DOI: 10.15439/2015F127
ACSIS, Vol. 5

Software Risk Assessment for Measuring Instruments in Legal Metrology

Marko Esche, Florian Thiel
Physikalisch-Technische Bundesanstalt
Abbestr. 2-12
10587 Berlin, Germany
Email: {marko.esche, florian.thiel}@ptb.de

***Abstract*—In Europe, measuring instruments subject to legal control are responsible for an annual turnover of 500 billion Euros and need to pass a conformity assessment with respect to European directives or national legislation before they can be used. Today, measuring instruments are frequently integrated into open networks and even branch into the areas of cloud computing and Internet of Things. Since software is one of the key components of such devices, Germany's national metrology institute, the Physikalisch-Technische Bundesantalt, is developing a method to assess the risks and evaluate current threats associated with software. The method uses the structure of and combines elements from the international ISO/IEC standards 27005 and 15408. It could be helpful for conformity assessment bodies and industry alike and supports the comparability of risk assessment results. Despite its focus on legal metrology, the method is applicable to other areas where software risk assessment is required, too.**

I. Introduction

CERTAIN types of measuring instruments, like gas meters, taximeters, fuel pumps and grain moisture meters are subject to legal control in the European Union. Before making them available on the market, such measuring instruments have to undergo a conformity assessment according to the Measurement Instruments Directive (MID) 2014/32/EU [1]. The entire area of measuring instruments even including individual measurements regulated by either national or European legislation is referred to as legal metrology. According to estimations, about four to six percent of the gross national income in European countries is accounted for by legal metrology. In Germany alone, 130 million of such instruments are installed. These are responsible for economic transactions worth roughly 157 billion Euros each year. For a more detailed description of the role of legal metrology in general see [2].

In most cases, the conformity assessment is performed by so-called Notified Bodies, which have proven that they have at their disposal ”(a) personnel with technical knowledge and sufficient and appropriate experience to perform the conformity assessment tasks, (b) descriptions of procedures in accordance with which conformity assessment is carried out, ensuring the transparency and the ability of reproduction of those procedures“ [1]. One such Notified Body that performs conformity assessments is the Physikalisch-Technische Bundesanstalt (PTB), Germany's national metrology institute. The assessment itself is conducted according to a combination of modules (A to H1) which encompass different roles for manufacturers and Notified Bodies [1]. For most of these modules, a new general requirement has been introduced in 2014 concerning the submitted manufacturer's documentation. It states, ”The documentation shall make it possible to assess the instrument's conformity to the relevant requirements, and shall include an adequate analysis and assessment of the risk(s).“ Such a risk assessment does not only need to cover the physical measuring instrument itself but also the metrologically relevant software running on it. In this context, harmonization between European Notified Bodies obviously becomes necessary to ensure fair and comparable software risk assessment within the common trade zone. In this paper, an approach for software risk assessment is presented that

- makes use of established international standards as far as possible and
- identifies risks with reproducible numeric values to better ensure comparability between evaluation results.

The remainder of the paper is structured as follows: In Section II, a literature overview covering other methods in the field of software risk assessment is provided. In order to obtain reproducible analysis results, a clear definition of assets and threats to these assets is required. A derivation of such assets from the requirements of the MID is, therefore, provided in Section III. An algorithmic description of the risk assessment approach proposed here, may be found in Section IV. Afterwards, the new approach is compared with other existing methods based on two real-world examples in Section V. Section VI summarizes the paper and provides an overview of planned future work.

II. Overview of existing methods

Before giving a list of reference approaches to software risk assessment, it is necessary to clearly identify what kind of risk assessment is required in the context of the MID. The directive establishes a common baseline by listing a number of essential requirements which all measuring instruments have to fulfill. Since the most important target of the MID is to ensure free and fair trade as well as to protect the consumer, these essential requirements are mainly targeted at protecting measuring results from accidental and intentional manipulation and to make both correct measuring results and detected manipulations traceable. Further details may be found in Section III. In this context, the term *risk* can be seen as the

978-83-60810-66-8/$25.00©2015, IEEE

product of the probability that the essential requirements are no longer met and the legal impact resulting from such a breach of the MID. It is important to note, that no financial loss needs to be associated with the risk, instead, the sole basis for the analysis are the essential requirements.

A. ISO/IEC 27005

Probably most important to mention is the ISO/IEC 27000 family of standards which covers all aspects of an information security management system (ISMS). According to ISO/IEC 27005 [3], ”risk is a combination of the consequences that would follow from the occurrence of an unwanted event and the likelihood of the occurrence of the event.“ Thus, three different components are needed to calculate risk, namely

- a list of unwanted events,
- consequences resultig from such events,
- and the likelihood of occurence of individual events.

In order to derive all three components [3] gives details on generalized procedures to conduct risk assessment. The standard places risk assessment in a logical chain comprising context establishment, risk assessment, and risk treatment, where risk assessment consists of risk identification, risk estimation, and risk evaluation.

During the risk identification phase, assets are to be identified first. These are derived from a ”list of constituents with owners, location, function“, resulting in a list of assets to be managed. Afterwards, for each possible asset, threats are collected based on information from reviewed incidents, accounts from asset owners, and possibly external threat catalogs. These threats correspond to the ”unwanted events“ mentioned above. The next step consists of identifying existing risk control mechanisms which could, for instance, be determined from the provided documentation. Risk identification is completed by an identification of vulnerabilities which can be used to implement certain threats. In this context, a vulnerability can only cause harm if it can be used to realize a threat. Equivalently, threats without a corresponding exploitable vulnerability do not pose a risk.

