Security Application of Failure Mode and Effect Analysis (FMEA)

Christoph Schmittner[1], Thomas Gruber[1], Peter Puschner[2], Erwin Schoitsch[1]

[1]Austrian Institute of Technology, Safety & Security Department, Vienna, Austria
{christoph.schmittner.fl, thomas.gruber, erwin.schoitsch}@ait.ac.at
[2]Vienna University of Technology, Department of Computer Engineering, Vienna, Austria
peter@vmars.tuwien.ac.at

Abstract. Increasingly complex systems lead to an interweaving of security, safety, availability and reliability concerns. Most dependability analysis techniques do not include security aspects. In order to include security, a holistic risk model for systems is needed. In our novel approach, the basic failure cause, failure mode and failure effect model known from FMEA is used as a template for a vulnerability cause-effect chain, and an FMEA analysis technique extended with security is presented. This represents a unified model for safety and security cause-effect analysis. As an example the technique is then applied to a distributed industrial measurement system.

Keywords: safety analysis, security analysis, combined analysis, FMEA, vulnerabilities, cause effect chain for security

1. Introduction

With interconnected and software intensive systems, availability and safety depend increasingly on security aspects. Threats to information security also threaten the availability or safety of a system [1]. Dependability of software intensive systems depends not only on the reliability of the used software but also on the security of the Information System. Information security is increasingly interwoven with all aspects of dependability [2]. Recent events like Stuxnet [1] or Duqu [2] demonstrated vulnerabilities in industrial or embedded IT-Systems. In order to remove or reduce these risks holistic analytical methods are necessary.

The Failure Mode and Effect Analysis (FMEA) is a structured technique which investigates failure modes and their effects. The aim is to identify potential weaknesses and improve reliability, availability or safety. A system or process is hierarchically decomposed into its basic elements and then the failure modes of the elements are examined for causes and effects [3]. FMEA was developed in the 1950s by the US Department of Defense to improve the reliability of military equipment[4]. Originally, FMEA was aimed at the reliability or safety of hardware.

[1] http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[2] http://www.symantec.com/connect/w32_duqu_precursor_next_stuxnet

The failure modes and probabilities for hardware components are normally well known. Although failure modes of software are more complex and coupled with a certain degree of uncertainty, Reifer and others [5], [6] showed the benefits of performing a Software-FMEA (SFMEA). As explained in [7] when an SFMEA is performed early in the design phase of software, activities for verification and validation of software are easier to execute and a more focused use of development effort is possible.

This paper describes an approach for the combined analysis of safety and security. The basic FMEA concept is extended to include vulnerabilities and attacks concerning the security of a system. A unified cause and effect model allows examining the combined risks for a system. The following method for a Failure Mode, Vulnerabilities and Effects Analysis (FMVEA) enables the analysis of complex mission critical systems. Similar to a Software-FMEA the benefits are the easier verification and validation and the ability to focus the development effort on critical areas.

After an overview of the state of the art, chapter 3 describes the new method in detail. Chapter 4 tries to prove the applicability of the FMVEA based on an example. Finally, Chapter 5 discusses the limitations and gives an outlook on further work.

2. State of the Art

For safety, IEC 61508 [8] is the basic functional safety standard, which covers the complete safety life cycle. It describes techniques and procedures for analysis, realization and operation of safety critical systems. With respect to security, IEC 61508 Ed 2.0 (2010) contains only a few requirements: Security threats are to be considered during hazard analysis in the form of a security threat analysis (IEC 61508, Part 1, 7.4.2.3). The ISO/IEC 27000-series describes best-practices advice for information security management. They consider classic security-critical systems such as databases, servers and corporate networks. Nevertheless, we use the terms as they are defined in the ISO/IEC 27000-series for this publication and those from IEC 61508 for safety.

Although IEC 62443 "Network and system security for industrial-process measurement and control" [9] is partially aimed at industrial security, safety concerns are outside its scope.

Summarizing, a standards review shows that critical control systems have been treated by well-established safety-standards for many years, while most available security standards aim at business applications with few exceptions. The analyzed effects and causes are, indeed, different in safety and security. So, what is definitively missing is a standard which considers both safety and security equally. Without a combined approach, there is a risk to miss critical and undesirable events: Security vulnerabilities which potentially lead to safety critical events could be overlooked.

In [10] FMEA was used for the dependability analysis of web services. The approach was based on a high level design FMEA. We propose here to extend the functional FMEA [11] in order to base the analysis on a functional system model. This enables a model based analysis of all the functions at the considered abstraction level.

