Qualitative and Quantitative Analysis of CFTs Taking Security Causes into Account

Max Steiner and Peter Liggesmeyer

Chair of Software Engineering: Dependability, University of Kaiserslautern
{steiner,liggesmeyer}@cs.uni-kl.de

Abstract. Component fault trees that contain safety basic events as well as security basic events cannot be analyzed like normal CFTs. Safety basic events are rated with probabilities in an interval [0,1], for security basic events simpler scales such as {low, medium, high} make more sense. In this paper an approach is described how to handle a quantitative safety analysis with different rating schemes for safety and security basic events. By doing so, it is possible to take security causes for safety failures into account and to rate their effect on system safety.

Keywords: safety analysis, security analysis, quantitative combined analysis, component fault trees, attack trees, security enhanced component fault trees

The final publication is available at http://link.springer.com/chapter/10.1007/978-3-319-24249-1_10

1 Introduction

Embedded systems are networked more and more, they evolve into cyber-physical systems, even if the initial design did not anticipate this. This networking creates new security problems that can lead to safety problems which have to be analyzed. The effects of such security problems are not taken into account in a traditional safety analysis. Thus consequences cannot be estimated correctly, which results in either insufficient or unnecessary and too expensive countermeasures.

This paper shows how to conduct a qualitative and quantitative safety analysis using component fault trees (CFTs), including the effects of security problems on system safety. In [16] the process as a whole was described. In this current paper the focus lies on the analysis of the security enhanced component fault trees (SECFTs). To achieve that, safety analysis methods of CFTs are extended to incorporate security problems as basic causes.

The paper is structured in the following way: After a short overview of related work, the overall analysis process is recalled. Then the foundations of an analysis are set by discussing rating scheme and calculation rules. And finally the qualitative and quantitative analysis procedure is shown using an example analysis of a generic patient controlled analgesia (GPCA) pump.

2 Related Work

The SECFTs used in this approach are CFTs [11] extended with additional elements from attack trees (ATs) [15] to model the effects of security attacks on the safety of a system. Based on established analysis methods for CFTs that are described in [18], adaptations were made to encompass the analysis of safety as well as security properties.

Other works concerning quantitative analysis of ATs like Mauw et al. [12] describe general calculating rules for predicates in ATs to compute the values for the top event (TE). Jürgenson and Willemson use those rules to calculate ratings in an AT in [10]. They use a combination of probabilities and costs/gain of the attacks. Fovino et al. propose in [6] a way how to combine quantitative analysis of fault trees (FTs) and ATs under the precondition that probabilities for all basic events (BEs) are available. But determining accurate probabilities for security attacks is often difficult or sometimes even not possible [17]. To circumvent that problem, Casals et al. use a scale with discrete values to rate security attacks in [5]. By those ratings they can compare different attack scenarios. The downside is that the accuracy is not as good as with probabilities for BEs in FTs.

Therefore, we decided to use a hybrid approach for the rating of the events to avoid the problem of assigning probabilities to security-related events. The overall process of the combined analysis was described in [7] and [16]. It is based on recommendations of standards as IEC 61025 [3] or IEC 60300–3–1 [2] to use a combination of inductive and deductive techniques to minimize the potential of omitted failure modes. Inductive techniques as failure mode and effects analysis (FMEA) [9] or hazard and operability study (HAZOP) [1] can be used to find the TEs. Deductive techniques as fault tree analysis (FTA) [3] are used to refine the analysis and to find causes and moreover combinations of causes that lead to the TE. The resulting graph is used to conduct qualitative and quantitative analyses.

The approach to introduce security aspects into safety analysis proposed in this work is based on CFTs. It extends the process described earlier by an additional step and modifies the analysis step [16]. After developing the CFT, it is extended by security attacks as additional causes that also could lead to the safety-related TE. Those security attacks are found by analyzing data flow and interface specifications, because most attacks are made at communication interfaces. Techniques such as STRIDE [8] and FMVEA [14] can be used to find possible attacks.

3 Analysis

To be able to conduct a quantitative analysis, a comprehensive rating of all of the events in a security enhanced component fault tree (SECFT) has to be available. Using a comprehensive rating for all events the individual impact of an attack on the occurrence of the top event (TE) can be determined.

