Proceedings of the Federated Conference on Computer Science and Information Systems pp. 763–771
DOI: 10.15439/2017F112
ISSN 2300-5963 ACSIS, Vol. 11

Representation of Attacker Motivation in Software Risk Assessment Using Attack Probability Trees

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

***Abstract*—Since software plays an ever more important role in measuring instruments, risk assessments for such instruments required by European regulations will usually include also a risk assessment of the software. Although previously introduced methods still lack efficient means for the representation of attacker motivation and have no prescribed way of constructing attack scenarios, attack trees have been used for several years in similar application scenarios. These trees are here developed into attack probability trees, specifically tailored to meet the requirements for software risk assessment. A real-world example based on taximeters is given to illustrate the application of attack probability trees approach and their advantages.**

I. Introduction

IN EUROPE certain kinds of measuring instruments, such as gas meters, electricity meters and taximeters are subject to requirements established in the European Measuring Instruments Directive (MID) [1]. The MID was originally published in 2004 with the aim of providing trust in measurements for both customers and users of measuring instruments by defining essential requirements that each measuring instrument used within the common European single market has to fulfill. These requirements cover everything from climatic operating conditions, electro-magnetic compliance testing to requirements on software and data protection. The entire economic sector of legally regulated measuring instruments is commonly referred to as Legal Metrology. In Germany, roughly 137 million such regulated instruments are currently in use and together are responsible for an annual turnover of around 150 billion Euros. For the entirety of the European Union, this amounts to more than 500 billion Euros per year. Each instrument regulated by European or national legislation first has to pass a conformity assessment before it can legally be put into use [1]. This conformity assessment is done according to certain modules, the most common of which is referred to as Module B and essentially comprises tests of a prototype instrument and of the associated documentation. In Germany, one of the conformity assessment bodies tasked with performing assessment according to Module B is the *Physikalisch-Technische Bundesanstalt* (PTB), Germany's national metrology institute. As software plays an ever more important role in measuring instruments, testing of the software nowadays constitutes an integral part of the conformity assessment process. Since April 2016, the documentation submitted by the manufacturer for Module B also has to include an "adequate analysis and assessment of the risks" [1] associated with the instrument type. To help manufacturers with this task, PTB has developed and published a risk assessment procedure based on ISO/IEC 27005 [2] and ISO/IEC 18045 [3], which also enables objective comparison between different instruments from different manufacturers [4]. The publication also derives detailed assets to be protected and their individual security properties from the legal text of the MID. In line with the definitions in the ISO/IEC 27005, the method defines the term risk as a combination of the consequences resulting from threats to assets and of the probability of occurrence of a threat. Any way to realize a certain threat to an asset is then usually referred to as an attack vector. Since the original method primarily focused on technical aspects of the instrument, PTB also published an extension to the method [5] that takes attacker motivation into account. Despite these improvements and even though the risk assessment method is now actively being used, it still harbors a number of deficiencies. Among these is the fact that there is no prescribed way of constructing above-mentioned attack vectors in a standardized way. In the past, so-called attack trees have been used to this end in similar fields of application, such as the design of cryptographic protocols and access control [6]. A second challenge is the fact that an efficient way to handle the impact of attacker motivation during risk assessment is also still missing. Both problems will be addressed here and a possible solution, based on modified attack trees, will be described. The remainder of the paper is structured as follows. Section II gives a brief overview on the history of attack trees, covers basic principles and describes other applications. Afterwards, Section III revisits the method originally described in [4], touches upon its extension to include attacker motivation and introduces attack probability trees (AtPT) as a way of constructing and evaluating attacks in a standardized manner. In the subsequent Section IV a real-world example from Legal Metrology concerning possibly manipulated taximeters is used to illustrate the AtPTs and their uses. Finally, Section V summarizes the paper and details the planned future work.

II. Literature Overview

The international standard ISO/IEC 27005 [2] defines three sub-processes that together form a complete risk assessment, namely risk identification, risk estimation and risk evaluation.

The first sub-process includes the identification of assets to be protected. Assets specifically tailored to Legal Metrology have been derived in [4] and will briefly be revisited in Section III. The risk identification phase also requires the definition or derivation of threats, which may invalidate certain security properties of an asset. Finding technical realizations of a threat, also referred to as attack vectors, is part of risk identification as well. Afterwards, a threat and its associated attack vector are evaluated with respect to probability of occurrence and resulting impact in the risk estimation phase.

For illustration purposes a brief example will be given here: If the integrity of a text document on a PC is to be protected, the document itself can be thought of the asset while its security property is integrity. Should such a file be write-protected due to measures realized in an operating system, a possible attack vector would be the retrieval of the administrator's credentials. With these, an attacker could delete or modify the file at will, thereby invalidating its integrity. To estimate the likelihood of such an attack, the password strength and the accessibility of the computer would, for instance, need to be taken into account. During risk evaluation, the estimated risk is either classified as tolerable or intolerable. In the latter case, countermeasures are selected and the entire risk assessment process is executed again until the risks are reduced to an acceptable level. An example for the selection of suitable countermeasures will be given in Section IV. Details on different risk assessment methods, which could be applied to software in measuring instruments, may be found in [4].

In this paper, the focus will be on the identification of attack vectors and on their efficient graphical and logical representation. While attack trees originally served the principal purpose of illustrating or identifying system vulnerabilities in a graphical manner easily understood by humans, they also show a number of mathematical properties which make them quite suitable for automatic analysis and processing.