In addition, we propose a generic set of security based failure modes (named threat modes), based on [12] and explain the correlations between the threat modes and the system quality attributes [13]. Generic threat modes allow anticipating potential threats first, assess the consequences and then identify potential causes.

3. FMVEA Concept

The basic approach to carry out an FMEA is described in IEC 60812. Based on this description, the flow chart includes the following steps.

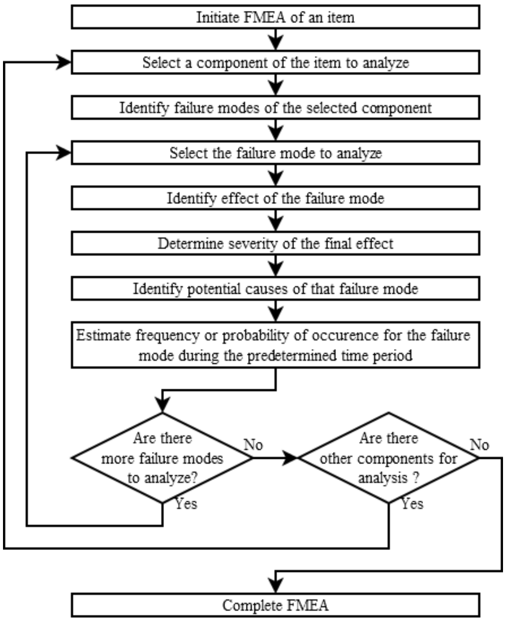
Initiate FMEA of an item
Select a component of the item to analyze
Identify failure modes of the selected component
Select the failure mode to analyze
Identify effect of the failure mode
Determine severity of the final effect
Identify potential causes of that failure mode
Estimate frequency or probability of occurence for the failure mode during the predetermined time period
Are there more failure modes to analyze?
No
Yes
Are there other components for analysis ?
No
Yes
Complete FMEA

Figure 1: FMEA - analysis flow chart, based on [3]

A system is divided into components, and failure modes for each component are identified. For each failure mode the effects, the severity of the final effect on the system and potential causes are examined. As far as possible, frequency or probability of the failure modes are estimated.

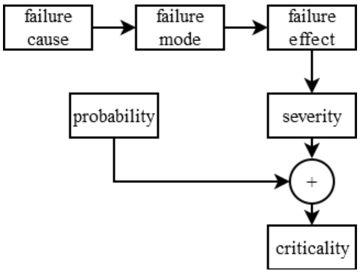

Figure 2: FMEA - cause-effect chain

The cause-effect chain analyzed with an FMEA is shown in Figure 2. Each failure mode has a failure cause, and each failure effect is associated with a failure mode that causes the effect. A failure effect leads to an unintended scenario. The severity describes the significance of the scenario. The frequency relates to failure cause and effect, and it describes how likely the event is.

Definitions according to IEC 60812 [3]:

- **Failure cause:** why did the item fail
- **Failure mode:** manner in which an item fails
- **Failure effect:** consequence of a failure mode in terms of the operation, function or status of the item
- **Failure severity:** significance or grading of the failure mode's effect on item operation, on the item surrounding, or on the item operator; failure mode effect severity as related to the defined boundaries of the analyzed system
- **Failure criticality:** combination of the severity of an effect and the frequency of its occurrence or other attributes of a failure as a measure of the need for addressing and mitigation

To include security in the analysis, a comparable cause-effect chain is necessary. It is possible to divide security-critical events into similar steps. The suggested parts of a security cause-effect chain are the following elements.

- Vulnerabilities
- Threat Agent
- Threat Mode
- Threat Effect
- Attack Probability

3.1. Vulnerabilities

The essential precondition for a successful security breach of a system is a weak spot or vulnerability. A vulnerability is comparable to a failure cause and represents the basic prerequisite in security. ISO/IEC 27002 defines vulnerability as "a weakness of an asset or group of assets that can be exploited by a threat" [14]. For information security ISO/IEC 27005 [15] divides vulnerabilities into categories:

- Hardware vulnerabilities
- Software vulnerabilities
- Network vulnerabilities

Additional vulnerability classifications are the Microsoft Security Development Lifecycle (SDL) [16] and the CWE[3] (Common Weakness Enumeration). The CWE is a detailed and community-developed list of common software weaknesses.
Figure 3 shows an overview of a possible categorization based on ISO/IEC 27005 [15].

[3] http://cwe.mitre.org/data/index.html

Following a top-down approach, vulnerabilities at a lower design level get more specific. The list is non-exhaustive. For software vulnerabilities the CWE lists additional software weakness types.

