TU WIEN Informatics

Kombinierte Modellierungstechniken für Safety und Security in der industriellen Automation: Eine Fallstudie für STRIDE-LM und FACT

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Technische Informatik

eingereicht von

Marta Chabrová
Matrikelnummer 01126520

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Kastner
Mitwirkung: Ing. Dipl-.Ing. Siegfried Hollerer, BSc

Wien, 1. September 2022

Marta Chabrová	Wolfgang Kastner

TU WIEN Informatics

Combined Modeling Techniques for Safety and Security in Industrial Automation: A Case study for STRIDE-LM and FACT Graph

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Computer Engineering

by

Marta Chabrová
Registration Number 01126520

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Kastner
Assistance: Ing. Dipl-.Ing. Siegfried Hollerer, BSc

Vienna, 1st September, 2022

Marta Chabrová | Wolfgang Kastner

Erklärung zur Verfassung der Arbeit

Marta Chabrová

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. September 2022

Marta Chabrová

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, ohne die die Anfertigung dieser Bachelorarbeit nicht zustande gekommen wäre. Zuerst gebührt mein Dank der Fakultät für Informatik an der TU Wien, wo ich die Möglichkeit bekommen habe, meine Bachelorarbeit zu schreiben. Für die Auswahl meines Themas möchte ich mich bei Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Kastner bedanken. Mein herzlicher Dank geht auch an meinen Betreuer Ing. Dipl.-Ing. Siegfried Hollerer, BSc, der mir immer zur Seite stand und mir konstruktive Kritik und hilfreiche Tipps zur Erstellung meiner Bachelorarbeit gegeben hat. Er war während des gesamten Schreibprozesses sehr nett und geduldig mit mir, ohne ihn würde die Arbeit nicht so vorliegen. Außerdem möchte ich mich bei Judita für das Korrekturlesen meiner Bachelorarbeit und ihre Anmerkungen bedanken. Ich möchte mich bei meiner Familie bedanken, dass sie mir das Studium ermöglicht und die ganze Zeit hinter mir gestanden hat. Zum Schluss möchte ich mich herzlich bei meinem Freund Matthias und meinen Freunden Peter und Terka dafür bedanken, dass sie mir auf dem ganzen Weg wichtigen emotionalen Rückhalt, Energie und Mut gegeben haben.

Acknowledgements

Here I want to thank all those without whom the writing of this Bachelor Thesis would not have been possible. My first thanks go to the faculty for informatics of the TU Wien, where I got the opportunity to write my Bachelor Thesis. For the selection of my research topic I would like to thank Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Kastner. My thanks also go to my tutor Ing. Dipl.-Ing. Siegfried Hollerer, BSc, who was always there to give me constructive feedback and helpful tips for the creation of my thesis. He was always very kind and patient with me and I don't think this thesis would have been possible without him. In addition I would like to thank Judita for the corrections and comments. I want to thank my family, who enabled me to study and stood by me the whole time. At last I want to thank my boyfriend Matthias and my friends Peter and Terka who gave me a lot of emotional support, energy and courage the whole way.

Kurzfassung

Durch die stetige Vernetzung von Automatisierungstechnik mit IT Systemen müssen Attacken frühzeitig erkannt und Sicherheitslücken ehestmöglich geschlossen werden. Durch die Modellierung von Bedrohungen (Threat Modeling (TM)) versucht man systematisch Schwachstellen zu erkennen und Antworten auf folgende Fragen zu finden:

- Welche Art von Attacken gefährden industrielle Automatisierungssysteme?
- Wie können diese Attacken auf eine methodische Art modelliert werden?
- Welche Methoden werden im Bereich der Information Technology (IT) Sicherheit eingesetzt um IT Threat Modelling (TM) Ansätze anzupassen?
- Welche Werkzeuge oder unterstützende Frameworks werden benutzt, um das TM in industriellen Automatisierungssystemen zu unterstützen?

Häufig sind TM Ansätze allerdings auf reine Daten- und Informationssicherheitsaspekte (Security) fokussiert und berücksichtigen kaum die Wechselwirkung zwischen Security und funktionaler Sicherheit (Safety), die gerade im Kontext von Automationstechnik von besonderer Relevanz ist. Die Arbeit versucht daher eine Beitrag zu liefern, der Antwort auf die folgende Frage gibt:

Wie kann die gegenseitige Abhängigkeit von Safety und Security bei der Bedrohungsmodellierung in industriellen Automatisierungssystemen berücksichtigt werden und welche Ansätze, Methoden und Tools eignen sich besonders.

In dieser Arbeit werden zwei unterschiedliche TM Methoden verglichen. Die Methode STRIDE-LM, welche nur auf den Security Aspekt spezialisiert ist, kombiniert mit dem Attackmodelling von MITRE ATT&CK (MA) wird mit der Einzelmethode Failure-Attack-CounTermeasure (FACT) Graph verglichen, welche sowohl Safety- als auch Security Aspekte abdeckt. Die beiden Methoden werden auf den gleichen Use Case angewendet. Der erzeugte Use Case enthält wichtige Komponenten der Automatisierungspyramide aus der Abbildung 3.1 und deckt verschiedene Industrie- und Automatisierungsbereiche ab. Als Inspiration für den Use Case wurde eine “Stakeholder Analysis” herangezogen [HKS21]. Die Ergebnisse zeigen, dass FACT Graph für die Modellierung besser geeignet

ist als STRIDE-LM. Im Rahmen der Untersuchung wurden jedoch auch Wege entdeckt, wie STRIDE-LM optimiert werden könnte, um auch mit dieser Methode potentiell bessere Ergebnisse zu erzielen. Diese wurden jedoch in dieser Arbeit nicht weiter untersucht.

Abstract

Because of the continuous integration of IT systems and automation technology, attacks have to be identified and security gaps closed as soon as possible. The modeling of threats (Threat Modeling (TM)) allows the systematic detection of vulnerabilities and answers the following questions:

- What kind of attacks threaten industrial automation systems?
- How can these attacks be modeled in a methodical way?
- Which methods are used in the field of Information Technology (IT)-security to customize IT-Threat Modelling (TM) approaches?
- Which tools or supporting frameworks are used to assist the threat-modeling in industrial automation systems?

TM - approaches are often focused on pure data- and information security aspects (security) and don't consider the interdependence between security and functional safety (safety), which is especially relevant in the context of automation technology. Therefore this Thesis tries to contribute to the answer of the following question:

How can the interdependence between safety and security be considered during the threat modeling in industrial automation systems and which approaches, methods and tools are best suited for this?

This thesis compared two different TM methods. The method STRIDE-LM, which focuses on the security aspect, combined with the attack modeling of MITRE ATT&CK (MA) is compared to the standalone method Failure-Attack-CounTermeasure graph, which covers both safety and security aspects. Both methods were used on the same use case. The use case created for this contains important components of the automation pyramid from Figure 3.1 and covers different industries and automation areas. A "Stakeholder Analysis" was used as inspiration for the use case [HKS21]. The results show that FACT graph is better suited for the modeling than STRIDE-LM. During the investigation some ways were discovered, to optimize STRIDE-LM and potentially receive better results but these approaches were not further investigated in this thesis.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Due to a fast growing demand, companies had to transform and modernize their production through the usage of automation systems. Various companies neglected the implementation of security in their automation systems during this process. There are several reasons why companies did not invest in the security of their systems, in many cases these included lack of time, money and manpower. Often companies acted according to the principle: Functionality first, cybersecurity later [BGK+18]. Only a small number of companies in the automation industry is working on the problem of the interdependence of security and safety. This problem could create critical "blind spots" that are attractive for attackers [HKS21]. This creates a bidirectional problem between security threats and safety hazards [SWT21]. These interdependencies require a lot of attention. Cyberattacks can have a wide range of consequences. Some of the most numerous consequences are loss of data or money, damage or destruction of goods and machinery and injury or even death of employees [HBB+21].

This thesis will investigate the interdependence between safety and security in the industrial automation sector. Two different Threat Modeling (TM) methods will be tested and their results analyzed regarding this interdependency. The first chapter introduces and explains the most important terms for this subject. It continues with an introduction into the dimensions of cyberattacks with attack structure and sequences and finishes with a summary of known attacks on an Industrial Control System (ICS). The second chapter provides information on TM in general and explains some individual methods. The methods used for the later comparison, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege and Lateral Movement (STRIDE-LM) and Failure-Attack-CounTermeasure (FACT) graph are analyzed in more detail. In the third chapter, the selected use case from the industrial sector is presented and necessary terms for the understanding of the modeling process in an ICS are introduced.

The fourth chapter covers the application of the selected methods to the use case and presents the achieved results. The results of the different methods are compared in the fifth chapter with an emphasis on the interdependence between safety and security. The sixth chapter presents the evaluation of the individual and combined methods.

1.2 Main terms

Threat: It is a danger of something affecting a system in a negative way [src].

Risk: Is the likelihood of a threat to realise its damaging potential. It is defined by two important parameters: source of risk (probability) and consequences (severity) e.g., financial, environmental or human [Bay15, SFS+11].

Weaknesses: Are errors, bugs or faults in software or hardware systems that might lead to a vulnerability. Commonly used glossaries, such as RFC 4949 and the National Institute of Standards and Technology (NIST) glossary do not define the term weakness [Enu21, src, Shi07].

Vulnerability: Is a weakness in a system that could be exploited or triggered by a threat. A system could contain multiple vulnerabilities, which can be divided into three different types, but not all vulnerabilities must lead to an attack [src, Shi07]:

- vulnerabilities in design or specification
- vulnerabilities in implementation
- vulnerabilities in operation and management

Attack: Is when a vulnerability is exploited to realise a threat. Attacks can be split into different groups: types of the attack, digital or physical and location of the attack, local or remote [HKS21].

Hazard: Is a source or a situation of potential damage of property or the environment or with the potential to harm in terms of human injury, ill-health or a combination of the two [fOHS22, oWA22],

Safety: A measure for the absence of risk, that would have a potential impact on the system's environment or humans affected by such impacts. Since the consequences associated with the risk can be unacceptably high, such as human losses, heavy material loss and nature damage, safety also considers hazards. A system would have a high level of safety if the risks and the associated hazards are low [KPCBH15, Shi07].

Security: Protecting information and systems from unauthorised access, disclosure, disruption, modification or destruction. Security considers threats and focuses on potential attacks and their impact on a system. A secure system would assure the integrity, confidentiality and accessibility of information and services [KPCBH15, src].

Interdependence between security and safety: Two random variables X and Y are considered interdependent, if X impacts Y or Y impacts X. Security and safety share many commonalities, for example both result in constraints, involve measures, create requirements and deal with risks. Both areas are important for the system and much can be gained by one adopting the knowledge, understanding, tools and techniques of the other and vice versa. Separating safety and security increases costs, implementation time and complexity of the system and dramatically reduces performance. Interaction between safety and security can exists in various kinds [KPCBH15]:

- Conditional dependency: Fulfillment of safety requirements conditions security or vice-versa.
- Mutual reinforcement: Fulfillment of safety requirements or safety measures contributes to security, or vice-versa, thereby enabling resource optimization and cost reduction.
- Antagonism: When considered jointly, safety and security requirements or measures lead to conflicting situations.

Threat modeling: Is a technique to describe a structured approach to analyse potential cybersecurity threats which impact a system because of its weaknesses, security vulnerabilities and design errors in the system layout. This structured approach optimizes the system, defining countermeasures and reducing the potential risk. The systematic identification of security threats provides the necessary security measures or security control. Various methods or combination of methods can be used to achieve this. TM is executed in four steps:

- Create a model of the system to by analyzed e.g., a data flow diagram
- Identify threats and create a threat list
- Specify threats and prioritize them
- Validate the effectiveness of the countermeasures

TM answers the following questions: Where are the vulnerabilities of the system which can be the target for a possible attack? What is the most relevant threat and what must be done to prevent attacks from occurring [Cob21, Gon20, Gra21, SWT21, MS16]?

Risk management: Is the ongoing process of identifying and handling risks in a cost-effective manner. The term handling means assessing, prioritizing, responding and controlling the threats. These risks stem from a variety of technological issues, accidents, strategic management errors or in the context of this work from the interdependence of safety and security requirements of the automation industry systems. This approach supports the critical and non-critical components of the system by providing convenient security solutions and considers the full range of

risks. The risk management process provides a necessary framework for the actions to be taken [HBB+21, SSM+16, SFS+11, Tuc21].

Risk assessment: A process combining risk identification, risk analysis and risk evaluation is risk assessment, which in itself is a part of risk management. It aims to break down threats into identifiable categories and defines all the potential impacts and likelihood of each risk. There are many ways available to conduct the risk assessment [HBB+21, SFS+11].

Risk modeling: Is a technique to quantify the likelihood of a cyberattack in an efficient manner. Risk modeling calculates risks and helps to identify which risks have the largest impact on the system [Gra21].

Attack vectors: An attack vector, or sometimes also referred to as a threat vector, is an avenue of approach that can be used by hackers as the means to gain unauthorised access to a system. Each attack vector targets one or more of the principal system vulnerabilities. A successful attack may lead to loss of control, loss of data or other changes undesirable to the system owner [HBB+21].

Attack modeling: This is a technique to identify and simulate possible attacks on the basis of found vulnerabilities. Modelling attacks helps the defenders to gain a better understanding of the behaviour, tactics and objectives of their adversaries. With attack modelling an organisation is able to save time, money and other resources. There are a number of different attack modelling techniques, for example: attack graph or tree, attack vector, attack surface, OWASP's threat model, Cyber Kill Chain (CKC) and diamond model [HQA+16].

Hazard analysis: The process of identifying the different types and sources of hazards that can have adverse impacts on people or the environment. It is a basic step towards risk assessment and risk management. A hazard analysis assesses the significance and degree of hazards to a specific system and then either eliminates or controls software component hazards [Bay15, CMN+17, WDY+14].

Malware: Short for "malicious software". Malware is software, firmware or hardware that is designed by cybercriminals to damage or destroy a system, ideally without the knowledge of the system's owner. There are many types of malware e.g., adware, spyware, viruses, botnets, trojans, worms, rootkits and ransomware. Each works differently to achieve its goals which could be to steal data or to leak private information. However all malware variants have two basic attributes in common: They are sneaky and work actively against the interests of the system's owner [Bel19, Cis21, src].

1.3 Cyberattack dimensions

1.3.1 Access vectors

Badly secured access vectors can be potential threat vectors. Attackers can exploit systems with a low rating in order to gain access to more critical systems connected to them. When vulnerabilities are exploited by using the mentioned attack vectors, undesired effects (scenarios) can be caused. In many cases, access vectors overlap with communication paths which frequently makes them critical points in a risk assessment. Common access vectors and possible threats are listed in Table 1.1.

Possible Access Vector	Possible Attack Vectors
Network (Ethernet, VPN)	ARP- spoofing, WLAN hooping
ICS Systems and Devices	Sensors, Actuators CPS, PLCs
Applications (Modbus, SSH)	Buffer overflow, manipulate User input
Physical access	USB Ports, HDMI, Display port
Users (social engineer-ing)	Phone, Emails, Internet browser, Social media application
Supply chain	Chip/hardware modification, Application code modification

Table 1.1: Common Attack Vectors based on [HKS21]

1.3.2 Type of attackers

Attackers fall into a few different categories that can be divided (distinguished) by features such as their goals (e.g., destroy, steal, disable), motivation (e.g., political, economical, socio-cultural, unauthorized access) and capabilities (e.g., technical skills, individual or organization, alter). Not all attackers are criminals, some are actually hired to find criminals or to test a system for potential vulnerabilities. This aggregation of hacker types ranging from beginner to professional is further described in the list below.

The first group are the hobbyist hackers. These are the “newbies” in the world of hacking. They are mostly harmless and do not cause heavy damage to the system. They can be further divided into two groups.

Script kiddies: These are amateurs, who learn by watching videos, reading online articles or forum discussions. They use existing malware, tools and scripts created by other hackers and use them for cyberattacks without having a complete understanding or knowledge of the tools functionalities or capabilities. The main motivation for script kiddies is to impress friends from their computer communities or to find new challenges [DeM19, Ins20, Sar21].

Green hat hackers: Like script kiddies, these are amateurs. The biggest difference to script kiddies is that green hat hackers want to know all details about their attacks.

They want to be full-fledged hackers and to constantly improve their technical skills. That is why they take skill development courses to learn new hacking techniques and programming [Ins20, Sar21, Sec21].

The next type of hackers belong to the advanced category and almost always conduct their attacks out of personal motivation.

Blue hat hackers: For this type of hackers, money and fame are not important. They use their capabilities to take revenge on people, employers, institutions or governments. Some attend special conferences or projects where companies often invite them to test new software to find security vulnerabilities before releasing it [Ins20, Sar21, Sec21].

Insider: They have enough information about vulnerabilities that can be exploited because they work for the organization they attack. The insiders threaten the security of the internal systems and work individually. Their motivation ranges from attacks for money, dissatisfaction with the company or finding and exposing illegal activities within the organization. There are different types of insider threats but there is no standard classification defined. As an example, Computer Emergency Response Team (CERT) defines malicious insiders and divides them into Information Technology (IT) sabotage, data theft and insider fraud [HKS21, Sar21, CMT12].

Cryptojackers: These are a very new type of hackers that emerged in the wake of the creation of cryptocurrencies. They exploit system vulnerabilities to steal computer and energy resources using them to mine cryptocurrencies. This is represented in their name which is derived from cryptocurrencies and hijacking. Because their malicious code should stay undetected as long as possible they try to work as silent as possible [Sec21].

The next group consists of hackers with various political agendas.

Hacktivist: This word is a combination of the words hacker and activist. These attackers are individuals or groups of hackers that target companies or organizations that are at odds with their religious beliefs, political agenda or social ideology. They use hacking as a form of protest and are not interested in financial gains. One of the most well-known groups of hacktivists is the Anonymous Collective [DeM19, HKS21, Sar21, Sec21].

Terrorist or Nation-State hacker: They are politically motivated attackers who are sponsored by a government or other vital businesses. The damage done by these hackers can be very high. They conduct sophisticated attacks like stealing highly sensitive information from other countries, damage critical infrastructure like traffic management systems or try to create international incidents [DeM19, HKS21, Sar21, Sec21].

The last group are professionals.

Cybercriminal: Sometimes also referred to as black hat hackers, they are attackers with a high level of knowledge and technical skills but with bad intentions. They steal sensitive data for financial gains or money directly and also either capture or destroy systems. These attackers work as individuals or in groups and can cause very high damage [DeM19, HKS21, Ins20, Sar21, Sec21].

White hat hackers: These are the opposite to black hat hackers and in contrast to their criminal counterparts they are authorized or certified professionals with expertise in cybersecurity. They protect a company's systems from cybercrimes by searching for and identifying vulnerabilities that can then be fixed. White hat hackers work according to rules and regulations defined by the government, they are therefore sometimes called ethical hackers [Ins20, Sar21, Sec21].

Red hat hackers: These are the "robin hoods" of the cybersecurity world. They are similar to white hat hackers in the way that they also try to find and disarm black hat hackers, but in contrast to white hat hackers, they often choose extreme and sometimes even the same illegal methods as black hat hackers [Ins20, Sar21, Sec21].

Gray hat hackers: These are experts in the "gray" zone between black hat and white hat hackers. They work just for fun with neither good nor bad intentions although they invade systems without their owners knowledge or permission but without the intention to rob or harm people or companies. They engage in hacking activities for fun, they love to find gaps in computer systems and experimenting with systems [Ins20, Sec21].