A. Foundations of Attack Trees

A very detailed introduction to attack trees and their background is given by Mauw and Oostdijk in [6]. There, the authors state that the root node of an attack tree usually represents an attacker's target or goal while child nodes are refinements of such an attack. The leaves of the tree then represent elementary or atomic attacks that can no longer be refined. As an example, Fig. 1 shows a very small attack tree which illustrates ways to manipulate the fare calculated by a taximeter. If two or more child nodes are connected by an arc, these refinements are to be seen as connected by an AND-statement, meaning that both of them have to be fulfilled before the respective parent node/goal itself can be reached. All other child nodes are OR-related so that only one of them needs to be fulfilled to achieve the parent goal. Mauw and Oostdijk refer to these to relations as "conjunctive aggregation" and "disjunctive refinements (choice)", respectively.

In the given example, the fare can either be manipulated by modifying the parameters of the taximeter itself or by manipulating the signal coming from the wheel sensor. This

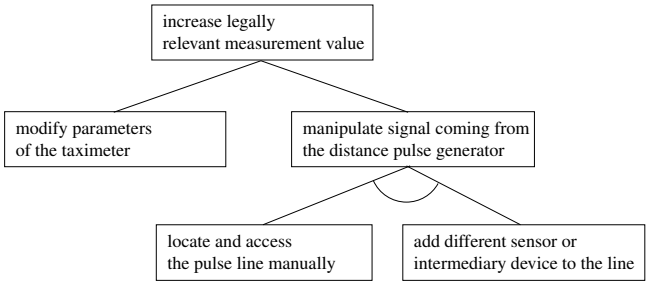

Fig. 1. Simple illustration of an attack tree that shows how the calculated fare/measurement value of a taximeter may be manipulated.

constitutes a simple OR-relationship as both attack vectors will help to achieve the desired goal. To manipulate the signal coming from the wheel sensor one could, for instance, obtain access to the pulse line connecting sensor and taximeter and then buy and install a pulse multiplier that automatically doubles the number of pulses transmitted on the line. Only if both steps have been taken, can the target be reached which corresponds to an AND-statement. In this context, it is not necessary for these actions to happen at the same time. AND-statements can also express a step-wise execution process.

Apart from giving basic definitions relevant to attack trees, Mauw and Oostdijk also state, that leaf nodes are usually given a number of predefined characteristics such as possibility, cost or special tools needed for their execution, also see [7] for further details. In order to estimate the attributes of the parent nodes and finally of the root node, rules for combining information originating from the child nodes are required. Mauw and Oostdijk also stress the point that these rules may usually be directly derived from the characteristics of an attribute. It should be obvious that these rules are generally used in a bottom-up fashion, requiring no additional loops or trace-backs. The results of an analysis are then either the attributes of a root node or a selected sub-tree, where the selected sub-tree may represent a set of likely attacks or may contain information not directly reflected by the values of the attributes but rather by its internal structure. Another important finding in [6] is the fact that individual nodes do not necessarily only have to occur once within an attack as they may have to be used several times. Nodes can subsequently have several copies whose attributes are linked to each other.

Mauw and Oostdijk then introduce the concept of an attack suite to represent a set of attacks which can all be used to achieve a goal without stating their individual branching structure. By means of this concept they are able to prove that attack trees with different structures may represent the same information despite their apparent structural differences. Nodes can also be connected to a multi-set of nodes, which Mauw and Oostdijk refer to as a bundle. Execution of all elements within the bundle will ensure that the goal is achieved. Cycles within an attack tree are not allowed, which restricts attack trees to be "rooted directed acyclic bundle graphs."

In [6] rules are defined for attack tree transformations. The first rule is "associativity of conjunction", meaning that a

sub-bundle can be lifted to the parent node if no other sub-bundles are connected to the parent node and the parent node is thus identical to the sub-bundle itself. An example for such a node will later be given and explained in Section IV, Fig. 6. The second rule is the distributivity of "conjunction over disjunction", meaning that a node with two sub-bundles/sub-trees can be replaced by two copies of the same node with one bundle/subtree each. Proofs for both rules are also detailed in [6]. Some additional remarks are targeted at attributes of attack trees: To calculate the value of an attribute, the semantics of the tree first need to be determined. Afterwards, the value of the attribute belonging to the equivalent attack suite is calculated. One basic assumption, which is going to be reused here, is that attributes of an attack node can always be calculated from the attributes of its attack components/child nodes.

B. Software Risk Assessment and Evaluation Process

In [8] Sadiq, Rahmani, Ahmad and Jung use a concept very similar to attack trees within their Software Risk Assessment and Evaluation Process (SREAP). The software fault trees (SFT) are again derived from a thorough examination and very detailed modeling of the target system. It is highlighted that despite attack trees being widely used, there is no standard way to construct such trees yet. However, once a tree has been identified, it can recursively be used to construct larger attacks, which might not have been obvious from the beginning of the investigation.

To rank certain attacks, Sadiq, Rahmani, Ahmad and Jung propose a key node safety metric. The metric is split into two parts, namely the impact of a node in the tree and the collective effect of the node consisting of the size of the underlying sub-tree, as well as the depth of the node.

C. Threat Risk Analysis for Cloud Security based on Attack-Defense Trees

Prior work on attack trees was focused on the description of envisioned attack profiles without taking defensive strategies into account. In [9] Wang, Lin, Kuo, Lin and Wang proposed a new modified version of such tries referred to as Attack-Defense Trees (ADT) which also incorporates defense concepts. The effectiveness of the proposed method has been tested according to a set of predefined metrics. The basic problem to be solved by their method is also referred to as Threat Risk Analysis (TRA), which describes the process of identifying realistic defense strategies based on vulnerability information and attack profiles.

