Security Assessments of Safety Critical Systems Using HAZOPs

Rune Winther[1], Ole-Arnt Johnsen[2], and Bjørn Axel Gran[3]

[1] Faculty of Computer Sciences, Østfold University College,
Os Allé 11, N-1757 Halden, Norway.
rune.winther@hiof.no
[2] moreCom, Norway.
oaj@morecom.no
[3] Institute for Energy Technology, Norway.
bjorn.axel.gran@hrp.no

Abstract. Concerned with serious problems regarding security as a safety issue, a HAZOP specifically suited for identifying security threats has been developed. Unfortunately, the emphasis placed on security issues when developing safety critical systems is to often inadequate, possibly due to the lack of "safety-compliant" security methods. Having had the opportunity to adapt the HAZOP-principle to the security context, a HAZOP was established which is well-suited for handling security issues in a *safety* context. Indeed, since the main modification of the method consists of establishing new guidewords and attributes, it is quite possible to handle security issues as part of the traditional hazard analysis. In addition, while presenting the modified HAZOP-method, its use on safety related systems will be demonstrated.

1 Introduction

Increasing dependence on programmable equipment (or Information and Communication Technology, ICT) is a well known fact. Systems used in, for example, transportation and process control systems involve exposure to the risk of physical injury and environmental damage. These are typically referred to as *safety-related* risks. The increased use of ICT-systems, in particular combined with the tendency to put "everything" on "the net", gives rise to serious concerns regarding security[1], not just in relation to confidentiality, integrity and availability (CIA), but also as a possible cause of safety problems. With the increasing dependence on ICT-systems saboteurs are likely to use logical bombs, viruses and remote manipulation of systems to cause harm. Some simple examples illustrate the seriousness:

- The result of a HIV-test is erroneously changed from positive to negative due to a fault in the medical laboratory's database system.

[1] Security is in this context interpreted as the systems ability to uphold confidentiality of information, integrity of information/systems and availability of information/services [5].

- The next update of autopilot software is manipulated *at the manufacturers site*.
- The corrections transmitted to passenger airplanes using differential GPS (DGPS) as part of their navigation system is manipulated in such a way that the airplane is sent off course.

Note that security might be a safety problem whether the system is real-time or not, and that it is not only the operational systems that need protection. Systems under development and software back-ups could also be targeted by an attacker.

It is our impression that in the past, and to a great extent at present, most safety-related assessments don't seem to include proper considerations of security as a safety problem. One possible reason might be the lack of methods which can be used to identify security threats in a safety context. Although there do exist analytical techniques from both the security and the safety traditions, the approaches used within these areas seem to be different. A "convergence" of methods would therefore be beneficial.

Based on experience from using safety-related techniques in security projects, we have had the opportunity to "think safety" in a security context. Related to the development of a risk analysis handbook (security issues) for Telenor[2] we evaluated methods such as Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Failure Mode Effect Analysis (FMEA) and HAZard and OPerability studies (HAZOP) for use in security. Regarding the identification of security threats our conclusion was that the HAZOP principle seemed well suited, assuming that adequate guidewords could be established.

We will in this paper present a "security-HAZOP" which has emerged from our experiences in practical projects and demonstrate its use in a safety context. Finally, we will point to a new EU-project which objective is to combine e.g. HAZOP with object oriented modeling and the use of UML in the development of security-critical systems.

2 What is HAZOP?

A HAZOP study [1, 2, 6] is a systematic analysis of how deviations from the design specifications in a system can arise, and whether these deviations can result in hazards. The analysis is performed using a set of guidewords and attributes. The guidewords identified in [6] for use when analysing programmable electronic systems (PES) are *no, more, less, as well as, part of, reverse, other than, early, late, before and after*. Combining these guidewords with attributes, such as *value* and *flow*, generic deviations can be described thus providing help in identifying specific safety related deviations.

A HAZOP study is typically conducted by a team consisting of four to eight persons with a detailed knowledge of the system to be analysed. The HAZOP-leader of the group will normally be an engineer with extensive training in the

[2] Telenor is Norway's largest telecommunication company.

use of HAZOP and other hazard analysis methods. The analysis itself is done by going systematically through all system components identifying possible deviations from intended behaviour and investigating the possible effects of these deviations. For each deviation the team sets out to answer a series of questions to decide whether the deviation could occur, and if so, whether it could result in a hazard. Where potential hazards are detected, further questions are asked to decide when it might occur and what can be done to reduce the risk associated with the hazard.

