HAL
open science

# Combination of Safety and Security Analysis - Finding Security Problems That Threaten The Safety of a System

Max Steiner, Peter Liggesmeyer

## To cite this version:

Max Steiner, Peter Liggesmeyer. Combination of Safety and Security Analysis - Finding Security Problems That Threaten The Safety of a System. SAFECOMP 2013 - Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems) of the 32nd International Conference on Computer Safety, Reliability and Security, Sep 2013, France. pp.NA. hal-00848604

**HAL Id: hal-00848604**

**https://hal.science/hal-00848604v1**

Submitted on 26 Jul 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combination of Safety and Security Analysis - Finding Security Problems That Threaten The Safety of a System

Max Steiner and Peter Liggesmeyer

AG Software Engineering: Dependability, TU Kaiserslautern,
{steiner,liggesmeyer}@cs.uni-kl.de

**Abstract.** In most cases in a safety analysis the influences of security problems are omitted or even forgotten. Because more and more systems are accessible from outside the system via maintenance interfaces, this missing security analysis is becoming a problem. This is why we propose an approach on how to extend the safety analysis by security aspects. Such a more comprehensive analysis should lead to systems that react in less catastrophic ways to attacks.

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

## 1 Introduction

Embedded systems that influence their environment, like factory control systems, have to be analyzed for safety to be certified by authorities. Security usually is not made an issue in embedded systems, because security is often only associated with data security in systems that store sensitive data. Also through the growing integration and networking of systems, especially by the Internet, new security problems arise in previously secure systems.

In this paper an approach will be described for the analysis of safety of a system considering the influence of security problems on system safety. As can be seen in recent incidents, like the worm Stuxnet or the attack scenario on airplane control systems[1], there are systems in which security flaws can lead to catastrophic system failures. These failures could be avoided if the influence of security problems on system safety is taken into account during the development of those systems.

In the following a process will be described how to conduct such an analysis using component fault trees (CFTs) and attack trees (ATs). A CFT will be extended by ATs which model attacks that can cause events in the CFT. Furthermore it will be shown how to adapt qualitative and quantitative analysis methods to such a combined tree.

[1] http://conference.hitb.org/hitbsecconf2013ams/hugo-teso/

## 2 Related Work

The basis of this work are CFTs as introduced in [4] and ATs [6]. CFTs are an extension of fault trees with an additional focus on system components and reusability. A system usually consists of several components which by themselves can consist of subcomponents. A CFT models one on these components following the component hierarchy of the system. CFTs model all failure modes of the component at once. That means a CFT can have more than one top level event. These top level events are also the outports of a component. The outports of one component can be connected to inports of other components. Despite these differences, the same analyses are possible as in a fault tree (FT). The component-wise construction of CFTs allows easier modeling of large systems than with FTs.

ATs were introduced by Schneier in [6]. They are similar to FTs concerning the structure. Instead of the probabilities of occurrence in FTs Schneier used values like attack costs, probability of success of a given attack and the likelihood that an attacker will try a given attack. In [5] Mauw et al. describe general rules for calculating with predicates in ATs to compute the values for the top level event.

In an analysis using a CFT with events that are caused by attacks that are detailed in an AT, this leads to the problem how to combine the different values. Fovino et al. propose in [1] a way how to combine FTs and ATs under the precondition that probabilities for both are available. Probabilities for attack events are usually not available. Even if they are available, one has to keep in mind that attack events are not necessarily stochastically independent as events in a FT should be.

In [2] we described how safety analysis and security analysis can be combined in general. We decided to use a hybrid approach for the rating of the events to avoid the problem of assigning probabilities to security-related events.

## 3 Analysis Process

In this section it will be described how to conduct a safety analysis using CFTs considering the influence of security problems on safety of a system. The first step is to develop a CFT for the system under study. The second step is to extend this CFT to consider security problems. The resulting tree is used to conduct qualitative and quantitative analyses.

### 3.1 Develop Component Fault Tree

The first thing to do for the analysis is to develop a CFT for the system to be analyzed. Per system component one CFT is modeled. The resulting CFTs are connected via their in- and out-ports. Out-ports are the top level events of the respective components. In-ports are basic events coming from other components. Since a component can have several top level events it can also have more than one out-port.

### 3.2 Extend Component Fault Tree

If the CFTs exist, the next step is to extend it to include security concerns which influence the safety of the system. To do that, additional information about the system is needed. Information about the data flow in the system and between the system and its environment is useful for security analysis because communication interfaces are preferred targets for an attack. This is why the specifications for the communication interfaces are also required.