A TRA encompasses both the impact of a realized attack and a precise description of the attack progression. This makes it possible to develop fitting defense strategies. Attack trees generally become very complex when trying to model all possible attacks at the desired level of detail. According to Wang, Lin, Kuo, Lin and Wang, stating both attack and defense strategies at the same time in an ADT is even more complex and usually beyond the scope of an attack tree. Whereas attack trees are used to model system weaknesses, protection trees offer the opportunity to identify protective strategies by migrating weaknesses. In Section IV it will be shown how good starting point for such a defense tree may be identified.

According to [9], it was historically assumed that attackers strategically plan the attacks based on the easiest available scenario, but this may not always be the case, as an attacker might not have all information necessary to make an informed choice. Equations for calculating the probability of occurrence and other metrics for AND- and OR-connections are also given. These include probability of success, attack cost, impact as well as revised attack cost and revised impact for the countermeasure stage. All metrics in [9] are first calculated for the leaf nodes and are then propagated up the tree.

Afterwards Wang, Lin, Kuo, Lin and Wang introduce the concept of "attack and defense actions". The first of which is to understand the vulnerabilities of the system. Information on vulnerabilities can, for instance, be collected from public databases. This is identical to the procedure described in [4], which is again used here. The next action is the collection of information on recognized attacks, e.g. identifying ways to implement an attack based on known vulnerabilities. Afterwards, an ADT is constructed by finding as many vulnerabilities as possible, which can be used to implement the considered threat. Once the ADT is finished, it is systematically evaluated. Wang, Lin and Kuo observe, that, while the goal is to minimize the probability of occurrence of an attack, the rate of occurrence and the associated impact may not always be available and thus a certain degree of uncertainty remains. It is postulated, that attack cost and defense cost are connected by a transfer function to map one to the other. An example covering Advanced Persistent Threat attacks called Operation Aurora is also included in [9].

D. Automated Generation of Attack Trees

As indicated above, currently no method exists that enforces a harmonized generation of attack trees. In [10] Vigo, Nielson and Nielson offer a solution to this problem by inferring attack trees from process algebraic expressions. They explain that attack trees are used by scientists as they are quantifiable and by the public since they are easily understandable. In their implementation, the root again represents a threat to be realized and internal nodes illustrate the manner in which attacks need to be combined to achieve a goal. As indicated above, there may be several attack trees that all describe the same attack logic. Vigo, Nielson and Nielson overcome this problem by resorting to the calculus used to describe the attack process. This is done by translating the attack process into propositional formulae. Since this step can be done automatically, it does not suffer from human interpretation errors.

After the modeling phase, atomic attacks, which constitute the leaves of a tree, need to be labeled with individual costs. Following the process-oriented idea, attacks are seen as interactions between attacker and target in terms of a communication process in [10]. Finally, the cheapest set of atomic attacks needed to achieve a goal is calculated. Section

III will show how a similar process could be realized for attack trees, specifically tailored for measuring instruments that follow harmonized technical requirements established by [11] as an interpretation of the MID.

III. Description of the Approach

The risk method assessment method, which will be described in this section, was originally published in [4], although some changes were adopted later on to reflect the experience gathered during the application of the method at PTB over the past two years.

A. Basic description of the risk assessment method

The basic procedure and all associated definitions were derived from ISO/IEC 27005 [2], where risk is defined as a combination of impact and probability of occurrence of a threat. Even though the international standard allows both quantitative and qualitative assessment of risks, only numeric representations of probability and impact (and therefore also risk) are used here. In order to be able to assign specific values to probability and impact, assets were defined in [4] by examining and interpreting the essential requirements of the legal text, i.e. of the Annex I of the MID. The interpretation led to the definition of a number of assets to be protected with associated security properties, all of which may be found in [4].

As only one such asset is going to be used for illustration purposes in Section IV, a single example will be given here: Essential requirement 8.4 of the Annex II 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]. In this simple requirement three assets are listed, namely measurement data, software critical for measurement characteristics and metrologically important parameters. Each of these can be assigned a number of security properties. Measurement data, for instance, are required to preserve their authenticity and integrity, i.e. measurement data should not be changed and an attacker should not be able to generate false measurement data. An example for a formal description of a threat could thus be given by the following sentence: *An attacker manages to invalidate the integrity of measurement data.*

To assess such a threat, values for impact and probability of occurrence are now required. In [4] five different levels were originally used for impact, but in practice only threats affecting a single measurement (impact of $\frac{1}{3}$) and affecting all future or all past measurements (impact of 1) are actively differentiated. As the threat itself is only a formal statement but gives no explicit instructions on how to realize it, a specific attack vector is needed next. To this end, all possible attack vectors, which could potentially be used to realize a threat, are examined in turn and their individual likelihood is checked. In order to estimate the probability of occurrence of an attack vector, a method called vulnerability analysis from ISO/IEC 18045 [3] is used.

TABLE I
Mapping of the so-called TOE resistance to the probability score used here, originally published in [4].