3 Adapting the HAZOP to a Security Context

In this chapter we will briefly discuss why we have chosen to use HAZOPs for identifying security threats and then present the proposed modifications to the original HAZOP.

3.1 Why HAZOP?

Even though HAZOPs originally were developed for use in a specific context, namely the chemical industry [1], experience over the years has shown that the basic principle is applicable in different contexts. [2, 6] presents modified HAZOPs for use on systems containing programmable electronics. The fact that HAZOPs see widespread practical use in diverse areas indicates that it is a good candidate for identifying security threats. After all, the aim is the same in security as in safety: We want to identify critical deviations from intended behaviour. There are also other arguments that lead to the same conclusion. Comparing HAZOPs to FMEA (which is a possible alternative) we see that FMEAs are best used in situations where the analysis can be performed by one or two persons and where the identification of possible failure modes is not to complicated. As the FMEA (at least in principle) requires that all failure modes of all components must be scrutinized equally thoroughly, we expect problems in using the method when dealing with complex computerized systems, having a multitude of possible failures, and usually requiring that more than two persons participate if all relevant aspects shall be adequately covered. This does not imply that we discard FMEA as a possible method for analysis of security threats. In situations where the possible failure modes are relatively obvious and the aim of the analysis is more focused on consequences, FMEA is probably a good candidate. Having a well structured description of the system might be one way of achieving this.

3.2 Modifying the HAZOP to Identify Security Threats

Since the HAZOP principle obviously should remain the same, our focus has been on identifying guidewords and attributes which will help us identify security-related deviations. As we are primarily focusing on the CIA of security, i.e. confidentiality, integrity and availability, the intuitive approach is to define these

as attributes and then continue by evaluating whether the guidewords defined in [6] (see Chapter 2) can be used, or if new ones are needed.

When systematically combining the guidewords in [6] with each of the CIA attributes, it is quickly realized that many combinations doesn't seem to be useful. For instance, although "more confidentiality" could be interpreted as too much confidentiality, implying that information is less available than intended, this deviation is more naturally identified through the statement "less availability". Since our prime concern is that the level of confidentiality, integrity and availability won't be adequate, a pragmatic evaluation suggests that only "less" is useful. However, only considering the applicability of preexisting guidewords is not enough. We need to see if there are *other* suitable guidewords. An interesting question is: "What *causes* inadequate CIA?" Firstly, the loss of CIA might happen both due to technical failures and human actions. Furthermore, typical security threats include deliberate hostile actions (by insiders or outsiders) as well as "trivial" human failures. It makes sense, therefore, to include guidewords encompassing the characteristics *deliberate, unintentional, technical, insider* and *outsider*. In order to be able to combine these with the CIA attributes in a sensible way, we have chosen to use *negations* of the CIA attributes, i.e. *disclosure, manipulation* and *denial*. Furthermore, since e.g. deliberate actions might be by both insiders and outsiders, we see that it might be beneficial to combine more than one guideword with each attribute. To accommodate this we have chosen to structure the HAZOP expressions as illustrated (with examples) in Table 1. Table 2 summarizes the guidewords and attributes we suggest as a starting point when using HAZOPs to identify security threats.

Table 1. A new way of combining guidewords and attributes, together with some simple examples.

Pre-Guideword	**Attribute**	*of*	**comp.**	*due to*	**Post-Guideword**
Deliberate	manipulation	of	firewall	due to	insider
Unintentional	denial	of	service	due to	technical failure

Table 2. Basic guidewords and attributes suitable for identifying security threats

Pre-Guideword	**Attribute**		**Post-Guideword**
Deliberate	Disclosure		Insider
	Manipulation	*of COMPONENT due to*	Outsider
Unintentional	Denial		Technical failure

It is important to note that in each analysis it will be necessary to evaluate what guidewords and attributes that are suited. Both the attributes and the post-guidewords given in Table 2 are rather generic and could be refined to better describe relevant deviations. For instance, the attribute *manipulation* could be replaced by *removal*, *alteration*, *fabrication*, etc. While the post-guidewords listed above define some generic threat agents, these could be replaced or augmented by describing the possible techniques these threat agents might use. *Spamming*, *social manipulation* and *virus* are relevant examples. Using these more specific attributes and guidewords, we obtain the examples in Table 3. In the first example the guidewords *unintentional* and *virus* are combined with the attribute *fabrication* and applied to the component *mail*.