The next part of ISO/IEC 27005, concerned with risk estimation, is likely the most relevant in the context of this paper. The standard considers both qualitative and quantitative approaches to calculate risk probability, where the quantitative approach ”uses a scale with numerical values for both consequences and likelihood, using data from a variety of sources.“ Such numerical values are a prerequisite for ensuring comparability among risk assessment results for different products conducted by different examiners. To derive at actual probabilities, ISO/IEC 27005 first assigns certain impacts or consequences to incidents that could result from realized threats by means of exploited vulnerabilities. Possible examples of impacts include loss of confidentiality of certain assets as well as a breach of asset integrity. In a final step, the probability, with which a threat is realized, is estimated. Important factors, in this context, are the frequency at which certain threats occur in real life and the difficulty of exploiting a vulnerability. For intentional exploitation of threats, ISO/IEC 27005 suggests a valuation of motivation and capabilities, resources available to the attackers as well as the perception of individual vulnerabilities. This approach will later be revisited in Section IV where certain aspects of ISO/IEC 27005 are reflected in the risk assessment approach presented here. Nevertheless, the standard does not prescribe a reference model to calculate individual numeric threat probabilities. The choice of such a model is instead left up to the user of the standard. One possible method to calculate risks quantitatively may, for instance, be found in [4], where the author proposes to define risks as probability functions that describe the likely gains or losses obtained from security incidents. The final components of risk assessment according to ISO/IEC 27005 are evaluation of the risk level and risk evaluation. The aim of these steps is to prioritize the identified risks according to the predetermined evaluation criteria.

B. WELMEC Guide 5.3

In order to harmonize the work of Notified Bodies in Europe, a number of non-mandatory guides have been established within the European Legal Metrology Cooperation (WELMEC). The guide 5.3 ”Risk Assessment Guide for Market Surveillance: Weigh and Measuring Instruments“ deals with risk assessment from a market surveillance perspective and originates from Regulation 765/2008/EC [5]. Its main goal ”is to understand the impact the instrument will have on the end user/consumer“ [6]. The guide establishes a list of evaluation criteria which should help ”the market surveillance authority to define priorities and to determine the choice of strategies to achieve their goals.“ Since the guide is solely targeted at market surveillance authorities, the expected impact is not clearly defined but rather encompasses everything from ”economic implications, public health, consumer confidence [to] legal issues“ [6]. Even if only legal issues are considered, the spectrum of the guide is still too broad to objectively evaluate software in measuring instruments. Instead, the guide provides a clear rule to eventually calculate the risk associated with non-compliance but does not provide means for calculating individual threat probabilities.

C. Van Deursen et al. ”Source-Based Software Risk Assessment“

One approach, that does not have this shortcoming, is the one by van Deursen et al. in [7]. There, risk assessment is defined as ”an independent assessment of the risks involved in building, operating or maintaining a software system.“ The method then calculates the risk based on so-called primary and secondary facts, where primary facts are data acquired through automatic source code analysis and secondary facts are obtained using user questionnaires. The primary facts are mainly needed to identify subsystems that show features not usually found in software systems. After the primary facts have been used to validate the secondary data, a final result can be computed. This method could readily be adapted for the use in legal metrology. However, source code is usually not a required part of the documentation for MID conformity assessment.

D. Foo et al. ”Software Risk Assessment Model“

Another method to objectively evaluate and compare risks associated with software was presented by Foo et al. in [8]. There the basic abstraction technique used by the authors is a software risk assessment model (SRAM) which is constructed based on an extensive questionnaire to be answered by the risk assessor of a software product. Within [8] the term risk is defined as ”factors that may cause late delivery, cost overrun or low quality of a software product.“ Nevertheless, the authors list the productive level of the staff, flexibility of the delivery schedule and most importantly complexity of software as having a significant impact on the risk evaluation. In the context of a MID conformity assessment, the first two sources of information are of no importance since the MID is not concerned with business processes. The complexity of the software can again not be used due to lack of available information. Moreover, the risk assessor required by the SRAM approach needs to have access to resources such as source code and error statistics that are not available to a MID evaluator. For comparison, a description of a risk assessment approach for measuring instruments with a similar scope covering the entire life cycle of a device can be found in [9].

E. Sadiq et al. ”Software Risk Assessment and Evaluation Process (SRAEP) using Model Based Approach“

In [10] a different software risk assessment method was proposed that is also model-driven. Sadiq et al. therein describe the Software risk assessment and evaluation process (SRAEP) which is based on the software risk assessment and evaluation model (SRAEM). Their approach is targeted at highlighting threats to the success of a software project rather than threats to a finished software product. Nevertheless, a number of useful formalized steps are included in their method which shall be reused later. For this reason, the basic steps of the SRAEP will be revisited here.

According to the authors the motivation for using a model-based assessment strategy is two-fold:

- With the help of a model, precise descriptions of the target system, its context, and security features can be formulated. These are prerequisites for performing risk assessments.
- The modeling technology facilitates a more precise documentation of risk assessment results and of the assumptions on which their validity depends. This is expected to reduce maintenance costs by increasing the possibilities of reuse of the documentation.

The SRAEP itself can be divided into two steps: the identification of a context for the analysis and the identification of risks themselves. Sadiq et al. here split the context identification into an identification of areas of concern, a description and evaluation of assets, and, finally, an identification of security requirements. These three steps will be used again during asset derivation, see Section III, and in the risk assessment method that is proposed here as described in Section and IV. Before beginning with the risk analysis, the SRAEP requires an evaluator to acquire detailed knowledge of the analysis target. Based on this knowledge, all security issues related to software should be discussed making reference to common vulnerabilities or the results of tool-based vulnerability checks.

F. ISO/IEC 15408 (Common Criteria)

An international standard for software security that explicitly does not address risk assessment is ISO/IEC 15408 also known as the ”Common Criteria“ (CC) [11]. In the CC, a set of functional security requirements is defined, which can be used to describe both product requirements in the form of Protection Profiles and product specifications in the form of Security Targets. An implemented Security Target, i.e. a product to be tested, is referred to as a Target of Evaluation (TOE). The standard also provides a list of assurance components, a chosen subset of which is also included in said Protection Profiles and Security Targets. These assurance components are then used to validate the design, the development, and finally the completed IT product itself. In which manner the assurance components are to be checked is not described in the CC themselves but rather in the Common Evaluation Methodology (CEM) [12] which accompanies the CC. Two building blocks from the CC with details provided in the CEM are of special interest here: Firstly, each Security Target includes a Security Problem Definition in which assets to be protected are identified. The CC initially list primary assets that represent objects or information of a given value whose authenticity, integrity or availability are to be protected. Certain aspects of an IT product can also become assets themselves when they play an integral role in realizing security functionality. These are referred to as secondary assets. Both types of assets are examined and listed in the security problem definition. Afterwards, possible threat agents and adverse actions that could be executed on the assets are investigated and described in a semi-formal manner. The combination of threat agent, asset, and adverse action is referred to as a threat. This construct will here again be used since it facilitates the implementation of reproducible risk assessment results.

