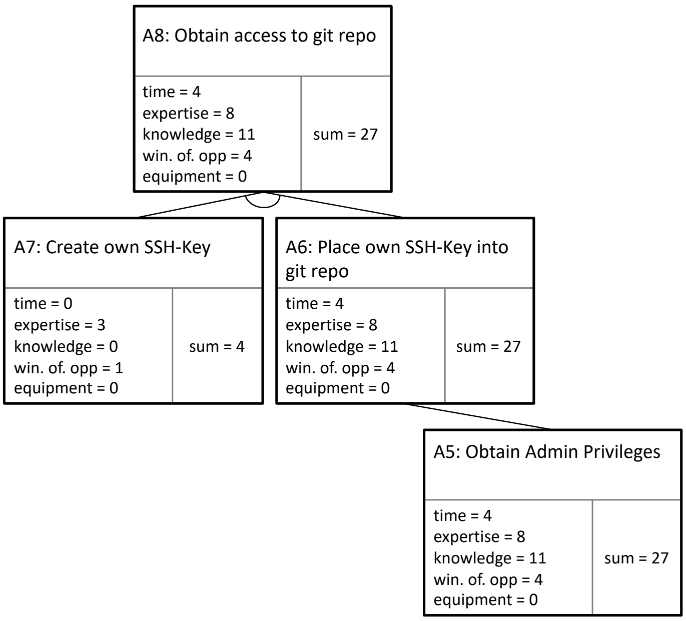

Fig. 8. AtPT for threat intention B4. View: Subtree of attack vector A8.

The score for subtree A5 has been described in the previous section, so that the evaluation starts with A6. All of the attributes stem from A5 and the difficulties to obtain administrator privileges, which are prerequisites for A6. To create an SSH-key (A7) takes less than a day (0) with proficient expertise (3). How to do this is public knowledge (0) and tutorials are easily to find on the Internet. Furthermore, assuming that an inside attacker is already in the system, the window of opportunity is easy to accomplish (1) and also no special equipment (0) is necessary. This yields a total sum of 4 points and a TOE resistance with no rating.

Attack vector A8 receives the point score from A6, since because of the AND-connection only the maximum of both attributes will be passed upwards. That leads to a total sum of 27 points for obtaining access to the original git repository and implies high resilience.

Altering property files can be done in less than a day (0), with only proficient expertise (3), an easy window of opportunity (1) and with any text editor (equipment = 0). That totals in 11 points and matches basic resilience for this attack.

A10 receives the attributes in total from A8 and thus defines the total score of 27 points and a high resilience for this attack path (see Figure 7).

The alternative attack vector for B4 is to deploy a fake git repository with already altered property files (A17). This attack vector splits into first creating and deploying a new git repository (A7) and then tricking the system into trusting and pulling the files from the fake git repository via IP spoofing (A15). The spoofing attack itself is subdivided into carrying

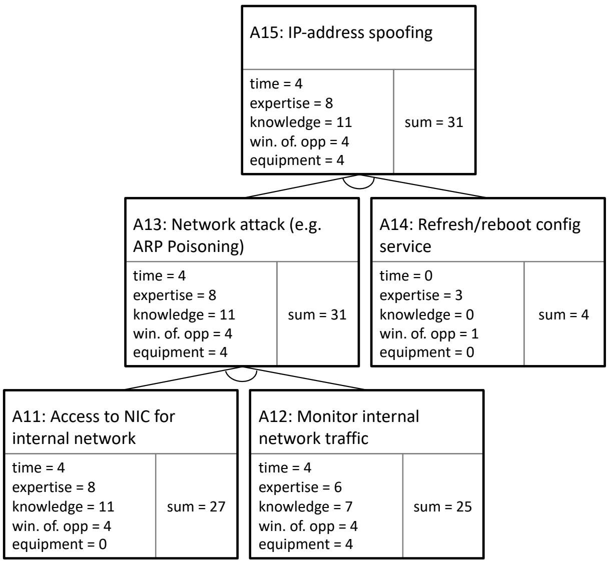

Fig. 9. AtPT for threat intention B4. View: Subtree of attack vector IP-address spoofing.

out a network attack such as Address-Resolution-Protocol-Poisoning (ARP-poisoning) (A13), in order to replace the IP address and then rebooting the configuration service (A14). To be able to carry out a network attack, the attacker is assumed to have full access to the internal network of the distributed measuring system (A11). To monitor the internal network traffic (A12), the attack vector A11 is necessary (see Figure 9). Once again subtree A5 is required to successfully implement A11.