3.1 Ratings

In component fault trees (CFTs) typical ratings for basic events (BEs) are probabilities or reliability values. These are used to calculate the respective values for minimal cut sets (MCSs) and the TE.

In an attack tree (AT) the same basic elements exist as in a CFT. Either Boolean or continuous values can be assigned to BEs. As Boolean values pairs such as *possible – impossible* or *expensive – not expensive* are used. Continuous values for BEs can be *costs to execute an attack*, *probability of success of a given attack* or *probability that an attack is actually conducted*.

A probability that an attack is successful could be determined from expert knowledge and experienced data just like failure probabilities. But even the success probability is difficult to determine. There is only a small portion of the data about successful attacks available. Most successful attacks are not published because companies fear the loss of trust of their customers.

The bigger problem is determining the probability that an attacker actually conducts an attack. First of all, this probability depends on different aspects: the attacker's motivation and experience, availability of assets/money, and accessibility of the system. And second of all, if this attack requires several distinct actions that are modeled as separate security events, these events are not independent, as it would be required for most calculation algorithms for CFTs.

Fig. 1 shows an example of an attacker modeled as a component with two output ports out_1 and out_2. For the output port out_2 it is basically an AT which consists of 4 gates and 5 BEs. Two MCSs for out_2 are present which represent two different attack scenarios: $\{e_1, e_2, e_5\}$ and $\{e_3, e_4, e_5\}$.

If an attack is consisting of several actions an attacker has to perform, like the ones for output port out_2 in the example, these actions are not stochastically independent. If an attacker plans to attack a given system and that attack requires him to execute different actions (security events, sub-attacks), it is most likely that he will at least try all sub-attacks that are necessary to reach his goal. In terms of the given example this means if an attacker chooses to try BE e_1 and he succeeds, he most probably will also try e_2 and e_5. In general this means, in an AT the events in a MCS are not independent from each other.

Therefore, it makes more sense to assign a probability to a whole MCS, which represents the attack, instead of the BEs. The other rating values (other than probabilities) can be calculated for the MCSs from their respective BEs. For the TE the same conditions hold than for CFTs: ratings are calculated from BEs or MCSs.

A first result from a safety analysis based on SECFTs is the set of MCSs as they are all combinations of BEs that lead to the analyzed system failure. To decide which of these combinations have to be mitigated to reach a certain safety level, this set of MCSs has to be prioritized. And of course to decide whether a system is complying to a given safety level from a standard or a requirement, the TE rating of the CFT has to be calculated.

Instead of trying to assign probabilities to security events, it is a better idea to use a more coarse scale with only a few discrete values. IEC 61025 [3] states

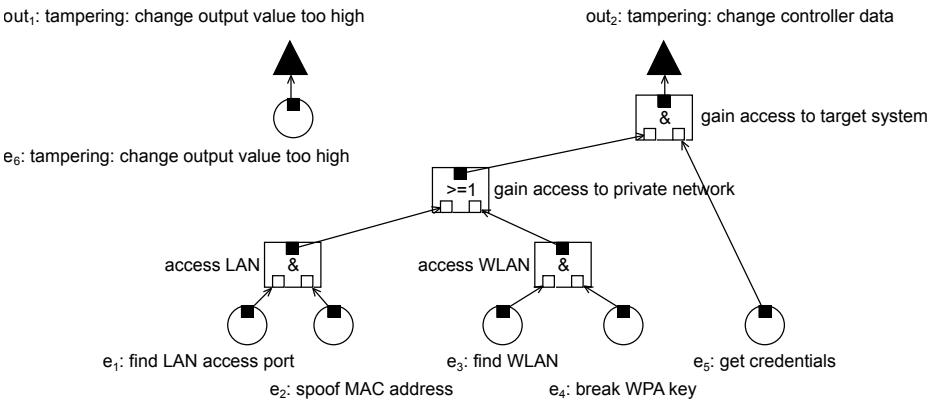

Fig. 1. Example attacker component.

for fault tree analysis that in case when probabilities cannot be assigned to BEs, a "descriptive likelihood of occurrence" can be used with values such as: "highly probable", "very probable", "medium probability", "remote probability", etc. Likelihood is defined as a qualitative probability for the occurrence of a security event. Security events are in most cases attacks conducted by an attacker. This likelihood can be used to rate security events in a SECFT.