An overview of attacker types is given in Table 1.2:

Category	Name	Motivation	Skills
Hobbyist hackers	Script kiddies	Challenge, experience, gain credit in computer enthusiast communities	Utilize scripts or programs developed by others
	Green hat hackers	Focus on gaining knowledge	Their own knowledge and help from more experienced hackers
Hacker with personal motivation	Blue hat hackers	Test a new software, weapon to gain popularity	Special conferences or projects, or at home
	Insider	Personal grudge (data, money), illegal activities in the organization	Through knowledge of the organization and authorized access

Category	Name	Motivation	Skills
	Cryptojackers	Free resources to mine for cryptocurrencies	Network vulnerabilities to steal resources
	Hacktivist	Hacking as a form protest	Network vulnerabilities to steal information
	Terrorist or Nation-State hackers	Money and nationalism	Network vulnerabilities to steal the information
Professionals hackers	Cybercriminal (black hat hackers)	Profit (money, data, identity)	Advanced technical knowledge to find vulnerabilities in computer systems and software (illegal)
	White hat hackers	Job, find vulnerabilities	Authorized cybersecurity experts
	Red hat hackers	Stop the attack of black hat hackers	The same tactics as black hat hackers
	Gray hat hackers	Personal (fun, improve own abilities), experimentation	Experts

Table 1.2: Attackers

1.3.3 Concept of a cyberattack

A cyber attack consists of various phases that can be represented with attack vectors where attack vectors are the avenues through which a threat source accesses the vulnerability. For a better understanding and reaction to adversary intrusions the life cycle of a cyberattack is represented in an attack model. The attack model CKC is presented in Figure 1.1. This model shows a sample Industrial Internet of Things (IIoT) Architecture that is roughly divided into two zones, end to end IT and Operational Technology (OT), and is further split into 5 levels. The target of an industrial attack in general can be any asset in the IT, the OT or the Demilitarized Zone (DMZ) but mostly the ICS in the control zones (level 0-2) is targeted. The life cycle contains the following phases [HB18, HKS21]:

- **Reconnaissance:** Identification of the target with the help of research (Internet, social media, interview or information on specific technologies) and data selection.
- **Weaponization:** Payload development (malicious code or binaries), it can be automated.
- **Delivery:** Transmission of the malicious code, breaking through the confidential barrier

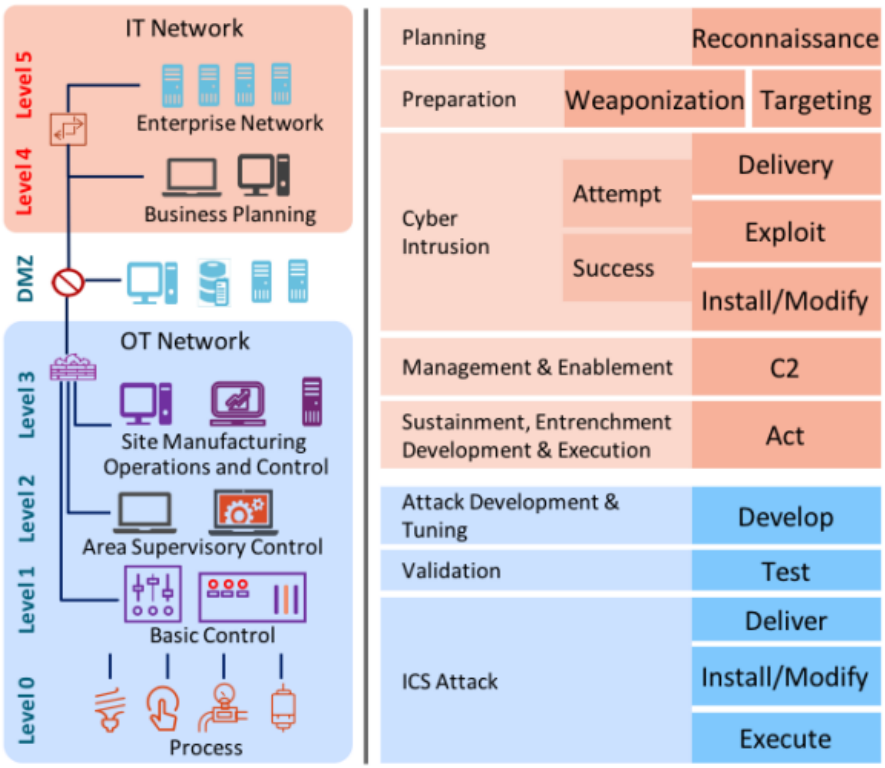

Figure 1.1: IIoT Zoned Architecture (left) and CKC for ICS based on [HB18]

- **Exploitation:** Execution or triggering of the code once successfully delivered to the target. The exploitation can be triggered automatically or manually by the system users without their knowledge. It uses the vulnerabilities of an application or operating system.
- **Installation:** Establishing the connection from the target to the attacker (remote access trojan or backdoor).
- **Command and Control (C2):** The attacker creates a connection to a command and control server.
- **Actions on Objectives:** The attack is successful and the attackers can now pursue their objectives, e.g., data exfiltration, manipulation of data integrity and availability or the attacked asset can be used to gain access to other network components.

It was discovered through observation that the attackers are often presented with ample time to execute their attacks. The duration to complete each phase can vary from seconds

up to several months, this is shown graphically in Figure 1.2. Most attacks on IIoT are multi-step attacks. The main attack vector focuses on access to the IT systems followed by multiple low and slow attacks that try to exploit system vulnerabilities as a way to gain access to the OT infrastructure. There are many reason why these attacks are not prevented [HB18]:

- lack of technology
- lack of knowledge and resources
- lack of adequate incident triage and investigation process

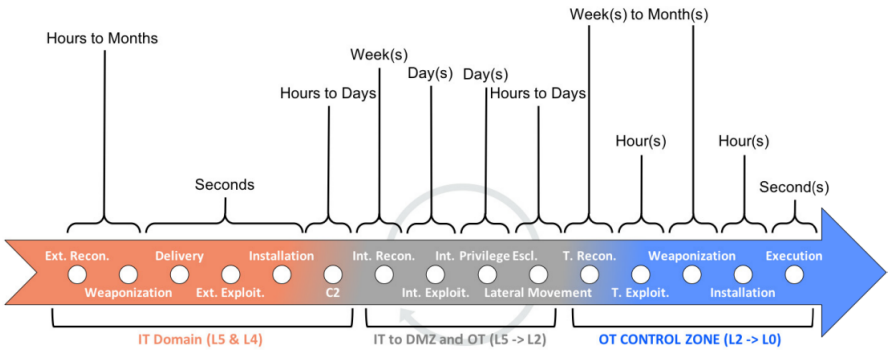

Figure 1.2: IIoT Attack Life Cycle [HB18]

1.3.4 Known attacks on ICS

For an attack to move forward, the attacker has to be motivated to attack the system and additionally the system needs to have at least one vulnerability. With the modernisation of ICSs the damage potential is high and the industrial sector has seen an enormous growth in attacks throughout the last years. The German federal office for information security (BSI) has created a list with the 10 most important threats [oCS19]:

1. Infiltration of malware via removable media and external hardware
2. Malware infection via Internet and intranet
3. Human error and sabotage
4. Compromising of extranet and cloud components
5. Social engineering and phishing
6. (D)DoS attacks

7. Control components connected to the Internet
8. Intrusion via remote access
9. Technical malfunctions and force majeure
10. Compromising of smartphones in the production environment

In Table 1.3 below, some successful, and well-known, attacks are listed:

Year	Name	What	Where	Reference
2017	TRITON	Led to a safety shutdown	Middle Eastern oil and gas petrochemical	[PDC18]
2017	Cyberattack Hits Deutsche Bahn - WannaCry ransomware	It has disrupted display of digital passenger information on station monitors.	Germany	[Sap17]
2018	Ryuk	Targeted ransomware that changes its demand depending on the victim's assumed ability to pay the ransom	Around the world	[apoG19]
2019	LockerGoga	Ransomware that blocked the ability to connect to the production systems	Norwegian aluminum manufacturing company (Norsk Hydra), US chemical enterprises Hexion and other	[ACS19]
2020	SolarWinds	Backdoor inserted into the product	A customer who downloaded the product	[sa21, fIS21]
2018	RMS and TeamViewer - Malware	Steal important documents to blackmail for money	Majority in Russia	[Kop20]
2021 January	Lazarus - Mata Malware	Cyber-espionage	Legitimate South Korean security software, a company developing asset monitoring solutions in Latvia	[CER21a]

Year	Name	What	Where	Reference
2021 May	Colonial Pipeline ransomware attack - DarkSide	Impacted computerized equipment managing the pipeline	Houston, Texas	[Fun21]
2021 June	Pseudo-Manuscrypt's Malware	Malware attack that created many access points to ICS-computers, mainly in mechanical engineering and automation	Whole world but mainly India, Vietnam and Russia	[Roo21, CER21b]
2022 January	Tesla	No damage done, potential vulnerability announced that could allow access to car systems	Origin Germany, target Tesla cars worldwide	[APO22]

Table 1.3: Attacks on Industrial Control Systems

CHAPTER 2

Threat Modeling Techniques

"Threat modeling is the key to a focused defense. Without threat modeling, you can never stop playing whack-a-mole."— Adam Shostack [Sho14]

2.1 State-of-the-art

There are various approaches for TM. These can be divided into four main groups displayed in Figure 2.1.

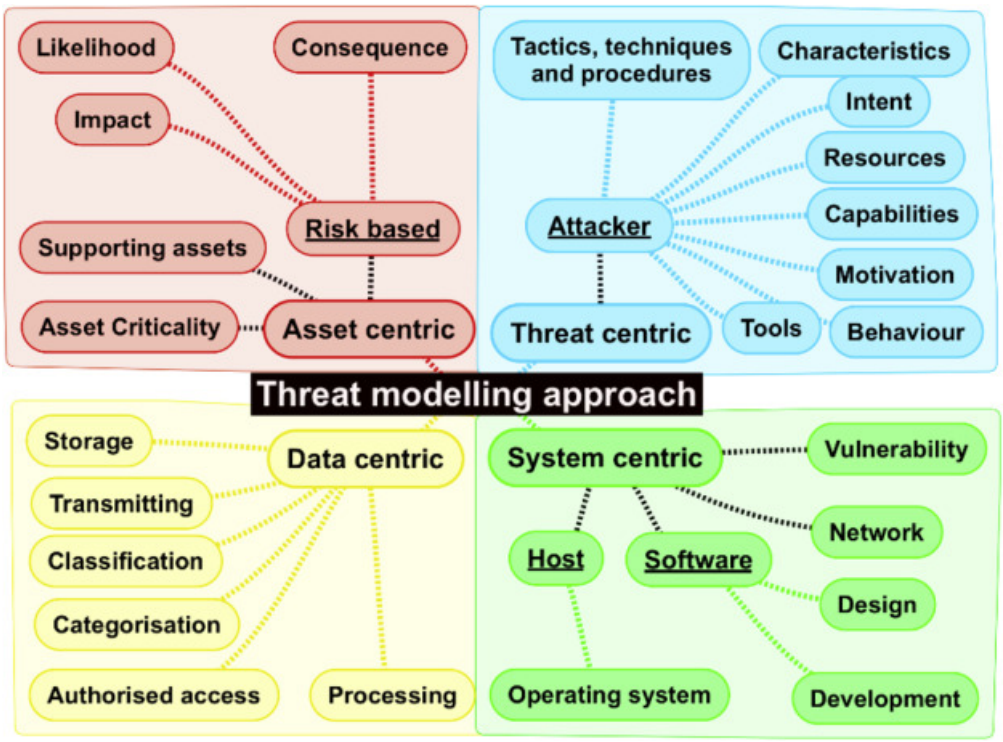

Figure 2.1: TM approaches [TSAK21]

The first group uses an asset centric approach that focuses on the elements like data, companies or an individual person. The next group uses a data centric approach and is based on a compilation of information from NIST, which does research that focuses on the protection of specific types of data. The third group uses a system centric approach to threat modeling. It tries to identify all components of the software and searches for their potential related attack vectors. The last group uses a threat-centric approach and focuses on the evaluation of potential targets for the attacker and investigates the corresponding attack vectors [TSAK21, SCO+18].

In order to decide which TM approach to use, the needs of the system project and its specificity should be considered. For example, if the project focus lies more on the safety or the security aspect of the system, the decision for a certain approach can have a big impact on the project strategy. One important parameter also is which hardware and software is used for the system and what level of complexity is required for certain operations. Finally, the field of operation should also be considered when choosing an approach [TSAK21].

Another way to divide the TM approaches into groups is by how they are created and displayed. The distribution of some well-known TM methods according to this distribution scheme is shown in Figure 2.2. The following text will shortly describe some important methods representing each quadrant of Figure 2.2. FACT graph and STRIDE-LM were selected for the modeling in this thesis and will be explained in more detail after this section [TSAK21].

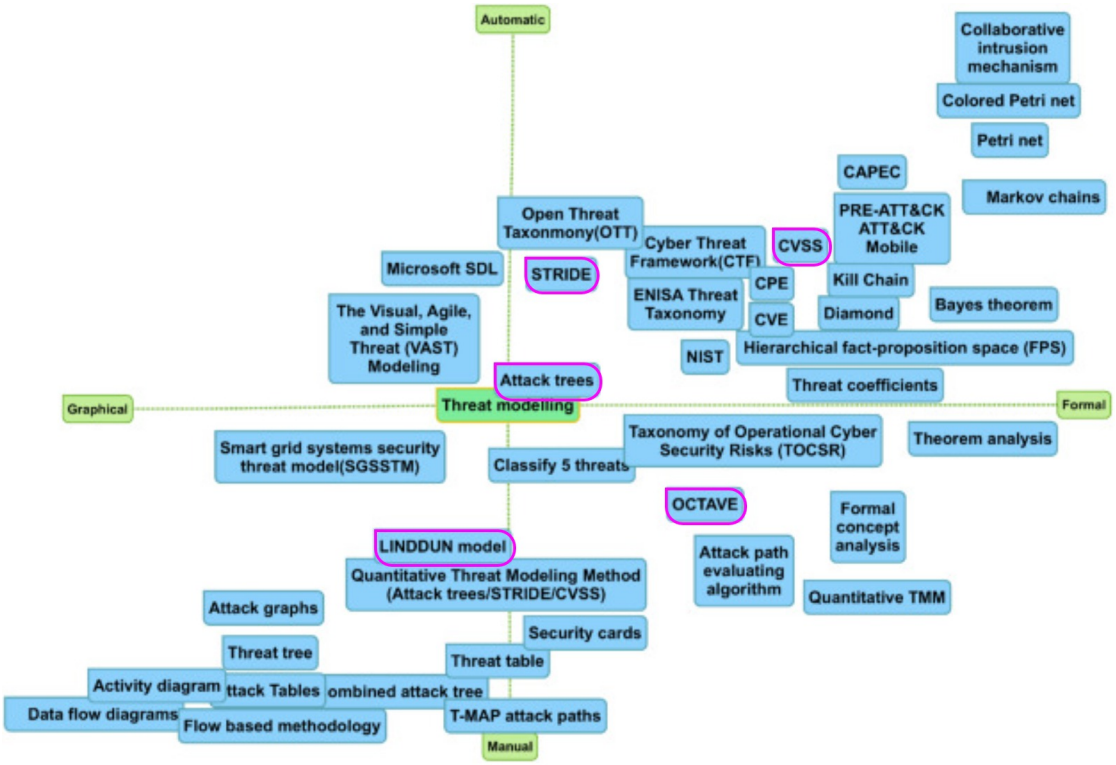

Figure 2.2: Quadrants identifying Automated/Manual and Formal/Graphical Threat Models [TSAK21]

ATASM: The Architecture, Threats, Attack Surfaces and Mitigation also referred to as ATASM is a TM approach for modeling threats from a security architect's perspective. Like the name suggests, this method consists of four parts. First, architecture decomposition into logical and functional components of the system. Second, identification and listing of possible threats. Third, cross-referencing threats with attack methods and system inputs to generate a list of credible attack surfaces. Fourth, application of new security controls to mitigate threat agents and threats to credible attack surfaces. This method uses only information about architecture [OEP20, BWCD+17].

CVSS: The Common Vulnerability Scoring System (CVSS) is an open framework developed by NIST and maintained by the Forum of Incident Response and Security

Team (FIRST) with support and contributions from the CVSS special interest group. It is used to communicate the main technical characteristics of software, hardware and firmware vulnerabilities. This approach consists of three metric groups: Base, Temporal and Environmental. These are shown in Figure 2.3.

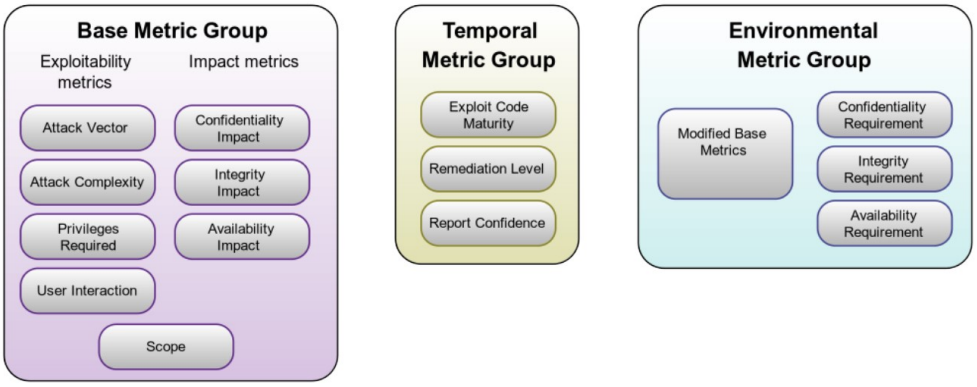

Figure 2.3: CVSS metric groups [FIR19]

Whereas the Base Metric Group contains the properties of vulnerabilities that can be considered constant regarding time and user environments, the Temporal Metric Group contains those that change with time. Finally, the Environmental Metric Group contains those properties that can be attributed to a special user environment. An online interface is available to calculate the system score. The results are consistent if the calculations are repeated but sources vary regrading the characteristics and scoring methodology of the calculations themselves and it is unclear if they can be considered transparent or not. The information gained through an CVSS analysis can be used as an input to an organizational vulnerability management process and the CVSS method is often used as support to other TM approaches [SCO+18, FIR19].

HAZOP: Hazard and Operability (HAZOP) was developed by the English ICI group of companies. A team of various internal and external experts with different backgrounds defines the systematic approach based on guide words. These guide words can be used by small teams to assess the possible hazards. The words define the deviation of a process, device or system from a given plan which can then be discussed in combination with their respective parameters. Similar to Fault Tree Analysis (FTA), the use of HAZOP is also recommended by ISO 14971 and is applicable to all types of specific operational sequences in planned or existing systems [Kle19, sü22].

LINDDUN: Stands for Linkability, Identifiability, Non-Repudiation, Detectability, Disclosure of Information, Unawareness and Non-Compliance (LINDDUN). This

method provides a systematic approach which focuses on privacy concerns and can be used for data security. This TM consists of six steps which are shown in Figure 2.4.

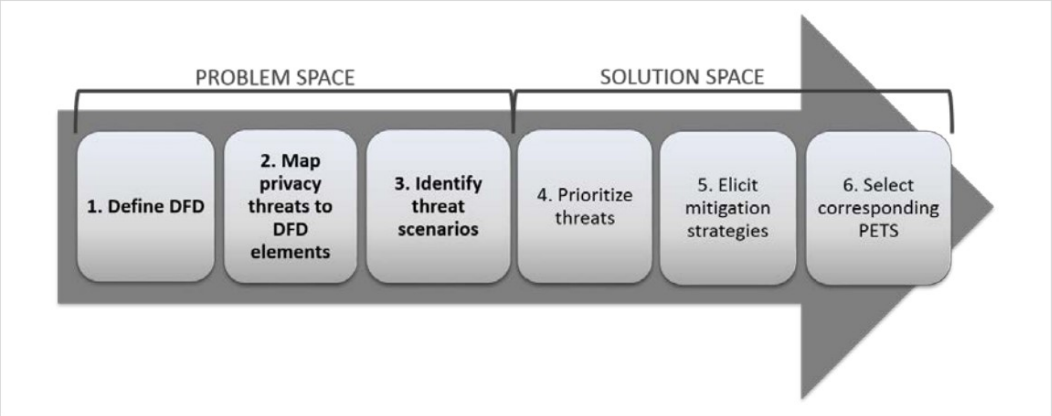

Figure 2.4: LINDDUN Methodology Steps [SCO+18]

It starts with the definition of a Data Flow Diagram (DFD) of the system considering communication, data stores, processes and external entity links to identify security threats. In the second step, threat categories are mapped to areas or parts of the system where they may appear, finally in a third step, scenarios are identified in which these threats could occur. This concludes the problem space with the following three steps focusing on finding solutions and mitigation strategies. A lot of documentation and an extensive privacy knowledge base is available for LINDDUN but it is also a labor intensive and time consuming method due to the rapidly growing number of threats especially with increasing system complexity. It shares this issue with the Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege (STRIDE) method [SCO+18, HKS21].

OCTAVE: The Operationally Critical Threat Asset and Vulnerability Evaluation (OCTAVE) is a risk based strategic approach to streamline and optimize the assessment of cybersecurity in an efficient way. It is a self-directed method that addresses organizational but not technological risks. This means that members of the organization need a comprehensive knowledge of the business and security processes at their disposal and also have professional experience because it is their responsibility to define the security strategy. OCTAVE was created in 2003 by CERT, a division of the Software Engineering Institute of Carnegie Mellon University. It was refined in 2005 and has vague documentation. The approach is divided into three phases which are displayed in Figure 2.5.

Phase 1 is organizational view, which builds asset based threat profile. Phase 2, technological view, focuses on the identification of infrastructure vulnerabilities and the last phase is security strategy and development.

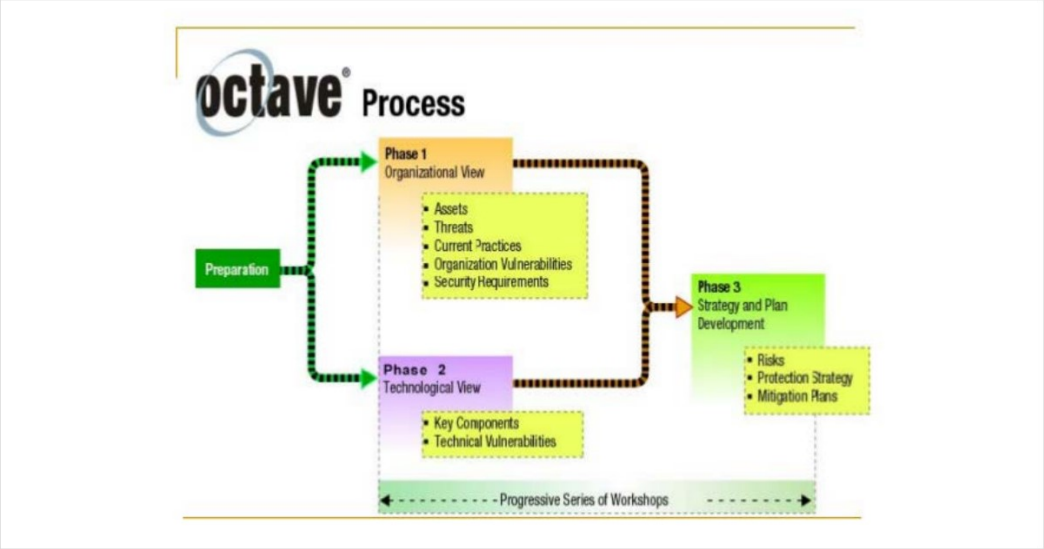

Figure 2.5: OCTAVE Phases [SCO+18]

OCTAVE was developed into two versions that are intended to be used by either large or small organizations using this approach. OCTAVE Allegro is the version intended for larger organizations which are structured into many divisions. OCTAVE-S is intended for smaller organizations with a flat hierarchical structure [SCO+18, Eni22, HKS21, CSYW07].

