A Semantic Security Model for Cyber-Physical Systems to Identify and Evaluate Potential Threats and Vulnerabilities

Andreas Aigner[a] and Abdelmajid Khelil[b]
Faculty of Computer Science, Landshut University of Applied Science, Landshut, Germany

Keywords: Security, Security Model, Security Engineering, Cyber-Physical Systems, Threat Model, System Model, Model-based Engineering, Security Analysis, Security Scoring, Security Metric.

Abstract: Establishing and sustaining a sufficient level of security in Cyber-Physical Systems (CPS) proposes a major challenge for engineers. Key characteristics, like heterogeneity, unpredictability and safety-relevance have the potential to significantly impact the overall level of security. However, exploited security-related vulnerabilities may cause malfunction of critical components or result in loss of sensitive information. Therefore, a toolkit, which is capable to identify vulnerabilities regarding security in CPS, would provide great benefit. Although a variety of security analysis frameworks exist, they mainly do not address the challenges proposed by CPS, which limits their applicability or accuracy. We aim to elaborate a more effective solution for CPS by analysing security on a Systems-of-Systems level. Moreover, we focus on the semantic relationships between essential security information, like attackers and attacks, towards the actual specification of the CPS. Our elaborated approach produces a quantitative expression of security, based on a variety of evaluation criteria and -policies. Ultimately, the generated output provides a quick indication about potential security-related threats and vulnerabilities. We utilize a prototypical, but realistic car-sharing application as a prime example for CPS, to illustrate the benefits and ease-of-use of our proposed solution.

1 INTRODUCTION

Industry 4.0, Smart Grid or Connected Vehicles represent prime examples of Cyber-Physical Systems (CPS). In these ecosystems, systems from the cyber-space- and embedded domain must work hand in hand to fulfill the desired business functionality or to provide the necessary infrastructure. For example, vehicles are now enhanced with the ability to communicate and interact with their environment to introduce new functionality for drivers.

Nevertheless, CPS also introduce a variety of challenges (Ashibani and Mahmoud, 2017) – especially for system engineers and administrators. One major aspect, refers to establishing and sustaining a sufficient level of security. The interaction of heterogeneous systems in a highly constrained and often unpredictable environment opens a variety of (new) attack surfaces. As CPS often implement critical, i.e. safety-relevant, functionality or provide the infrastructures for such services, exploited systems may cause harm to life of involved humans or end in a loss of sensitive information (O`Neil, 2016).

To counter this issue, security- or threat analysis frameworks are used to provide information about security-related weaknesses of a specification. In general, they try to express the abstract term of security based on actual measurements, parameters or a dedicated rule set. Recent surveys (Pendleton et al., 2017. Rudolph and Schwarz, 2021) illustrate that several security analysis frameworks and –metrics are available. However, the characteristics of CPS, like (a) the interaction of heterogonous systems in a dynamic and scalable environment, (b) the need of reliability in an unpredictable environment or (c) distinctive physical characteristics, the applicability and accuracy of existing solutions may be limited.

Although solutions tailored to the application areas of CPS have been proposed (Shevchenko et al., 2018), they are not capable to comprehensively address all the aspects regarding security of a highly

[a] https://orcid.org/0000-0001-8990-4775
[b] https://orcid.org/0000-0002-4536-8058

Aigner, A. and Khelil, A.
A Semantic Security Model for Cyber-Physical Systems to Identify and Evaluate Potential Threats and Vulnerabilities.
DOI: 10.5220/0011086300003194
In *Proceedings of the 7th International Conference on Internet of Things, Big Data and Security (IoTBDS 2022)*, pages 249-257
ISBN: 978-989-758-564-7; ISSN: 2184-4976

constrained System-of-Systems (SoS). In this work, we want to address this research gap by elaborating a security model for CPS, which can effectively be used in environments to comprehensively analyse and express security. To achieve this goal, we set our focus on the semantic relationships between the interacting systems and their dependencies towards security-related entities, like attackers and attacks.

In detail, we focus on the most critical aspects and relationships, which have the ability to affect the level of security in a negative way from one particular systems point of view. To obtain this information, we identify the semantic relationships between all core security objects on a SoS level, while considering the actual business logic. The used data is semantic, as it is (a) directly taken from the business logic and (b) depends on the currently analyzed system's role, meaning and behavior in the SoS context. In contrast to other solutions, we do not want to interpret the impact of attacks for all potential types of systems in a generic way, but focus on the impact one particular type of attack has on one specific system, within a given (operational) context. To illustrate and underline the thinking process of the definitions made in this work, we utilize a prototypical car sharing application. The usage of realistic CPS specification should also allow us to verify the correctness of our solution, while showcasing the benefits for engineers.

The reminder of the paper is structured as follows. In Section 2, we start by illustrating the problem statement regarding security engineering for CPS. In Section 3, we take a look at existing solutions and evaluate their effectiveness and accuracy in CPS. In Section 4, we introduce our concept of how to analyze security in CPS by elaborating a semantic security model. Here, we also give insight into our conducted case study along the made explanations and definitions. We sum up our work in Section 5.

2 PROBLEM STATEMENT

In order to follow a hands-on approach, we first define a concrete specification of a prototypical CPS. This will enable us to identify the most critical characteristics and aspects, which must be addressed, as they potentially influence security. To this end, we use a realistic car sharing application. Figure 1 illustrates a high-level model of this application.