The approach described in this paper can work with different numbers of distinct values. In the following a three-value scale is selected for simplicity. More fine-grained scales only make sense if more distinct values are needed explicitly. One has to keep in mind that assigning more precise numerical values might only add fictitious precision which can be misleading in the interpretation of the results. This also has to be considered when calculating values for combinations of events that are rated with likelihood values.

The values of that likelihood are determined from several indicators as: attack cost, attack resources, attacker motivation, accessibility, or attacker type. Casals et al. describe in [5] one possibility to determine likelihood values. The scale represents the likelihood of a security event to occur. To each value a numerical value is assigned for easier comparisons. From this follows that the likelihood would be mapped to integer values from the interval $[1, m]$, where $m \in \mathbb{N}$ is the number of discrete values.

One possibility to achieve a common rating, other than probabilities, is to use the likelihood for both safety and security events. The advantage of this approach is that values for all BEs can be determined relatively easy and comparisons of likelihood are easily performed. The disadvantage is that the accuracy coming from rating safety events with probabilities is lost.

To use the advantages of both, probabilities for safety events and likelihood for security events, an approach using a combination of both probability and likelihood is used. Hence in a SECFT there can be both likelihoods and probabilities for different events.

When MCSs are determined in a SECFT that includes safety as well as security events, there can be three types of MCSs as defined in the following:

Definition 1 *MCS types:*

1. *A safety MCS contains only safety events (BEs which occur due to random faults in the system).*
2. *A security MCS contains only security events (BEs which are triggered from outside the system by a malicious attacker or a random fault).*
3. *A mixed MCS contains safety events as well as security events.*

The TE will most certainly depend on safety as well as security events. Therefore a combination of both probabilities and likelihood is needed to calculate ratings for MCSs and TEs.

Events in a safety MCS are rated with probabilities. Therefore, the overall rating of a safety MCS is also a probability. Events in a security MCS are rated with likelihoods. So the overall rating of a security MCS is also a likelihood. In a mixed MCS however, there are both probabilities and likelihoods. As they are not directly comparable, the rating of a mixed MCS is a tuple consisting of the overall probability of all included safety events and the overall likelihood of all included security events. For TEs in a SECFT, the same holds as for mixed MCSs.

The next section will introduce the extensions for the calculation rules needed for a SECFT to handle the tuples of probabilities and likelihoods.

3.2 Calculation Rules for Likelihood and Probability Values in SECFTs

For the calculation of the ratings from Section 3.1 at least calculation rules for the gates `AND`, `OR`, and `NOT` are required. Other gates such as `XOR` or voter gates can be constructed from these three basic gates. Their calculation rules result from the combination accordingly.

Definition 2 (Likelihood of an `AND`-gate) *All subordinate events have to occur in order that the combined event occurs. Therefore, the event with the lowest likelihood determines the likelihood L of the combined event. This is explained by the fact, that if all events of an `AND`-gate have to occur, the one with the lowest likelihood also has to occur, which then determines the overall likelihood of the `AND`-gate.*

Definition 3 (Likelihood of an `OR`-gate) *At least one subordinate event has to occur in order that the combined event occurs. If there are alternatives to attack a system to trigger the same event, an attacker will execute the one that has the highest outcome while requiring the lowest effort. In other words he will execute the attack action with the highest likelihood.*

Definition 4 (Likelihood of a `NOT`-gate) *A subordinate event must not occur in order that the resulting event occurs. If the likelihood L is defined as an interval* $[1, m]$ *of integer values with* $m \in \mathbb{N}$*, the value of a* ***`NOT`*** *gate is defined as follows:* $L(\overline{A}) = (m + 1) - L(A)$

The outcome of `AND`, `OR` and `NOT` gates with independent input events A, B, or more general n independent input events X_i, is calculated as follows in Table 1 with $i, n, m \in \mathbb{N}$:

Table 1. Probability and likelihood calculation for `AND`, `OR`, and `NOT`-gates.