PASTA: Process for Attack Simulation and Threat Analysis (PASTA) is a risk centric approach developed in 2012 by Tony UcedaVélez with very rich documentation to help with this laborious and extensive process. It contains seven steps each with multiple activities, this is shown in Figure 2.6. PASTA combines two different areas, business objectives and technical requirements. In this method, various tools are used in single steps to find the optimal strategic asset-centric output in the form of threat enumeration and scoring. A vital role is played by input information about operations, governance, architecture and development and also by people that can make decisions [SCO+18, Uce12].

SAHARA: Security Aware Hazard and Risk Analysis (SAHARA) is a combined approach of Hazard Analysis and Risk Management (HARA) according to automotive safety standard ISO26262 and encompassed threats of the security domain STRIDE. SAHARA enables the quantification of probability, occurrence and impacts of security issues on the safety concept. It is a systematic approach but it can only be used to identify threats or hazards and does not provide solutions for the identified problems. Further it is limited to the automotive industry [MSB+15, HKS21].

VAST: The Visual, Agile, Simple Threat modeling (VAST) method is based on the automated TM platform ThreatModeler. VAST focuses on covering the entire Soft-

Figure 2.6: PASTA Stages [SCO+18]

ware Development Lifecycle (SDLC) across an organization. Due to its scalability and usability this method is easily adopted by large companies for their entire infrastructure. The method provides reliable and usable results for different stakeholders, this is represented by the three important pillars: Automation, Integration and Collaboration. For the implementation of VAST it is necessary to create two different types of models [She18, Moh21].

Application threat models: uses process-flow diagrams and represents the architectural point of view.

Operational threat models: uses DFD and represents the threat from the attacker's perspective

2.2 STRIDE-LM

The STRIDE threat modeling method was developed in 1999 by Praerit Garg and Loren Kohnfelder for the Microsoft company and was originally only intended to be used for the internal security software. STRIDE is an acronym for six different threat types: spoofing, tampering, repudiation, information disclosure, denial of service and escalation of privilege. These individual threat types are shown in Figure 2.7 and will be explained in more detail in the following text.

STRIDE-LM	Threat	Property	Definition	Controls
S	Spoofing	Authentication	Impersonating someone or something	Authentication Stores, Strong Authentication mechanisms
T	Tampering	Integrity / Access Controls	Modifying data or code	Crypto Hash, Digital watermark/ isolation and access checks
R	Repudiation	Non-repudiation	Claiming to have not performed a specific action	Logging infrastructure, full-packet-capture
I	Information Disclosure	Confidentiality	Exposing information or data to unauthorized individuals or roles	Encryption or Isolation
D	Denial of Service	Availability	Deny or degrade service	Redundancy, failover, QoS, Bandwidth throttle
E	Elevation of Privilege	Authorization / Least Privilege	Gain capabilities without proper authorization	RBAC, DACL, MAC; Sudo, UAC, Privileged account protections
LM	Lateral Movement	Segmentation / Least Privilege	Expand influence post-compromise; often dependent on Elevation of Privilege	Credential Hardening; Segmentation and Boundary enforcement; Host-based firewalls

Figure 2.7: STRIDE-LM - Threat Categorization, Security Properties and Controls [MF14]

Using STRIDE as a mnemonic for identifying computer security threats stayed popular in the community and was embedded into a loose threat modeling methodology. In 2019, Michael Muckin and Scott C. Fitch added Lateral Movement extending the TM method to STRIDE-LM, which is primarily a system-of-systems type of threat [BMF18, MF14]. This approach shows the threats from the perspective of the software developer and answers the questions “What are you building?” and “Are my components protected by the trust boundaries?”. When using this method, the first step is to create a DFD that identifies system entities, events and boundaries of the system. A more detailed DFD model leads to a more precise result of the STRIDE-LM analysis. In the next step, the system is divided into the relevant components. The individual components are analyzed regarding their susceptibility to attack vectors and to find mitigations for identified vulnerabilities. This process can be repeated until the remaining threats are deemed acceptable. [Sho14] provides a more detailed look at the topics of property violations, describing threats, typical victims and what an attacker does. Finally, the results should be documented and individual vectors out of the pool of potential attack vectors should be prioritized [SCO+18, BMF18].

Threat types, examples and how to treat them are described in the following.

- **Spoofing:** Is impersonating another user or using other machine authentication information to gain access to a system. Spoofing typically compromises authenticity. A typical example for spoofing is the misuse of a username and password to create various items like fake files or machine processes under a false identity. This threat can be minimized with the use of strong authentication [Mic09, itGtCCRAfCII20, Too09].
- **Tampering:** Is the permanent change of data without authorisation, which is almost always malicious in nature. Tampering typically compromises integrity. Examples for tampering are changing files, databases, memory spaces, network configurations or network communications. Tampering can be minimized by using proper access checks and encryption [Mic09, itGtCCRAfCII20].
- **Repudiation:** is the claim that certain actions were not carried out by the user and therefore no responsibility for the changes is taken. An example is the illegal operation in a system where tracking is not possible but a testimony of innocence would seem plausible. A system can be protected against repudiation by introducing robust logging and digital signatures [Mic09, itGtCCRAfCII20, Too09].
- **Information disclosure:** Publishing information to people who are not authorized to access this particular information. Such attacks are usually breaches in confidentiality. Frequent cases are granting access to data stores with sensitive data but it can also occur on secondary channels like processes or networks. Information disclosure can be suppressed by using encryption for data storage, use or transit [Mic09, itGtCCRAfCII20, Too09].
- **Denial of Service (DoS):** Is a threat that obstructs or impairs a service for the valid user. DoS is usually an attack on availability. This can happen when on exceptional amount of resources is used which makes services like memory, processing power, network resources or data storage unusable or unavailable. A system can be protected against DoS attacks by proper checking of legitimate traffic and putting redundancy mechanisms in place [Mic09, itGtCCRAfCII20, Too09].
- **Elevation of privilege:** Is when unauthorized users gain access to the system in order to create a privileged access account which gives them the possibility to access, change, damage or destroy any part of the whole system. To avoid this problem, it is advised to apply the principle of least privilege and to have strong privileged account protection mechanisms [Mic09, itGtCCRAfCII20, Too09].
- **Lateral Movement:** This classification was added later for better applicability. Lateral movement is pivoting across the network, often in cooperation with elevation of privilege to gain access to other parts of the network and to find possibilities to access important data or damage the system. A mitigation for this threat is the definition of strict firewall rules and proper system segmentation [itGtCCRAfCII20].

STRIDE-LM is a very popular approach in security software engineering. The following paragraphs provide a closer look at the advantages and disadvantages for this method.

2.2.1 Disadvantages

Like every other TM method STRIDE-LM is also unable to provide a 100% security guarantee. It is therefore important to feed STRIDE-LM with information about the system that is as detailed as possible. This leads to another negative point because with the increase in complexity the number of threat issues increases also and the required processing time is increased accordingly. The approach is not focused on safety and does not use a common metric. The STRIDE-LM method should not be the sole focus in the search for potential threats because it cannot perform an attack modeling or risk analysis. The study by Scandariato et al. about the STRIDE technique concluded that it produces results with a moderately high rate of false negatives [HKS21, itGtCCRAfCII20, SCO+18, HLOS06].

2.2.2 Advantages

STRIDE-LM is a popularly applied method because it incorporates the accepted Confidentiality, Integrity, and Availability triad (CIA) and because it is efficient. This method has low overall cost because there are freeware tools available (e.g., Microsoft Threat Modeling Tool (MS TM Tool), OWASP Threat Dragon), easy to use and therefore does not require the employment of specialized security experts. Developers can easily identify and change the found security vulnerabilities in a cost-efficient way. STRIDE-LM provides technical and organizational countermeasures and can be used for both cyber-only and Cyber-Physical Systems (CPS). The study by Scandariato et al. about the STRIDE technique concluded that it produces results with a low rate of false positives [HKS21, JAMH22, Mic22].

2.3 Failure-Attack-CounTermeasure

FACT is a graphical TM method. FACT is one of the few methods that covers the interdependencies between safety and security. This is a very important feature that allows easy analysis of CPS activities. The method is based on the alignment of the safety and security standards ISA84 (IEC 61511) [oA04] and ISA 99 (IEC 62443) [Tr07], which were developed by the International Society of Automation (ISA) and International Electrotechnical Commission (IEC). The integration of these standards is achieved by combining the corresponding security and safety lifecycle phases creating a unified lifecycle with 14 individual phases. This unified lifecycle is shown in Figure 2.8.

The first four phases cover the safety risk assessment and design phases while the security assessment and design phases are covered in phases 5 through 9. The FACT graph is part of phases 1-9 and is explained in detail in the following text. In phase 10, safety and security are aligned and finally safety and security lifecycles are merged in phases 11-14.

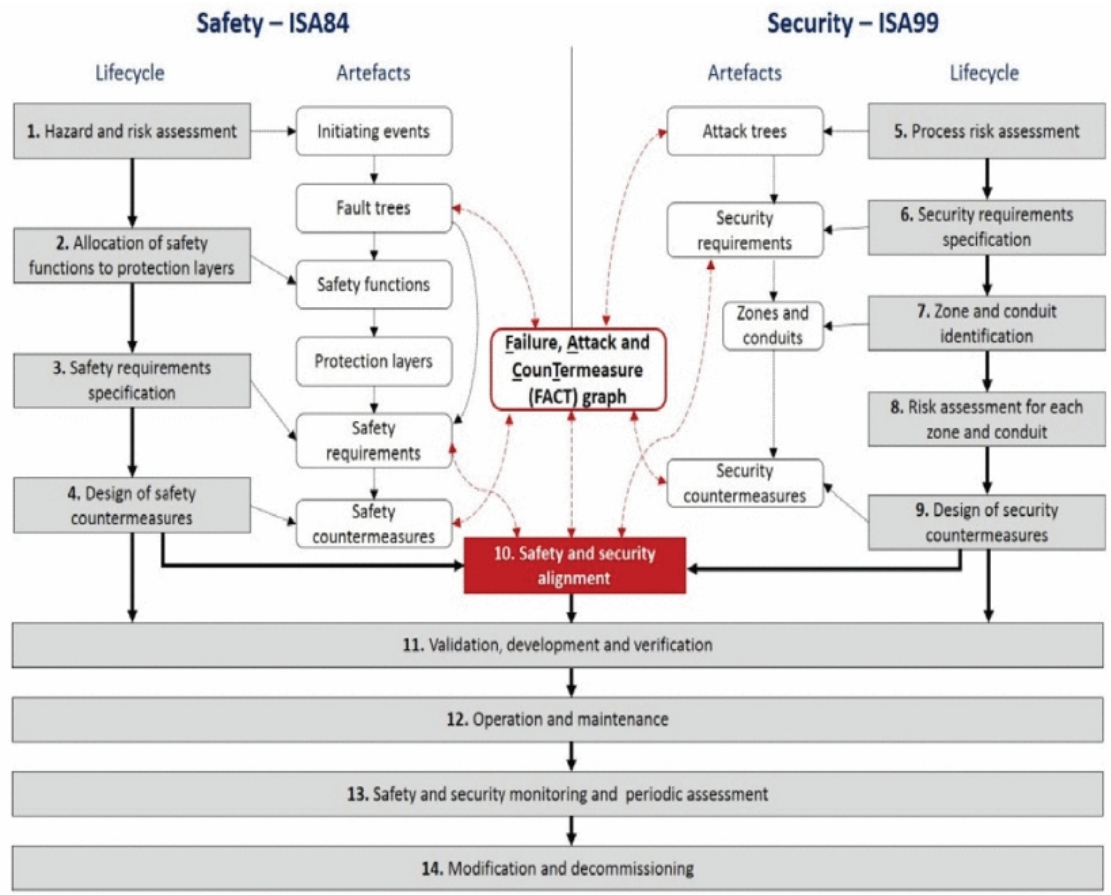

Figure 2.8: Corresponding security and safety lifecycle phases [GSO17]

These include validation, development and verification, operation and maintenance, safety and security monitoring and periodic assessment, modification and decommissioning related activities [KKG20, CS17, SM15, GSO17]. FACT incorporates safety artifacts (fault trees and safety countermeasures) and security artifacts (attack trees and security countermeasures) which are used to prevent outages and attacks and to keep up the required protection standard on all system levels even during CPS operations. The suggested alignment of safety and security allows the creation of a unified implementation for the real time vulnerability analysis. This coordination is very important for the good cooperation between safety and security which could otherwise result in an inefficient CPS development and a partially unprotected system [GSO17, SNRM17, SM15].

FTA: Is a popular graphical safety failure analysis tool for hazard and risk management, which uses a special notation to display the logical connections between specific system failures and their respective causes. With FTA, Boolean Logic is used to analyze the system to pathways that lead to the failure cause, as Figure 2.9

demonstrates. Starting from a single point at the top, events are tested based

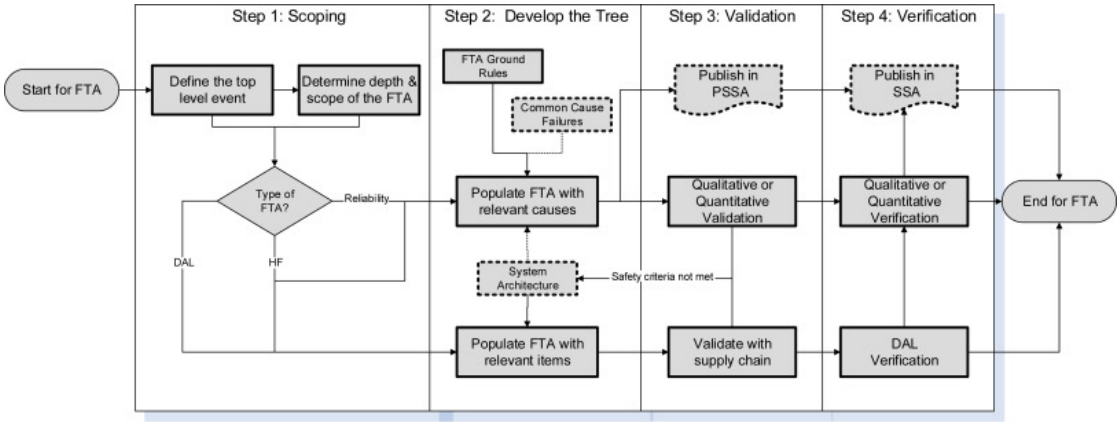

Figure 2.9: An FTA process flow [Kri17]

on true/false statements and sorted in chains. This creates a logic diagram that displays the root of the failure. The use of FTA is recommended by ISO 14971. It has the ability to display both normal and fault events, which can cause undesired events. The graph consists of three different components [Kri17, Kle19, SM15]:

- **Nodes:** undesired events in the system
- **Gates:** AND or OR for relations between nodes
- **Edges:** path of the undesired events through the system

Attack Tree: Is a graphical technique used for security risk management that describes the individual steps of the attack cycle. The graph consists of three different components which are identical to the three FTA components [SM15]:

- **Nodes:** represent attacks
- **Gates:** AND or OR for relations between nodes
- **Edges:** path of attacks through the system

The FACT TM method is designed in four steps, which are presented here in detail [KKG20, CHP+17, SM15]:

1. Import analyzed failure trees. The fault trees are connected using AND and OR gates to create a complete picture of the possible errors of the system.

2. Attach the safety countermeasures to the failure nodes which they are supposed to prevent. This technique gives an overview over the coverage of safety failures by safety countermeasures.

3. Import analyzed attack trees to the corresponding safety failure nodes in the FACT graph with the help of OR gates, which indicate that a failure may be caused either by accidental failures, or by intentional attacks.

4. Attach the security countermeasures to the attack nodes. This can be done on the basis of the Attack Countermeasure Tree (ACT)-technique [RKT12] which allows the attachment of security countermeasures to any node of the attack tree.

2.3.1 Advantages

FACT graph is a widely used approach to identify potential cyberattacks. It has a broad range of ways in which it can be used like CPS development and maintenance, safety and security verification or monitoring and periodic assessment. The FACT graph helps you to identify error misalignment, double or missing countermeasures to both safety and security issues. The FACT graph provides information that allows the user to associate countermeasures to certain faults and attacks which in turn makes development much easier [CS17, SNRM17, SM15].

2.3.2 Disadvantages

The FACT approach does not include the creation of a risk analysis and is therefore unable to present a risk evaluation. It is not easy to combine this approach with others because the graph does not use common metrics for threat evaluation [CHP$^+$17, HKS21].

CHAPTER 3

Use case

This chapter will introduce the use case which is shown in Figure 3.2. For the modeling of the use case it is necessary to know two terms, which are important for automated industrial systems. They will be explained in the following chapter.

3.1 Automation pyramid

The structure was originally developed in the 1980th for the definition and integration of strategic, tactical and operative IT systems. The industrial automation pyramid is built from five different layers using the ANSI/ISA-95 standard which is shown in Figure 3.1. The levels represent the hierarchy of various devices in a semi-automated or automated system. Each device has its own specific attributes like requirements, types of requirements, reaction time. These will now be further explained with examples from the use case. The structure of the pyramid is also used to graphically display the information exchange between individual devices on each level and levels themselves [RPC19, KBK+19, CRV+20, MPMM21].

Field level: also referred to as sensor/actor level, is the lowest level of the pyramid where real-time behavior, low latency and low jitter for control applications take place [RPC19, CRV+20].

Control level: The next level coordinates sensors and actuators. Information exchange and processing between field level and control level is done in milliseconds [RPC19, CRV+20].

Supervisory level: Is the middle level and the central part of the automation system-unit and therefore has the highest priority regarding interaction with the lower levels. The information from the field and control level is filtered and sent to the supervisory

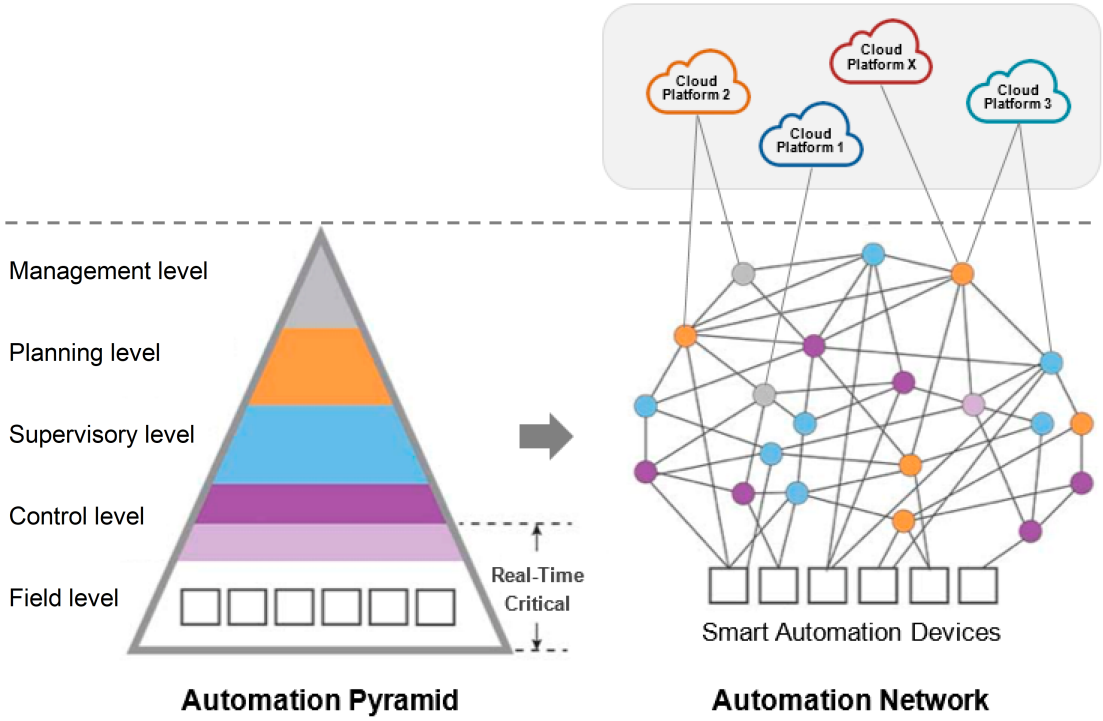

Figure 3.1: Evolution of the hierarchical model toward an integrated network [ZSM+19]

level in a given timeframe. This information is received by the supervisory level in milliseconds but larger intervals are also possible [RPC19, CRV+20].

Planning level: At this level, latency and real-time becomes less important for the data exchange because the systems at this level are dedicated to the monitoring of key process indicators. The level serves as an interface for the detailed planning of the manufacturing process, quality management, data collection, material management and KPI documentation covering the complete arc from the raw materials to the finished product [RPC19, CRV+20].