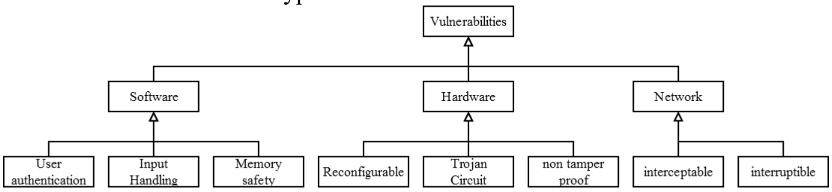

Figure 3: Classification of vulnerabilities

Vulnerabilities can be located at network, hardware and software level. Hardware vulnerabilities are especially a challenge for security engineering if parts of the embedded systems are employed in a potentially not trustworthy environment. In addition, hardware could generally be equipped with additional malicious components [17]. Reconfigurable means a microcontroller is reprogrammable. This could be because one step in the commissioning process has not been executed. With not tamper proof hardware, an attacker could access hardware components and execute direct attacks on the hardware.

If an attacker (= *threat agent*) exploits a vulnerability, the security of the system is at risk. Vulnerabilities without a threat agent do not lead to an effect and have a negligible risk attached. A threat agent is a necessary extension for the safety cause-effect chain.

3.2. Threat Agents

Threat agents represent the active element which is trying to exploit the vulnerability. Examples for possible threat agents are hacker, computer criminals, terrorists, industrial espionage or insiders [15]. For now, inmate threat agents like viruses are not considered.

The closest corresponding element in safety would be the random event that causes an element to fail. In contrast to the random element a threat agent has a motivation and an objective. Table 1 lists different threat agents with objectives and characteristics.

Table 1: Threat agents, based on [15]

Threat agent	Objective / Aim	Characteristic
Hacker, Cracker	Challenge, Ego, Rebellion, Status, Money	Limited resources Random attacker
Computer criminal	Destruction of information, Illegal information disclosure, Monetary gain, Unauthorized data alteration	Monetarily motivated
Terrorist	Blackmail, Destruction, Exploitation, Revenge, Political Gain, Media Coverage	Ideologically motivated
Industrial espionage	Competitive advantage, Economic espionage	Purposeful attacker
Insiders	Curiosity, Ego, Intelligence, Monetary gain, Revenge, Unintentional errors and omissions	Internal knowledge Easy access

3.3. Threat Mode

Threat mode classifies the way in which vulnerabilities are exploited. Vulnerabilities can be exploited in various ways, each with different effects and prerequisites. Potential threat modes depend on the system and on the capabilities of the threat agent. Threat modes can be simple like jamming a connection or elaborate operations such as exploiting an injection flaw vulnerability, which requires access to the input system and sending an exactly formulated input signal. In general, this can be mapped to a violation of a security attribute.

The mapping of security attribute to system quality attribute varies for each individual system. Depending on the system, every threat mode could affect any dependability attribute (Reliability, Availability, Maintainability, and Safety) - or not.

A common model for the categorization of threats is STRIDE [18]. As described in Table 2, an exploited vulnerability leads to one of the following generic effects.

Table 2: Threat modes

Threat mode	**Description**	**Violated security attribute (generic effect)**
Spoofing identity	Accessing a system, disguised as another actor	Authenticity
Tampering with Data	Unauthorized modification of data	Integrity
Repudiation	Actions can be assigned to one actor	Non-repudiation
Information disclosure	Accessing restricted data	Confidentiality
Denial of Service	Restricting or preventing access to a service or function	Availability
Elevation of privilege	Actors may perform actions with a higher authority level	Authenticity

A threat mode is similar to the failure mode of safety and describes the manner in which the security fails.

For a classification of threat and failure modes the approach described in [19] could be used. Different properties of failure modes are described and sorted.

3.4. Threat Effect

Similar to the failure effect in safety the threat effect is the consequence in terms of the operation, function or status. While the threat mode characterizes the violated security attribute, the threat effect describes the violated system quality attribute [13]. Violated attributes are not limited to security. All dependability attributes may be affected. Which attribute is actually violated in a particular case depends on the system, its environment and the operational state.

3.5. Attack Probability

In order to assess the criticality of a security attack, severity and probability of the attack needs to be evaluated. While the severity can be assessed with the help of domain experts, probability is defined differently for safety and security.

For a safety based event the probability describes the probability of failure of hardware or software. For a security based element the attack probability describes the probability of the threat agent to accomplish the threat effect. This depends not only on the threat agent itself but also on system properties and the system environment. If a system is not connected to a public network and located in a restricted area, a successful attack is relatively improbable. In addition to the technical probability of an attack, each threat agent has different motivating factors and capabilities. Capabilities are an umbrella term for financial resources and knowledge or possibly other resources of the threat agent used to exploit the vulnerability. Motivation and capabilities characterize the threat agent and their *sum* constitutes the *threat properties*.