Secondly, one part of validating a Security Target consists of a so-called vulnerability analysis which is specified in the CC’s AVA_VAN class. The assurance components associated with this class allow an examiner to execute both white box and black box tests on the Security Target based on the knowledge acquired during the evaluation procedure. The vulnerability analysis uses a point score, where each adverse action to be executed on an asset is evaluated with respect to five different aspects ranging from the time required to the equipment needed to implement an attack, for details see Section IV. In each category mentioned a point score is determined. Based on the total sum of all points the TOE’s resilience to an attack is checked. This is done with the aid of matrix, details on which will also be provided in Section IV. More information concerning the general mechanisms of the vulnerability analysis will be given there as well.

G. ETSI TS 102 165-1

The European Telecommunications Standards Institute (ETSI) in an international non-profit organization that publishes industrial standards in the area of telecommunications systems. These are targeted at manufacturers of communication equipment and network operators. One of these standards is ETSI TS 102 165-1 ”Telecommunications and Internet converged Services and Protocols for Advanced Networking“ [13] (TISPAN), where in part 1 ”Method and proforma for Threat, Risk, Vulnerability Analysis“ are detailed. The so-called technical specification describes a risk assessment approach consisting of nine individual steps that comprise everything from a definition of the device to be examined (TOE) up to the establishment of risks and an identification of countermeasures. The method starts by defining clearly the boundaries of the TOE and by identifying its security functionalities using terminology from the common criteria. Afterwards, all assets are identified. In [13] these have to fall into one of the following categories: equipment, human assets or information stored. It will be shown in Section IV that this definition is to narrow for most applications outside the area of communication systems. Moreover, the standard does not describe a way to derive abstract assets resulting, for instance, from legal requirements. Next, possible ”attack interfaces“ are identified that a threat agent can use to implement a threat. In order to assess the likelihood of occurrence for an individual threat, elements from the CC’s AVA_VAN class are used as described earlier. Details on the method may be found in Section IV. According to [13], ”threats to a telecommunications system are fairly restricted and fall into a small set of easily identified operations.“ Consequentially, [13] only lists a very small number of possible threats namely interception, manipulation, repudiation, and denial of service. While well suited to the area of telecommunication networks, these are to limited for general-purpose IT devices. The same is true for the definition of threat agents where [13] only allows a very small number of different roles. Finally, impact in the context of TISPAN is defined as a function of the intensity of an attack. For measuring instruments, as discussed here, a different definition is needed which will be given in Section III. Nevertheless, the method has certain properties which are of use to the scenario discussed here:

- calculation of the probability of an attack based on the AVA_VAN class from the CC,
- evaluation of impact and attack likelihood based on simple numeric scores (1 to 3 points),
- extension of the AVA_VAN class to account for multiple attacks being executed simultaneously.

III. Formal derivation of security requirements from the Directive 2014/32/EU

Before beginning with the algorithmic description of the new risk assessment method for software itself, a specific set of assets and associated security properties for measuring instruments will be derived here. These will later be reused in the experimental evaluation. The derivation should be seen as an example on how to formalize legal or contractual requirements with respect to software. In application scenarios not related to the conformity assessment of measuring instruments, other assets such as human health or monetary values etc. would, of course, be used. The latest version of the MID lists several requirements relating to software as plain text, which will be formalized here.

A. Exemplary Asset Derivation

The actual procedure of defining security requirements based on legal specifications will be highlighted with an example: Annex I of the MID lists so-called essential requirements for measuring instruments that have to be fulfilled before putting them on the European market. As an example clause 8.4 will be used here. It reads, ”Measurement data, software that is critical for measurement characteristics and metrologically important parameters stored or transmitted shall be adequately protected against accidental or intentional corruption.“ [1, L 96/173]

The requirement specifically mentions three asset candidates, namely measurement data, software that is critical for measurement characteristics, and metrologically important parameters stored or transmitted. All three assets are required to be protected against accidental or intentional corruption. Firstly, this can be interpreted as a requirement for guaranteeing integrity of these assets. Secondly, however, an intentional replacement of a parameter set also represents a viable way to invalidate parameter integrity. Thus, authenticity of said assets also appears to be required. This is not specifically mentioned in the MID but is common understanding among Notified Bodies [14]. Consequentially, the assets measurement data, software critical for measurement characteristics, and metrological parameters are associated with the security properties of integrity and authenticity. Availability of the software, however, is not mandatory since an instrument with no running measurement software cannot produce false measuring results.

Another requirement related to software can be found in Annex I of the MID, clause 7.6. It states, ”When a measuring instrument has associated software which provides other functions besides the measuring function, the software that is critical for the metrological characteristics shall be identifiable and shall not be inadmissibly influenced by the associated software.“[1, L 96/173] Again, two assets are specifically mentioned. The first is the identification of the software. The second one is an inadmissible influence by other software which is not a physical object or an IT object itself but rather is a property of the software. In the language of the CC, prohibiting external influence on the software can be expressed by stating, that the inadmissible influence shall be unavailable. This again enables the use of a fixed scheme to describe security functionality by identifying dedicated security properties associated with an asset.

B. Complete List of Assets

An overview of all MID requirements for software in measuring instruments may be found in Table I. The two examples discussed here are listed there for completeness as well.

Apart from the measurement software itself, the assets to be protected include an identifier for the software, measurement, results and parameters that determine the behavior of the instrument. In addition, the presentation of the measurement result as well as the presentation of the identifier for the software have to meet special requirements. Details on how these requirements are usually fulfilled may be found in [14] where the paragraphs from the MID are translated into implementation specific requirements and into so-called acceptable (technical) solutions. Even though [14] is usually of great value for both software developers and software examiners, it will not be used here since the new risk assessment procedure (see Section IV) aims to be generic and independent from a limited number of established technical realizations.