	probability	likelihood
`AND`	$P(A \wedge B) = P(A) \cdot P(B)$	$L(A \wedge B) = \min[L(A), L(B)]$
	$P\left(\bigwedge_{i=1}^{n} X_i\right) = \prod_{i=1}^{n} P(X_i)$	$L\left(\bigwedge_{i=1}^{n} X_i\right) = \min_{i=1}^{n}[L(X_i)]$
`OR`	$P(A \vee B) = P(A) + P(B) - P(A) \cdot P(B)$	$L(A \vee B) = \max[L(A), L(B)]$
	$P\left(\bigvee_{i=1}^{n} X_i\right) = 1 - \prod_{i=1}^{n}(1 - P(X_i))$	$L\left(\bigvee_{i=1}^{n} X_i\right) = \max_{i=1}^{n}[L(X_i)]$
`NOT`	$P(\overline{A}) = 1 - P(A)$	$L(\overline{A}) = (m + 1) - L(A)$

If the `NOT` gate is used it has to be considered that it has an unusual semantics in CFTs: The lower the probability/likelihood of occurrence of an event that is attached to a `NOT` gate, the higher is its effect on the TE, and vice versa. Whereas normally high probabilities of single events lead to a higher probability for the TE. From this follows that to reduce the TE probability/likelihood, an event or even a whole component that is connected via a `NOT` has to fail with high probability/likelihood.

To have a uniform rating scheme over all events in a CFT, all ratings of BEs are interpreted as tuples (P, L), where P is a probability and L a likelihood. For safety events there is no likelihood leading to $(P_e, -)$ with an undefined L_e, and for security events there is no probability value leading to $(-, L_e)$ with an undefined P_e. Undefined values will be ignored in the calculation of the rating.

This has to be explained further: The alternative to undefined values would be values that do not influence the order between the events. To achieve this, an *identity element* or *neutral element* for all possible gate-operations would be needed. This would mean in terms of probabilities, a value is needed, which is the identity element for addition and multiplication. Such a value does not exist because the identity element for the addition is 0, and the identity element for the multiplication is 1. The same problem arises for the likelihood operations: The identity element of the min-function is the maximum value, and the identity element of the max-function is the minimum value. These values exclude each other, so no value is selected and the undefined values are ignored during the calculation.

The tuple elements of a combination of events by logic gates are calculated independent of each other according to the rules established earlier. The following example illustrates this in more detail. Fig. 2 shows a high-level view of a

SECFT of a generic patient controlled analgesia pump [4] with all modeled components. The security part is inspired from the security flaw of the infusion pump Hospira PCA3 that was published recently [13]. To include the attack, extra input ports were added to vulnerable components. The actual attack is modeled in component `Attacker`. Suitable ratings for all basic events were chosen. The required Safety Integrity Level for the individual components was estimated and used as order of magnitude for the rating of the safety events. The rating of the security events was chosen due to the simplicity of the physical access (attach an Ethernet cable) and telnet access (no authentication necessary to get root access). The likelihood for the security events in this example is a three-level scale of {low, medium, high} with corresponding values of {1,2,3}. The model has 7 MCSs. Their resulting ratings are shown in Table 2.

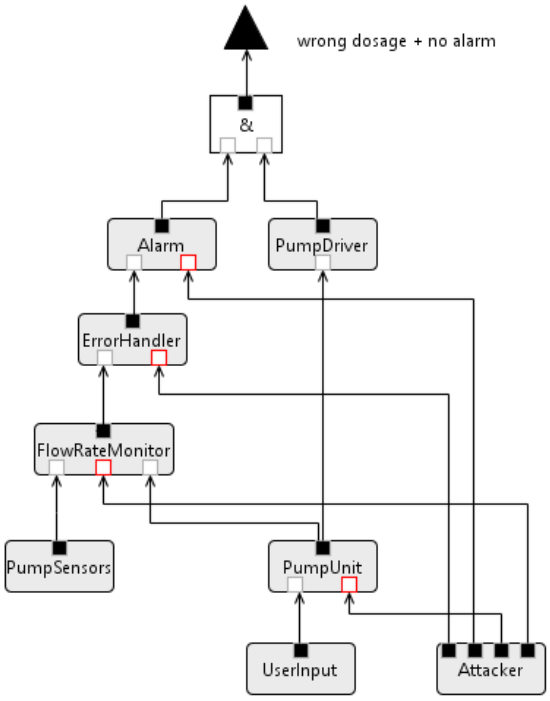

Fig. 2. High-level SECFT model of a generic infusion pump.