Management level: This is the top level of the pyramid where all elements of a process are managed. The fundamental role of the management level in an automated process is to provide information about the availability of resources, building elements and spare parts or time management for the predictive and corrective maintenance. An Enterprise Resource Planning (ERP) is typical for this level, where the business information is processed and documented e.g., supply chain, amount of products and accounting information. The management level is also responsible for the communication outside of the system with customers and business partners [RPC19, CRV+20].

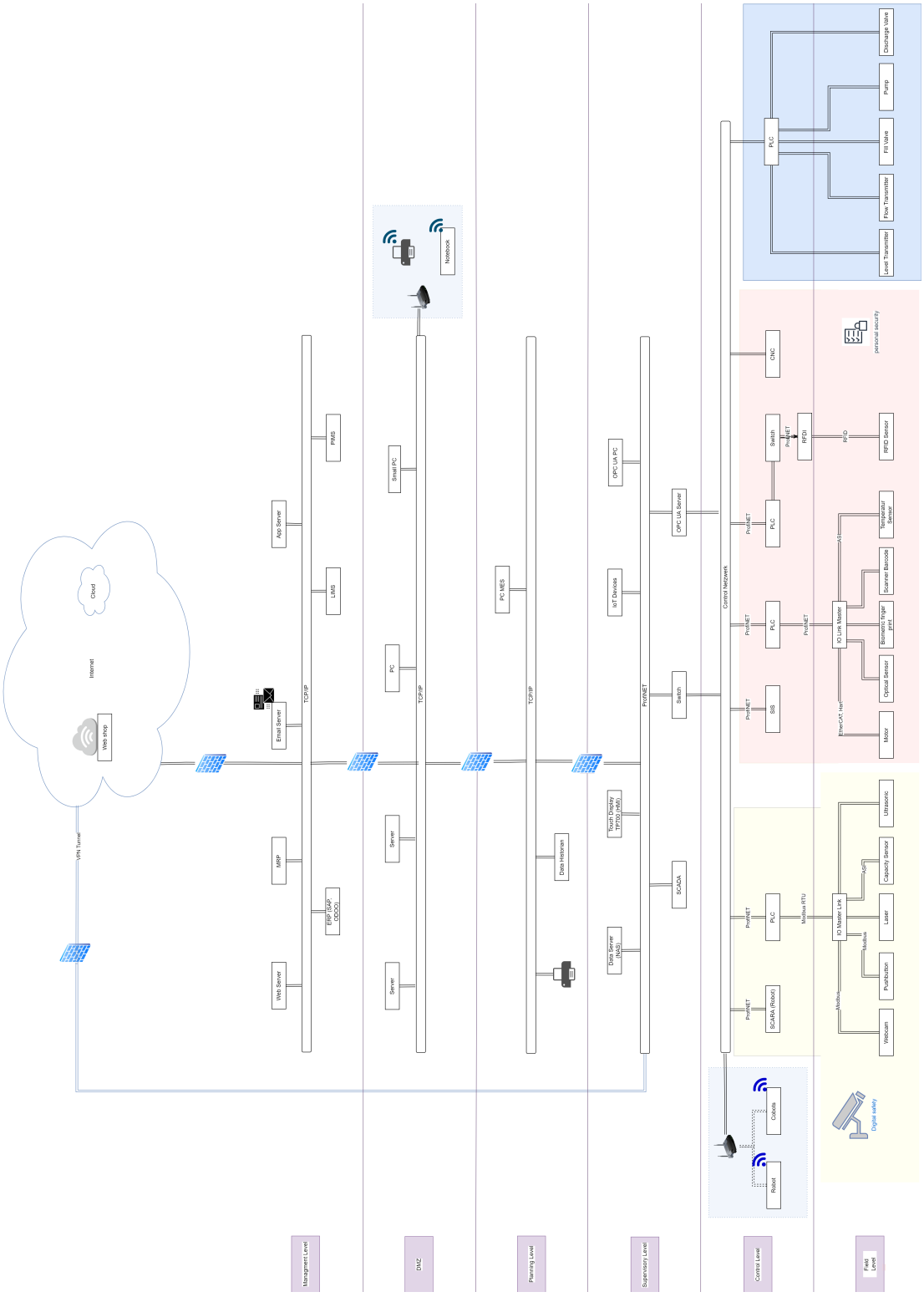

Figure 3.2: Use case

3.2 Industry 4.0

Named like the 4th industrial revolution, this phrase was first used by the German government when they announced their new high tech strategy project in the year 2011. It is a collective term for a multitude of actual concepts but the exact distinction between these concepts is often blurred or not possible for each individual case [CRV+20, LFK+14, HPO+15].

The components are [LFK+14]:

- **Smart factory:** Manufacturing is completely equipped with sensors, actuators and systems capable of a high level of autonomy, so called smart factories.
- **CPS:** The physical and digital levels are merged and can no longer be separated from each other in a sensible way. This also includes the production and product level.
- **Self organization:** The manufacturing systems developed in a decentralized way and the classical product hierarchy was abandoned.
- **Individualisation:** Acquisition and distribution become new systems and the corresponding processes are operated through a multitude of different channels. The challenge is to use product intelligence for innovative product and service development.
- **Adaptation to the human being:** The fundamental needs of human beings should be respected in the design of manufacturing systems.
- **Corporate social responsibility:** Sustainability and resource efficiency are the basic conditions for a successful industrial manufacturing system.

3.3 Use case introduction

The focus during the modeling of the use case, as presented in Figure 3.2, was to describe an architecture that represents current industrial standards. It is therefore important that the use case contains elements from each level and typical connections between each level of the automation pyramid but also structured network architecture known from industry 4.0. These two structures are displayed in Figure 3.1. Table 3.1 shows which elements were chosen for each automation layer. Because the use case shall be able to support other research projects in the fields of safety and security in the future, it was created using elements and structures common for the automation industry. This way it can easily be adapted to represent a certain industrial system [JB21].

Automation pyramid level	Devices
Field Level	Webcam Pushbutton Laser Capacity Sensor Ultrasonic Motor Optical Sensor Biometric fingerprint Scanner Barcode Temperature Sensor Radio-frequency Identification (RFID) Sensor Level Transmitter Flow Transmitter Fill Valve Pump Discharge Valve
Control Level	Robot Cobots Selective Compliance Assembly Robot Arm (SCARA) Programmable Logic Controller (PLC) Computerized Numerical Control (CNC) RFID SIS
Supervisory Level	Data Server (NAS) Supervisory Control and Data Acquisition (SCADA) Touch Display TP700 (Human Machine Interface (HMI)) Internet of Things (IoT) Devices Open Platform Communications Unified Architecture (OPC UA) PC OPC UA Server
Planning Level	PC Manufacturing Execution System (MES) Data Historian Printer
Management Level	Web Server ERP(SAP, ODOO) Material Requirements Planning (MRP) Email Server

Automation pyramid level	Devices
	Laboratory Information Management System (LIMS) App Server Product Information Management System (PIMS)

Table 3.1: Use case devices from automation pyramid

CHAPTER 4

Application of methods to the use case

4.1 Application

In this chapter, two TM methods will be applied on the use case constructed in the previous chapter. The methods used for this analysis are STRIDE-LM and FACT which were explained in detail in Chapter 2. These methods were selected intentionally in order to contribute to future scientific studies.

4.1.1 STRIDE-LM

The freeware tool MS TM Tool [1] was selected for the execution of the STRIDE-LM method. This tool works in multiple steps. First, a graphical representation of the use case is created. To help with this process, MS TM Tool provides generic elements from 6 predefined stencil categories. Each group contains a list with predefined and universal objects. Stencil categories and example are listed in Table 4.1.

A list of attributes is defined for each group of objects with predefined values for each attribute which in most cases is boolean (yes or no). With the help of these predefined objects, a use case was created using the MS TM Tool. For a better perception, some parts of the resulting graphic are displayed in Figure 4.1. The second step is to assign values to all attributes for each object. The predefined attributes are mostly relevant for security aspects. Because this thesis focuses on the interdependence of safety and security, it was necessary to define additional attributes for each object with a focus on the safety aspect. The safety attributes with their respective questions and answers are introduced in detail in the following paragraph and are summarized Table 4.2.

[1] https://aka.ms/threatmodelingtool

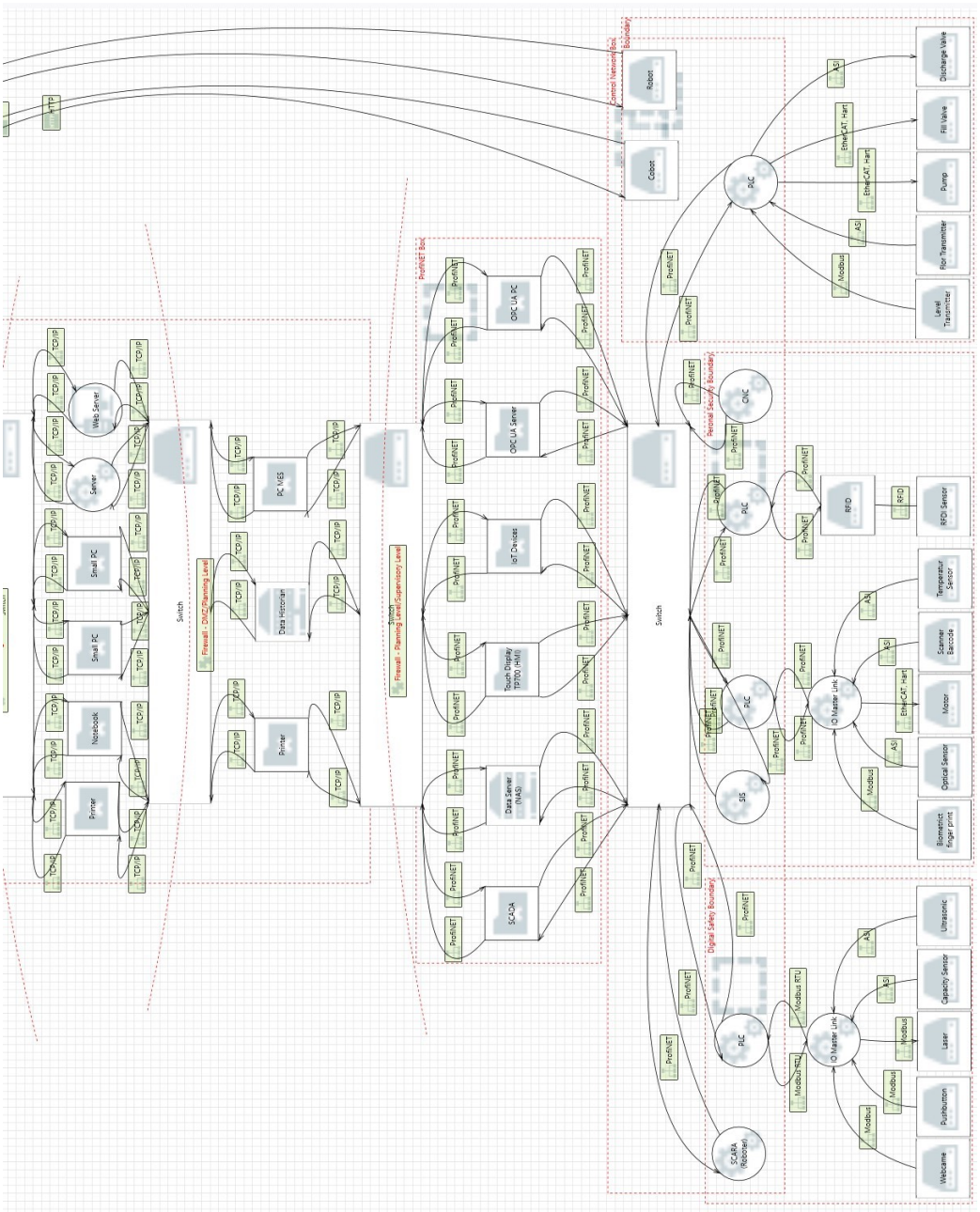

Figure 4.1: Use case in MS TM Tool

Stencil categories	Example
Generic process	Web server, LIMS, Managed Application
Generic external interactor	Human user, Small PC, Printer, IoT Devices
Generic data store	Cloud storage, Data Historian, Email Server, NAS
Generic data flow	HTTP, ProfiNET, UDP
Generic trust line boundary	Internet boundary, Machine Trust Boundary
Generic trust border boundary	Sandbox trust boundary border

Table 4.1: Stencil categories

Safety Attribute	Possible values
Mechanical adjustment required/possible	yes / no
Safety critical for employees	yes / no
Cascading impact	yes / no
Spare parts are available	directly / hours / days / month / none
Redundancy available	yes / no
Support / maintenance	internal / external
Support / maintenance availability	directly / hours / days / month / none
Fail-safe operation	yes / no / partially
Powered by	electricity / battery / human force
Local human service required	yes / no
Safety critical according to SIL-Level	0 / 1 / 2 / 3 / 4
Authorization required for physical access	yes / no
Dedicated responsibilities	yes / no
Hardware interfaces	USB, USB-C, PCI-Bus, fireWire, NIC, Bluetooth, AGP, SCSI

Table 4.2: Safety attribute

- Mechanical adjustment required/possible: Is it necessary for an employee to adjust a machine or program locally before executing its orders?
- Safety critical for employees: In case of unexpected behavior, is it possible that a machine or program can injure the employees?
- Cascading impact: Can a malfunction of this OT component lead to impacts on other OT components?
- Spare parts are available: How long does it take until the malfunctioning part is replaced? Low availability leads to higher criticality if part is damaged.

- Redundancy available: Are machines or programs outfitted with redundant components?
- Support / maintenance: Who is responsible for support and maintenance? Maintenance staff has access to critical areas and this could lead to sabotage of the system. The security risk for the system is increased when using an external company.
- Support / maintenance availability: How long does it take to contact support or maintenance? Is the component only maintainable by special suppliers or are suitable employees available inside the company. Low maintenance availability for a part leads to higher criticality if this part is damaged.
- Fail-safe operation: Is the component able to recognize a critical situation and does it have the capabilities to terminate the process in this case?
- Powered by: Is the component able to fulfill its tasks in case of a power outage? Is the component independent from electrical energy supply? Does the system remain in a save state if power is cut off?
- Local human service required: Can the component operate independently from local human operation? Is it necessary for an employee to supervise the process and operate the component at all times?
- Safety critical according to SIL-Level: Safety Integrity Level (SIL) based on the IEC 61508 standard, SIL 4 is the most critical and SIL 1 the least.
- Authorization required for physical access: Is authorization required to access the component physically (e.g., keys, smartcards)?
- Dedicated responsibilities: Are dedicated employees available that can take responsibility for the component?
- Hardware Interfaces: Which hardware interfaces are available? Are interfaces available that offer potential attack vectors?

In a third step, the method STRIDE-LM is applied to the use case. The tool uses predefined threats and corresponding conditions for certain predefined stencil attributes and this enables it to conduct the step automatically. In case the included attributes are not sufficient, customized stencils, attributes, conditions and threats can be added and connected respectively, otherwise the program will ignore the customized attributes when generating the PDF with the results. The corresponding conditions must therefore be created for each extra safety attribute and assigned to a certain predefined or added threat. For the use case created for this thesis, only attributes were added.

4.1.2 Mitre ATT&CK

MITRE ATT&CK (MA) is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations [ATT22]. This technique was used in this thesis on the results of the STRIDE-LM method to generate and model the potential attack areas. MITRE ATT&CK Navigator (MA Navigator) [2] provides multiple environments, Enterprise, Mobile and ICS. For this use case, the ICS environment was used. MA provides 12 tactic sets and each tactic set contains varying numbers of techniques which are designated with a name and an identification number. The model, which was created for this thesis, is based on the threats from the STRIDE-LM and is compared with other known attacks like Triton, Stuxnet, Ryuk, Locker Goga or Attack Group Lazarus from Table 1.3 and represents areas for general attack possibilities from the use case.

4.1.3 Failure-Attack-CounTermeasure (FACT) graph

For this thesis, the free web based tool draw.io [3] was used to apply the FACT method to the previously created use case. This tool is able to create various graphics, which is very helpful for the FACT graph. As introduced in Chapter 2, this TM method is a combination of two different methods: FTA and Attack tree. The use case from Chapter 3 was used as a basis to create the FACT graph. Because no fully automatic tool for the creation of the FACT graph is available at the moment, only certain sections of the use case were used to create the FACT graph. The relevant components are shown in Table 4.3 and the individual building components for the FACT graph are shown in Figure 4.3

As written in Chapter 2, at first a FTA was created to implement the FACT method. For the root cause of the problem at the top of the graph, a case was created which has the largest damage potential for an organization. For the example use case, a stop in the production line was chosen as the root cause. With the use of an FTA, the probable causes and failures that could lead to this problem were investigated. The FTA graph in Figure 4.2 shows a part of the FTA with the following groups:

[2] https://mitre-attack.github.io/attack-navigator/
[3] https://app.diagrams.net/

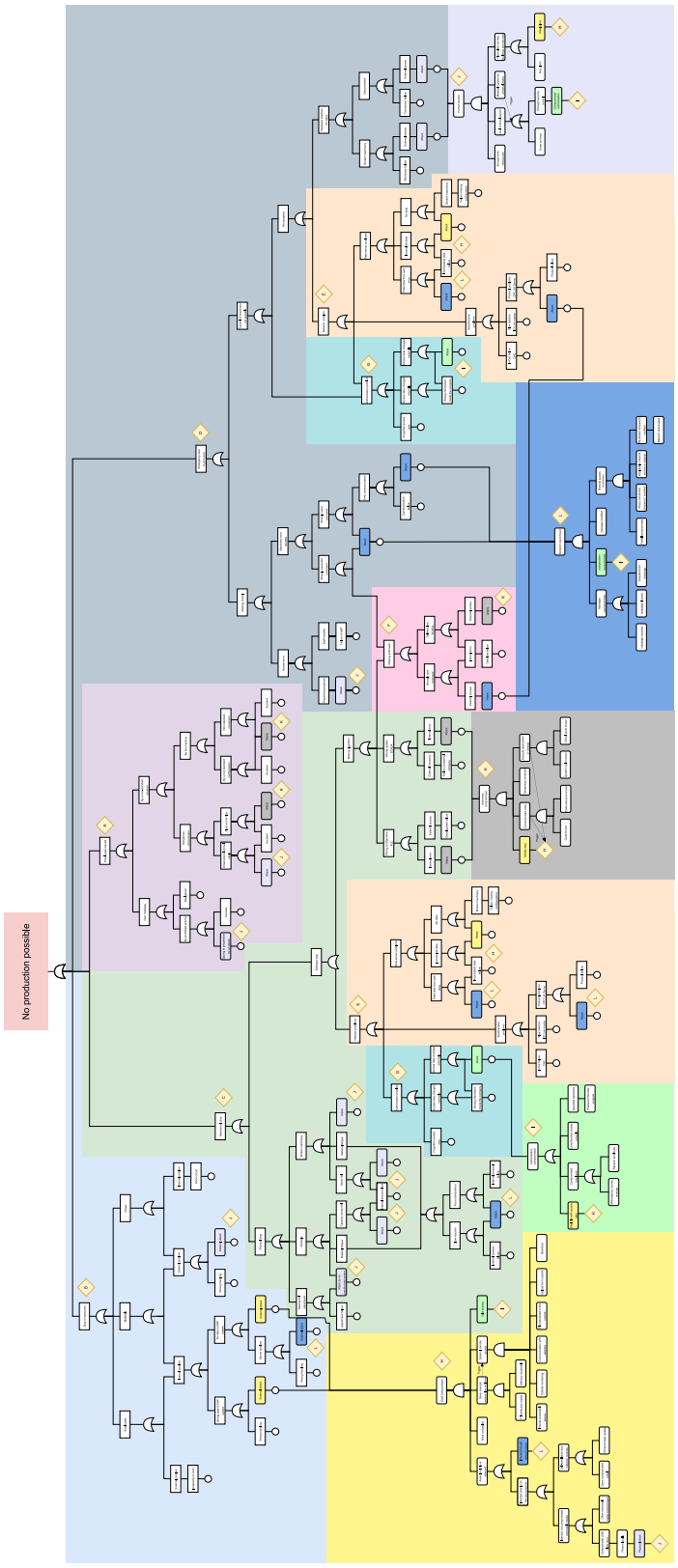

Figure 4.2: FACT graph all groups

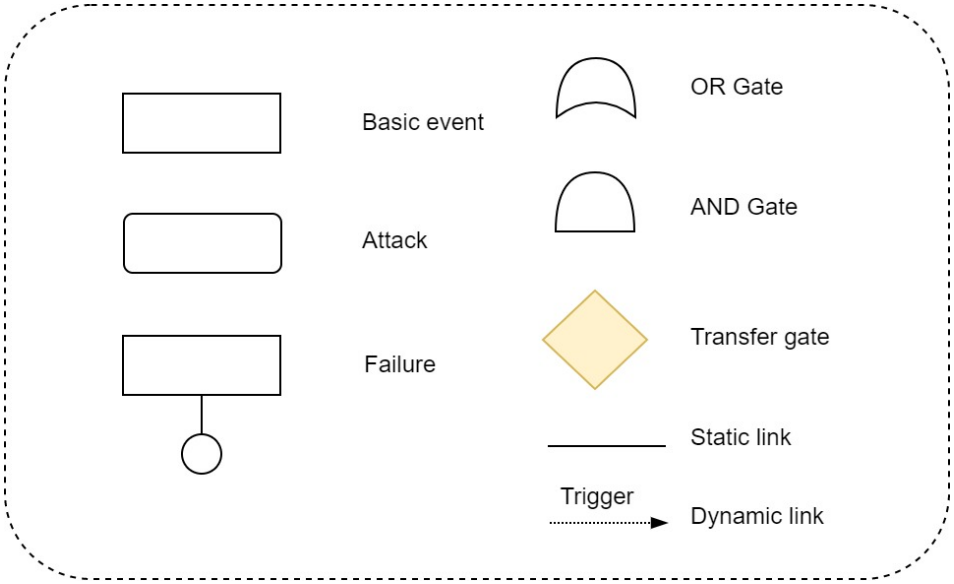

Figure 4.3: FACT explanation

Automation level	Use case components
Field level	Digital security, Push-button, Capacity sensor, Personal security, Motor, Temperature sensor, IO Link master
Control level	SCARA, PLC, SIS, CNC
Supervisory level	SCADA, OPC UA Server, Data Server
Planning level	PC MES
DMZ	Server
Management level	ERP
Internet	Web shop

Table 4.3: Components of use case to FACT graph

- A - Employees Injured (light purple)
- B - Stop of Resources (light blue)
- C - Mechanical Stop (light green)
- D - Emergency Stop - Push button (light blue-grey)
- E - Network Problem (light orange)
- F - Missing Arithmetic (light pink)
- G - Authorization Problem (light turquoise)

These groups were created to give a better overview and ease the modeling process. Each group represents a certain problematic area from the use case which could endanger the production flow. The error description is generalized and does not describe each instance individually. The individual groups are shown in Figures 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12. In the next step, the countermeasures should be defined. Because it does not fit the thematic scope of this thesis and requires extensive work, this part was left out. The third part is the creation of the attack tree and its connection to the FTA. The same grouping strategy was used as in the previous text. Figure 4.2 shows the FACT graph and the following groups which are parts of it.