IV. Algorithmic Description of the Risk Assessment Procedure

The risk assessment method described in this paper follows the structure defined in [3] and consists of three main parts, namely identification of assets, identification of attack vectors, and calculating the probability of occurrence for an individual attack. Each of these will now in turn be described. A flowchart that links all three parts and incorporates details for each step may be found in Figure 1.

A. Identification of Assets

As has been detailed in Section III, assets to be protected can be derived directly from the legal requirements for measuring instruments as laid down in the MID. This is in accordance with [3], which states that the risk evaluation process can take ”legal and regulatory requirements, and contractual obligations“ into account and should also consider the ”criticality of the information assets involved“. In addition to the asset definition, one or more attacker models are needed, see upper right corner of Figure 1. In the simplest case, a Notified Body will assume all market players to be untrustworthy with equal motivation to manipulate measurement results and measuring instruments. This includes manufacturers or distributors of such devices, users or maintainers, and customers. These only differ in their respective capabilities to implement an attack. Subsequently, the risk assessment procedure can use the market player with the most detailed knowledge and with the highest skills (normally manufacturer or user) as a representative model. The basic structure for both the attacker model and the formulation of adverse actions may be found in the Security Problem Definition as described in [11]. When examining an individual measuring instrument, it may make sense to individually differentiate between attackers/authenticated users with different levels of access. If the authentication data is available to any of the market operators mentioned, then the highest access rights granted to any of these will be allocated to the modeled attacker. The only entity to be considered trustworthy in this context is the market surveillance which may hold administrator level authentication data for certain devices.

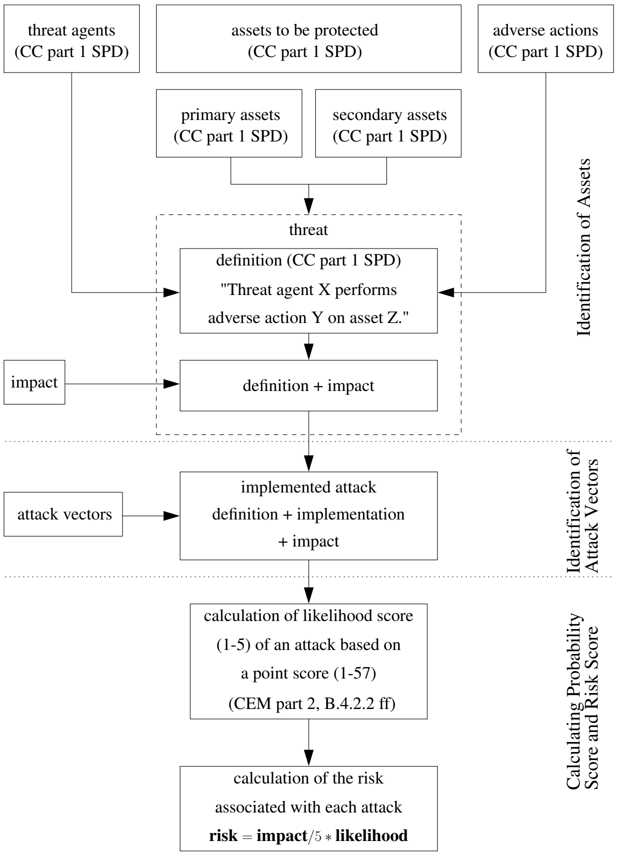
threat agents (CC part 1 SPD)
assets to be protected (CC part 1 SPD)
adverse actions (CC part 1 SPD)
primary assets (CC part 1 SPD)
secondary assets (CC part 1 SPD)
threat
definition (CC part 1 SPD) "Threat agent X performs adverse action Y on asset Z."
impact
definition + impact
Identification of Assets
attack vectors
implemented attack definition + implementation + impact
Identification of Attack Vectors
calculation of likelihood score (1-5) of an attack based on a point score (1-57) (CEM part 2, B.4.2.2 ff)
calculation of the risk associated with each attack
risk = impact/5 * likelihood
Calculating Probability Score and Risk Score

Fig. 1. Flowchart of the proposed risk assessment procedure. The notes on the right hand side indicate the division into the three main steps of the method.

Yet another action to be performed during the asset identification phase is the collection of certain adverse actions that can cause harm to the assets. Here again, a generic approach from the CC [11] can be used: Each modeled attacker may harm any of the identified assets by invalidating one or more of their security properties, i.e. availability, integrity or authenticity as applicable. Such an adverse action may then read for example, ”An attacker with the access rights of a local administrator manages to invalidate the availability of the proof of an intervention.“ The formulation of a complete threat may be found within the dashed box in Figure 1. The complete set of possible adverse actions derived from these basic combinatorics then only has to be checked for consistency and for possible duplicates. In a final step, the implemented attack, consisting of an adverse action and an attack vector, will be assigned an individual impact score between 1 and 5. Since all legal requirements are generally assumed to be equally important, the highest score (5) will usually be used. A smaller score will only be chosen if the attack only applies to a single measurement or can later be

TABLE I
Requirements from the MID [1] relating to software and their formalization as assets and security properties. The numbers in brackets after each asset (A1 to A10) represent there unique identifier.

Requirement in the MID [1] Annex I	Asset	Security Property
7.6 ”When a measuring instrument has associated software which provides other functions besides the measuring function, the software that is critical for the metrological characteristics shall be identifiable and shall not be inadmissibly influenced by the associated software.“	identification of the software (A9)	availability, integrity
	inadmissible influence on the software (A5)	unavailability
8.1 ”The metrological characteristics of a measuring instrument shall not be influenced in any inadmissible way by the connection to it of another device, by any feature of the connected device itself or by any remote device that communicates with the measuring instrument.“	inadmissible influence on the software (A5)	unavailability
8.3 ”Software identification shall be easily provided by the measuring instrument.“	presentation of the software identification (A10)	availability
8.3 ”Evidence of an intervention shall be available for a reasonable period of time.“	evidence of an intervention (A2)	availability, integrity
8.4 ”Measurement data, software that is critical for measurement characteristics and metrologically important parameters stored or transmitted shall be adequately protected against accidental or intentional corruption.“	measurement data (A3)	integrity, authenticity
	software critical for metrological characteristics (A1)	integrity, authenticity
	metrologically important parameters (A4)	integrity, authenticity
10.1 ”Indication of the result shall be by means of a display or hard copy.“	indication of the result (A6)	availability, integrity
10.2 ”The indication of any result shall be clear and unambiguous and accompanied by such marks and inscriptions necessary to inform the user of the significance of the result.“	marks and inscriptions (A7) accompanying the indication of a result	availability, integrity
11.1 ”A measuring instrument other than a utility measuring instrument shall record by a durable means the measurement result accompanied by information to identify the particular transaction, when: the measurement is non-repeatable; and the measuring instrument is normally intended for use in the absence of one of the trading parties.“	record of a measurement result (A8)	availability, integrity, authenticity

detected by market surveillance.