To extend the tree systematically, it is necessary to go through it starting with basic events searching for events that can also be caused by directed manual interventions. Components that deserve special interest are the ones with interfaces to the environment of the system as they are the ones that are most probably attacked. To find out which attacks are possible, the STRIDE classification is used [3]. STRIDE maps threats to security properties: Spoofing—Authentication, Tampering—Integrity, Repudiation—Non-repudiation, Information Disclosure—Confidentiality, Denial of Service—Availability, Elevation of Privilege—Authorization. From this list the properties that fit to the system are analyzed further. Also, communication between two system components that goes through a channel through the environment has to be taken under special consideration. No events should be left out from the beginning unless the causes for that are documented in the tree or in a separate document that is linked to the tree. In the following, events in a CFT that are caused by internal faults will be called *safety events* (the traditional events in a CFT). Events that are caused by an outside influence on the system will be called *security events*.

If an event is found that can also be caused by an attack, the tree is extended by an OR gate to which the previous subtree and the new security event are attached (see Fig. 1).

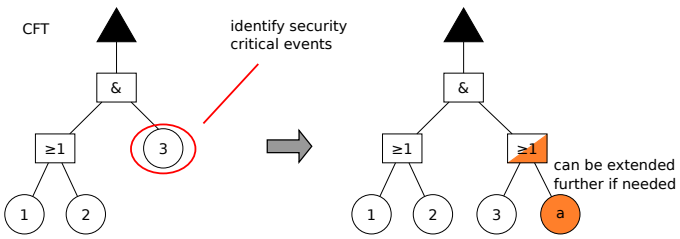


**Fig. 1.** Extension of a CFT.

When the whole tree is analyzed the result is a CFT that contains safety events as well as security events.

### 3.3 Qualitative Analysis

After extending the CFT, a qualitative safety analysis can be conducted. The result of such an analysis are ordered lists of minimal cut sets (MCSs). A cut

set is a set of basic events which together cause the top level event of the tree. A cut set is called MCS if it has no subsets that are also cut sets. The first step is to calculate the MCSs per top level event. The second step is to sort the MCSs according to their size. The smaller the MCSs, the more critical they are because lesser basic events have to happen to cause the top level event.

It can also make sense to sort them according to safety events and security events. Then, one receives three lists of MCSs: MCSs containing only safety events, MCSs containing only security events and MCSs containing both.

MCSs containing only security events can be seen as more critical than the rest under certain circumstances. The top level event only depends on actions from outside the system which cannot be influenced by the current system itself. The safety of the system depends on attacker capabilities and motivation which can change over time. Necessary tools become better available and cheaper over time which can make an attack more probable in the future. Countermeasures against that can be additional system components that add safety events to those security MCSs to create mixed MCSs.

The probability for a mixed MCS to cause the top level event 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.

The probability of statistically independent safety events is multiplied to obtain the probability of the top level event. That means, the more statistically independent safety events a MCS contains the less probable it is to cause the top level event. With this in mind, adding more events to a MCS increases the safety of the system without knowing the exact probabilities.

To summarize the qualitative analysis, MCSs containing only one event (single points of failure) should be avoided by adding more (safety) events to these MCSs.

### 3.4 Quantitative Analysis

If the results from the qualitative analysis are not enough, a quantitative analysis can be done. The first step here is to assign values to basic events. These can be probabilities for safety events. For security events, probabilities are not feasible because statistical data is not available or conditions are changing depending on attacker capabilities and motivation. Another reason is that events in one MCS are usually statistically dependent on each other. This is why security events are rated using a simple ordinal scale like {low, medium, high} or {0,1,2} or something similar. A high rating corresponds with a high probability of occurrence and a low rating with a low one. For the creation of the rating different attributes of an attack can be used, like costs, attacker capabilities, attacker motivation, simplicity, access difficulty, etc.

The next step is to calculate the compound values for each MCS. 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 rating of a MCS the minimum of all ratings of the included events is determined.

Coming back to our three classes of MCSs: safety, security and mixed. Safety MCSs have a probability $P$, security MCSs have a rating $R$, and mixed MCSs have a tuple of probability and rating $(P, R)$. $P$ is calculated from the individual probabilities of the included safety events and $R$ is calculated from the individual ratings of the included security events.

Each class of MCSs can be ordered by itself. The tuples of the mixed MCSs can be ordered first by probability or by rating. A complete order is not possible for all cases (see Table 1).