- H - Data Compromised (yellow)
- I - Authorization Compromised (clear light green)
- J - Human Access (lavender)
- K - Arithmetic Compromised (light brown-grey)
- L - Network Compromised (blue)

The individual groups are shown in Figures 4.13, 4.14, 4.15, 4.16 and 4.17. In this case, the groups represent common attack possibilities which can be used on the use case. MA was taken as an inspiration for these attack possibilities. Figure 4.18 presents an attack tree of the ICS Attack TRITON which is also mentioned in in Chapter 2. The particular attack possibilities were taken from the MA - Triton S1009 and added to the attack tree. A final fourth step would typically define the security countermeasures but similar to step two this was left out in this thesis because it does not fit the thematic scope and requires extensive work.

4.2 Results

In this paragraph, the results of the individual methods are presented.

4.2.1 STRIDE-LM

The resulting PDF has 131 pages and 879 threats. Due to this large amount of data only a summary of the most important threats is shown in Table 4.4.

Threat ID	Name	Category	Description
10	Material damage	Safety	PLC and their configurations could be responsible for material damage.

Threat ID	Name	Category	Description
74	Injuries/death of an employee	Safety	Motor and their configurations could be responsible for an injury or even death of an employee.
79	Spoofing of Source Data Store Cloud Storage	Spoofing	Cloud storage may be spoofed by an attacker and this may lead to incorrect data delivered to the cobot.
82	Injuries/death of an employee	Safety	Cobot and their configurations could be responsible for an injury or even death of an employee.
97	Data Flow HTTP Is Potentially Interrupted	Denial of Service	An external agent interrupts data flowing across a trust boundary in either direction.
162	Spoofing of Destination Data Store Email Server	Spoofing	Email server may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Email Server.
306	Elevation Using Impersonation	Elevation Of Privilege	PLC may be able to impersonate the context of IO Master Link in order to gain additional privilege.
369	External Entity OPC UA PC Potentially Denies Receiving Data	Repudiation	OPC UA PC claims that it did not receive data from a process on the other side of the trust boundary.
386	Spoofing of Source Data Store Switch	Spoofing	Switch may be spoofed by an attacker and this may lead to incorrect data delivered to OPC UA server.
453	No production	Safety	SCADA and its configurations could be responsible for a total loss of production.
490	Potential Process Crash or Stop for SCARA	Denial of Service	SCARA (Robot) crashes, halts, stops or runs slowly; in all cases violating availability.
508	Collision Attacks	Tampering	Attackers who can send a series of packets or messages may be able to overlap data. For example, packet 1 may be 100 bytes starting at offset 0. Packet 2 may be 100 bytes starting at offset 25. Packet 2 will overwrite 75 bytes of packet 1.

Threat ID	Name	Category	Description
509	Weak Authentication Scheme	Information Disclosure	Custom authentication schemes are susceptible to common weaknesses such as weak credential change management, credential equivalence, easily guessable credentials, null credentials, downgrade authentication or a weak credential change management system.
673	Weak Access Control for a Resource	Information Disclosure	Improper data protection of RFID can allow an attacker to read information not intended for disclosure. Review authorization settings.
674	Potential Data Repudiation by PLC	Repudiation	PLC claims that it did not receive data from a source outside the trust boundary.
687	Data Flow Sniffing	Information Disclosure	Data flowing across ProfiNET may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations.
702	Elevation by Changing the Execution Flow in SIS	Elevation Of Privilege	An attacker may pass data into SIS in order to change the flow of program execution within SIS to the attacker's choosing.
729	No production	Safety	Notebook and its configurations could be responsible for a total loss of production.

Table 4.4: Components of use case

4.2.2 MITRE ATT&CK

Table 4.5 shows the mapping of the results from the STRIDE-LM method with the techniques of the MA Navigator. Figure 4.4 shows the results of the attack model from the MA Navigator. The same figure also includes a comparison with other known attacks like Triton, Stuxnet, Ryuk, Locker Goga or Attack Group Lazarus from Table 1.3. MA Navigator provides multiple layers to compare attacks. To make distinguishing the layers easier, different goal values were assigned to the various layers. The potential threats from the mapping table were grouped as a general model and rated with a goal value of 1, all other known attack techniques, like Triton, Stuxnet, Ryuk, Locker Goga or Attack Group Lazarus from Table 1.3), were each rated with a goal value of 2. A color scheme displays the results on a scale from 1 to 5 as shown in the upper right part of the picture. This scale displays the sum of goals from individual attacks. The results from the general model with no overlap with known attacks have a value of 1 and are shown in red. This shows that these attack possibilities are not included in known attacks. All other uneven values, like 3 (yellow) or 5 (blue), show overlaps of the general model with at least one known attack. This shows that the analyzed system is vulnerable to known attacks in these areas. The even values show areas where only known attacks overlap and those that were not shown by the mapping from STRIDE-LM. This could mean that the system is safe against those techniques or that STRIDE-LM does not provide enough information.

Tactic	Techniques Name	Techniques ID	Threat ID
Initial Access	External Remote Services	T0822	82
	Internet Accessible Device	T0883	872
	Transient Cyber Asset	T0864	509
	Wireless Compromise	T0860	79, 162
	Replication Through Removable Media	T0847	872
Execution	Change Operating Mode	T0858	10, 79, 82, 162
Persistence	Valid Accounts	T0859	369, 674
	System Firmware	T0857	729
Evasion	Change Operating Mode	T0858	10, 79, 82, 162
	Masquerading	T0849	673, 702
	Spoof Reporting Message	T0856	79, 162, 186
Discovery	Network Sniffing	T0842	687
	Remote System Discovery	T0846	
Lateral Movement	Program Download	T0843	79, 162
	Remote Services	T0886	79, 162
	Valid Accounts	T0859	369, 674
Collection	Data from Information Repositories	T0811	369, 674
	Detect Operating Mode	T0868	79, 162, 306
	I/O Image	T0877	306

Tactic	Techniques Name	Techniques ID	Threat ID
	Program Upload	T0845	79, 162, 306
Command and Control	Commonly Used Port	T0885	463
	Standard Application Layer Protocol	T0869	97, 453
Inhibit Response Function	Alarm Suppression	T0878	508
	Block Command Message	T0803	386
	Data Destruction	T0809	10, 74
	Manipulate I/O Image	T0835	306
	Service Stop	T0881	10, 74, 82
	System Firmware	T0857	79,162
Impair Process Control	Spoof Reporting Message	T0856	79, 162, 186
	Unauthorized Command Message	T0855	79,162
Impact	Denial of Control	T0813	10, 74, 82, 490
	Denial of View	T0815	10, 74, 82
	Loss of Control	T0827	10, 74, 82, 702
	Loss of Safety	T0880	74, 82, 490, 702
	Loss of View	T0829	10, 74, 82
	Manipulation of Control	T0831	10, 74, 82

Table 4.5: Mapping of the results from the STRIDE-LM method with the techniques of the MA Navigator

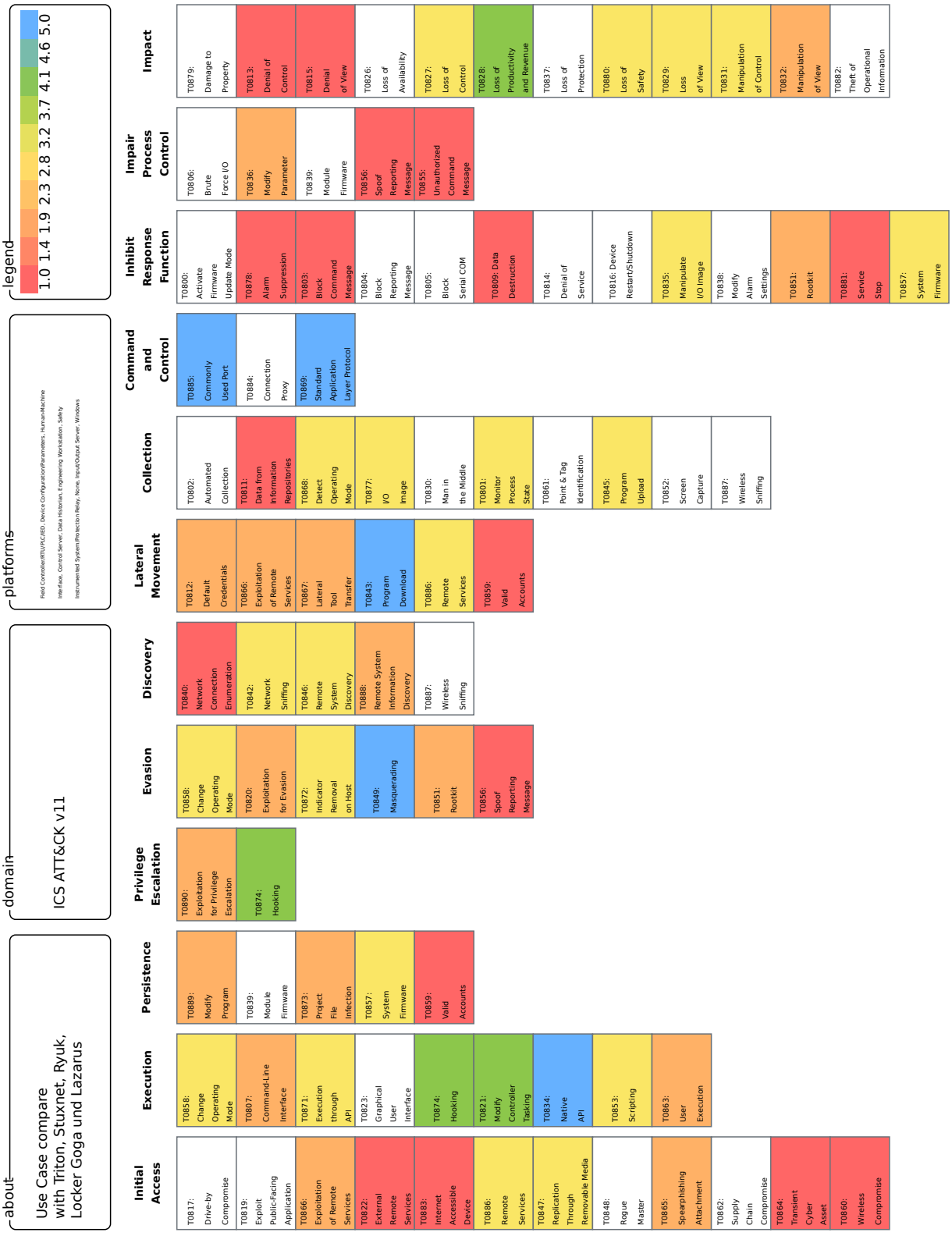

Figure 4.4: MA Navigator results

4.2.3 FACT graph

The completed FACT graph is shown in Figure 4.2 and the root in Figure 4.5. At the top sits the root cause which in this case is the problem No Production Possible. The upper portion of the graph represents the FTA with its seven groups A through G while the lower part represents the attack tree with five groups H through L. FTA and the attack tree are connected through various scenarios. Exemplary for all groups, two groups from the fault tree and one group from the attack tree will be explained in detail in the following text. All other groups in the FACT graph were created in a similar manner.

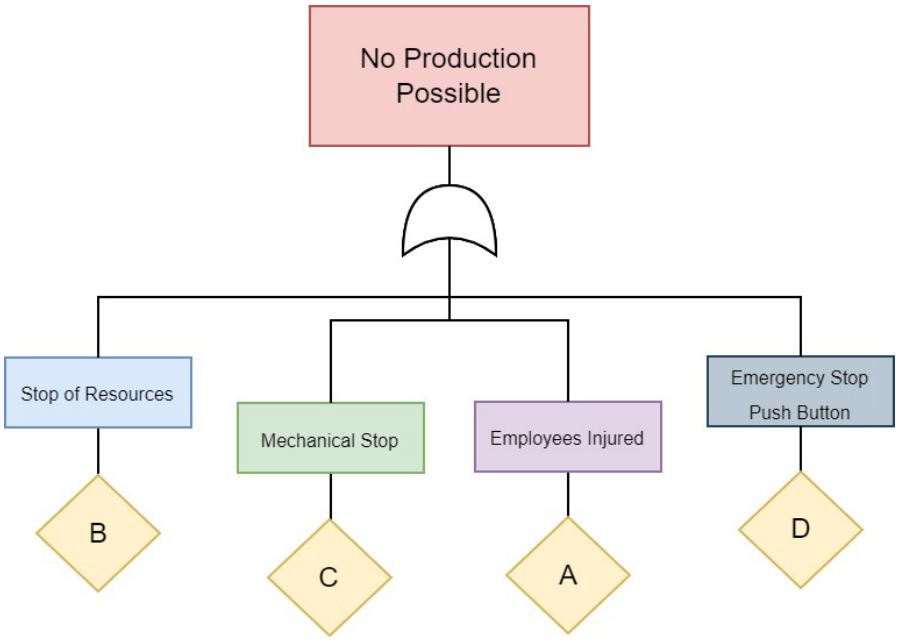

Figure 4.5: FTA root

- **Group A - Employees injured:** This group investigates the potential harm to humans. The part of the FACT graph associated with this group is shown in Figure 4.6. On the top of the graph sits the event Employees Injured which can be triggered by either of the two events Own Mistakes or By a Non-standard Situation. In the branch of the event Own Mistake the next lower level is formed by the events Negligence and By Prohibited Action which are also connected with an "or" gate. The event Negligence ends its sub-branch while the event By Prohibited Action can be triggered by the events Curiosity or Spying in Preparation for an Attack. This last event is the top level event of the attack graph associated with group J and therefore here a connection is formed from a fault tree to an attack tree. The second branch of Employees Injured is the Non-standard Situation which can be triggered by a Dangerous Environment or By the Machine. The sub-branch Dangerous Environment is divided into Unsecured Electrical Cable or Fluid on the Floor. Both

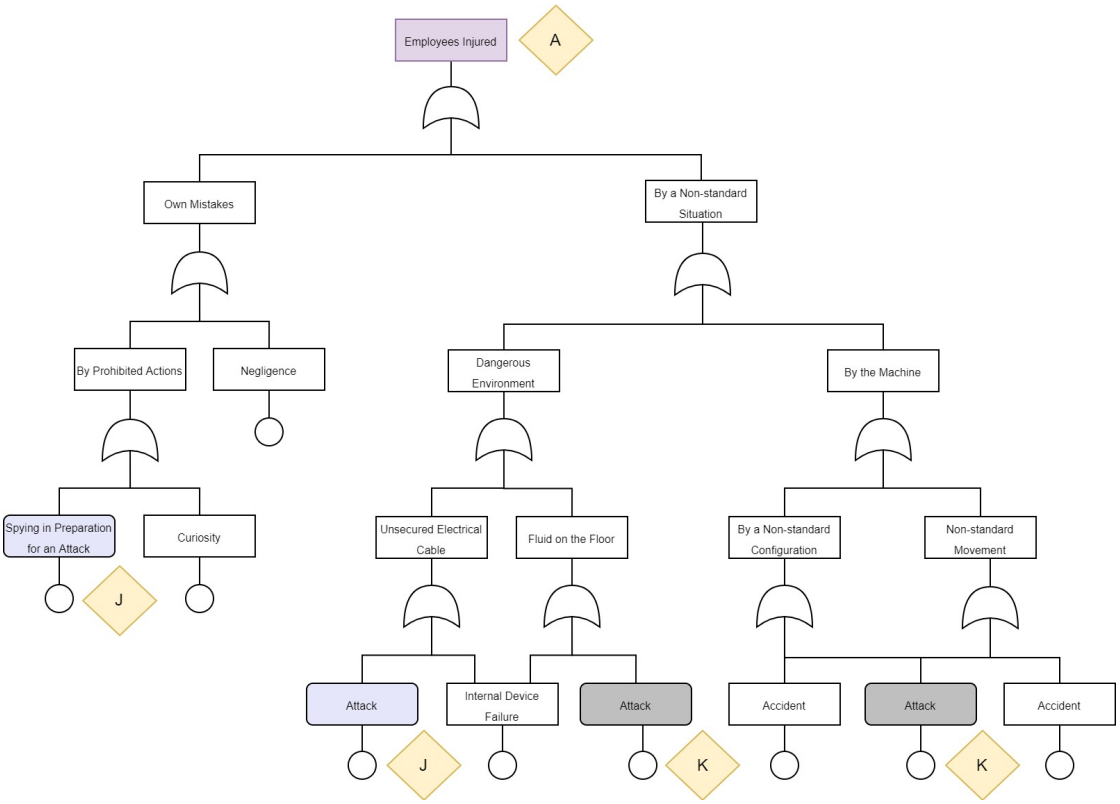

Figure 4.6: FTA employees injured

cases can end with the end failure event Internal Device Failure or are connected with the group J - Human Access or group K - Arithmetic Compromised from the attack tree. For example a compromised pump could increase pressure to level that causes pipe leakage or even rupture. The second sub-branch of Non-standard Situation, By the Machine, is divided into Non-standard Movement and By a Non-standard Configuration. In both cases, these events are connected with the group K - Arithmetic Compromised from the attack tree or with the end failure event Accident.

- **Group B - Stop of Resources:** This group focuses on the problem of missing resources. The part of the FACT graph associated with this group is shown in Figure 4.7. The resources were divided into Employees, Material and Power and further distinguished into Internal or External Problem for all three cases. An External Problem with Employees can cause a failure through the Incapacity to Work. An Internal Problem has the same substructure for Employees and Materials and is distinguished between Wrong Data in ERP System and No Data in ERP System. In the first case, the failure can be caused by Wrong Handling of the Program or through an External Attack from group H, aimed at the program. No Data in the ERP System can be caused by a Server Problem or Compromised Data because of an External Attack from group H. A Server Problem can be caused by Wrong Handling

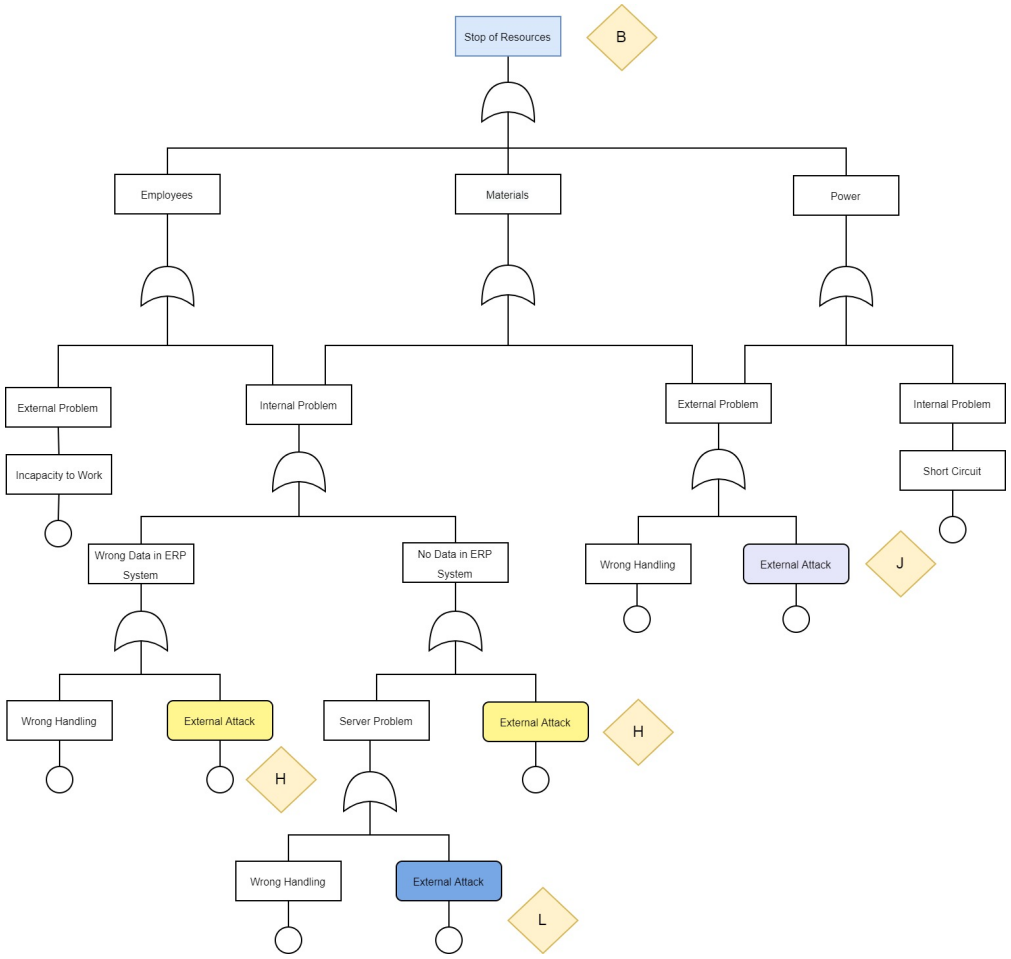

Figure 4.7: FTA stop of resources

or a Compromised Network because of an External Attack from group L. External Problems are similar for Materials and Power. In this case, resources can become unavailable through Wrong Handling in other companies or an External Attack from group J. An Internal Problem that can cause a loss of power could be a Short Circuit.