Table 3. Examples of more specific expressions.

Expression	**Possible security threats**
Unintentional fabrication of mail due to virus	Improper handling of mail attachments. Inadequate virus protection.
Deliberate disclosure of patient records due to social manipulation	Improper handling of requests for information from unknown persons.

If we replace *unintentional* with *deliberate* in the first example, achieving the expression *Deliberate fabrication of mail due to virus*, we immediately associate this with an attacker using viruses of the "I LOVE YOU" type to cause harm. Although the threats we have identified for the component *mail* are closely related, changing from *unintentional* to *deliberate* moves our focus from sloppy internal routines to hostile actions. Table 4 provides an extended list of guidewords and attributes compiled through various projects.

Table 4. An extended list of guidewords and attributes suitable for identifying security threats

Attributes	**Post-Guidewords**
disclosure, manipulation, disconnection, fabrication, delay, corruption, deletion, removal, stopping, destabilisation, capacity reduction, destruction, denial	insider, outsider, technical failure, virus, ignorance, fire, faulty auxiliary equipment, sabotage, broken cable, logical problems, logical attack, planned work, configuration fault, spamming, social manipulation

Going through the list of attributes and guidewords one will see that some of them have similar meanings. However, although some of the words can be considered quite similar, they might give different associations for different people. Furthermore, words that have similar meanings in one context might have different meanings in another. Taking "deletion" and "removal" as an example, it is easily realized that in an analysis of physical entities the word "removal" might give sensible associations, while "deletion" doesn't.

Having discussed possible guidewords and attributes we also need to discuss what a typical *component* will be in the context of the security-HAZOP. In the original HAZOP the components typically are physical entities such as valves, pipes and vessels. In the "PES-HAZOP" described in [6], the entities might be both physical and logical. Since the main focus of the security-HAZOP is on possible threats to confidentiality, integrity and availability, components must constitute entities for which these attributes are meaningful. While confidentiality is relevant for information, integrity and availability are relevant in respect to both information and functionality. Hence, we suggest that the focus of the security-HAZOP should be on the various types of information handled by the system, and on the functions the system perform. In fact, it could be argued that we could limit the selection of components to the *information components,* since any failure of an ICT system must in some way be related to changed, erroneous or missing information. In practice, however, we will include both information, functions and in some cases even physical entities in our list of components as they provide different perspectives. In some situations it might be more intuitive for the experts to consider physical entities than the more abstract information types. Deciding which components to analyse should be done pragmatically, where available time and experience are relevant factors to consider.

An important difference between physical entities and information is that the latter are not bounded to be at a single place at any one time (although they can be). Information might be stored in several locations as well as being in transition between various nodes in a network. Although functionality is naturally associated with physical entities they are not necessarily limited to a single entity either. The function "file transfer", for instance, is a functionality that involves at least two physical entities. The reason for pointing out these more or less obvious facts, is that they have affected the analyses we have performed. Let's illustrate this with a simple example: Consider a system consisting of a client and a server where the client requests a download of a patient record. Relevant components in this scenario are the *patient record* (information) and *information transfer* (function). Physical entities are the two computers, which the client and server software are running on, and the network[3] which connects the two. If we were to specifically cover all physical entities, as well as information types, we would have to evaluate possible threats to the patient record at the client computer, the server computer and in the network. Since threats exists for all of these this is not irrelevant. However, it might become tedious in cases where there are many physical entities. An alternative approach consists of

[3] Which can be subdivided into a number of components.

evaluating threats to the information *detached from the physical entities*. Since threats *have* to be related to the information we don't necessarily miss relevant threats, although more emphasis will have to be put on the cause-consequence analysis. It should be noted that unavailability of information, which could be caused by threats to the physical entities, must be included in the list of threats to the information. In practice, a pragmatic approach will be to use a mixture, thus putting specific focus on selected parts of the system by considering some physical entities in more detail than others.

4 Practical Use of the "Security-HAZOP"

In this chapter we will briefly present projects where the security-HAZOP has been used and then demonstrate the method's applicability on a safety related system.

4.1 Non-safety Projects

As mentioned earlier the method was developed while working on security projects for Telenor. Although these are not related to safety it might be interesting to see a list of projects where the method has been applied.