**Table 1.** Conditions for an order of mixed MCSs according to two tuples $t_1 = (P_1, R_1)$ and $t_2 = (P_2, R_2)$.

| | $P_1 < P_2$ | $P_1 = P_2$ | $P_1 > P_2$ |
|---|---|---|---|
| $R_1 < R_2$ | $\texttt{MCS}_1 < \texttt{MCS}_2$ | $\texttt{MCS}_1 < \texttt{MCS}_2$ | undefined |
| $R_1 = R_2$ | $\texttt{MCS}_1 < \texttt{MCS}_2$ | $\texttt{MCS}_1 = \texttt{MCS}_2$ | $\texttt{MCS}_1 > \texttt{MCS}_2$ |
| $R_1 > R_2$ | undefined | $\texttt{MCS}_1 > \texttt{MCS}_2$ | $\texttt{MCS}_1 > \texttt{MCS}_2$ |

## 4 Example

In this section the analysis process from Section 3 is demonstrated by an example analysis. As example, a part of an adaptive cruise control (ACC) system was chosen (see also [7]). The analyzed scenario is: vehicle VB follows vehicle VA and VB drives against VA due to a failure in the ACC.

Four wheel speed sensors provide the own speed of the vehicle to the speedometer (SM). The distance to the other vehicle is provided by front and rear distance sensors. Each vehicle receives the measured velocity and distance values from the other vehicle by an antenna component. Distance values and foreign velocities are used by the communication system (CS) which provides them to the control logic unit.

From the distance and the velocity the new target velocity is calculated by the ACC component which then increases or decreases the vehicle speed. The received distance and the measured one are fused in the CS. The received velocity is used as is.

In Fig. 2 one can see the high level overview of the CFT model of the ACC. As an example CFT the front antenna component is highlighted. This CFT contains two top level events: `velocity_too_high` (the velocity of vehicle VB) and `distance_too_high` (the measured distance is higher than the actual distance) which can cause vehicle VB to drive against VA. Both top level events are directly caused by safety basic events: the received values from the other vehicle (white circles).

The CFT is extended by security events (shaded circles). They are additional causes for the top level events and are connected to the CFT via OR-gates. Both values (velocity and distance) can be manipulated by an attacker by sending a

tampered radio signal (transmission of noise) or spoofing the antenna component.

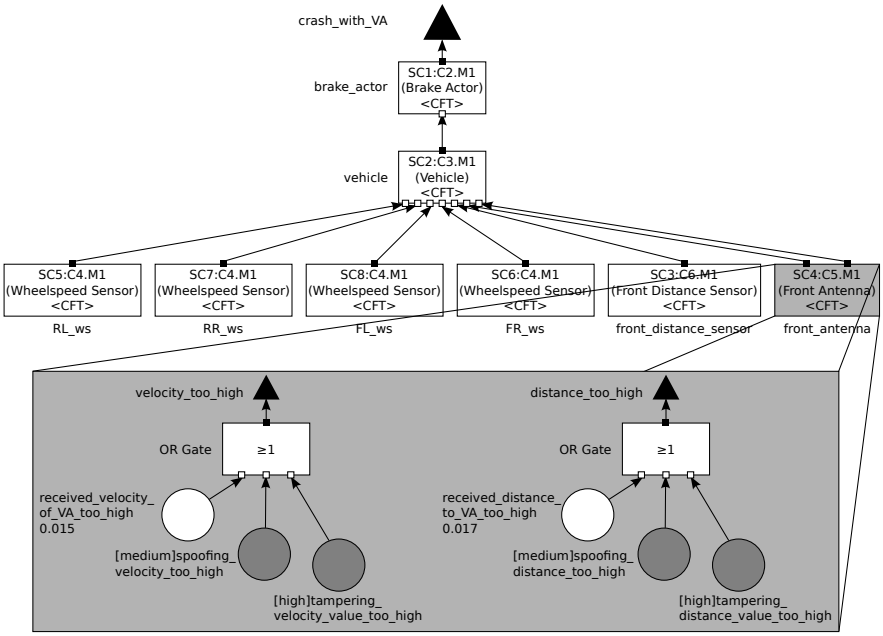


**Fig. 2.** CFT with security events of the ACC.

In a qualitative analysis of the CFT for the top level event `crash_with_VA` 82 MCSs were found. 14 MCSs contained only one basic event and 68 contained two basic events. Single event MCSs or single points of failure are especially critical and should be avoided by altering the system to add more basic events into those MCSs.

The classification according safety and security events results in 2 MCSs consisting of safety events and 14 of security events for MCSs of size 1. Of the MCSs of size 2, 26 consist of safety events, 12 of security events, and 30 MCSs of safety and security events.

In Table 2, MCSs of size 1 with elements from the example MCS from Fig. 2 are shown. Table 3 lists the MCSs of size 2 corresponding to the example MCS.