- **Group J - Human Access:** This group presents the conditions which need to be fulfilled in order for the attack to be carried out successfully in this case, particularly by Physical Human access from the use case. This part of the attack tree is shown in Figure 4.15. The root event Human Access can be triggered when the conditions for the events Enough Time Resources, Local Criminal Person, Material Support for Physical Attack and Floor Plans of the Site, Buildings and Devices are fulfilled. These events are connected with an "and" gate. The sub-branch Local Criminal Person is divided into Entry by Force or Through Security Control and is triggered by the event Material Support for Physical Attack. The event Through

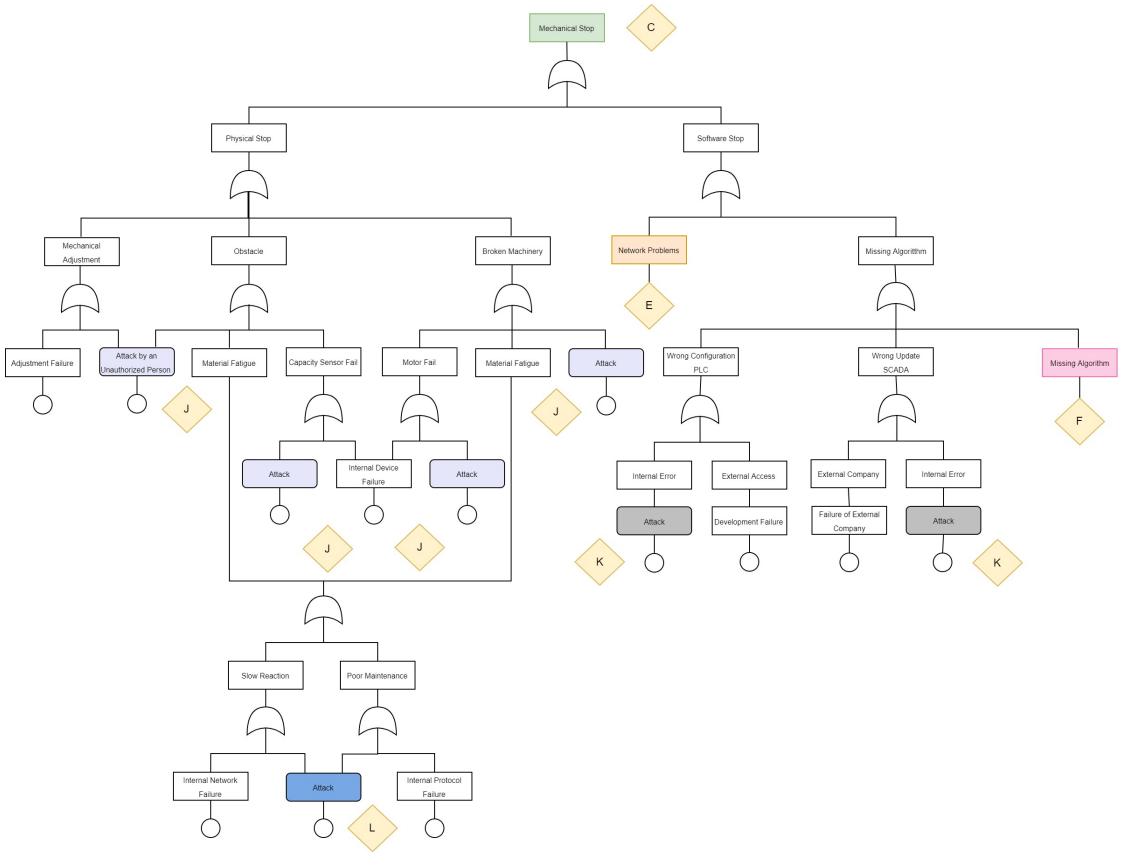

Figure 4.8: FTA mechanical stop

Security Control is connected to the next group from the attack tree, group I - Compromised Authorisation and by this the conditions for the attack are extended. The next sub-branch Floor Plans for Site, Buildings or Devices is divided into Buy Plans or Steal Plans. This last event is connected with group H - Data Compromised from the attack tree.