The rating of the TE can be calculated as a conservative estimate from a disjunction of all MCSs. The undefined values P_2 and $L_1, L_3, L_4, L_5, L_6, L_7$ are

Table 2. Minimal cut sets and ratings.

id	basic events	BE rating	MCS rating
1	PumpUnit.pump unit sets wrong values	10^{-7}	$(10^{-7}, -)$
2	Attacker.physical access the Ethernet interface of the pump	3 (high)	$(\ -\ , 3)$
	Attacker.access telnet service of the pump	3 (high)	
3	UserInput.user sets wrong values	10^{-6}	$(10^{-13}, -)$
	PumpUnit.check of user input fails	10^{-7}	
4	PumpDriver.pump driver fails	10^{-8}	$(10^{-15}, -)$
	Alarm.alarm fails	10^{-7}	
5	PumpDriver.pump driver fails	10^{-8}	$(10^{-15}, -)$
	ErrorHandler.error handler fails	10^{-7}	
6	PumpDriver.pump driver fails	10^{-8}	$(10^{-15}, -)$
	FlowRateMonitor.flow rate monitor fails	10^{-7}	
7	PumpDriver.pump driver fails	10^{-8}	$(10^{-15}, -)$
	PumpSensors.sensors provide wrong values	10^{-7}	

ignored in this calculation.

$$\begin{aligned} P_{TE} &= P\left(\bigvee_i MCS_i\right) \\ &= 1 - \prod_i (1 - P(MCS_i)), i \in 1,3,4,5,6,7 \\ &= 1.00000103975262 \cdot 10^{-7} \\ L_{TE} &= \max(L(MCS_2)) \\ &= \max(3) = 3 \end{aligned}$$

That results in a rating of $(P_{TE}, L_{TE}) = (1.00000103975262 \cdot 10^{-7}, 3)$ for the TE *wrong dosage and no alarm*.

3.3 Qualitative Analysis

The most important activity of a qualitative analysis in CFTs and SECFTs is the determination of MCSs. MCSs also are used to derive a coarse classification of the criticality of failure scenarios and BEs, and they allow to make statements about the general endangerment of the analyzed system. MCSs are also an important starting point for a following quantitative analysis. Based on the MCSs also a basic importance analysis of BEs and MCSs can be conducted.

This Section deals with necessary extensions of the qualitative analysis of CFTs to cope with additional security events in the SECFT. The first step of the analysis is the determination and analysis of the MCSs. The interpretation of a MCS is the same as in CFTs: a MCS is a minimal safety failure scenario (but possibly depending also on security attacks). A CFT (and therefore a SECFT as well) can be transformed into a set of MCSs that represents all failure scenarios which are relevant for the system. In general, a tree contains multiple MCSs corresponding to different failure scenarios.

In addition to size, an analysis of MCSs of a SECFT takes also the type of the MCSs into account. The result of a qualitative analysis are ordered lists of MCSs.

As discussed in detail in Sections 3.1 and 3.2, ratings of safety and security events cannot be compared directly. Therefore, it makes sense to sort them according to safety events and security events. Then, one receives three lists of MCSs (Definition 1):

1. safety MCS
2. security MCS
3. mixed MCS

Safety MCSs are analyzed as usual: A qualitative analysis starts with an ordering according to the size of the MCS. The smaller a MCS the more critically it should be analyzed. This is explained by the fact that all events in a MCS have to occur, so that the TE occurs and the system fails. The lesser events have to occur, the more the TE depends on individual events. So events in smaller MCSs deserve more attention in an analysis. An especially critical case is a MCS with only one event – a single point of failure which itself can directly lead to the system failure.

Security MCSs are a more special case: In this case a system failure only depends on external influences and does not depend on failures of internal system components. Pure security MCSs are not more critical per se than pure safety MCSs, but the uncertainty of the modeling of an attack is relatively high. Depending on the threat scenario and the attacker type the likelihood value changes. Necessary tools become better available and cheaper over time which can make an attack more probable in the future. Also, the attacker type, the attacker's motivation and capabilities can and will change over time – potentially to the disadvantage of the system. This is why pure security MCSs should be avoided by adding suitable countermeasures which convert security MCSs to mixed MCSs.