The values for the safety events for the quantitative analysis are taken from [7]. Security events were rated using an estimation for difficulty of access to the system and difficulty of conducting the attack (high difficulty results in a low rating). The final rating is the minimum of both values. The resulting rating values for the MCSs of the example are listed in Tables 2 and 3.

The MCSs in the tables are sorted first by security, mixed and safety MCSs and then by their rating. Without further knowledge how to compare security

**Table 2.** MCSs with one element from the example of Fig. 2.

| MCS | Event | Rating |
|---|---|---|
| 9 | front antenna.tampering velocity value too high | high |
| 7 | front antenna.spoofing velocity too high | medium |
| 8 | front antenna.received velocity of VA too high | 1.50E-02 |

ratings and safety probabilities an order considering both security and safety at the same time is not possible. The results show that the velocity value coming from the front antenna is critical because it can be tampered with easily and it can be measured wrong with a high probability. It is even a single point of failure and can lead to a crash of two vehicles. The other value in the front antenna component, the distance to the other car, is not as critical as the velocity because it is measured additionally by the own car. But it is shown that both values can be manipulated by an attacker and there are no countermeasures in the shown system design.

## 5 Conclusion

In this paper the process of an extended safety analysis was shown which considers influences of security problems on the safety of a system. To accomplish that, CFTs were extended by ATs to model attacks that can cause system failures. Qualitative and quantitative analysis methods were adapted to the new combined tree. The problem of the missing or hard to obtain probabilities for security events was avoided by the use of a hybrid rating scheme: probabilities for safety events and a simple rating (low, medium, high) for security events. The resulting analysis should give a more comprehensive analysis of embedded systems than the classical safety analysis.

**Acknowledgement.** This work was funded by the German Ministry of Education and Research (BMBF) in the context of the "Virtuelle und Erweiterte Realität für höchste Sicherheit und Zuverlässigkeit Eingebetteter Systeme – Zweite Phase" (ViERforES II) project.

## References

1. Fovino, I.N., Masera, M., Cian, A.D.: Integrating cyber attacks within fault trees. Reliability Engineering and System Safety 94, 1394–1402 (2009)
2. 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)
3. Hernan, S., Lambert, S., Ostwald, T., Shostack, A.: Uncover security design flaws using the stride approach (November 2006), `http://msdn.microsoft.com/en-us/magazine/cc163519.aspx`

**Table 3.** MCSs with two elements from the example of Fig. 2.

| MCS | Events | Rating |
|---|---|---|
| 45 | front distance sensor.denial of service (jamming)<br>front antenna.tampering distance value too high | high |
| 41 | front distance sensor.spoofing sensor values<br>front antenna.tampering distance value too high | medium |
| 42 | front distance sensor.spoofing sensor values<br>front antenna.spoofing distance too high | medium |
| 46 | front distance sensor.denial of service (jamming)<br>front antenna.spoofing distance too high | medium |
| 43 | front distance sensor.spoofing sensor values<br>front antenna.received distance to VA is too high | (0.017, medium) |
| 47 | front distance sensor.denial of service (jamming)<br>front antenna.received distance to VA is too high | (0.017, high) |
| 49 | front distance sensor.echo time too high<br>front antenna.spoofing distance too high | (0.001, medium) |
| 48 | front distance sensor.echo time too high<br>front antenna.tampering distance value too high | (0.001, high) |
| 52 | front distance sensor.assumed sonic velocity too high<br>front antenna.spoofing distance too high | (0.00002, medium) |
| 51 | front distance sensor.assumed sonic velocity too high<br>front antenna.tampering distance value too high | (0.00002, high) |
| 50 | front distance sensor.echo time too high<br>front antenna.received distance to VA is too high | 1.70E-05 |
| 53 | front distance sensor.assumed sonic velocity too high<br>front antenna.received distance to VA is too high | 3.40E-07 |


4. Kaiser, B., Liggesmeyer, P., Mäckel, O.: A new component concept for fault trees. In: 8th Australian Workshop on Safety Critical Systems and Software. Canberra (October 2003), `http://dl.acm.org/citation.cfm?id=1082051.1082054`
5. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Information Security and Cryptology - ICISC 2005 (2006)
6. Schneier, B.: Attack trees. Dr. Dobb's Journal (December 1999), `http://www.schneier.com/paper-attacktrees-ddj-ft.html`
7. Spang, S., Adler, R., Hussain, T., Eschbach, R.: Scrutinizing the impact of security on safety on an communicating vehicle platoon. Tech. rep., Fraunhofer IESE (2010)