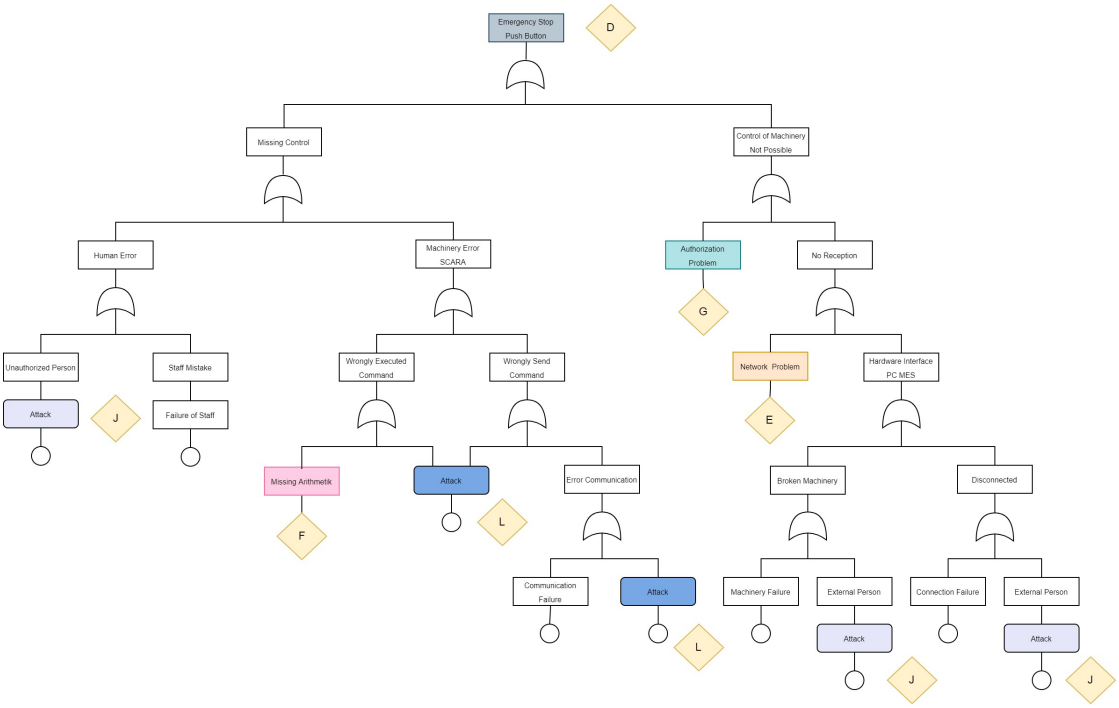

Figure 4.9: FTA emergency stop push-button

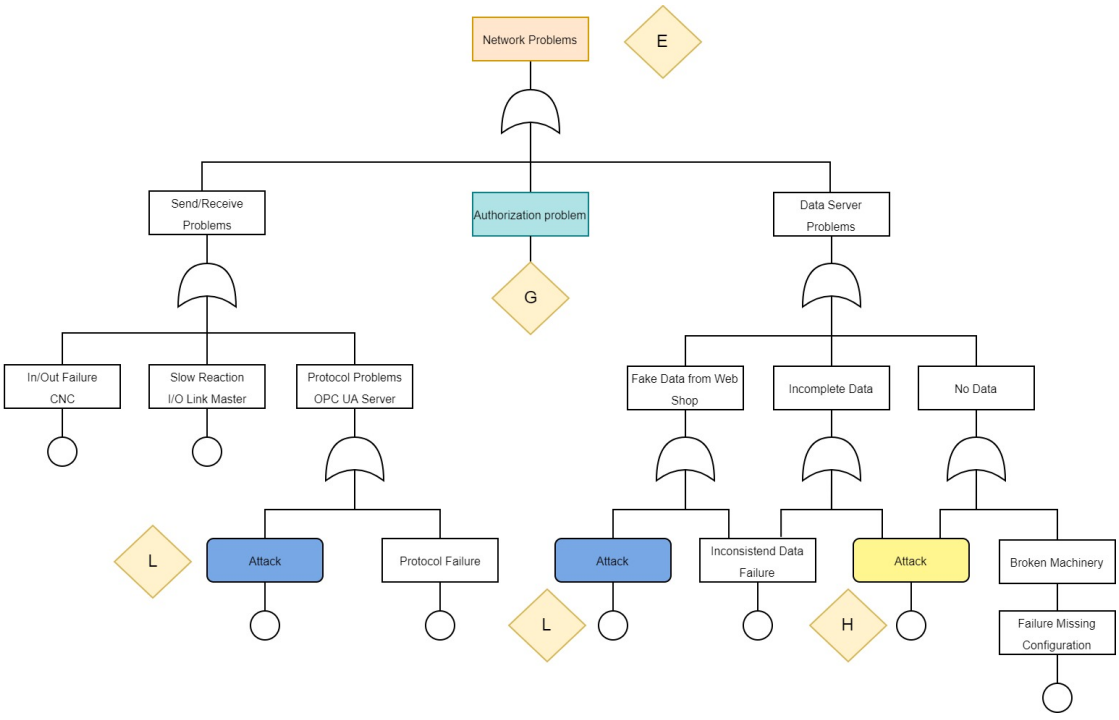

Figure 4.10: FTA network problems

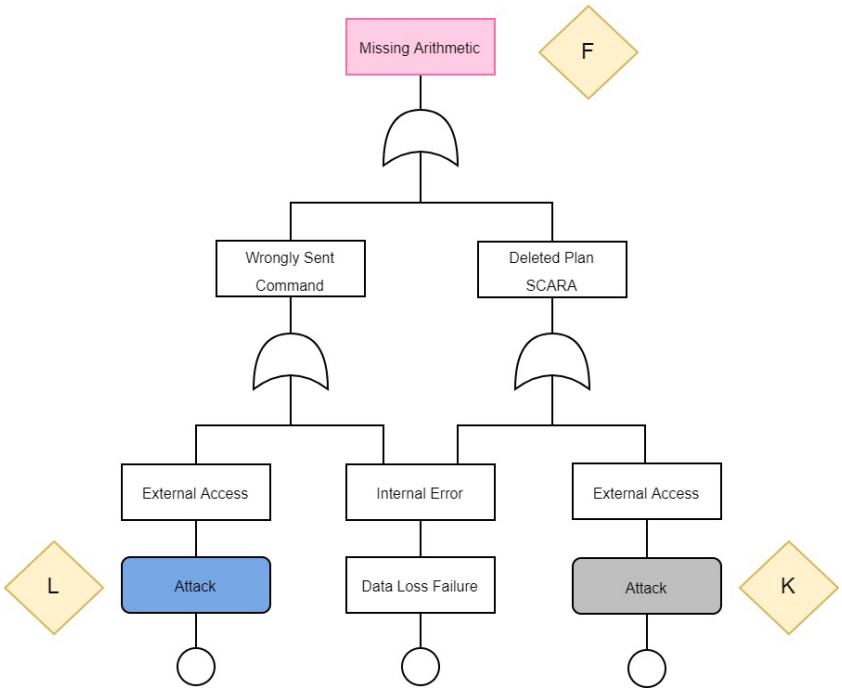

Figure 4.11: FTA missing arithmetic

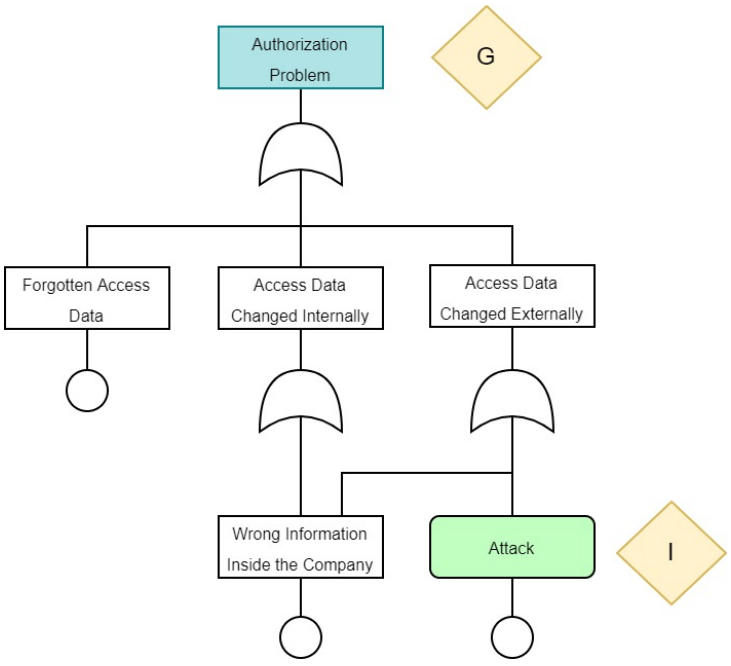

Figure 4.12: FTA authorization problem

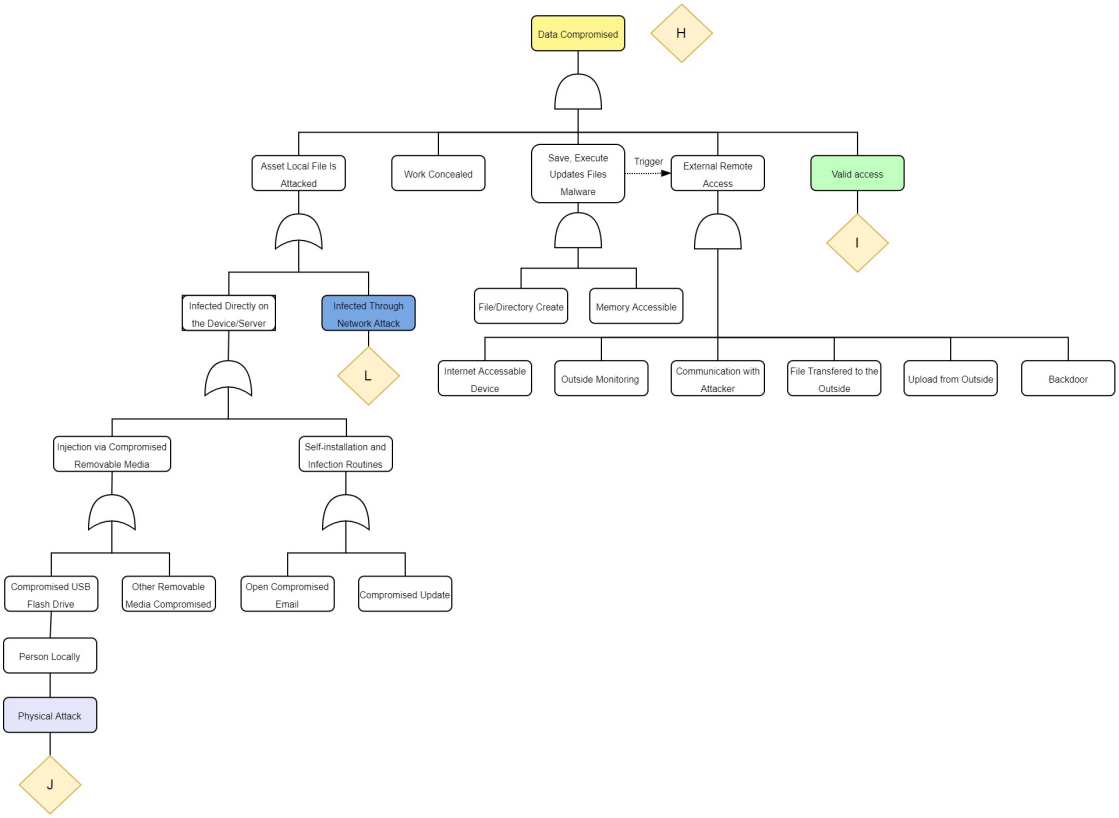

Figure 4.13: Attack tree data compromised

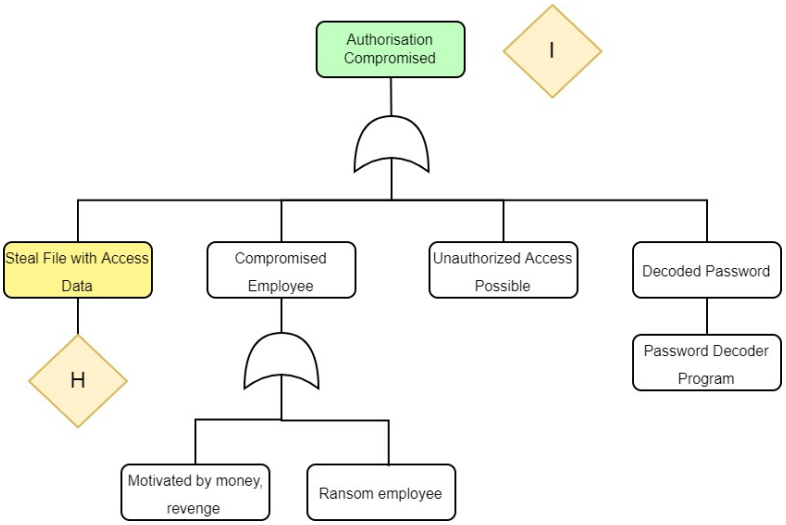

Figure 4.14: Attack tree authorisations compromised

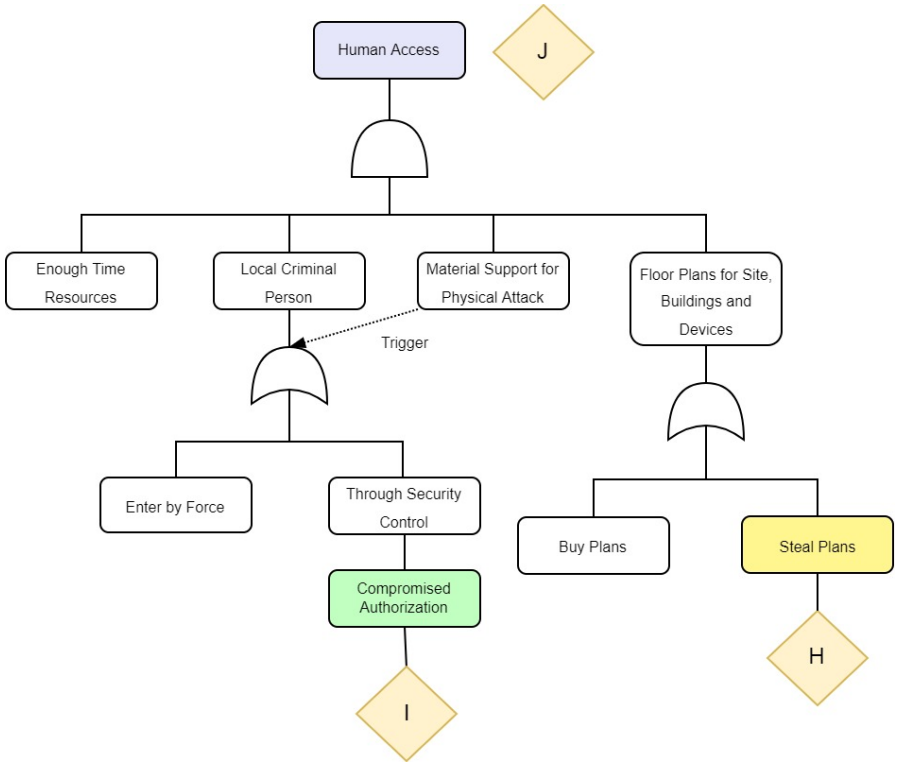

Figure 4.15: Attack tree human access

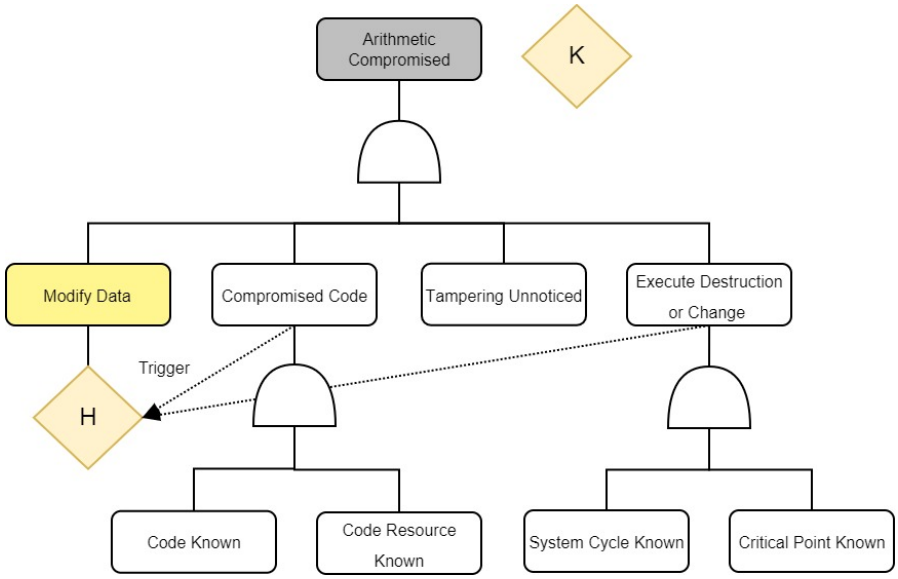

Figure 4.16: Attack tree arithmetic compromised

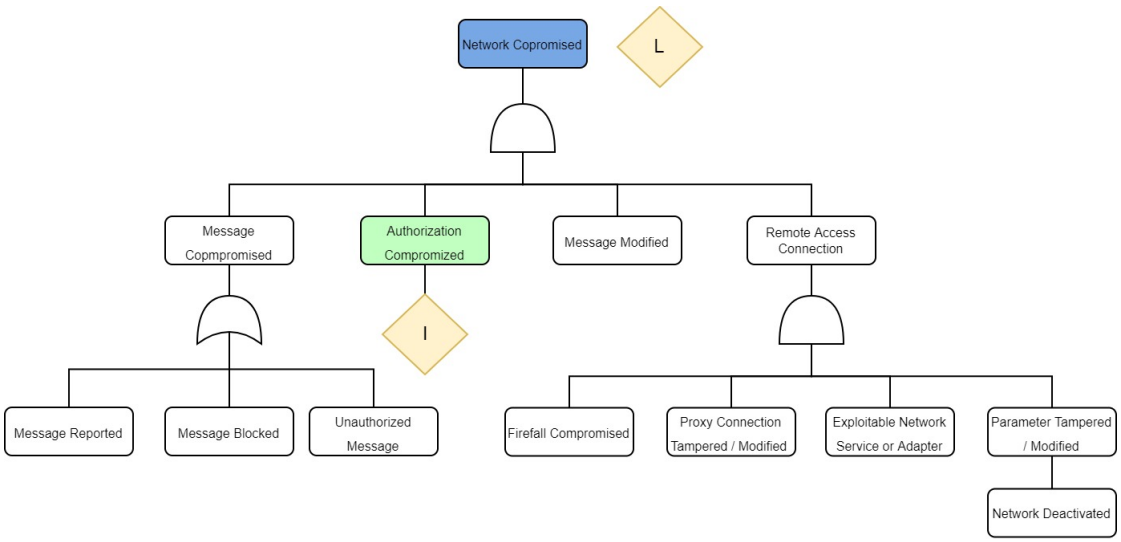

Figure 4.17: Attack tree network compromised

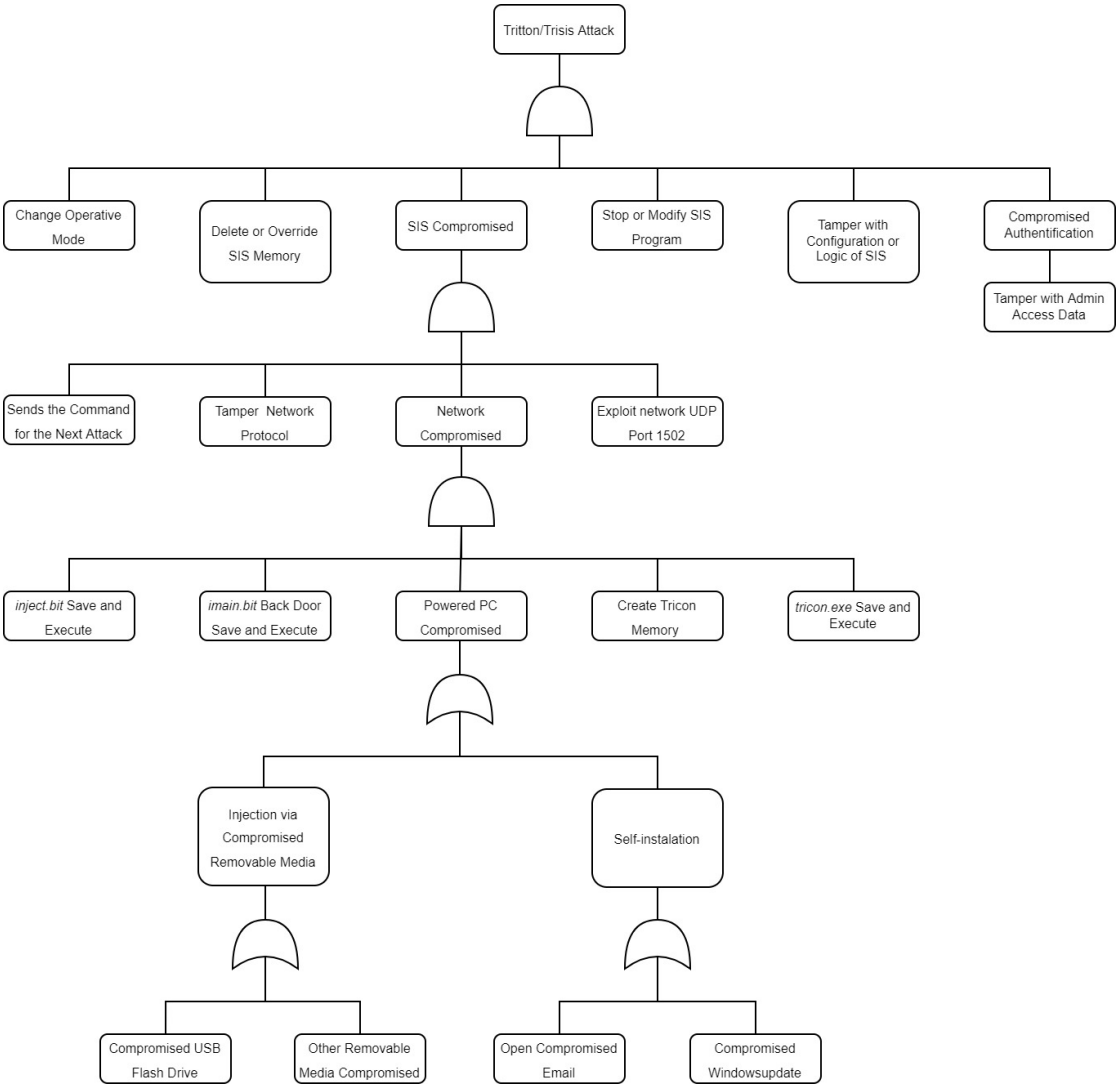

Figure 4.18: Attack tree Triton based on MA Framework

CHAPTER 5

Method comparison and assessment

In this chapter, the methods used in Chapter 4 are assessed and subsequently compared. The assessment is split into three major groups. The first group assesses the efficiency of each method. The second group assesses the results from each method and the last group assesses how each method can handle the interdependence of safety and security.

5.1 Efficiency

The comparison results regarding efficiency of the modeling process are dependent on the selection of certain tools used for the individual threat modeling methods.

5.1.1 STRIDE-LM

The MS TM Tool was selected to be used with the STRIDE-LM method. At a first glance it is a very user friendly tool but before using it the corresponding documentation should be read and assessed if the predetermined templates are sufficient for the use case because changes during the modeling process are very time consuming. If the predefined templates fit the use case well then modeling is fast and easy. Creating individual templates outside the provided catalog increases the necessary amount of work significantly because all associated events, attributes, threats and situations have to be defined. This was the case for the use case in this thesis because the provided templates are built for software systems and deal only with the security protection goal. The area available to model the use case in the tool is limited which narrows the field of view for large examples and makes it hard to have the full overview. Larger examples with a lot of data also significantly degrade the reaction time of the tool proportional to the increase in data. Each object class, called stencil by the tool, has its own window with properties like

name, a picture and attributes. This allows the user to clearly structure and summarize the available information on the object class. The more properties are defined by the user the better and more complete is the result data. A report is automatically generated in the form of an HTML file in which the individual threats of the use case are clearly structured and labeled including the associated pictures.

Originally OWASP Threat Dragon [1] was selected as a tool for this thesis, but during the first modeling attempts its possibilities seemed limited and it was discarded in favor of another tool. This does not mean that OWASP Threat Dragon might not be the tool of choice for certain applications or even would have been for the example use case in this thesis, since no full review was done in this thesis. It was simply discarded based on the initial user experience.

Modeling the use case created for this thesis with the MS TM Tool using the STRIDE-LM method was very time consuming, because the part considering the safety aspect was not included in the provided templates and had to be adapted individually.

5.1.2 MITRE ATT&CK

The Web-based tool MA Navigator was selected for the attack modeling. This tool is intuitive to use during the first steps. For the subsequent modeling, a lot of helpful information and tips are available on the official website of MA like workshop videos, papers and links to blogs on the subject. Short texts with information on each tactic, technique, threat group, software or mitigation are provided to make it easy for the user to understand. MA is a dynamic database which is constantly updated with new information by its community, providing up to date data to the user at all times. The results from the selected techniques are easy to compare with the techniques used in known attacks. The attacks are divided into different groups like threat group, software or mitigation which gives the user multiple perspectives on the attack problematic. Changes or adjustments can be implemented with little effort. Results can be viewed online or downloaded in different formats like json, xlsx or svg and stored locally for later review. Applying MA Navigator to the use case created for this thesis was easy and qualitatively good results were achieved.

5.1.3 FACT graph

The FACT graph modeling was done with the Web-based diagram program draw.io by diagram.net. This application can be used via a desktop client or directly through the Internet browser and is therefore easily available for all users. Using the program is intuitive which makes it easily accessible for inexperienced users. The user should be proficient in FACT graph modeling because draw.io provides only simple objects like events, fails and gates and no general structural templates. The exact event- and connection-structure have to be defined by the user. Good preparation saves a lot of time

[1] https://github.com/OWASP/threat-dragon/releases

during the modeling process which for large use cases can also lead to a lot of preparation work especially if one wants to keep a good overview. On the other hand, the FACT graph is easily split into thematically sorted groups which can be displayed separately for a better overview. In draw.io, the size of the graphical working environment is scalable, which also contributes to keep a good overview. When defining the FACT graph, it is important to determine how detailed the use case should be analyzed, otherwise there is a risk to create an endless graph with reoccurring cases. Changes and adjustments are easily done with draw.io. During the modeling process, the user often becomes aware of new threats, which were not thought of during the first sketches.

FACT graph is not an ideal solution for large use cases. The individual approaches of different users can lead to inconsistent data in large projects with multiple users. The method is efficient for small cases and it helps to discover potential failure- and attack-sources.

5.1.4 Comparison

At a first glance, the MS TM Tool appears to be user friendly and time saving. In the end, it was shown that the diagram program draw.io is more comfortable to use. With MA Navigator there were no surprising complications. Since all three tools are freeware, there is no additional cost for the modeling process. MS TM Tool and MA Navigator provide the user with templates, which saves time and can even lead to finding previously unknown structures. On the other hand, the adaptability of the templates to a deviating use case is often limited which also limits the flexibility of these tools. This is not the case for the FACT graph.

In general, adapting the model to the use case was easier with FACT graph. This is mostly due to the method structure, FACT graph is already intended to cover both safety and security threats. STRIDE-LM on the other hand is only intended for security threats and therefore provides no templates fitting for ICSs. This required some adaptation work to include the safety part in the model which with the tools possibilities was time consuming.

A reason for this lies in the method structure because the STRIDE-LM method was intended for the search for security vulnerabilities and therefore provides no safety-aware templates that fit an ICS. For this thesis, using FACT graph turned out to be the most efficient because the developed use case was meant to cover a broad spectrum which was not fully covered by the templates of the other tools. The implementation of the necessary adaptations was most efficient when using the FACT graph.

5.2 Method result quality

5.2.1 STRIDE-LM

The results are automatically exported into a file as already written above. In this file, the maximum number of results is the number of threats multiplied with the number of elements from the use case. A threat is written into the result file when the conditions for the particular threat are met. When the threat and its conditions are defined too generally, the same threat problem is repeatedly written into the result file. This means that the same text will be written as a description for multiple threats just with a different event name, which in turn could lead to an unprecise and limited modeling in the further development of the threats. The data is structured according to a predefined template and can not be presented in a different manner. It is therefore important to already take care during the template creation that the threats are well defined. This makes the method somewhat user unfriendly when using the MS TM Tool because adjustments during the model creation take a lot of time.

For the example use case in this thesis, a large number of potential threats was found, which makes the further assessment, in this case attack modeling, very time consuming. This causes the user to see only individual problems listed and not the connections between the threats.

5.2.2 MITRE ATT&CK

The results from the potential attack possibilities are clearly marked in color in the table provided by MA Navigator. When using the online version, the user can access additional information on the individual techniques, procedure examples, mitigation, detection and references which is helpful for the further assessment. For this thesis, the results were compared to known attacks from the ICS sector and it was shown in Figure 4.4, which known attacks would have the potential to be successful when aimed at the example use case.

5.2.3 FACT graph

The results from the FACT graph analysis are displayed in a Figure 4.2. Because the example use case from this thesis is large, it was not possible to display all elements in a single graph. For this thesis, the example use case was therefore only used as inspiration for the modeling process. Despite this limitation the graph is large and unusable as a detailed overview. It is only possible to get an understanding of the connections between the individual groups. Detailed Figures of individual groups are necessary to detect and follow the failure- and attack-paths. This can lead to a bad understanding of the complete system. The FACT graph method is therefore most efficient for small use cases.

5.2.4 Comparison

Each user prefers a different presentation of the results, which could play an important role when selecting a method. The graphical presentation from the FACT graph method can give a good overview of general threats, MS TM Tool and MA Navigator on the other hand present more information in a text, which increases precision and decreases the danger of misunderstandings. With FACT graph and MA Navigator, the user has the possibility to adjust the model and its results after the modeling with little effort. With MS TM Tool, the same procedure is time consuming and thus the model does not encourage the user to do so.

The big difference between both methods is that FACT graph searches for potential threats based on the indicated situation while STRIDE-LM evaluates the threats based on the properties of certain object classes. In the example used for this thesis, the STRIDE-LM method led to a detailed evaluation of the individual object classes while FACT graph was focused on the entire system. A reason for this is also the difference in data analyzed with both methods. STRIDE-LM was used on a complete representation of the use case while FACT graph was used on a reduced representation with only basic elements as shown in Table 4.3. This was necessary to limit the scope of this thesis.

Compared to the FACT graph method, results from the MA Navigator are more detailed due to the database support of MA. A user is also able to find other attacks which use similar tactics. Such a comparison with known attack profiles is not possible with the FACT graph method.

5.3 Interdependence of safety and security

5.3.1 STRIDE-LM

The STRIDE-LM method is only intended for security investigations, therefore the safety part had to be artificially created. During this adaptation, it is important how the safety threats are connected to the corresponding conditions and attributes by the user and which situation is represented. In the example for this thesis, the attribute conditions assigned each threat to either the safety or the security category and not both at the same time. The results of this method always show the information about individual threats for a certain object class. It is not directly visible which cascading effects might be caused by a threat and which object classes are affected by this. The user is only presented with information on a single point of the use case and is not aware if the threat source is in the examined object class or if this object class is only a part in the problem chain. This leads to a bad overview of the threat level for the system overall which makes it difficult to assess it. This makes it possible that the exact interdependence between safety and security is not visible in the results because the individual threats are divided into smaller independent groups. When using STRIDE-LM on the example use case from this thesis, this turned out to be a disadvantage of this method.

5.3.2 MITRE ATT&CK

MA Navigator provides a special layer for ICS, which is a good base for the interdependence of safety and security. The results from STRIDE-LM are grouped into safety and security threats. This made it necessary to join some threats in order to connect the results from STRIDE-LM with certain techniques from MA Navigator. The techniques cover both safety and security attack possibilities but the security part dominates here.

5.3.3 FACT graph

FACT graph is a method which is officially suitable for both safety and security investigations. This is why safety and security threats are connected and alternate in the tree diagram. As shown in Figure 4.2 in the results of this thesis, many attacks and failures occur, when safety and security are threatened at the same time. The user is able to see this interdependency between safety and security directly in the graph because the exact path of a failure or attack is shown. Through this illustration, it is easy to recognize what the source of the failure or attack is and what the function of the object class is for the threat. As already shown in the results above, the FACT graph analysis done for the example use case shows interdependencies of safety and security mostly for the overall system because a detailed investigation would have been too time consuming. The method works well with the interdependence between safety and security and the user gets a good base for further investigations in this direction, this is also good to see in the FACT graph in Figure 4.2.

5.3.4 Comparison

The FACT graph method is able to display the interdependence between safety and security naturally which it was designed for. STRIDE-LM is unable to display results for this topic at the same level as the FACT graph method. This is mainly due to the required adjustments to model the safety part in STRIDE-LM which take a lot of time and effort and also lead to results of lower quality. The results from the MA Navigator show that the method works well when analyzing the interdependency of safety and security and that it can compete with the attack modeling section of the FACT graph method.

5.4 Comparison

A summary of the individual groups and their respective methods is shown in Table 5.1. The scale for the assessment is “-” bad, “∼” ok and “*” good.

Major groups	Method name	Results
Method efficiency	STRIDE-LM	∼
	MITRE ATT&CK	*
	FACT graph	∼

Major groups	Method name	Results
Tools efficiency	Microsoft Threat Modeling Tool	-
	MITRE ATT&CK Navigator	*
	draw.io	∼
Method result quality	STRIDE-LM	∼
	MITRE ATT&CK	*
	FACT graph	*
Interdependence of safety and security	STRIDE-LM	∼
	MITRE ATT&CK	*
	FACT graph	*

Table 5.1: Method comparison

CHAPTER 6

Conclusion

This last chapter concludes the comparison of the different TM methods STRIDE-LM, MA and FACT graph and present the derived insights of this analysis.

The idea behind this thesis was to use different TM methods on a single use case and compare the results. The focus was laid on how the different methods handled the interdependence between safety and security and how this was displayed in the results of each method.

The created use case was intended to be a general use case that is usable by as many industrial automation systems from as many different sectors as possible in the future. It also includes all important components from and connections between each level of the automation pyramid from Figure 3.1 and covers multiple industries and automation areas. Additionally, it also covers structured network architecture known from industry 4.0. Inspiration for this use case was taken from a "Stakeholder Analysis" [HKS21].

After a short investigation, certain tools were selected for the individual methods. The selected tools for the combined methods STRIDE-LM and MA were MS TM Tool and MA Navigator, respectively. For the standalone method FACT graph, draw.io was selected. The first tool created the most difficulties during the work on the example use case. This was partially caused by the method itself because it was only designed for the analysis of security threats and there might have been other tools better suited to the particular task investigated in this thesis. For the other two methods, the tools worked well and the time required to model the use case in each of them was reasonable considering the amount of data analyzed.

The results confirmed what the modeling already indicated before, the FACT graph method requires less time and effort than the STRIDE-LM method for failure modeling. STRIDE-LM automatically generates detailed results, therefore no cases previously defined as threats will be forgotten. But in contrast to FACT graph, STRIDE-LM does not provide an overview over the complete system. Because FACT graph is not automatic,

automatic, it is the users responsibility to model all cases and it is possible that certain cases might be forgotten.

MA and FACT graph produced clearly structured results from the attack modeling in an adequate timeframe. But these results are not exactly the same since both methods worked with a different base. MA took the threats from STRIDE-LM and converted them into a format compatible with the attack database to compare them with the techniques from currently known attacks and attack-groups. The MA method is therefore limited to the detection of known techniques if applied alone and potential newer vulnerabilities might be overlooked because they are not included in the database. This can be mitigated to a degree by the combination with STRIDE-LM, because STRIDE-LM will detect unknown threats that can then be mapped with techniques from MA and used with the MA Navigator. It is beneficial to combine the two methods because they partially neutralize each other's weaknesses.

FACT graph works with the resulting situations from the evaluation of the particular use case. This method is therefore not limited to the known scenarios and it is possible to find entirely new vulnerable areas. But on the other hand, it is also possible to miss known threats when the modeling is done carelessly or in an imprecise way and to miss potential attacks when the modeler decides to stop at the attack modeling and not go into detail with the threats on a lower level. Unlike MA, FACT graph is unable to compare the results with known attacks and a new attack model has to be set up for a certain attack.

The results show that the application of the FACT graph method is more efficient than using the combination of the methods STRIDE-LM and MA. But during the analysis of the results from the STRIDE-LM method it was discovered that the selection of a different tool for the modeling and a different approach to the implementation of the threat definitions might have yielded better results.

Overall MA turned out to be the most efficient method for the attack modeling and it was also shown that it can be used for the attack detection in an industrial automation system. This is an important point that can be valuable for further research in the future. The pressure to digitalize the industrial sector is ever increasing and attackers are always fast in finding new attack vectors. To keep up with this development and to strengthen the security and safety sector, it is therefore important to continue the research in this field. This includes the investigation which methods, combinations or optimisations have the highest value and are the most efficient to use and how safety and security best support each other.