B. Identification of Attack Vectors

The second stage of the risk assessment phase is certainly the least formalized one. It begins with a careful study of the submitted manufacturer’s documentation of the measuring instrument to be examined. This process is shown in the middle section of Figure 1. The evaluator then collects possible attack vectors consisting of actions to be performed, that would enable an attacker to realize any of the previously identified threats. This represents a clear difference to the TISPAN method detailed in Section II. Some of these attack vectors may be as simple as trying a number of password combinations on a keypad in order to gain a higher level of access. Others may comprise complex cross-site-scripting (XSS) attacks in conjunction with the preparation of a root kit to take over a device in the field and subsequently install unapproved software on the device. One relatively simple attack vector from this category is the execution of a denial-of-service (DoS) attack on a measuring instrument connected to the Internet. In many cases, such an attack will lead to the generation of an arbitrary number of error messages written to an audit log which is subject to legal control. Should the log be restricted in size, an earlier intervention may no longer be traceable later if the log is flooded with a huge amount of automatically generated errors. This would be a direct breach of the essential requirements as laid down in Section III.

C. Calculating Probability Score and Risk Score

Once an adverse action with one or more associated attack vectors has been identified, it remains to calculate the likelihood with which the attack will actually be implemented, see lower part of Figure 1. A similar activity is in detail described in the vulnerability analysis (class AVA_VAN) in [12]. There, an evaluator estimates the resistance of an IT product (TOE in the language of the CC) to certain attacks. The evaluation in [12] is done based on five different scores describing the resources needed for the attack:

- 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 score for elapsed time represents the amount of time required by the selected attacker to implement the chosen attack. A score of 0 usually signifies work of less than a day. Required work of less than a week would give a point score of 1, whereas a score of 19 represents an estimated work period of more than half a year. Further examples will be given in Section V. A table with details on all five score criteria and additional explanations for the choice of the criteria may be found in [12, p. 429].

The logarithmic progression of the scores ensures that with every additional point assigned to an attack, it becomes significantly more complex to implement. This also means that the score is more easily reproducible since evaluators

TABLE II
CALCULATING THE RESISTANCE OF A TOE. THE THIRD COLUMN MAPPING THE TOE RESISTANCE LEVEL TO THE APPROPRIATE PROBABILITY SCORE IS NOT PART OF THE ORIGINAL TABLE AS GIVEN IN [12].

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

will certainly come up with estimated attack times of the same magnitude even if the actual times differ slightly. In the second category (expertise), between 0 and 8 points can be assigned, where 0 represents layman capabilities and 8 points are given when the attacker needs to be an expert in more than one field. The third category refers to the required knowledge concerning the attacked device. Again, a score of 0 is given if only publicly available knowledge is needed, such as information easily available from the web. 3 points represent restricted knowledge as might be found in the user documentation. The maximum of 11 points would stand for critical inside information only available to employees of the manufacturer. One very important score criterion is the window of opportunity available to the respective attacker. In the case of unlimited access, as would be usual for devices connected to the Internet, 0 points will be given, where 1 point signifies easy access. Should access, however, be difficult to obtain, 10 points can be assigned. In the ideal case, where access is impossible, no rating is done and the respective attack vector is removed from the list of candidates. At this point, there exists a simple way to include the motivation of threat agents into the score. Should an attacker lack the motivation to implement a threat even though he is able to realize the attack vector, the respective threat should be removed from the list.

When the assignment of score points has been done according to the five categories mentioned, the sum total of all scores is calculated. During a CC evaluation the so-called TOE resistance is then derived as indicated by Table II. A score between 10 and 13 points would, for instance, demonstrate a basic resistance to attacks, while a score above 24 indicates high resilience. In the context of the CC, the resistance to attacks would then be used to validate the selected evaluation assurance level (EAL). Here, however, the resistance rating is mapped to a probability score between 1 and 5, where 5 represents high probability of occurrence for an attack and 1 states that an attack is very unlikely to occur. The mapping of TOE resistance to probability score is also shown in Table II.

Calculating the risk associated with a threat subsequently consists of multiplying the impact score (between 1 and 5) for the given threat with the probability score of the most probable attack vector, that could realize the threat:

$$\text{risk score} = \frac{\text{impact score}}{5} \cdot \text{probability score} \qquad (1)$$

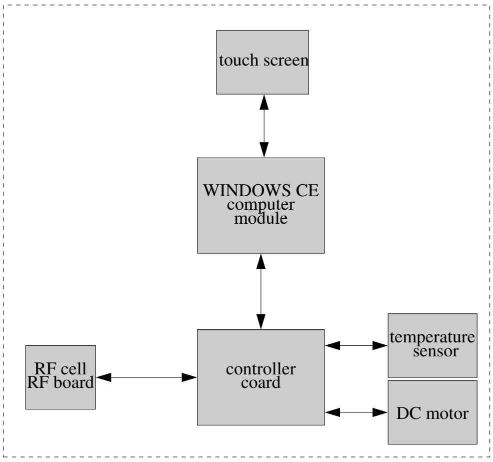

Fig. 2. High-level schematic for the grain moisture analyzer that was evaluated as an illustrative example. Parts lying physically inside the instrument are surrounded by the dotted line.

Dividing the impact score by 5 simply ensures, that the risk score is in the range between 1 and 5, too. As will be shown in the experimental evaluation, the risk score thus calculated can easily be used to rank risks associated with a single instrument or even to compare different instruments and their risks with one another.