- Analysis of security in the top level design phase of a new Wide Area Network concept.
- Analysis of threats related to Signaling System No. 7.
- Analysis of security related to the use of mobile office solutions.
- Analysis of threats related to the installation and operation of a large customer-order-invoice system planned to handle in the range of 1.000.000.000 euros per year.
- Analysis of threats related to the implementation and operation of GPRS.

4.2 Using the Security-HAZOP to Analyse Safety

In this section we will illustrate the use of the security-HAZOP in a safety context.

The basic elements of the system referred to as a "Train Leader (TL) Telephone System" (TLT-system) are shown in Figure 1. The TL's main task is to supervise the traffic and ensure that signals etc. are correct. The safety aspects of this system have been analysed using both HAZOP and FMEA and the experiences are in the proceedings of an earlier SAFECOMP [4]. The analysis of security illustrated below was never done in practice, although the results might indicate that it should have been.

The TLT-system's main functions are:

- To present incoming calls from trains at the train leader's (TL's) terminal (PC), including information regarding the trains' positions.

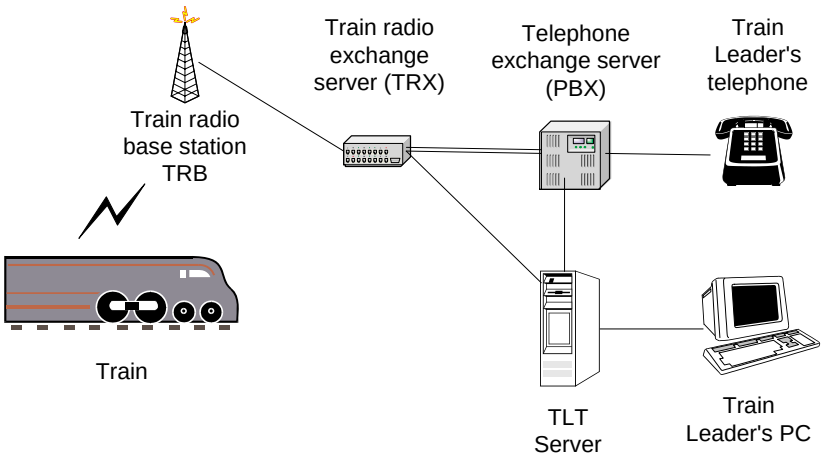

Fig. 1. Basic elements in the TLT-system

- Connect TL's telephone to whatever incoming call the TL selects.
- Set up outgoing calls as requested by TL.
- Route incoming calls from trains to the TLs responsible for the various trains.

The TL's main task is to supervise the train traffic and ensure that signals etc. are correct. Since the TLs are authorized to give "green-light" to trains in the case of signal system failure it is important that calls are connected to the *correct* TL and that the information presented to the train leader is also correct. Erroneous routing of calls, or misleading information regarding the trains' identity or position, could cause serious accidents.

In this illustration, we will focus on one specific scenario, namely: "Train driver initiates a train radio call to TL and TL answers call". The analysis will be performed by going through the following steps:

1. Identify relevant components.
2. Construct relevant expressions based on the suggested guidewords, attributes and components.
3. Evaluate whether the expressions identify relevant security threats.

In the scenario we are investigating we have the following sequence of messages:

1. Train identifier (ID) and train position are sent from TRB to TRX. Train ID is obtained from the on-board radio while train position is determined from sensors placed along the tracks.
2. Train ID and train position are sent from TRX to TLT-server.
3. Train ID and train position are sent from TLT-server to the appropriate TL PC.

4. When TL decides to answer the call the TL PC sends a *connect* command to the TLT-server.
5. TLT-server commands TRX and PBX to connect the incoming call to TL's telephone.
6. TLT-server updates TL's PC-screen.

As noted in Section 3.2, typical components for the security-HAZOP are information types and functions. From the scenario above we see that we have three types of information: Train ID, train position and voice. The most critical functions are routing of voice and call information to correct TL, and to present call information at the TL's terminal. Train ID originates from the train itself and is sent through TRB, TRX and TLT-server before it is presented to the TL. The train ID is used by the TLT-server to decide which TL should receive the call.

For simplicity we have chosen to ignore the post-guidwords and to focus on deliberate actions related to manipulation and denial of train ID and voice. Table 5 presents both the constructed expressions, security related hazards and possible causes. It should be noted that this table is not a complete list of relevant hazards. Erroneous train position is obviously another critical failure that could potentially be caused by an attacker.