List of Figures

List of Tables

Acronyms

ACT Attack Countermeasure Tree. 25

BSI German federal office for information security. 10

C2 Command and Control. 9

CERT Computer Emergency Response Team. 6, 17

CIA Confidentiality, Integrity, and Availability triad. 22

CKC Cyber Kill Chain. 4, 8, 9, 67

CNC Computerized Numerical Control. 31

CPS Cyber-Physical Systems. 22, 23, 25, 30

CVSS Common Vulnerability Scoring System. 15, 16, 67

DFD Data Flow Diagram. 17, 19, 20

DMZ Demilitarized Zone. 8

DoS Denial of Service. 21

ERP Enterprise Resource Planning. 28, 31

FACT Failure-Attack-CounTermeasure. 1, 15, 22–25, 33, 37, 39, 40, 46, 47, 58–63, 65–67, 69

FIRST Forum of Incident Response and Security Team. 15

FTA Fault Tree Analysis. 16, 23, 24, 37, 40, 46–51, 67

HARA Hazard Analysis and Risk Management. 18

HAZOP Hazard and Operability. 16

HMI Human Machine Interface. 31

ICS Industrial Control System. 1, 8–10, 12, 37, 40, 59, 60, 62, 67

IEC International Electrotechnical Commission. 22

IIoT Industrial Internet of Things. 8–10, 67

IoT Internet of Things. 31

ISA International Society of Automation. 22

IT Information Technology. 6, 8, 10, 27

LIMS Laboratory Information Management System. 32, 35

LINDDUN Linkability, Identifiability, Non-Repudiation, Detectability, Disclosure of Information, Unawareness and Non-Compliance. 16, 17, 67

MA MITRE ATT&CK. 37, 40, 55, 58, 61, 65, 66, 68

MA Navigator MITRE ATT&CK Navigator. 37, 43–45, 58–62, 65–67, 69

MES Manufacturing Execution System. 31

MRP Material Requirements Planning. 31

MS TM Tool Microsoft Threat Modeling Tool. 22, 33, 34, 57–61, 65, 67

NIST National Institute of Standards and Technology. 2, 14, 15

OCTAVE Operationally Critical Threat Asset and Vulnerability Evaluation. 17, 18

OPC UA Open Platform Communications Unified Architecture. 31

OT Operational Technology. 8, 10, 35

PASTA Process for Attack Simulation and Threat Analysis. 18, 19, 67

PIMS Product Information Management System. 32

PLC Programmable Logic Controller. 31, 39–42

RFID Radio-frequency Identification. 31, 42

SAHARA Security Aware Hazard and Risk Analysis. 18

SCADA Supervisory Control and Data Acquisition. 31

SCARA Selective Compliance Assembly Robot Arm. 31

SDLC Software Development Lifecycle. 18

SIL Safety Integrity Level. 36

STRIDE Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege. 17, 18, 20, 22

STRIDE-LM Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege and Lateral Movement. 1, 15, 20, 22, 33, 36, 37, 43, 44, 57–59, 61–63, 65–67, 69

TM Threat Modeling. 1, 3, 14–18, 20, 22, 24, 33, 37, 65, 67

VAST Visual, Agile, Simple Threat modeling. 18, 19

Bibliography

[ACS19] Alexander Adamov, Anders Carlsson, and Tomasz Surmacz. An Analysis of LockerGoga Ransomware. In *2019 IEEE East-West Design Test Symposium (EWDTS)*, pages 1–5, 2019.

[APO22] APO. A Teen Tech Genius Figured Out How to Hijack Teslas. https://www.derstandard.at/story/2000132518550/19-jaehriger-hacker-brachte-weltweit-25-teslas-unter-seine-kontrolle, 2022.

[apoG19] National Cyber Security Centre a part of GCHQ. Advisory: Ryuk ransomware targeting organisations globally. https://www.ncsc.gov.uk/news/ryuk-advisory, 2019.

[ATT22] MITRE ATT&ACK. Mitre att&ack. https://attack.mitre.org/, 2022.

[Bay15] Paul Baybutt. Competency requirements for process hazard analysis (PHA) teams. *Journal of Loss Prevention in the Process Industries*, 2015.

[Bel19] Ivan Belcic. Was ist Malware? https://www.avast.com/de-de/c-malware, 2019.

[BGK+18] Michael Bartsch, Lukas Gentemann, Prof. Timo Kob, Christoph Krösmann, Marco Mille, Axel Petri, Teresa Ritter, Peter Rost, Swantje Schmidt, Marco Schulz, Dr.Dan Trapp, Lars Wittmaack, and Torsten Wunderlich. Spionage, Sabotage und Datendiebstahl – Wirtschaftsschutz in der Industrie. https://www.bitkom.org/sites/default/files/file/import/181008-Bitkom-Studie-Wirtschaftsschutz-2018-NEU.pdf, 2018.

[BMF18] Deborah J Bodeau, Catherine D McCollum, and David B Fox. Cyber threat modeling: Survey, assessment, and representative framework. https://www.mitre.org/sites/default/files/publications/pr{_}18-1174-ngci-cyber-threat-modeling.pdf, 2018.

[BWCD+17] Josh Brown-White, Loren Brent Cobb, Jim DelGrosso, Ehsan Foroughi, Afshar Ganjali, Souheil Moghnie, Nick Ozmore, Ragavendran Padmanabhan, Brook Schoenfield, and Izar Tarandach. Tactical Threat Modeling. https://safecode.org/wp-content/uploads/2017/05/SAFECode{_}TM{_}Whitepaper.pdf, 2017.

[CER21a] Kaspersky ICS CERT. Advanced persistent threat actor Lazarus attacks defense industry, develops supply chain attack capabilities. https://www.kaspersky.com/about/press-releases/2021{_}advanced-persistent-threat-actor-lazarus-attacks-defense-industry-develops-supply-chain-attack-capabilities, 2021.

[CER21b] Kaspersky ICS CERT. Tausende ICS-Computer weltweit von neuer Spyware-Kampagne betroffen. https://www.kaspersky.de/about/press-releases/2021{_}tausende-ics-computer-weltweit-von-neuer-spyware-kampagne-betroffen, 2021.

[CHP+17] Sabarathinam Chockalingam, Dina Hadžiosmanović, Wolter Pieters, André Teixeira, and Pieter van Gelder. Integrated Safety and Security Risk Assessment Methods: A Survey of Key Characteristics and Applications. In Grigore Havarneanu, Roberto Setola, Hypatia Nassopoulos, and Stephen Wolthusen, editors, *Critical Information Infrastructures Security*, pages 50–62, Cham, 2017. Springer International Publishing.

[Cis21] Cisco. What is Malware? https://www.cisco.com/c/en/us/products/security/advanced-malware-protection/what-is-malware.html, 2021.

[CMN+17] Ian Cameron, Sam Mannan, Erzsébet Németh, Sunhwa Park, Hans Pasman, William Rogers, and Benjamin Seligmann. Process hazard analysis, hazard identification and scenario definition: Are the conventional tools sufficient, or should and can we do much better? *Process Safety and Environmental Protection*, 2017. Loss prevention and safety promotion in the process industries: issues and challenges.

[CMT12] Dawn Cappelli, Andrew Moore, and Randall Trzeciak. *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley Professional, 2012.

[Cob21] Michael Cobb. Threat modeling. https://www.techtarget.com/searchsecurity/definition/threat-modeling, 2021.

[CRV+20] Daniel Cortés, José Ramírez, Luis Villagómez, Rafael Batres, Virgilio Vasquez-Lopez, and Arturo Molina. Digital Pyramid: an approach to relate industrial automation and digital twin concepts. In *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–7, 2020.

[CS17] Jin Cui and Giedre Sabaliauskaite. On the alignment of safety and security for autonomous vehicles. *Proc. IARIA CYBER*, pages 1–6, 2017.

[CSYW07] Richard A Caralli, James F Stevens, Lisa R Young, and William R Wilson. Introducing octave allegro: Improving the information security risk assessment process. Technical report, Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, 2007.

[DeM19] Katie DeMatteis. The 4 Types of Attackers and Their Motives. `https://blogs.vmware.com/security/2019/08/the-4-types-of-attackers-and-their-motives.html`, 2019.

[Eni22] Enisa. Octave. https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/rm-ra-methods/m_octave.html, 2022.

[Enu21] Common Weakness Enumeration. About CWE. `https://cwe.mitre.org/about/index.html`, 2021.

[FIR19] FIRST. Common Vulnerability Scoring System version 3.1: Specification Document. `https://www.first.org/cvss/v3-1/cvss-v31-specification{_}r1.pdf`, 2019.

[fIS21] Center for Internet Security. The SolarWinds Cyber-Attack: What You Need to Know. `https://www.cisecurity.org/solarwinds/`, 2021.

[fOHS22] Canadian Centre for Occupational Health and Safety. Hazard and Risk. `https://www.ccohs.ca/oshanswers/hsprograms/hazard{_}risk.html`, 2022.

[Fun21] Brian Fung. Colonial Pipeline says ransomware attack also led to personal information being stolen. `https://edition.cnn.com/2021/08/16/tech/colonial-pipeline-ransomware/index.html`, 2021.

[Gon20] Cynthia Gonzalez. 6 Threat Modeling Methodologies: Prioritize and Mitigate Threats. `https://www.exabeam.com/information-security/threat-modeling/`, 2020.

[Gra21] Kaitlyn Graham. Cyber Security Risk Modeling: What Is It And How Does It Benefit Your Organization? https://www.bitsight.com/blog/cyber-security-risk-modeling, 2021.

[GSO17] Angelito Gabriel, Juan Shi, and Cagil Ozansoy. A Proposed Alignment of the National Institute of Standards and Technology Framework with the Funnel Risk Graph Method. *IEEE Access*, 5:12103–12113, 2017.

[HB18] Amin Hassanzadeh and Robin Burkett. SAMIIT: Spiral Attack Model in IIoT Mapping Security Alerts to Attack Life Cycle Phases. *Accenture Technology Labs Arlington, Virginia, USA*, 2018.

[HBB+21] Siegfried Hollerer, Bernhard Brenner, Pushparaj Bhosale, Clara Fischer, Ali M. Hosseini, Sofia Maragkou, Maximilian Papa, Sebastian Schlund, Thilo Sauter, and Wolfgang Kastner. Challenges in OT Security and their Impacts on Safety-related Cyber-physical Production Systems. *TUWien*, 2021.

[HKS21] Siegfried Hollerer, Wolfgang Kastner, and Thilo Sauter. Safety und Security–ein Spannungsfeld in der industriellen Praxis. *e&i Elektrotechnik und Informationstechnik*, 138(7):449–453, 2021.

[HLOS06] Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. THREAT MODELING Uncover Security Design Flaws Using The STRIDE Approach. https://web.archive.org/web/20070303103639/http://msdn.microsoft.com/msdnmag/issues/06/11/ThreatModeling/default.aspx, 2006.

[HPO+15] Mario Hermann, Tobias Pentek, Boris Otto, et al. Design principles for Industrie 4.0 scenarios: a literature review. *Technische Universität Dortmund, Dortmund*, 45, 2015.

[HQA+16] Al-Mohannadi Hamad, Mirza Qublai, Namanya Anitta, Awan Irfan, Cullen Andrea, and Disso Jules. Cyber-Attack Modeling Analysis Techniques: An Overview. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2016.

[Ins20] Infosec Insights. Different Types of Hackers: The 6 Hats Explained. https://sectigostore.com/blog/different-types-of-hackers-hats-explained/, 2020.

[itGtCCRAfCII20] CSA issued the Guide to Conducting Cybersecurity Risk Assessment for Critical Information Infrastructure. GUIDE TO CYBER THREAT MODELLING. https://www.csa.gov.sg/-/media/csa/documents/legislation{_}supplementary

{_}references/guide-to-cyber-threat-modelling.pdf, 2020.

[JAMH22] Mohammad Jbair, Bilal Ahmad, Carsten Maple, and Robert Harrison. Threat modelling for industrial cyber physical systems in the era of smart manufacturing. *Computers in Industry*, 137:103611, 2022.

[JB21] Charlotta Johnsson and Dennis Brandl. Beyond the Pyramid: Using ISA95 for Industry 4.0/Smart Manufacturing. *InTech*, pages 14–20, 2021.

[KBK+19] Marc-Fabian Körner, Dennis Bauer, Robert Keller, Martin Rösch, Andreas Schlereth, Peter Simon, Thomas Bauernhansl, Gilbert Fridgen, and Gunther Reinhart. Extending the Automation Pyramid for Industrial Demand Response. *Procedia CIRP*, 81:998–1003, 2019. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.

[KKG20] Georgios Kavallieratos, Sokratis Katsikas, and Vasileios Gkioulos. Cybersecurity and Safety Co-Engineering of Cyberphysical Systems—A Comprehensive Survey. *Future Internet*, 12(4), 2020.

[Kle19] Mario Klessascheck. HAZOP – Risikoanalyse konform IEC 61882. *Johner Institut*, 2019.

[Kop20] Vyacheslav Kopeytsev. Attacks on industrial enterprises using RMS and TeamViewer: new data. https://ics-cert.kaspersky.com/media/Kaspersky-Attacks-on-industrial-enterprises-using-RMS-and-TeamViewer-EN.pdf2, 2020.

[KPCBH15] Siwar Kriaa, Ludovic Pietre-Cambacedes, Marc Bouissou, and Yoran Halgand. A survey of approaches combining safety and security for industrial control systems. *Reliability Engineering & System Safety*, 139:156–178, 2015.

[Kri17] Duane Kritzinger. 4 - Fault tree analysis. In Duane Kritzinger, editor, *Aircraft System Safety*, pages 59–99. Woodhead Publishing, 2017.

[LFK+14] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industrie 4.0. *Wirtschaftsinformatik*, 56(4):261–264, 2014.

[MF14] Michael Muckin and Scott C Fitch. A threat-driven approach to cyber security. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Threat-Driven-Approach.pdf, 2014.

[Mic09] Microsoft. The STRIDE Threat Model. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN, 2009.

[Mic22] Microsoft. Microsoft Threat Modeling Tool. https://docs.microsoft.com/de-de/azure/security/develop/threat-modeling-tool, 2022.

[Moh21] Ramya Mohanakrishnan. What Is Threat Modeling? Definition, Process, Examples, and Best Practices. https://www.spiceworks.com/it-security/network-security/articles/what-is-threat-modeling-definition-process-examples-and-best-practices/, 2021.

[MPMM21] Edwin Mauricio Martinez, Pedro Ponce, Israel Macias, and Arturo Molina. Automation pyramid as constructor for a complete digital twin, case study: A didactic manufacturing system. *Sensors*, 21(14):4656, 2021.

[MS16] Zhendong Ma and Christoph Schmittner. Threat modeling for automotive security analysis. *Advanced Science and Technology Letters*, 139:333–339, 2016.

[MSB+15] Georg Macher, Harald Sporer, Reinhard Berlach, Eric Armengaud, and Christian Kreiner. SAHARA: A security-aware hazard and risk analysis method. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 621–624, 2015.

[oA04] International Society of Automation—ISA. Technical report, ANSI/ISA 84.00. 01-2004, Application of Safety Instrumented Systems for the Process Industries. https://sis-tech.com/wp-content/uploads/2011/05/ANSIISA{_}84.00.01-2004{_}and{_}Existing{_}Safety{_}Instrumented{_}Systems.pdf, 2004.

[oCS19] BSI Publications on Cyber-Security. Industrial Control System Security - Top 10 Threats and Countermeasures. https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS{_}005E.pdf?{_}{_}blob=publicationFile{&}v=1, 2019.

[OEP20] Bamidele Ola and Iyobor Egho-Promise. Cybersecurity Threat Modelling: A Case Study of An Ecommerce Platform Migration to the Public Cloud. *European Journal of Electrical Engineering and Computer Science*, 4(4), 2020.

[oWA22] Goverment of Western Australia. What is a hazard and what is risk? https://www.dmp.wa.gov.au/Safety/What-is-a-hazard-and-what-is-4721.aspx, 2022.

[PDC18] Alessandro Di Pinto, Younes Dragoni, and Andrea Carcano. TRITON: The First ICS Cyber Attack on Safety Instrument Systems . *NOZOMI Network*, 2018.

[RKT12] Arpan Roy, Dong Seong Kim, and Kishor S. Trivedi. Scalable optimal countermeasure selection using implicit enumeration on attack countermeasure trees. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pages 1–12, 2012.

[Roo21] Enoch Root. PseudoManuscrypt's nonstandard industrial attack. https://www.kaspersky.com/blog/pseudomanuscrypt-industrial-malware/43177/, 2021.

[RPC19] Theofanis P Raptis, Andrea Passarella, and Marco Conti. Data management in industry 4.0: State of the art and open challenges. *IEEE Access*, 7:97052–97093, 2019.

[sa21] Cybersecurity & Infrastructure security agency. Guidance on Remediating Networks Affected by the SolarWinds and Active Directory/M365 Compromise. https://www.cisa.gov/uscert/ncas/current-activity/2021/03/09/guidance-remediating-networks-affected-solarwinds-and-active, 2021.

[Sap17] Josephine Cordero Sapién. Global Cyber Attack Hits Deutsche Bahn. https://railway-news.com/global-cyber-attack-hits-deutsche-bahn/, 2017.

[Sar21] Ajay Sarangam. 10 Types of Hackers To Be Aware Of In 2021. https://www.jigsawacademy.com/blogs/cyber-security/different-types-of-hackers/, 2021.

[SCO+18] Nataliya Shevchenko, Timothy A Chick, Paige O'Riordan, Thomas P Scanlon, and Carol Woody. Threat modeling: a summary of available methods. https://resources.sei.cmu.edu/asset{_}files/WhitePaper/2018{_}019{_}001{_}524597.pdf, 2018.

[Sec21] Panda Security. 14 Types of Hackers to Watch Out For. https://www.pandasecurity.com/en/mediacenter/security/14-types-of-hackers-to-watch-out-for/, 2021.

[SFS+11] Keith Stouffer, Joe Falco, Karen Scarfone, et al. Guide to industrial control systems (ics) security. *NIST special publication*, 800(82):16–16, 2011.

[She18] Nataliya Shevchenko. Threat Modeling: 12 Available Methods. https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/, 2018.

[Shi07] Robert W. Shirey. Internet Security Glossary, Version 2. *FYI 36*, 2007.

[Sho14] Adam Shostack. *Threat modeling: Designing for security.* John Wiley & Sons, 2014.

[SM15] Giedre Sabaliauskaite and Aditya P. Mathur. Aligning Cyber-Physical System Safety and Security. In Michel-Alexandre Cardin, Daniel Krob, Pao Chuen Lui, Yang How Tan, and Kristin Wood, editors, *Complex Systems Design & Management Asia*, pages 41–53, Cham, 2015. Springer International Publishing.

[SNRM17] Giedre Sabaliauskaite, Geok See Ng, Justin Ruths, and Aditya Mathur. A comprehensive approach, and a case study, for conducting attack detection experiments in Cyber–Physical Systems. *Robotics and Autonomous Systems*, 98:174–191, 2017.

[src] NIST Computer security resource center. Glossary terms and definitions. https://csrc.nist.gov/glossary/.

[SSM+16] Sven Schrecker, Hamed Soroush, Jesus Molina, JP LeBlanc, Frederick Hirsch Marcellus Buchheit, Andrew Ginter, Robert Martin, Harsha Banavara, Shrinath Eswarahally, Kaveri Raman, Andrew King, Qinqing Zhang, Peter MacKay, and Brian Witten. Industrial Internet of Things Volume G4: Security Framework. *Industrial Internet Consortium*, 2016.

[SWT21] Hollerer Siegfried, Kastner Wolfgang, and Sauter Thilo. Towards a Threat Modeling Approach Addressing Security and Safety in OT Environments. In *2021 17th IEEE International Conference on Factory Communication Systems (WFCS)*, 2021.

[sü22] TÜV süd. HAZARD and operability study (Hazop) procedure. https://www.tuvsud.com/en/industries/energy/conventional-power/hazop-procedure, 2022.

[Too09] CSF Tools. STRIDE-LM Threat Model. https://csf.tools/reference/stride-lm/, 2009.

[Tr07] ANSI/ISA-99-00-01-2007 Technical report. Security for industrial automation and control systems part 1: Terminology, concepts, and models. *International Society for Automation*, 10, 2007.

[TSAK21] Matt Tatam, Bharanidharan Shanmugam, Sami Azam, and Krishnan Kannoorpatti. A review of threat modelling approaches for APT-style attacks. *Heliyon*, 7(1):e05969, 2021.

[Tuc21] Linda Tucci. What is risk management and why is it important? `https://searchcompliance.techtarget.com/definition/risk-management`, 2021.

[Uce12] Tony UcedaVelez. Real World Threat Modeling Using the PASTA Methology. `https://owasp.org/www-pdf-archive/AppSecEU2012{_}PASTA.pdf`, 2012.

[WDY+14] Xiaomin Wei, Yunwei Dong, Mengmeng Yang, Ning Hu, and Hong Ye. Hazard analysis for AADL model. In *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2014.

[ZSM+19] Abe Zeid, Sarvesh Sundaram, Mohsen Moghaddam, Sagar Kamarthi, and Tucker Marion. Interoperability in smart manufacturing: Research challenges. *Machines*, 7(2):21, 2019.