- Motivation (1 = opportunity target, 2 = mildly interested, 3 = main target)
- Capabilities (1 = low, 2 = medium, 3 = high)

In addition to the properties of the threat agent, different system properties influence the probability of an attack. Reachability is characterized by a number between 1 and 3 and describes how easy it is to connect to the system. Examples for reachability 3 are systems which are directly connected to the internet and discoverable with tools like SHODAN[4]. If a system is not directly connected to the internet but accessible by an internet connected network, then it is assigned reachability 2. Systems with no network connection at all have reachability 1.

In addition to reachability, another factor that describes the susceptibility of a system is the unusualness of its components and architectures. It can be assumed that potential threat agents have less knowledge about unusual systems and the effort to find flaws and exploit them is higher. The *sum* of both properties characterizes the *system susceptibility*.

- Reachability (1 = no network, 2 = private network, 3 = public network)
- Unusualness (1 = restricted, 2 = commercially available, 3 = standard)

The combination of *system susceptibility* and *threat properties* for attack probability is influenced by the DREAD Risk assessment model [12] and the OWASP Likelihood assessment method. Like in the DREAD approach, we estimate the probability by summing up system susceptibility and threat property values. In combination, the four properties allow a semi-quantitative assessment of the probability.

Table 3: Estimation table for attack probability

System Susceptibility						
6	8	9	10	11	12	
5	7	8	9	10	11	
4	6	7	8	9	10	
3	5	6	7	8	9	
2	4	5	6	7	8	
	2	3	4	5	6	Threat properties

[4] SHODAN is a search engine for internet connected SCADA systems.

It should be noted that the probability table lacks a calibration with the failure probability in order to introduce it in a common safety and security method.

While this approach should bring reasonable results, the results are based on an assumption about attack frequency and not on empirical data. In order to get better estimates for frequency or probability, empirical values for attack probabilities for different systems could be useful. Since security for industrial information systems is a relatively new area there is not much empirical data. Besides historical incident data, different projects try to collect information about attack probability [20]. To gather information about attack probability or frequency, honeypot systems are used. A honeypot is a closely monitored decoy system which acts as a trap for potential threats. In a research experiment from Trend Micro Inc. three different honeypot systems were used [21]. One was a simulated water supply facility with connected pumps and purification systems running on Apache web server. For the next setup an internet connected programmable logic controller (PLC) was set up to imitate a temperature controller in a factory which had temperature, fan speed, and light settings. The last Honeypot Setup was a server running PLC control software and a web server to imitate a human machine interface (HMI) connected to a PLC.

Trend Micro Inc. concentrated on planned and targeted attacks and integrated measures like firewalls in order to filter automated “drive-by” attacks. In the first 28 days of the experiment 39 attacks were reported [21]. Unfortunately no long-term study for industrial information systems is known at this time. Therefore all results should be taken with a pinch of salt. They may help in estimating the magnitude of threats but for now they shouldn’t be used as valid numbers for quantitative analysis. But in the future both approaches will yield better estimations for attack probability. In any case, probability and frequency of successful attacks my change over time depending on the evolution of methods, the increase of knowledge about control and protection systems, and other causes. Therefore, security measures have a much shorter lifetime than safety measures and need unfortunately more frequent updates.

3.6. FMVEA Cause-Effect Chain

With the single components for a security cause-effect chain described in the previous sections, we are able to generate a combined cause-effect chain for safety and security. The combined approach includes safety and security causes for a negative effect on system quality attributes.

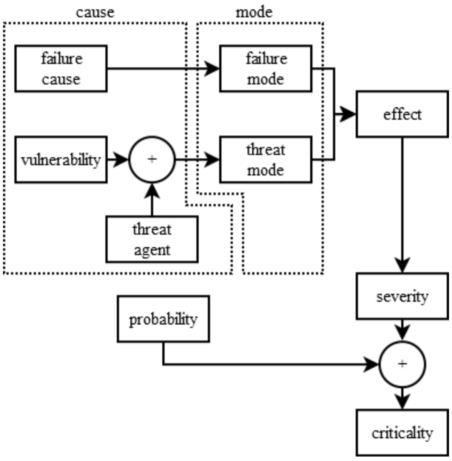

Figure 4: FMEVA – cause-effect chain

The extended flow chart for an FMVEA in Figure 5, includes security in the analysis. As described in [1] there are different ways in which security or safety properties of a system can influence security or safety risks. Therefore, while the consideration of failure or threat modes of an item is split, the analysis of effects and causes combines both viewpoints.