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

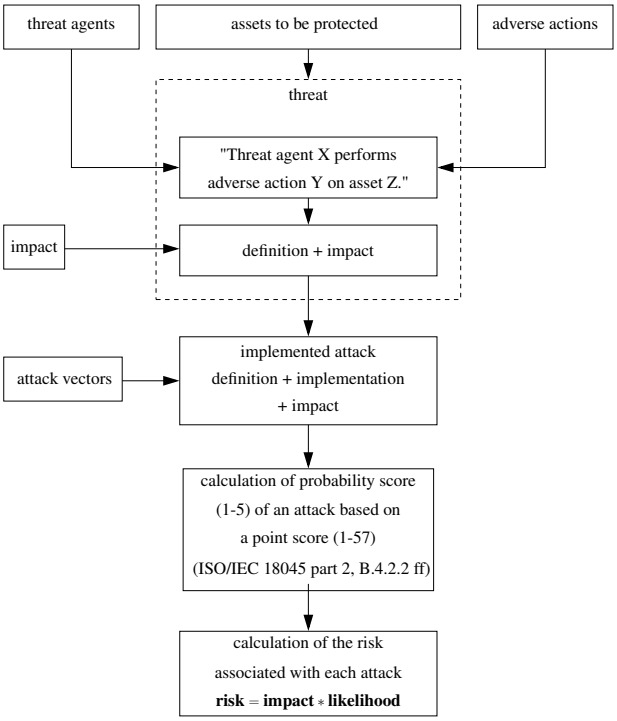

Fig. 2. Flowchart of the basic risk assessment procedure, adopted from [4].

The analysis consists of assigning a point score to the attack vector in five different categories, namely required time, expertise and knowledge of the attacked target of evaluation (TOE) as well as the window of opportunity and special equipment needed. A time score of 1, for instance is given if an attack requires more than a day, but less then a week for its execution. A full example for such a point score in all five categories will be given in Section IV. Explicit instructions on how to assign the scores may be found in [3]. In general, a higher sum score expresses a higher resistance of the TOE to attacks, i.e. an attack is less likely if its sum score is high. In line with this notion, the sum score is mapped to a probability score as given in Table I. Once the probability score has been calculated it is multiplied with the impact score (between 0 and 1) to form the final risk value. A complete flowchart of the entire basic method is given in Fig. 2.

B. Description of the extension for attacker motivation

One of the shortcomings of the original method described in [4] was the inability to take attacker motivation into account. As a motivated attacker will certainly be willing to invest

more resources into the execution of an attack, there should definitely be some sort of correlation between motivation of an attacker and likelihood of an attack. In [3], it is specifically stated that attacker motivation will have an influence on the used resources e.g. equipment, expertise, as both may be acquired if sufficient monetary funds are available. Other factors, like the time required for an attack and a possible window of opportunity however, cannot be influenced.

In [5], the original method was extended by a motivation score (0 for high motivation and 9 for no motivation), that acts as a lower limit for the expertise and equipment scores introduced above. Whenever the initially assigned score for one of these categories is smaller than the motivation score, its value is replaced with that of said score. It should be instantly obvious that the scenario with a highly motivated attacker is then identical to the originally calculated score as described in [4] and now it takes on the role of a theoretical upper limit to the probability of occurrence. A lower level of motivation will automatically result in a smaller probability, as the sum score will be increased due to the replacement values for expertise and equipment. A more in-depth discussion and a complete example may be found in [5].

C. Attack probability trees

In Section I the two main objectives of this paper were stated: to design a method that enables a standardized derivation of attack vectors and also efficiently represents the effect of attacker motivation on the risk assessment results. To this end, attack trees as defined by Mauw and Oostdijk in [6] will be extended here into attack probability trees. These extended attack trees do not only represent the logical relationship between parent and child attacks, but are now also labeled with all the attributes defined in the vulnerability analysis in [3], i.e. each node has its own score for time, expertise, knowledge, window of opportunity and equipment. Based on these values, each node is given a sum score and, subsequently, a probability score. To fully reflect all variables from the method described in [4] each node could also be labeled with an impact score. However, since every node within an AtPT aims at realizing the same threat with a fixed impact, the impact score can safely be omitted. The final attribute of the root node thus only represents the probability of an attack, which can later be turned into a risk if combined with the respective impact score.

Nodes may be linked with each other to either form AND- or OR-statement. As suggested by Mauw and Oostdijk, information will enter the AtPT only via the leaves. The attributes for the parent nodes and finally for the root node can be calculated in a bottom-up fashion by observing the following stated rules. The rationale for each rule is also given.

- Time
  - **AND**: Time scores are logarithmic (1 for more than a day, 2 for a one week to two weeks, 17 for half a year), therefore the maximum of both scores needs to be chosen which is a good approximation for the logarithm of two added time spans.
  - **OR**: The smaller sum-score indicates which time score is to be chosen.
- Expertise
  - **AND**: If expertise in different areas is required (HW/SW), the scores are added with a maximum of 8 in accordance with ISO/IEC 18045. Otherwise, the maximum is chosen.
  - **OR**: The smaller sum-score indicates which expertise score is to be chosen.
- Knowledge of the TOE
  - **AND**: The maximum of both knowledge scores is chosen.
  - **OR**: The smaller sum-score indicates which knowledge score is to be chosen.
- Window of opportunity
  - **AND**: A smaller window of opportunity (higher score) for one node will also affect the other node. Therefore, the maximum is selected.
  - **OR**: The smaller sum-score indicates which window of opportunity score is to be chosen.
- Equipment
  - **AND**: If equipment from different areas is required (HW/SW), the scores are added with a maximum of 9 in accordance with Common Evaluation Methodology [3]. Otherwise, the maximum is chosen.
  - **OR**: The smaller sum-score indicates which equipment score is to be chosen.