The score of gaining full access to the internal network interface card (NIC) and thus to the internal network (A11) is inherited from A5 and the struggle of obtaining administrator privileges. To gain a full picture of the structure of the internal network with its services (A12) takes less than a month (4) with an assumed expertise in networking (6) and a sensitive knowledge of the measuring system (7). A moderate window of opportunity (4) is predicted, because it is difficult to explore a supervised internal network undetected. Furthermore, specialized software is needed to monitor network traffic (4). This yields in total 25 points and maps to a high resistance to attacks with probability score of 1 and a risk of 1.

Carrying out network attacks, such as ARP-poisoning (A13), requires less than a month (4) for experts on several fields (8) with sensitive knowledge of the system (11), a moderate window of opportunity (4) and specialized software (4). For most network attacks, it is not necessary any more to write specialized software. There exists publicly available grey software, that can be used to detect vulnerabilities or can be misused to attack computer systems. This attack vector combines to 31 points and a high resistance factor. From here on, no significant changes to the resilience are contributed until the final attack vector A17. Minor actions are required to finally deploy a fake git repository but both acquire only 4 points in total with a negligible threat resistance (see Figure 7).

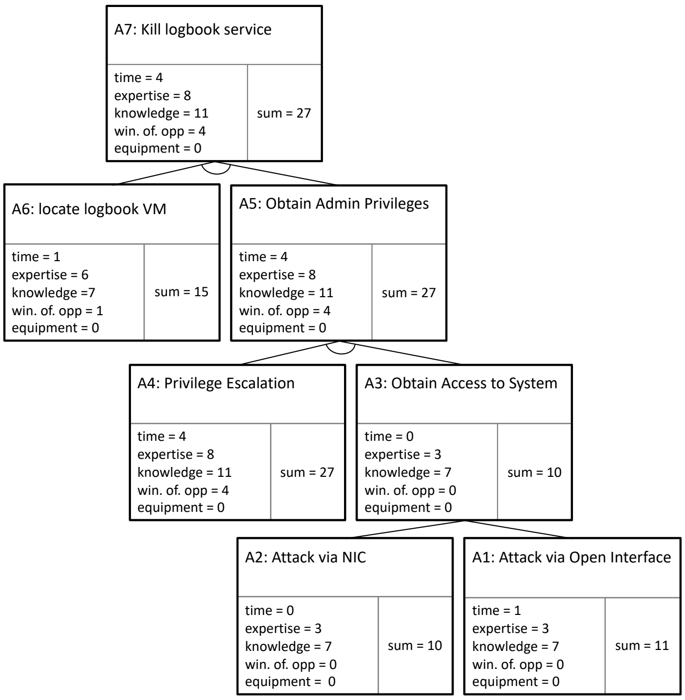

Fig. 10. AtPT for threat intention B5 to violate the availability security property.

The most probable attack path will be via A10, changing the properties files in the original repository, since it is the least complex one and without the hassle of deploying software and monitoring traffic etc. This is also reflected in the total score of 27 against 31 points.

F. Availability of Service

Threat intention B5 targets the availability of a legally relevant logbook service. In Figure 10 the complete AtPT is illustrated with the already introduced subtree A5, that enables access to the measuring system and comes along with an escalation of privileges. This subtree in conjunction with the localization of the logbook service's virtual machine (A6) enables the final attack vector that kills the logbook service (A7).

The final score stems from the difficulty to gain access to the system and to elevate the privilege level (subtree A5), which totals 27 points. With an associated risk level of 1 and a probability score of 1. Locating the logbook's virtual machine is with 15 points in total an enhanced basic TOE resistance level, but has no significant influence on the final score of killing the logbook service (A7).

G. Effect of Attacker Motivation

Esche et al. described in [10] possibilities to represent attacker motivation during risk assessment. The presented AtPTs are created for a highly motivated attacker. In order to reconsider these trees with a low or medium motivated attacker, the expertise and equipment score have to be replaced

TABLE VII
MAPPING OF EXPERTISE AND MOTIVATION LEVEL ACCORDING TO [10]

Expertise	Score	Motivation	Score
Layman	0	no motivation	9
Proficient	3	low	6
Expert	6	moderate	3
Multiple Expert	8	high	0

with a higher motivation score according to Table VII if they are originally smaller. This will result in a decreased probability score for a lower motivation and vice versa for a highly motivated attacker. It is noteworthy, that the likeliest attacker path can shift, when the motivation is adjusted.

H. Suitable Countermeasures

To find the best suitable place for countermeasures in an AtPT, it is recommended to locate an inverted subtree for mitigating attack vectors and increasing the impact of applied countermeasures. An inverted tree is usually any leaf that is connected to more than one node of the previous level. Subsequently, the size of an inverted tree matters, since the greater it is, more parent nodes are impacted. In the trees for B3 and B4, A7 and A16 depend on A5 as well as A6 and A11 depend on A5. Subtree A5 is of general importance, because it describes the unauthorized access to the measuring system and privilege escalation. A countermeasure specifically tailored for A5 will exacerbate to obtain administrator rights. This node will have the biggest impact on all three threat scenarios from B3-B5.