Table 5. Examples of the use of guidewords together with typical results

Expression	**Threat**	**Causes**	**Consequences**
Deliberate manipulation of train ID.	Train ID is altered.	TRB, TRX or communication links between TRB/TRX or TRX/TLT-server has been manipulated.	Call information is wrong. Call is routed to wrong TL.
Deliberate denial of train ID	Communication between train driver and train leader is inhibited.	TRB, TRX or cabling has been manipulated, destroyed or in any other way forced to fail.	Train cannot be given a manual "green-light". Emergency calls cannot be made.
Deliberate manipulation of voice.	Unauthorized person responds to call from train and impersonates a TL.	TRB, TRX, PBX or com. links in between has been manipulated to connect unauthorized person to a call from a train.	Manipulation of train driver to perform unsafe action.

Having identified security threats in the TLT-system, the next activity is to evaluate whether these can cause hazards to occur. From this example we see that the threat "Train ID is altered", which has the possible consequences "Call information is wrong" and "Call is routed to wrong TL", is naturally associated with the hazard "Wrong train receives green-light". In the analysis actually carried out for the TLT-system the possible causes for this hazard were limited to internal software and hardware failures, thus illustrating the limited scope of the analysis.

Let us now make a simple comparison with some combinations of guidewords/attributes for the TLT-system based on the PES-HAZOP [6]:

- Train ID combined with *Other Than*
- Train ID combined with *No*

Clearly, applying these guidewords does not mean that we will not identify security threats. Manipulation of Train ID is one possible cause of getting an erroneous Train Id. The benefit of applying the security specific guidewords and attributes is that our attention is specifically directed to the security issues, thus reducing the possibility of missing out on important security threats. While the guidewords in the PES-HAZOP tends to focus on system failures, the security-HAZOP emphasizes the systems vulnerability to human actions and incorrect information.

5 A Framework for Efficient Risk Analysis of Security-Critical Systems

The successful employment of HAZOP and FMEA to identify and analyse safety risks in the TLT system [4] is one of the arguments for the IST-project CORAS [3]. The CORAS main objective is to develop a practical framework, exploiting methods for risk analysis developed within the safety domain (such as HAZOP), semiformal description methods (in particular, methods for object-oriented modelling), and computerised tools (for the above mentioned methods), for a precise, unambiguous and efficient risk analysis of security-critical systems. One hypothesis considered in the project is that a more formal system description can make it easier to detect possible inconsistencies. Another main objective is to assess the framework by applying it in the security critical application domains telemedicine and e-commerce. We believe that the security-HAZOP presented in this paper, sketching out how new guidewords, attributes and a new template could be made, can be another input to this project. The fact that the project has got funding from January 2001, and will run for 30 months, is also an example of a growing awareness with respect to the identification of security threats in safety critical systems.

6 Conclusions

We have shown that it is possible to adapt the HAZOP-principle for analysis of security. The adaptation required new guidewords, new attributes and a new

template for combining guidewords and attributes. Since the HAZOP-principle is well known in the "safety world", extending the HAZOPs already in use with the modifications presented in this paper should enable a relatively easy incorporation of security analyses in safety contexts.

We have argued that relevant components to be analysed could be limited to the information types handled by the system. Since the same information might be stored in several locations, as well as being in a state of transfer, systematically going through all physical entities for each type of information quickly becomes tedious, without necessarily improving the threat identification.

References

1. Chemical Industries Association: A guide to Hazard and Operability Studies (1992).
2. Chudleigh M.F., Catmur J.R.: Safety Assessment of Computer Systems Using HAZOP and Audit Techniques. Proceedings of Safety of Computer Control Systems, SAFECOMP (1992). Pergamon Press
3. CORAS IST-2000-25031: A Platform for Risk Analysis of Security Critical Systems. http://www.nr.no/coras.
4. Dahll, G.: Safety Evaluation of a Train Leader Telephone System. Proceedings of Computer Safety, Reliability and Security, 18th International Conference, SAFECOMP (1999). Springer-Verlag.
5. Laprie J.-C. (Ed.),: Dependability: Basic Concepts and Terminology. IFIP WG 10.4 Dependable Computing and Fault Tolerance, vol. 5. Springer-Verlag (1992).
6. Ministry of Defence: Interim Defence Standard 00-58/1: Hazop Studies on Systems Containing Programmable Electronics. Directorate of Standardization (1994).