When the assumed motivation is changed, the most probable path within the attack tree will also take on a different shape and a simple evaluation of the attributes of the root node does not suffice anymore. Previously, every individual attack then had to be reevaluated individually. With an AtPT in place, the time required for this can be reduced, since many attacks share common nodes whose attributes only have to be recalculated once. This will be illustrated in detail in Section IV. The rules established above should be applicable to methods apart from the one examined here, since the attributes are also used in the vulnerability analysis of the AVA_VAN class in [3] and in the risk assessment method, which is part of the ETSI standard [12]. As shown by Mauw and Oostdijk, many different attack trees may all be interchangeable representations of each other, therefore, the design of an attack tree is a very subjective procedure. However, in Legal Metrology at least, all devices/measuring instruments share some basic characteristics, due to the fact that most instruments are based on the same acceptable technical solutions described in [11]. It may, therefore, be possible to construct attack probability trees in a reproducible manner by applying the following rules.

1) For each user interface of the measuring instrument, collect all known vulnerabilities that may lead to a realization of the threat.
2) For each communication interface of the measuring instrument, collect all known vulnerabilities that may lead to a realization of the threat.

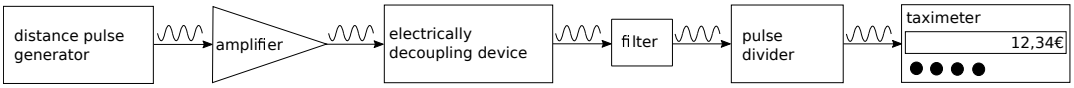

Fig. 3. Illustration of the analog signal path connecting a pulse generator at the wheel with a taximeter.

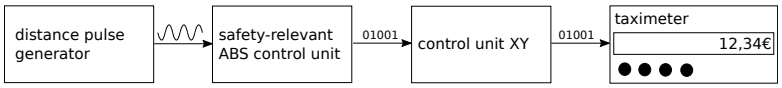

Fig. 4. Illustration of the digital signal path connecting a pulse generator at the wheel with a taximeter.

3) For each hardware protection mechanism of the measuring instrument, collect all known vulnerabilities that may lead to a realization of the threat.
4) Logical connections between the vulnerabilities gathered in steps 1) to 3) are then expressed by means of boolean expressions, afterwards transformed into the structure of the AtPT. Single attack vectors requiring no other support actions, for instance, will always be represented by direct child nodes of the root node that represents the examined threat/goal.

As these rules still leave plenty of room for subjective interpretation, a full expansion of the steps into formalized instructions to construct an attack probability tree will remain an objective for future work.

IV. Experimental Example

In the following Section, the combination of the attack probability trees introduced in this paper and the risk assessment method from [4] and [5] will be illustrated with the help of a real-world example from Legal Metrology:

Taximeters are one kind of measuring instrument subject to the requirements of the MID widely used all over Europe. However, the protective measures for such instruments agreed upon on the European level only affect the taximeter unit itself, which consists of display, user interface and open communication interface for a sensor. The signal path between the distance pulse generator at the wheel (i.e. the sensor itself) and the taximeter is only subject to national regulations.

Therefore, some countries do not require any protection for the signal path while others have introduced different protective mechanisms, each being designed with particular attack vectors in mind. Before taking a look at some of these countermeasures, the simple case of an open communication path between signal generator and taximeter will be discussed.

A. Formal case analysis for taximeters

The general installation of a taximeter in a car can be described by two very basic configurations, as shown in Fig. 3 and Fig. 4. The first typical installation consists of an analog pulse generator at the wheel of the taxi whose output are distance pulses. Each of these pulses represents a fixed distance traveled. The rate of the pulses is thus proportional to the speed of the car. These pulses may be filtered and amplified several times on their way to the taximeter, with additional electrically decoupling devices being used for safety. Pulse dividers may be used as well, if the rate expected by the taximeter does not fit the rate of the wheel sensor. This is usually the case when a car is fitted with a taximeter that does not come from the same manufacturer as the car itself. It is important to note, that the signal in this scenario is fully analog all along the signal path.

The second typical installation represents a digital signal path, see Fig. 4. There, too, an analog pulse signal is generated at the wheel. The signal is, however, converted within the safety-relevant controller of the anti-blocking system (ABS) to a digital datagram on the CAN-bus as defined by ISO 11898-1 [13]. The CAN-bus is a well-known bus protocol widely used by car manufacturers around the world to connect different digital systems within a vehicle. Attacks on the analog signal between signal generator and ABS controller are not considered in this paper, as they would very likely result in failures of brakes or acceleration control and could thus not safely be used to influence the calculated taxi fare or the distance traveled. The remainder of the signal path is purely digital, but no protective measure are realized, except for simple checksums, which may be used to test the integrity of received datagrams.

B. Attack probability tree for the analog signal path

As mentioned in Section III, the only threat investigated here is an inadmissible increase of the legally relevant measurement value, i.e. the distance traveled or the calculated taxi fare. The root node (A) as given in Fig. 5 reflects this. Since all attacks examined here have an impact on all future measuring values, they are assigned an impact score of 1. Once the sum score of a node has been calculated its respective probability score can be identified using Table I. If this score is then multiplied with the impact score of 1 to calculate the actual risk, it becomes obvious that probability score and risk are identical. For other threats, this will of course be different and the respective tree should be appropriately labeled with the assigned impact score.