Mixed MCSs on the other hand can be managed better: For the occurrence of the TE all events of a mixed MCS have to occur, which means regular safety events have to occur. These occurrences of safety events can be managed with the usual methods like redundancy or monitoring. The probability for a mixed MCS to cause the TE has an upper bound: the probability of the contained safety events. This way the criticality of security events can be mitigated by safety events with low probability. That means, the more statistically independent safety events a MCS contains the less probable it is to cause the TE.

To summarize the qualitative analysis of MCSs: There are three types of MCSs which differ in the level of controllability of their BEs. Controllability in this context means how much a failure scenario (a MCS) depends on faults of the system as opposed to external factors as e.g. attacks. In descending order according to their controllability these are: safety MCSs, mixed MCSs and security MCSs. Resulting from that, additionally to MCSs containing only one event (single points of failure) also plain security MCSs should be avoided by adding more (safety) BEs. Also, the more MCSs exist in a given SECFT, the

more opportunities for the TE exist, which indicates a higher vulnerability of the system with respect to this TE.

Another goal of an analysis is to determine the importance of BEs. The importance shows how much of an impact a BE has on a TE. BEs that are part of more than one MCS are more important than the ones that are only part of one MCS. But the size of MCSs is also a factor. BEs in smaller MCSs are more important than the ones in larger MCSs. More accurate importance analysis is possible within a quantitative analysis.

3.4 Quantitative Analysis

A quantitative analysis is conducted if more accurate statements about the system safety are necessary than the results from a qualitative analysis, which are mainly the determination and preliminary ordering of MCSs. A quantitative analysis, therefore, has several goals [18,3]:

- to determine the rating of the TE under consideration to compare it to the given requirements from standards or customers,
- to determine ratings of the individual MCSs to determine the MCS that has the biggest contribution to the TE (the most probable failure scenario),
- and derived from the previous ones: to determine where countermeasures would have the most effect.

A quantitative analysis of a SECFT starts with a quantitative evaluation of its MCSs. The first step here is to assign probabilities to safety events and likelihoods to security events. (During the assignment of likelihood values to security events it should be kept in mind that those security events belonging to the same MCS can influence each other.)

After the determination of the MCSs there are two possibilities to order them: according to size and type (see qualitative analysis in Section 3.3) or according to type and ratings (probability and likelihood). An ordering simply according to the ratings is not possible for all MCSs in general because of the incomparability of probabilities and likelihoods (see also Section 3.1). For each MCS a tuple rating (P, L) is calculated according to the rules described in Section 3.2. For probabilities this means the value for the MCS is the product of all probabilities of the contained events. (Under the precondition that all events are independent, which is usually given for safety events.) For the likelihood of a MCS the minimum of all likelihoods of the included events is determined.

Each type of MCSs can be ordered by itself. To compare two minimal cut sets MCS_1 and MCS_2 with tuple ratings (P_1, L_1) and (P_2, L_2), the ordering has to be prioritized either according to probability or to likelihood. The resulting ordered list of MCSs reflects the relative criticality of the represented failure scenarios. Higher ratings here correspond to a higher criticality and vice versa. To find out if the system complies with the given requirements, the list of MCSs is filtered according to the requirements (e.g.: "show me all MCSs with size ≤ 2", "$P > 10^{-7}$" or "$L \geq 2$(medium)"). The results are the failure scenarios that require countermeasures.

As mentioned earlier, requirements can define boundary values for MCSs in size or rating, but usually the main requirement is a boundary value for the rating of the TEs: "the system shall not fail with a probability more than ..." The TE probability can be calculated as the sum of the probabilities of all MCSs if only AND and OR gates are used. This defines an upper bound for the probability:

$$P(TE) \leq \sum_{i=1}^{n} P(MCS_i) \qquad , i, n \in \mathbb{N}, n \text{ number of MCSs} \tag{1}$$

The other variant is to calculate $P(TE)$ using the binary decision diagram (BDD) algorithm which returns the exact probability value. To adapt the BDD algorithm to SECFTs only the BEs with an assigned probability value are considered for the calculation as already discussed in Section 3.2.

The likelihood of the TE $L(TE)$ is simply calculated as the maximum of the likelihoods of all MCSs as defined in the equations for the `OR`-gate:

$$L(TE) = L\left(\bigvee_{i=1}^{n} X_i\right) = \max_{i=1}^{n}[L(X_i)] \quad , i, n \in \mathbb{N}, n \text{ number of MCSs} \tag{2}$$

With the described extensions of the calculation rules and the different types of MCSs SECFTs can be used to conduct safety analysis with additional consideration of security problems.