A suitable countermeasure is to strengthen the access rights and to enforce a least privilege policy. For example, one could implement Security Enhance Linux (SELinux) for virtual machines (VM), that provides a mandatory access control system and security policies. Instead of using a standard Linux, the kernel extension SELinux provides by default a least privilege policy that denies everything except if it is specifically allowed by access policies (enforcing mode). All violations against these rules are logged and an alarm can be triggered. To obtain administrator privileges by an escalation of access rights would need significantly more time (less than 2 months (7)) with SELinux in place. Furthermore, if the attacker is able to bypass SELinux via switching form enforcing to permissive mode it needs to be done on every VM with a bespoke software (7). However, rolling out SELinux to the measuring system would mean a lot of configuration overhead, but it would elevate the security score by 10 points to 37. This security enhancement would propagate via the inverted tree to the top of each AtPT.

IV. SUMMARY

In this paper, a secure cloud reference architecture for distributed measuring instruments under legal control was presented and subjected to a especially tailored risk assessment method for software in Legal Metrology. After formally introducing the risk analysis, five threats for the reference architecture were described and evaluated extensively. The first two threats were assessed using the traditional method via tables. However, this approach seemed infeasible for more complex threats. Therefore, the Attack Probability Tree (AtPT), that eases the handling of more complex attacks, was introduced and applied. It was shown that adequate protection of the essential requirements formulated by the MID is provided by the secure cloud reference architecture. Therefore, the architecture is qualified to be implemented in measuring systems under legal control.

The detailed analysis of the threat intentions using AtPTs revealed for all formulated threats and attacked security properties a high resilience factor. Nevertheless, through the inverted subtree method for AtPTs the optimal entry point for countermeasures was identified. The implementation of countermeasures reduced the risk to the level provided by physical sealing and increases the resilience to attacks.

Future work will focus on different attacker motivation and therewith diverse attack paths. Furthermore, the formalization of creating AtPTs has to be optimized and standardized.

ACKNOWLEDGMENT

This Paper would not exist without the help and fruitful discussions with our colleague Federico Grasso Toro.

REFERENCES

[1] European Parliament and Council, “Directive 2014/32/EU of the European Parliament and of the Council,” *Official Journal of the European Union*, 2014.

[2] A. Oppermann, J.-P. Seifert, and F. Thiel, “Secure cloud reference architectures for measuring instruments under legal control.” in *CLOSER (1)*, 2016, pp. 289–294.

[3] “WELMEC 7.2 Software Guide,” *WELMEC European cooperation in legal metrology, Welmec Secretariat, Delft, Standard*, 2015.

[4] A. Oppermann, A. Yurchenko, M. Esche, and J.-P. Seifert, “Secure cloud computing: Multithreaded fully homomorphic encryption for legal metrology,” in *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*. Springer, 2017, pp. 35–54.

[5] M. Esche and F. Thiel, “Software risk assessment for measuring instruments in legal metrology,” in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1113–1123.

[6] M. Esche, F. G. Toro, and F. Thiel, “Representation of attacker motivation in software risk assessment using attack probability trees,” *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on (pp. 763-771). IEEE.*, 2017.

[7] C. Gentry *et al.*, “Fully homomorphic encryption using ideal lattices.” in *STOC*, vol. 9, 2009, pp. 169–178.

[8] ISO27005:2011(e), “Information technology - security techniques - information security risk management.” *International Organisation for Standardisation, Geneva, CH*, vol. Standard, Jun. 2011.

[9] “Welmec 5.3 Risk Assessment Guide for Market Surveillance: Weigh and Measuring Instrument,” *WELMEC European cooperation in legal metrology, WELMEC Secretariat, Ljubljana*, May 2011.

[10] M. Esche and F. Thiel, “Incorporating a measure for attacker motivation into software risk assessment for measuring instruments in legal metrology,” *18. GMA/ITG-Fachtagung Sensoren und Messsysteme 2016,Nürnberg, Germany*, vol. 1, no. 1, pp. 735–742, Mai 2016.

[11] ISO/IEC18045:2012, “Common Methodology for Information Technology Security Evaluation,” *International Organisation for Standardisation, Geneva, CH*, Sep. 2012.

[12] S. Mauw and M. Oostdijk, “Foundations of attack trees,” in *International Conference on Information Security and Cryptology*. Springer, 2005, pp. 186–198.