For the purely analog signal path, two known attack vectors exist: the manual feeding of additional pulses into the pulse line by means of a needle (node (B) in Fig. 5) and the installation of a different pulse generator or other intermediary device into the signal path (node (C) in Fig. 5). As these two attack vectors are alternatives of one another, they are linked to the parent node (A) by an OR-connection expressed by two simple edges. An arc between two or more edges

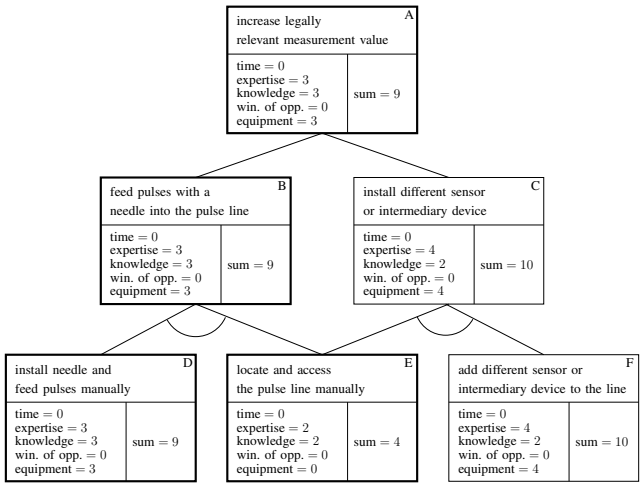

Fig. 5. Exemplary attack probability tree for the analog scenario. This also corresponds to the state of the tree for a highly motivated attacker. Highlighted nodes form the likeliest sub-tree to the root node after successful execution of the algorithm.

would represent an AND-connection. Such AND-statements may be found in the next level of the AtPT. The feeding of pulses by means of a needle (node (B)) requires both access to the pulse line (node (E)) and the manual feeding of pulses itself (node (D)). If a different sensor is to be installed (node (C)), again access to the pulse line is required (node (E)). In addition, the installation itself needs to be realized (node (F)). Again nodes (E) and (F) are linked by an AND-statement. Interestingly, node (E) plays a role in both attacks and thus offers the possibility of functioning as a possible entry point for a countermeasure. To calculate the probability score of the original threat (A), the leaf nodes (D), (E) and (F) are each assigned point scores in the aforementioned five categories. Tables listing all possible scores and an explanation for each may be found in [3].

The actual values given in Fig. 5 can be explained as follows: Finding and accessing the pulse line (node (E)) takes less than a day and is thus given a score of 0 for time. Especially in cars with a greater number of signals lines connecting arbitrary devices, finding the right spot to access the correct cable without the change later being obvious to market surveillance will require only proficient expertise, which corresponds to a score of 2. In addition, only restricted knowledge of the taximeter's installation is necessary to carry out this step, resulting in a score of 2 for knowledge. If the attacker is also the car's owner, he or she will have unlimited access, which is expressed by a value of 0 (unlimited) for the window of opportunity. Also, only standard equipment is necessary for this step (score of 0). The situation is slightly different for the actual installation and usage of the needle to feed additional pulses (node (D)). The expertise and knowledge score are slightly increased as one has to understand the internal algorithm of the taximeter to feed the right amount of pulses while also performing the task without being noticed by customer/passenger. Finally, the situation is slightly different if some additional sensor or store-bought intermediary device is installed (node (F)). Connecting the device correctly will require a slightly higher level of expertise (score of 4) but no additional information about the actual taximeter (knowledge score of 2). As the additional sensor or intermediary device cannot be considered standard equipment, that score is raised to a value of 4.

Once the attributes of the leaf nodes have all be initialized, these values can now propagate up the tree according to the rules established in Section III. It should be noted that nodes (B) and (C) both have AND-connections to their respective child nodes and thus the maximum value for each score is copied to the next level. At the root node, however, an OR-relationship between both alternative attack vectors exists. Here, scenarios (B) and (C) compete with each other. As (B) has a slightly smaller sum score it is considered more likely and the root node (A) thus becomes a direct copy of (B). The most probable attack scenario resulting from the algorithm is a sub-tree consisting of nodes (A), (B), (D) and (E), which are highlighted in Fig. 5. The sum score of 9 at node (A) corresponds to a very high probability score of 5 and, after multiplication with the impact of 1, to a risk value of 5 which would require changes to the system, before it can pass conformity assessment. Since no limits are imposed here on the expertise and knowledge scores, this scenario corresponds to the case of a highly motivated attacker. For a detailed explanation see Section III and also reference [5].

C. Attack probability tree for the digital signal path

The AtPT for the digital signal path is given in Fig. 6. In this scenario, there are three alternative attacks ((D), (E) and (F) in Fig. 6) that could all be used to realize an illegal increase of the legally relevant measurement value. Two of them ((D) and (E)) require access to the field bus of the taxi first (node B). Once physical access to the field bus has been established, an attacker could either install an additional signal source that transmits its own datagrams over the bus (node (D)) or the attacker could install a so-called car hacking device (node (E)), which jams the dataflow from other sources before transmitting its own signal and is thus more difficult to detect. Nodes (C) and (F) represent an attack on the control unit which converts accumulated distance pulse counts from the ABS unit into a physical distance in meters. The conversion factor for this operation is usually referred to as k. Once the configuration interface of the control unit has physically been accessed, k can be changed using equipment available to most car mechanics.