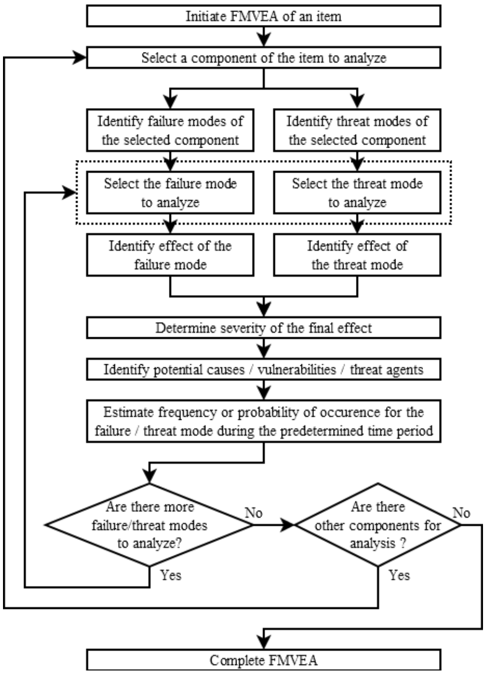
Initiate FMVEA of an item
Select a component of the item to analyze
Identify failure modes of the selected component
Identify threat modes of the selected component
Select the failure mode to analyze
Select the threat mode to analyze
Identify effect of the failure mode
Identify effect of the threat mode
Determine severity of the final effect
Identify potential causes / vulnerabilities / threat agents
Estimate frequency or probability of occurence for the failure / threat mode during the predetermined time period
Are there more failure/threat modes to analyze?
No
Yes
Are there other components for analysis ?
No
Yes
Complete FMVEA

Figure 5: FMVEA - analysis flow chart

4. Example Application of the FMVEA

As an example, an engine test stand in an industrial plant is analyzed. The engine test stand consists of one or more measurement devices with smart maintenance features. The measuring devices are configurable for different engines. The measurement and configuration data should be only locally readable and writeable. For maintenance, lifetime data (= device conditions) is stored on the measurement device. The device itself can check its conditions in order to schedule maintenance activity. The device conditions can also be checked from a remote side. In order to check the conditions remotely the measurement devices are directly connected to a public network like GSM. Mission statement for the system is: Dependable measurements of different systems

4.1. Functional Analysis

In order to get the individual components, a functional analysis [22] at system level is conducted. Useful results for the FMVEA of a functional analysis are the functional tree, the functions / device matrix and the connection matrix. The functional tree (see Figure 6) identifies the functions of the system based on the mission statement.

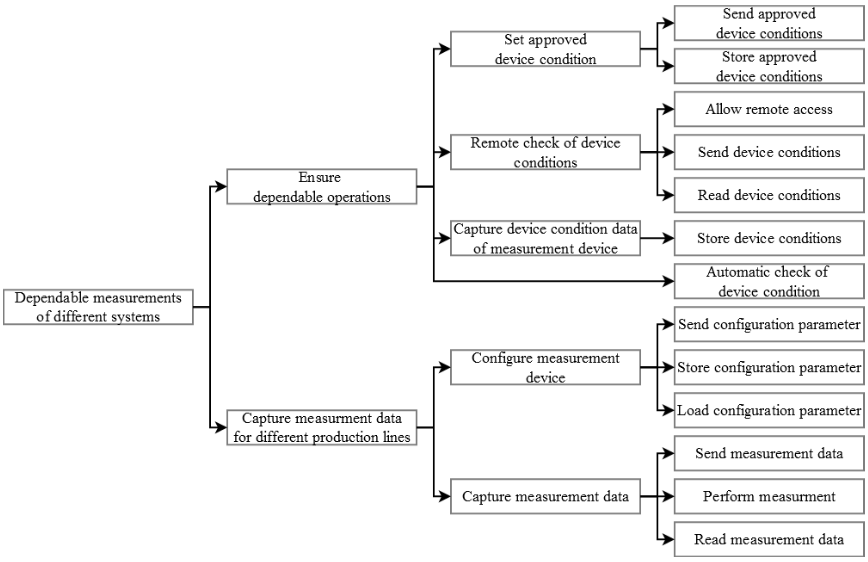
Dependable measurements of different systems
Ensure dependable operations
Capture measurment data for different production lines
Set approved device condition
Remote check of device conditions
Capture device condition data of measurement device
Configure measurement device
Capture measurement data
Send approved device conditions
Store approved device conditions
Allow remote access
Send device conditions
Read device conditions
Store device conditions
Automatic check of device condition
Send configuration parameter
Store configuration parameter
Load configuration parameter
Send measurement data
Perform measurment
Read measurement data

Figure 6: Functional tree of the example system

The functions / device matrix (see Table 4) maps the system functions to physical devices. In our example, the system consists of Measurement Devices, the Local HMI and the Backend System.