V. Experimental Evaluation and Comparison with Other Methods

A. Grain Moisture Meter

The first measuring instrument that was examined during evaluation of the proposed software risk assessment method is a grain moisture analyzer. Such devices usually take a small sample of grain and calculate the moisture level within the sample by submitting it to infrared light and observing the absorbed wavelength spectrum. The relative moisture is economically important since it has a significant influence on the price of the grain. In this example, the measuring instrument is a stand-alone unit that is physically closed except for a touch screen, the sample inlet, as well as a serial and a USB port. As an operating system Windows CE is used. Certain types of grain can be selected via the touch screen, which is also used to start the measurement process and to show the current and past measurement results. In addition, the instrument contains a so-called audit log in which changes to both software and relevant measurement parameters are recorded. If an empty USB stick is plugged into the unit, it will write all available measurement results together with the respective date and time of the measurement to the stick. The measuring process can also be started via the serial port, which uses a proprietary protocol. Through this protocol measurement results can be read out, too. Access to the relevant system parameters and to the operating system are protected by a 6-digit password. A high-level schematic of the system may be found in Figure 2.

Based on the described system architecture and the documentation supplied by the manufacturer, a list of possible attack vectors can be compiled. The following list is just a short extract from the complete one and is used here for illustration purposes:

- **A_PASSWORD**: An attacker gains access to the administrator password by trying all 6-digit combinations.
- **A_SW_REPLACE**: An attacker retrieves the administrator password and replaces the legally relevant software.
- **A_INT_SERIAL**: An attacker exploits a vulnerability of the proprietary serial protocol and causes the instrument to malfunction.
- **A_INT_SERIAL_VALUE**: An attacker exploits a vulnerability of the proprietary serial protocol and manipulates a measurement value by interrupting the measurement.
- **A_INT_USB**: An attacker manages to install malicious code on the measuring instrument by disabling the USB-port's protection.

For each threat, as described above, an evaluator now has to go through the list of attack vectors and select those vectors that can realize the threat. In some cases, a combination of attack vectors might be necessary. An excerpt from the complete mapping between threat scenarios and attack vectors can be found in Table III. Each threat can then be rated individually. Both the point score for each aspect and its meaning are supplied in the table as well. The first threat (T1) here is a replacement of the legally relevant software by a local attacker. Since the only individual with adequate access to the measuring instrument is the operator of the device, he is also assumed to be the most likely attacker. This, of course, has an influence on the assigned point scores (see Table III). To realize T1, the attacker first has to retrieve the password for the operating system. In addition, a software needs to be written that mimics the behaviour of the approved one without raising suspicion from customers. The development of such a software is deemed to be very complex, giving it a time rating of more than half a year with an associated point score of 19. In addition, the attacker needs to be an expert in the area of software development or needs to have access to somebody who has such skills. The expertise score is therefore set to 6. Restricted knowledge of the device such as a description of the system behavior and its components is also required. Finally, the owner of the measuring instrument has unlimited access to it and to write software no special equipment aside from an off-the-shelf PC is required. The sum score for this scenario is 29, which even in the context of the CC is so high, that virtually no attack likelihood remains. Table II subsequently assigns the lowest probability score for this threat.

Subsequently, threats T2 to T5 are rated in the same manner. Threat T5 has a probability score of 2 which is identical to those of threats T3 and T4. Nevertheless, the associated risk score is only 1 since T5 will only affect one single measurement result and thus has a fairly low impact. For all other threats, there is no difference between risk and probability score since they were classified as having the highest possible impact score of 5.

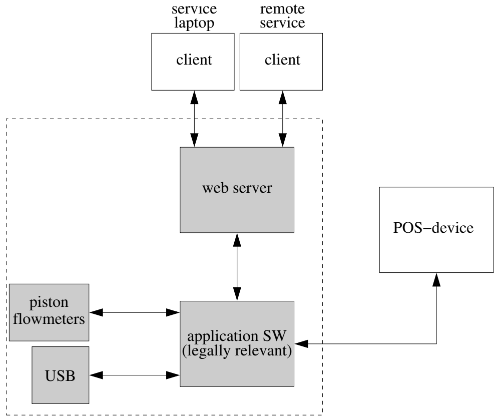

Fig. 3. High-level schematic for the fuel pump calculator that was evaluated as an illustrative example. Parts lying physically inside the instrument are surrounded by the dotted line.

B. Fuel Pump

The second measuring instrument, that was evaluated according to the new scheme proposed here, is the calculator unit of a fuel pump. The device communicates externally with a point-of-sales (POS) device and reads data from a number of flow piston meters. As an operating system a common Linux distribution is used. Measurement results are displayed locally on a seven-segment-display and are also transmitted over a LAN to the POS device. The communication with the POS device is unidirectional. Parameters that influence the metrological behavior of the system and the operating system can be changed and accessed when a USB stick with a 32-bit key is plugged into the unit. This key is usually only in the possession of an authorized inspector. In addition, the instrument possesses a web server that can be accessed over the Internet. Through the web interface, the status of the machine can be queried and parameters can be set.

A rough schematic of the measuring instrument and its surroundings can be found in Figure 3. With reference to the documentation supplied by the manufacturer a list of possible attack scenarios can again be identified. The easiest way of deriving meaningful attack vectors is focusing on interfaces available to the outside world. Here, these include both the USB port and the communication with the POS device as well as the web interface. The communication with the POS device is physically sealed since it is also under legal control. The USB port is easily accessible for the owner of the pump, while the web interface is freely accessible for anybody in possession of the IP address. The web server in question, as a commonly used IT product, has several entries in the public CVE database [15] which is maintained by MITRE, a non-profit company operating multiple research and development centers financed by the US government. The database provides an extensive list of known vulnerabilities for virtually all software components

TABLE III
Evaluation of a small number of selected threats according to their attack vector for the example no. 1 ”grain moisture analyzer“