It should be noted that (F) could also have been split into two seperate nodes for accessing the port and changing the configuration, which was avoided here due to space limitations. Again, each leaf node can be assigned point scores for all of its five attributes. The time required for (D) and (E) will still be less than a week, corresponding to a value of 1. In both cases an expert is required to install the new device or signal source (expertise score of 6). However, the

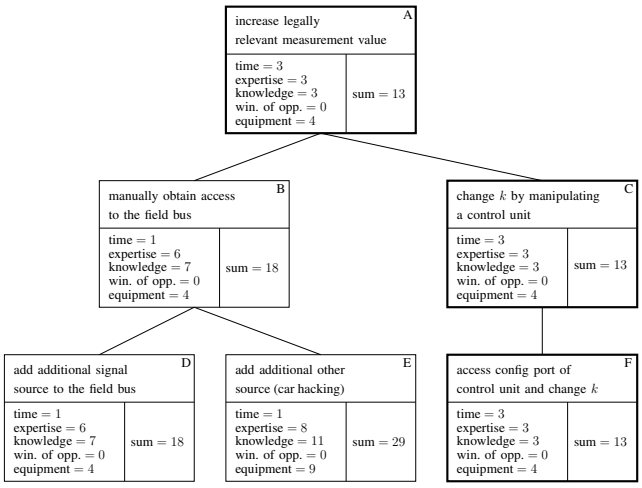

Fig. 6. Exemplary attack probability tree for the digital scenario. This also corresponds to the state of the tree for a highly motivated attacker.

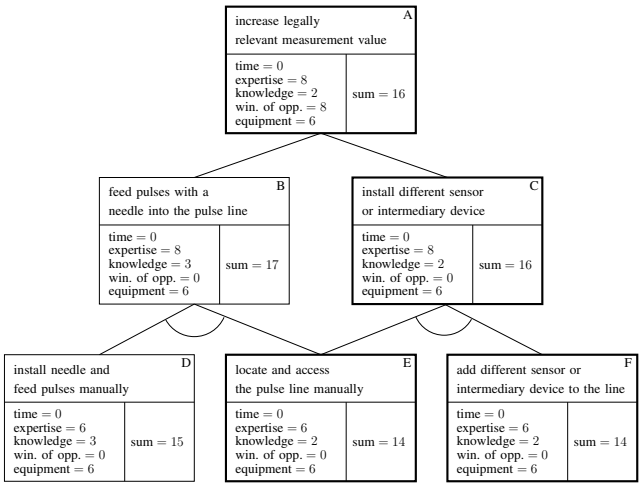

Fig. 7. Exemplary attack probability tree for the analog scenario for an attacker with low motivation.

programming of the car hacking device will also require programming skills, which increases the score to 8 (multiple expert). In order to be able to add an additional source to the field bus, sensitive knowledge concerning the addresses used on the bus and possible manufacturer-specific protocol extensions is needed (knowledge score of 7 for node (D)). If a car hacking device is programmed, the exact behavior of all other devices connected to the bus needs to be known first, which increases the knowledge score to 11 (critical knowledge). The window of opportunity is identical to that of the analog scenario. While specialized equipment (score of 4) is needed to install a new signal source (node (D)), multiple bespoke equipment, that cannot be bought legally on the market (score of 9 for node (E)) is required for car hacking. The attack on the configuration port as described by nodes (C) and (F) may take considerably longer (score of 3 for time), since software for breaking the car's security mechanisms (password protection) may be needed. The software, however does not require expert knowledge to be operated (score of 3) and only restricted knowledge, available to most mechanics, is needed to identify the correct port and execute the attack (score of 3). Again, there may be an unlimited window of opportunity and the equipment level is comparable to the one for attack (D).

The algorithm for propagating attributes values is executed in the same manner as before. At node (B) attacks (D) and (E) compete with one another as they are linked by an OR-statement. Since (D) has a smaller sum score and is thus more likely, its values are propagated to (B). (F) is a simple copy of (C) and its values are simply copied to the next level. At the root node the likeliest attack scenario (C) is again selected according to the sum score. The resulting sub-tree with the highest probability of occurence (nodes (A), (C) and (F)) is highlighted in Fig. The score for probability of occurrence can finally be derived from the sum score of 13 for the root node (A) and takes on a rather high value of 4. The risk associated with the threat is also 4, due to the impact score of 1.

D. Effect of attacker motivation

To examine the effect of attacker motivation on an AtPT, the taximeter example with an analog signal path will be used again. As mentioned above, the AtPT given in Fig. 5 corresponds to a scenario with a highly motivated attacker willing to invest virtually limitless amounts of resources. If an attacker with low or medium motivation is considered instead, a lower bound for expertise and equipment is imposed. For medium motivation, this limit takes on a value of 3, see [5] for details. The effect of a low level of motivation (lower limit of 6) can be seen in Fig. 7. Again, the attributes of the leaves constitute the input to the algorithm.

Here, the scores for expertise and equipment are automatically set to a value of 6. Afterwards, the values are propagated up the tree. Node (B) is thus assigned a sum score of 17 while node (C) receives a sum score of 16 due to their expertise value of 8, see expertise rule for AND-statement in Section III. Compared to the original state of the AtPT, node (C) suddenly becomes more probable, which shifts the likeliest sub-tree to the constellation (A), (C), (E), (F). Thus, the properties of (C) are finally copied to the root node (A). It follows, that the most probable attack vector does not only depend upon technical specifications but also on the level of motivation of an attacker. This finding should play an important role when designing and selecting countermeasures to attack vectors.