Table 4 : Functions / Device Matrix

		Devices	
Functions	Measurement Device	Local HMI	Backend System
Perform measurement	X		
Send measurement data	X		
Read measurement data		X	
Send configuration parameter		X	
Store configuration parameter	X		
Load configuration parameter	X		
Send approved device conditions		X	
Store approved device conditions	X		
Store device condition	X		
Allow remote access	X		
Send device conditions	X		
Read device conditions			X
Automatic check of device conditions	X		

With the connection matrix (see Figure 7), necessary connections between devices are identified and marked with “x”.

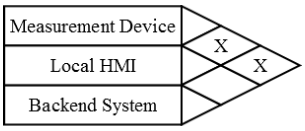

Figure 7: Connection Matrix

The analyzed system (see Figure 8 for a system overview) has connections between Measurement Device and Local HMI and between Measurement Device and Backend System.

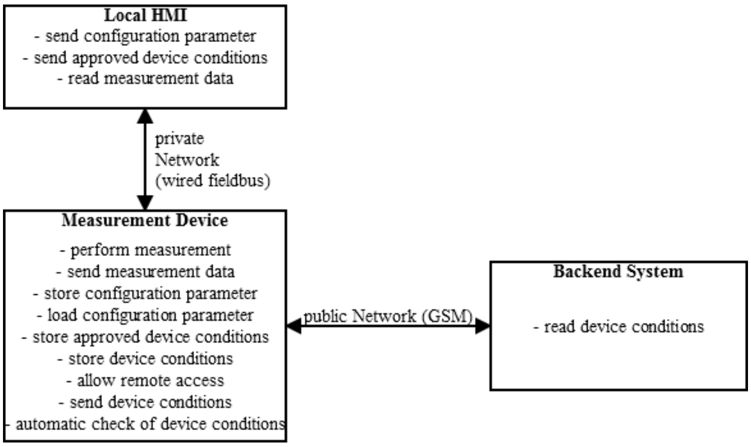
Local HMI
- send configuration parameter
- send approved device conditions
- read measurement data
private Network (wired fieldbus)
Measurement Device
- perform measurement
- send measurement data
- store configuration parameter
- load configuration parameter
- store approved device conditions
- store device conditions
- allow remote access
- send device conditions
- automatic check of device conditions
public Network (GSM)
Backend System
- read device conditions

Figure 8: System overview

4.2. Failure and Vulnerability Analysis

The system is analyzed according to the flowchart of Figure 5. The first chosen component is the measurement device. In order to identify potential threat modes the generic threat modes from STRIDE [18] are used. Applying the general concept of “spoofing of identity” to the Measurement Devices, a potential threat mode is that an attacker masks himself as the measurement device and communicates with other devices.

FMVEA Table. Table 5 shows a short excerpt from the FMVEA table for the described system.

Table 5: FMVEA example

	ID	Vulnerability	Threat mode	Threat effect	System status	System effect	Severity	System susceptibility	Threat properties	Attack probability
measurement device	1	No device verification, man in the middle attack with physical access to measurement device or connection	Attacker is pretending to be the measurement device	send false measurement data	Normal operation	System is no longer reliable	Critical	4	Insider: 4	8
									Hacker: 3	7
	2	GSM connection, Base station emulation, man in the middle attack	Attacker is pretending to be the measurement device	send false device condition data	Remote query of device status	System is no longer available (unnecessary maintenance)	Marginal	5	Insider: 4	9
									Hacker: 3	8
	3	GSM connection, Base station emulation, man in the middle attack	Attacker is pretending to be the measurement device	intercept user credentials	Remote query of device status	System integrity is hurt	Insignificant	5	Insider: 4	9
									Hacker: 3	8
	4	No device verification, man in the middle attack with physical access to measurement device or connection	Attacker is pretending to be the measurement device	Intercept configuration changes	Configuration change	System is unreliable and potentially unsafe	Catastrophic	4	Insider: 4	8
									Hacker: 3	7

In the following section, the elements of the table are explained in detail. If an ID is used it refers to the ID column.

Threat mode:

- The Attacker is pretending to be the measurement device. Spoofing of identity attacks only work at connected devices and for functions with a communication aspect. Potential effects are submitting wrong measurement data, submitting wrong device condition data, intercepting user credentials or blocking out configuration changes.