4 Conclusion

Based on SECFTs a qualitative and quantitative safety analysis is extended to include influences of security problems on the safety of a system. To avoid the problem how to assign probabilities to security events, a scale of discrete values (e.g. {low, medium, high}) is used to rate security events while retaining the higher accuracy of probabilities for safety events. Existing analysis techniques are extended to work with probabilities for safety events as well as discrete likelihoods for security events. As a result, a hybrid rating scheme is used to rank the different MCSs according to the tuple of probability and likelihood, and to calculate TE ratings that can be used to check the compliance of requirements.

Acknowledgement. The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement n° 621429 (project EMC^2) and from the respective national funding authorities.

References

1. IEC 61882: Hazard and operability studies (HAZOP studies) — Application guide (2001)
2. IEC 60300-3-1: Dependability management - Part 3-1: Application guide; Analysis techniques for dependability; Guide on methodology (May 2005)

3. IEC 61025: Fault tree Analysis (FTA) (2006)
4. Arney, D., Jetley, R., Zhang, Y., Jones, P., Sokolsky, O., Lee, I., Ray, A.: The generic patient controlled analgesia pump model. Website (2009), http://rtg.cis.upenn.edu/gip.php3
5. Casals, S.G., Owezarski, P., Descargues, G.: Risk assessment for airworthiness security. In: Ortmeier, F., Daniel, P. (eds.) Computer Safety, Reliability, and Security, Lecture Notes in Computer Science, vol. 7612. Springer Berlin Heidelberg (2012)
6. Fovino, I.N., Masera, M., Cian, A.D.: Integrating cyber attacks within fault trees. Reliability Engineering and System Safety 94, 1394–1402 (2009)
7. Förster, M., Schwarz, R., Steiner, M.: Integration of modular safety and security models for the analysis of the impact of security on safety. Tech. rep., Fraunhofer IESE, Technische Universität Kaiserslautern (2010), http://publica.fraunhofer.de/dokumente/N-151512.html
8. Hernan, S., Lambert, S., Ostwald, T., Shostack, A.: Uncover security design flaws using the stride approach. MSDN Magazine (nov 2006), http://msdn.microsoft.com/en-us/magazine/cc163519.aspx
9. IEC/TC 56 Reliability and maintainability: IEC 60812: Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA) (Jan 2006)
10. Jürgenson, A., Willemson, J.: Computing exact outcomes of multi-parameter attack trees. In: On the Move to Meaningful Internet Systems (2008)
11. Kaiser, B., Liggesmeyer, P., Mäckel, O.: A new component concept for fault trees. In: 8th Australian Workshop on Safety Critical Systems and Software. Canberra (oct 2003), http://dl.acm.org/citation.cfm?id=1082051.1082054
12. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Information Security and Cryptology - ICISC 2005 (2006)
13. Scherschel, F.: Root-Shell im Krankenhaus: Hospira-Infusionspumpe mit Telnet-Lücke. Website (2015), http://heise.de/-2633529
14. Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E.: Security application of failure mode and effect analysis (fmea). In: Bondavalli, A., Di Giandomenico, F. (eds.) Computer Safety, Reliability, and Security, Lecture Notes in Computer Science, vol. 8666, pp. 310–325. Springer International Publishing (2014)
15. Schneier, B.: Attack trees. Dr. Dobb's Journal (dec 1999), http://www.schneier.com/paper-attacktrees-ddj-ft.html
16. Steiner, M., Liggesmeyer, P.: Combination of safety and security analysis - finding security problems that threaten the safety of a system. In: ROY, M. (ed.) Proceedings of Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems) of the 32nd International Conference on Computer Safety, Reliability and Security (2013), http://hal.archives-ouvertes.fr/hal-00848604
17. Verendel, V.: Quantified security is a weak hypothesis: a critical survey of results and assumptions. In: NSPW '09: Proceedings of the 2009 workshop on New security paradigms workshop. pp. 37–50. ACM, New York, NY, USA (2009)
18. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: Fault Tree Handbook. U.S. Nuclear Regulatory Commission (1981)