E. Identifying suitable countermeasures

As countermeasures will specifically target one or more attack vectors, they can directly be linked to one or more nodes within an AtPT. With the aim of finding the best node for a countermeasure, inverted sub-trees within an AtPT need to be found. An inverted tree could be any leaf with more than one connected node from the preceding level. The bigger such an inverted tree is, the more parent nodes depend upon the selected leaf. In Figure 5, both (B) and (C) depend upon

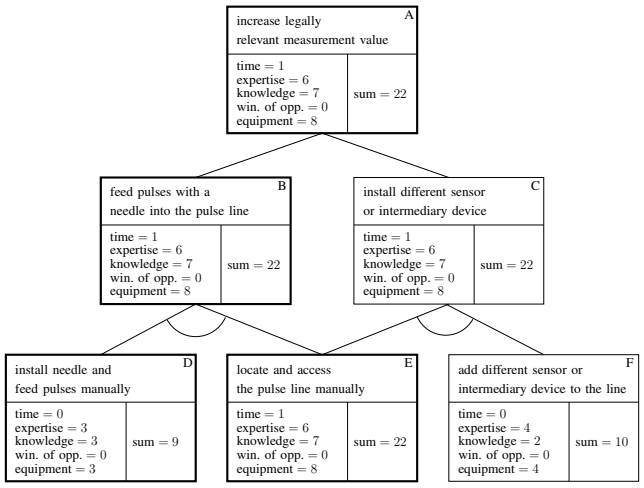

Fig. 8. Exemplary attack probability tree for the analog scenario after the implementation of the armored cable as a countermeasure.

node (E) and the biggest inverted sub-tree is constituted by nodes (E), (B) and (C). A countermeasure specifically targeted at preventing access to the pulse line (node (E)) will thus have the biggest impact in this scenario.

One such countermeasure is the installation of so-called armored cable that will either prevent access to the pulse line with layers of wire mesh or will stop working if one of the meshes is cut from the outside. The effect of the countermeasure on the attack tree is shown in Fig. 8. With the armored cable in place, the time required to access the pulse line is raised to at least a week (score of 1 for node (E)). In addition, expert knowledge (expertise score of 6) and sensitive details about the protective mechanism of the armored cable (knowledge score of 7) are needed. While the window of opportunity remains unchanged, multiple bespoke equipment is needed to successfully obtain access to the pulse line without detection (score of 8). Once these attributes have been propagated up the tree, both nodes (B) and (C) are now only influenced by node (E), with nodes (D) and (F) having no significant effect. Subsequently, the root node (A) also takes on the properties of node (E) and the sum score of 22 expresses the considerably decreased probability of occurrence, which results in acceptable probability and risk scores of 2.

V. Summary

In this paper, attack probability trees (AtPT) have been introduced specifically tailored to the risk assessment method for software in Legal Metrology described in [4] and extended in [5]. Nevertheless, the rules established here for the propagation of attributes within the AtPT should be applicable for a number of other methods ([3], [12]).

A detailed example was discussed to illustrate the *bottom-up* approach of the algorithm. In addition, the effect of attacker motivation on the assessment results was also examined and it was shown that the most likely attack cannot be identified by examining technical features alone. Instead, the attacker motivation will have a significant affect on the sub-tree that finally defines the properties of the root node. Finally, it was illustrated how countermeasures may be identified from a complete AtPT by searching for the biggest inverted tree.

Future work will firstly focus on the definition of strict rules to derive attack probability trees from the documentation supplied for conformity assessment, according to Module B and proving their correctness. For this, a general model of measuring instruments derived from [11] may be of some use. Secondly, a formalized method is still needed for identifying optimal countermeasure entry points, standardizing risk and probability measurements. This kind of development will require a sufficient amount of empirical data that could be tested by means of existing Bayesian strategies. Finally, it will be investigated how information originating from the field may be used to validate the risk assessment results.

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

[3] "ISO/IEC 18045:2008 Common Methodology for Information Technology Security Evaluation," International Organization for Standardization, Geneva, CH, Standard, September 2008, Version 3.1 Revision 4.

[4] M. Esche and F. Thiel, "Software risk assessment for measuring instruments in legal metrology," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, Lodz, Poland, September 2015, pp. 1113–1123, DOI: 10.15439/978-83-60810-66-8.

[5] ——, "Incorporating a measure for attacker motivation into software risk assessment for measuring instruments in legal metrology," in *Proceedings of the 18th GMA/ITG-Fachtagung Sensoren und Messsysteme 2016*, Nuremberg, Germany, May 2016, pp. 735 – 742, DOI: 10.5162/sensoren2016/P7.4.

[6] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *Proceedings of the 8th international conference on Information Security and Cryptology*. Seoul, Korea: IEEE, December 2005, pp. 186–198, DOI: 10.1007/11734727_17.

[7] B. Schneier, *Secrets and lies: digital security in a networked world*. Indianapolis, Indiana: Wiley Computer Publishing, 1963.

[8] 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.

[9] W.-H. Lin, P.-T. Kuo, H.-T. Lin, and T. C. W. and, "Threat risk analysis for cloud security based on attack-defense trees," in *Proceedings of the International Conference on Computing Technology and Information Management*. Seoul, Korea: IEEE, April 2012, pp. 106–111, ISBN: 978-89-88678-68-8.

[10] R. Vigo, F. Nielson, and H. R. Nielson, "Automated generation of attack trees," in *Proceedings of the IEEE Computer Security Foundations Symposium*. Seoul, Korea: IEEE, 2014, pp. 337–350, DOI: 10.1109/CSF.2014.31.

[11] "WELMEC 7.2 Software Guide," European cooperation in legal metrology, WELMEC Secretariat, Delft, Standard, 2015.

[12] "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.

[13] "ISO 11898-1:2015 Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling," International Organization for Standardization, Geneva, CH, Standard, December 2015.