Threat effects:

1. The attacker is able to send false measurement data to the Local HMI. This leads to wrong measurements and an unreliable system.
2. While the Backend System executes a remote query of the device conditions, the attacker is able to send wrong device conditions. If the sent device conditions are worse than reality, unnecessary maintenance is caused. If the sent device conditions are better than reality, a defect of a Measurement Device may remain undetected.
3. While the backend system tries to login on the Measurement Device an attacker disguised as the Measurement Device is able to intercept the login credentials. This may cause no direct severe consequences but it hurts system integrity and enables a malicious user to access the measurement device.
4. The attacker intercepts configuration data and acknowledges the configuration change to the Local HMI. If multiple measurement devices are employed for one test stand different configurations could lead to inconsistent and incompatible demands

Severity:

For the severity assessment, the classification from IEC 61812 was used. This classification does not include privacy as a factor, only consequences for the dependability are considered.

1. Wrong measurement data means that the reliability of the whole measurement system is endangered. The attacker has the choice to send positive data for bad devices or negative data for good devices. This could lead to an increased rate of misproduction or to an increased liability for defective products. (Severity = critical)
2. Submitting of wrong device condition data. While wrong device condition data does not directly influence the measurement data, the attacker could use it to trigger or delay maintenance. A prematurely triggered maintenance reduces the availability of the whole system. While an early maintenance leads to production downtime, it does not endanger the quality of the products. (Severity = marginal)
3. Intercepted login credentials could be used as a first step enabling further attacks. But they do not cause immediate danger to dependability attributes. (Severity = insignificant)

4. Different configuration on measurement devices could lead to a potentially dangerous situation[5]. Incompatible demands and commands on the engine under test could potentially destroy the engine and the test stand. (Severity = catastrophic)

Vulnerabilities:

Different vulnerabilities can give an attacker the opportunity to masquerade as a Measurement Device. In order to pose for the Backend System, an attacker could exploit vulnerabilities in the connection between these devices and conduct a man-in-the-middle attack.

1&2. The attacker could exploit flaws in the GSM connection and intercept the GSM connection with his own fake base station [23] (= “International Mobile Subscriber Identity (IMSI) catcher”). After this, any communication between Backend System and Measurement Device will be routed via his system.

3&4. In order to pretend to be the Measurement Device in a connection with the Local HMI physical access to the device or the connection is necessary. This reduces potential threat agents to insiders or intruding attackers. Then the attacker could integrate his own device in the communication or intercept and replay commands. Most field buses have negligible security features or none at all; if an attacker achieves physical access to it, this part of the system is endangered. Possible solutions would be a change to a protocol with integrated security features or a physical protection of the connection.

Probability:

Attack probability depends not only on the attacked system element but also on the attacker. In order to quantify threat properties two different threat agents are described:

- Insider: Inside attacks are among to the most dangerous attacks. While they may not have hacking experience they have knowledge of the system and easy access to critical elements of a system. In addition they are highly determined and focused on their target.
  - Motivation (3 = main target)
  - Capabilities (1 = low)
  - Threat Properties: 4
- Hacker: Hacker describes a person who seeks and exploits weaknesses in different information systems. While they are motivated by a multitude of factors and mostly don’t aim to cause direct harm their action may have negative consequences. They have good technological knowledge and other resources useful for an attack.

[5] The engine test stand is able to test lubrication in curves. For this the engine is tilted. If a measurement device performs other measuring cycles at this time the engine and the test stand could be destroyed.

  - Motivation (1 = opportunity target)
  - Capabilities (2 = medium)
  - Threat Properties: 3

For a spoofing threat mode the attacker needs to target a connection. In case of the Measurement Device a threat agent could either aim at the GSM connection between Measurement Device and Backend System or at the internal connection between Measurement Device and Local HMI.

- GSM connection: While a GSM connection is not that common for non-commercial applications components are commercially available. As a wireless connection a GSM connection is publicly accessible.
  - Reachability of the system (3 = public network)
  - Unusualness (2 = commercially available)
  - System Susceptibility: 5
- Internal connection: internal connections are not publicly accessible and best described as a private network. Most fieldbus systems are not common for non-commercial applications but components are commercially available.
  - Reachability of the system (2 = private network)
  - Unusualness (2 = commercially available)
  - System Susceptibility: 4

5. Limitations and Further Work

Similar to the SFMEA a FMVEA is best suited for a qualitative high level analysis of a system in the early design phases. A general limitation of the failure mode and effects analysis is the restriction to analyze only single causes of an effect. Because of this some multi-stage attacks could be overlooked. This concern could be particularly relevant for security event chains, if several systems have to be compromised in order to reach a target system. Recent developments in combining FTA with Attack-Trees for a combined analysis could support a FMVEA in considering all security risks [24].