Threat	Description	Impact	Attack Vector	Elapsed Time	Expertise	Knowledge of the TOE	Window of Opportunity	Equipment	Sum	Score	Risk
T1	Local admin (S1) invalidates integrity or authenticity of the metrological software (O1).	5	A_SW_REPLACE	(>180d) 19	(expert) 6	(restricted) 3	(unlimited) 0	(standard) 0	28	1	**1**
T2	Local admin (S1) invalidates the availability of the evidence of an intervention (A2).	5	A_INT_SERIAL	(>30d) 4	(proficient) 3	(sensitive) 7	(unlimited) 0	(specialized) 4	18	3	**3**
T3	Local admin (S1) invalidates the integrity of the metrological parameters (A4).	5	A_INT_SERIAL_VALUE	(>60d) 7	(expert) 6	(sensitive) 7	(unlimited) 0	(specialized) 4	24	2	**2**
T4	Local admin (S2) invalidates the availability of the evidence of an intervention (A2) by deleting the evidence.	5	A_PASSWORD	(>180d) 19	(layman) 0	(restricted) 3	(unlimited) 0	(standard) 0	22	2	**2**
T5	Local admin (S2) invalidates integrity, authenticity or availability of a measurement result (A8).	2	A_INT_USB	(>60d) 7	(expert) 6	(restricted) 3	(unlimited) 0	(specialized) 4	20	2	**1**

publicly available. A short excerpt from the compiled list of possible attack vectors will be supplied here:

- **A_USB_SCRIPT**: An attacker fakes an authorized key on a USB stick thus gaining access to the operating system.
- **A_WEB_XSS**: An attacker utilizes CVE-2011-4273 for a XSS attack to execute arbitrary javascript code on the web server and to subsequently download a root kit to the system.
- **A_WEB_DOS**: An attacker exploits CVE-2009-5111, CVE-2003-1568 or CVE-2002-2429 by executing a DoS attack via partial HTTP requests.
- **A_WEB_SOCKET**: An attacker executes arbitrary malicious code while establishing a connection making use of CVE-2002-2431.

As was the case with the first example, all known threats are iteratively examined. For each of them, the evaluator has to decide whether there are any attack vectors that could be used to realize the respective threat. Afterwards, the combination of threat and attack vector is again evaluated using the point score from the CC. Here too, the threat T1 consists of a replacement of the legally relevant software after gaining access to the operating system. Since the password is entered via the USB port, there is the possibility to execute and automated brute-force attack on the authentication data. This will, however, require significant resources. In addition, a replacement software needs to be written that mimics the original one. Again, the time needed for implementing of T1 is very high, resulting in a point score of 19. Also, the attacker in question needs to be an expert software engineer (point score 6) with detailed knowledge of the measuring instrument (point score 3). Should the operator of the pump also be the attacker, unlimited access to the device is obviously given resulting in a respective score of 0. The sum total of 29 points is still so high, that the resulting probability score of 1 is negligible. This has to be seen in conjunction with the fact that the selected combination of attacker and capabilities is highly improbable in any case.

Much more likely appears a local attack on the web server (see threat T2). Since exploits for the servers vulnerabilities can be downloaded from the web, no significant implementation time (point score 4) is needed. The attacker still needs to be an expert (point score 6) in the field of software engineering to correctly use the exploit. A certain amount of detail with respect to the measuring instrument such as its IP address (point score 3) is also required. But since the attack is web-based, access is virtually unlimited. Given specialized equipment like a platform to test the attack, the resulting total point score of 17 is relatively low. This results in an medium threat probability. The evaluation of the remaining threats progresses in a similar manner. The evaluation results may be found in Table IV. Again, it is important to mention, that some threats (T5 and T6) have a low risk score despite a medium probability score since their impact is limited to one single measurement at a time.

C. Comparison with WELMEC Guide 5.3

WELMEC Guide 5.3 uses the same definition of risk as the approach discussed here: a product of impact and probability of occurrence for a threat. Additionally, the guide proposes to calculate an average impact score based on economic

TABLE IV
EVALUATION OF A SMALL NUMBER OF SELECTED THREATS ACCORDING TO THEIR ATTACK VECTOR FOR THE EXAMPLE NO. 2 ”FUEL PUMP“

Threat	**Description**	**Impact**	**Attack Vector**	**Elapsed Time**	**Expertise**	**Knowledge of the TOE**	**Window of Opportunity**	**Equipment**	**Sum**	**Score**	**Risk**
T1	Local admin (S1) invalidates integrity or authenticity of the metrological software (A1).	5	A_USB_ SCRIPT	(>180d) 19	(expert) 6	(restricted) 3	(unlimited) 0	(standard) 0	28	1	**1**
T2	Local admin (S1) invalidates the integrity of the metrological parameters (A4).	5	A_WEB_ SOCKET	(>30d) 4	(expert) 6	(restricted) 3	(unlimited) 0	(specialized) 4	17	3	**3**
T3	Remote admin (S2) invalidates the availability of the evidence of an intervention (A2).	5	A_WEB_ SOCKET	(>30d) 4	(expert) 6	(restricted) 3	(unlimited) 0	(specialized) 4	17	3	**3**
T4	Remote admin (S2) invalidates the integrity or the authenticity of the metrological software (A1).	5	A_WEB_ DOS + A_WEB_ XSS	(>180d) 19	(expert) 6	(sensitive) 7	(unlimited) 0	(specialized) 4	36	1	**1**
T5	Local admin (S1) invalidates the availability or the integrity of the indication of the result. (A6)	2	A_USB SCRIPT	(>30d) 4	(expert) 6	(restricted) 3	(easy) 1	(standard) 0	14	3	**1**
T6	Remote admin (S2) invalidates availability or integrity of the indication of a result (A6).	2	A_WEB_ DOS	(>30d) 4	(proficient) 3	(restricted) 3	(unlimited) 0	(specialized) 4	20	3	**1**

implications, public health, consumer confidence, and legal issues. The assets identified in Section III, which fall into the latter category, could thus also be used in the context of the guide. However, the likelihood estimation in WELMEC Guide 5.3 clearly has the aim of assessing ”the probability of non-compliance“. It addresses both behavior of manufacturer and consumer as well as the production cycle of the instrument. The assessment is clearly focused on the manufacturer of the instrument. Unintended, implementation-based vulnerabilities of a measuring instrument are not within the scope of WELMEC Guide 5.3 since technical details with respect to the instrument's components are not taken into account at all. Instead, much more emphasis is placed on the perception of legal requirements and statistics concerning malfunctions observed in the field. While the latter may provide helpful hints, it cannot be used to assess risks associated with a new product in advance. The Guide is thus not able to produce comparable evaluation results.