Further research is also needed to achieve a reliable assessment of the risk related to security concerns. Especially proven data for attack probability or frequency is needed. This would allow a calibration of the criticality of security threats in order to obtain results comparable with safety criticality. The semi-quantitative approach which distinguishes four factors for probability, split into system and attacker properties which influence attack probability is only a first approach to tackle this challenge. The attacker properties also only works for human threat agents, approaches to include inmate threat agents need further research.

Acknowledgment. The work presented in this paper has been supported by the European Commission through the FP7 Joint Technology Initiatives (Call ARTEMIS-2012-1, Project Arrowhead, Grant Agreement Number 332987).

References

1. F. L. Dong-bo Pan, “Influence between Safety and Security,” *2nd IEEE Conference on Industrial Electronics and Applications, 2007. ICIEA 2007*, pp. 1323–1325, 2007.
2. S. Lautieri, “De-risking safety [military safety systems],” *Computing and Control Engineering*, vol. 17, pp. 38–41, 2006.
3. *IEC 60812: ANALYSIS TECHNIQUES FOR SYSTEM RELIABILITY – PROCEDURE FOR FAILURE MODE AND EFFECTS ANALYSIS (FMEA)*. International Electrotechnical Commission.
4. *MIL-P-1629: Procedures for Performing a failure mode, effects and Criticality analysis*. Department of Defense (US).
5. D. J. Reifer, “Software Failure Modes and Effects Analysis,” *IEEE TRANSACTIONS ON RELIABILITY*, vol. 28, no. 3, pp. 247–249, 1979.
6. N. J. S. Jacob J. Stadler, “Software Failure Modes and Effects Analysis,” *Reliability and Maintainability Symposium (RAMS), 2013 Proceedings-Annual*, pp. 1–5, 2013.
7. H. A. Haapanen Pentti, “FAILURE MODE AND EFFECTS ANALYSIS OF SOFTWARE-BASED AUTOMATION SYSTEMS,” *STUK-Y TO-TR-19 0, August*, vol. 2, p. 2, 2002.
8. *IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems (E/E/PE, or E/E/PES)*. International Electrotechnical Commission, 2010.
9. *IEC 62443: Industrial communication networks - Network and system security*. International Electrotechnical Commission.
10. A. Gorbenko, V. Kharchenko, O. Tarasyuk, and A. Furmanov, “F (I) MEA-technique of web services analysis and dependability ensuring,” in *Rigorous Development of Complex Fault-Tolerant Systems*, Springer, 2006, pp. 153–167.
11. P. Haapanen and A. Helminen, “Failure mode and effects analysis of software-based automation systems,” Radiation and Nuclear Safety Authority, Helsinki (Finland), 2002.
12. W. S. Frank Swiderski, *Threat Modeling*. Microsoft Press, 2004.
13. J.-C. Laprie, “Dependable Computing: Concepts, Limits, Challenges,” *Digest of Papers FTCS-15*, pp. 2–11, 1985.
14. *ISO/IEC:27002: Information technology - security techniques - Code of practice for information security management*. International Organization for Standardization (ISO), International Electrotechnical Commission (IEC).
15. *ISO/IEC 27005, Information technology — Security techniques — Information security risk management*. International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), 2008.
16. Microsoft, “Security Development Lifecycle,” Microsoft, 2010.
17. M. Tehranipoor and F. Koushanfar, “A survey of hardware Trojan taxonomy and detection,” 2009.
18. A. S. Scott Lambert Tomasz Ostwald Shawn Hernan, “Uncover Security Design Flaws Using The STRIDE Approach,” *MSDN Magazine*, 2006.
19. D. Powell, R. Stroud, and others, “Conceptual model and architecture of MAFTIA,” *TECHNICAL REPORT SERIES-UNIVERSITY OF NEWCASTLE UPON TYNE COMPUTING SCIENCE*, 2003.
20. J. L. Eric Byres, “The Myths and Facts behind Cyber Security Risks for Industrial Control Systems,” British Columbia Institute of Technology, 2004.
21. K. Wilhoit, “Who’s Really Attacking Your ICS Equipment,” Trend Micro Incorporated, 2013.
22. N. Viola, S. Corpino, F. Stesina, and M. Fioriti, “Functional Analysis in Systems Engineering: methodology and applications,” 2012.
23. U. Meyer and S. Wetzel, “On the impact of GSM encryption and man-in-the-middle attacks on the security of interoperating GSM/UMTS networks,” in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, 2004, vol. 4, pp. 2876–2883.
24. M. Steiner and P. 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*, 2013.