D. Comparison with ISO/IEC 27005

The most significant difference between the approach presented here and ISO/IEC 27005 [3] is the addition of the probability calculation based on the vulnerability analysis from the CC and the CEM. ISO/IEC 27005 explicitly states that likelihood estimation techniques should take into account ”the motivation and capabilities, which will change over time, and resources available to possible attackers“. Both motivation and capabilities (including equipment, required skills, and knowledge) can clearly be mapped to the CC-based probability estimation as detailed in Section IV. The new risk assessment approach can thus be seen as a practical realization of ISO/IEC 27005, which does itself not specify ways of calculating individual threat probabilities. The identification of assets as described in Section III is also clearly compatible with ISO/IEC 27005, since it follows the same three-step approach of risk identification, risk estimation and risk evaluation. Nevertheless, a number of additional hints can be found in the standard on how to improve the proposed method further:

- Motivation or resources of attackers may change over time, so a risk assessment for a specific measuring instrument may have to be conducted again after a certain time interval to keep it up to date.
- Even though a possible attacker may have access to a device, he might lack the motivation or the skills to carry out an attack. Thus, each implemented threat with an associated attack vector could be checked against a list of likely attackers. Unlikely combinations of skill and window of opportunity could then be removed from the evaluation table, resulting in a more clearly defined risk scenario.

VI. SUMMARY AND FUTURE WORK

The method for software risk assessment for measuring instruments in legal metrology described here follows the guidelines of ISO/IEC 27005 [3]. In addition, elements from ISO/IEC 15408 [11] and ISO/IEC 18045 [12] were used to derive meaningful probability scores for certain threats. In order to make the evaluation process more objective, legal requirements for measuring instruments as laid down in [1] have been formalized, resulting in a list of assets to be protected and their respective security properties. With the aim of showing the feasibility of the approach, two real-world examples of measuring instruments were examined giving results that could be used as a feedback into the manufacturers design and production phase. Even though all application scenarios discussed here are from the sector of legal metrology, the approach may be of interest to evaluators of software in general. The formalization of assets and security requirements can, of course, be adapted to fit other legal or contractual obligations apart from the MID. The evaluation scheme can then be used in the same manner as was demonstrated here.

The two examples used for demonstrative purposes showed that the scheme can indeed provide meaningful results based on the information available to a Notified Body when assessing a manufacturer's design. If the source code also were available, the method from [7] could be applied to further validate the determined risk scenario. Even without additional information there are a number of steps that could be taken to improve the proposed approach:

In a first step, different evaluators will be asked to assess generic measuring instruments in a field test. To this end, the approach is currently being tested in a subgroup of WELMEC Working Group 7 "Software“. During testing, the reproducibility of the assessment results can be investigated under realistic circumstances. Secondly, better developed attacker models will be incorporated to include more information about a measuring instrument's field of usage in the assessment. In this step, the motivation of certain attackers could also be added as another individual evaluation component. This change would also necessitate a modification of the point scores from the CC and will thus require very careful adjustments.

References

[1] "Directive 2014/32/EU of the European Parliament and of the Council of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of measuring instruments," European Union, Council of the European Union ; European Parliament, Directive, February 2014.

[2] D. Peters, U. Grottker, F. Thiel, M. Peter, and J.-P. Seifert, "Achieving software security for measuring instruments under legal control," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, vol. 3, Warsaw Poland, September 2014, pp. 123–130, DOI: 10.15439/2014F460.

[3] "ISO/IEC 27005:2011(e) Information technology - Security techniques - Information security risk management," International Organization for Standardization, Geneva, CH, Standard, June 2011.

[4] G. Geiger, "Ict Security Risk Management: Economic Perspectives," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, vol. 3, 2014, pp. 119–122, DOI: 10.15439/2014F439.

[5] "Regulation (EC) No 765/2008 of the European Parliament and of the Council of 9 July 2008 setting out the requirements for accreditation and market surveillance relating to the marketing of products," European Union, Council of the European Union ; European Parliament, Regulation, July 2008.

[6] "WELMEC 5.3 Risk Assessment Guide for Market Surveillance: Weigh and Measuring Instrument," European cooperation in legal metrology, WELMEC Secretariat, Ljubljana, Standard, May 2011.

[7] A. van Deursen and T. Kuipers, "Source-based software risk assessment," in *Proceedings of the IEEE International Conference on Software Maintenance*. IEEE, September 2003, pp. 385–388, DOI: 10.1109/ICSM.2003.1235448.

[8] S.-W. Foo and A. Muruganantham, "Software risk assessment model," in *Proceedings of the IEEE International Conference on Management of Innovation and Technology*, vol. 2. IEEE, November 2000, pp. 536–544, DOI: 10.1109/ICMIT.2000.916747.

[9] N. Greif and G. Parkin, "An international harmonised measurement software guide: the need and the concept," in *Proceedings of the IMEKO World Congress Fundamental and Applied Metrology*, Lisbon, Portugal, September 2009, pp. 2440–2443.

[10] M. Sadiq, M. K. I. Rahmani, M. W. Ahmad, and S. Jung, "Software risk assessment and evaluation process (sraep) using model based approach," in *Proceedings of the IEEE International Conference on Networking and Information Technology*. IEEE, June 2010, pp. 171–177, DOI: 10.1109/ICNIT.2010.5508535.

[11] "ISO/IEC 15408:2012 Common Criteria for Information Technology Security Evaluation," International Organization for Standardization, Geneva, CH, Standard, September 2012, Version 3.1 Revision 4.

[12] "ISO/IEC 18045:2012 Common Methodology for Information Technology Security Evaluation," International Organization for Standardization, Geneva, CH, Standard, September 2012, Version 3.1 Revision 4.

[13] "ETSI TS 102 165-1 Telecommunications and Internet converged Services and Protocols for Advanced Networking; Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis," European Telecommunications Standards Institute, Sophia Antipolis Cedex, FR, Standard, March 2011, v4.2.3.

[14] "WELMEC 7.2 Software Guide," European cooperation in legal metrology, WELMEC Secretariat, Delft, Standard, March 2012.

[15] "CVE - Common Vulnerabilities and Exposures," https://cve.mitre.org/, Accessed 04|17|2015.