It couples different types of systems from the cyber-space- and physical domain, while showcasing the key CPS aspects of heterogeneity, unpredictability in an unreliable environment and critical assets. The car sharing domain consists of various system types, which offer different (physical) characteristics. Additionally, human actors are also involved, who use a mobile phone to enter their credentials to perform an authentication. A dedicated application on the mobile phone prepares an authentication request, which is sent towards the connected vehicle via Bluetooth. The connected vehicle forwards the received requests via a 5G/LTE uplink it has towards the backend service. The backend service verifies, whether the received authentication request is valid and prepares an according response (authentication token), which is sent back towards the vehicle. Based on the verification result, the vehicle will be unlocked and the end user can enter the connected vehicle, as it unlocks its doors.

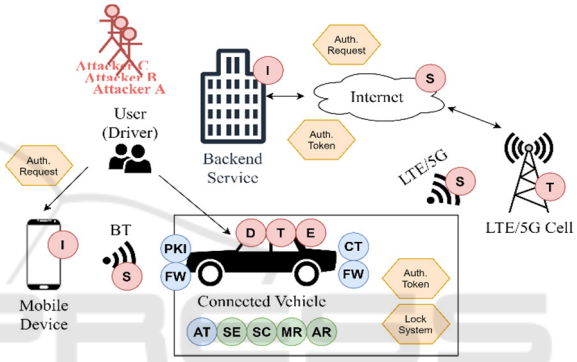

Figure 1: A prototypical Car Sharing Application.

Besides the actual business logic, we also integrate security-related information to learn about the dependencies between these two types of data. First, we have added a list of threats on a SoS level. To this end, we use the well-known STRIDE paradigm, which defines Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service and Elevation-of-Privileges as classification from a high-level perspective, to insert a list of potential attacks.

In contrast to the threats, we also added several mitigation concepts. To apply an objective baseline, we first adopt a list of security requirements (green background in Figure 1) defined AUTOSAR (Fürst and Bechter, 2016), i.e. access restriction (AR) at the physical components, separation (SE) of memory, processes and persistence and secure channels (SC). Additionally, we integrate a set of mitigations (blue background in Figure 1), which seem meaningful from our point of view for a system type *Vehicle*. In detail, we adopt a public-key-infrastructure (PKI), using ciphered-transmission (CT), installing a firewall (FW), and implementing an anti-tamper (AT) mechanism. Later, we illustrate the benefit of our

proposed solution, which is the ability to compare multiple specifications of the same CPS.

We now elaborate the challenges proposed by CPS. First, CPS spawn a SoS environment (Challenge-01), which consists of multiple systems. These systems implement various dependencies to provide the overall functionality. In contrast to embedded systems, a CPS is not deployed to a closed environment, but is equipped with the ability to communicate beyond the area they operate in.

This high level of connectivity in combination with the increased number of interfaces also extends the options an attacker might choose from to exploit one or multiple systems within the SoS. Moreover, the number of interacting systems cannot be foreseen in terms of scalability (Challenge-02). Additionally, CPS consist of heterogeneous systems (Challenge-03), referring to distinctive physical attributes and capabilities, e.g. computation power or available bandwidth. As CPS couple systems from the cyber- and physical domain, both types must be aware of the issues regarding security of the other type, as they might face the same attacking scenarios.

As CPS usually build the backbone of critical infrastructure, fulfil safety-relevant functionality or process sensitive data, exploited vulnerabilities potentially cause major harm. Therefore, the given multi-criticality (Challenge-04), e.g. real-time processing, significantly stresses the need to achieve a sustainable level of security. In the same context, human actors (Challenge-05) are often directly affected by the functionality our output of CPS. On one side, control loops in CPS do often depend on input or triggers provided by the cyber-space domain which leads to a critical dependency. On the other side, they must operate in a highly unpredictable environment. The disjunctive characteristics of dependability in a potentially unknown environment (Challenge-06) may create new attack surfaces, as communication links may break down spontaneously or systems do not provide input as expected.

As the systems within CPS are often not fixed to a geographical area, they must also be context aware (Challenge-07). As CPS often operate in an unknown or unsupervised environment, attackers may face no hurdles to gain (physical) access to the interfaces of the (geographically) deployed systems.

All these key characteristics of CPS have the potential to significantly impact the level of security throughout the SoS.

3 RELATED WORK

As illustrated in Section 2, the characteristics of CPS-like environments propose critical challenges for security engineering in CPS engineering, as various, often distinctive, characteristics and side effects must be considered simultaneously. Although security has traditionally been a point of emphasis in cyber-space engineering, it may not have played a major role during the design of embedded systems. Although a number of security analysis frameworks and have emerged in the cyber-space domain over time, solutions for CPS (including the physical aspects) are relatively new on the horizon.

In this work, we want to evaluate a selection of available security analysis frameworks in terms of evaluating their applicability, effectiveness and accuracy in a CPS-like environment. In detail, we use the elaborated list of key challenges (Challenge-01 to 07) proposed by CPS as evaluation criteria. The selection of evaluated solutions has been made based on the first impressions provided by the referenced surveys (Pendleton et al., 2017. Rudolph and Schwarz, 2021. Shevchenko et al., 2018), alongside the potential challenges proposed by a CPS.

Table 1: Evaluation of Related Work in terms of their effectiveness and accuracy in CPS security engineering.

Framework	C 1	C 2	C 3	C 4	C 5	C 6	C 7
Common Vulnerability Scoring System (Chandramouli, 2006)	☹	☹	😐	☹	☹	☹	☺
Attack Trees (Nagaraju, 2017)	😐	☹	☺	☹	😐	😐	😐
Attack Surface (Manadhata, 2011)	😐	☹	☺	☹	😐	☹	😐
STRIDE-based (Khan, 2017)	☹	☹	😐	☹	☹	☹	☹
hTTM (Mead, 2018)	😐	😐	😐	😐	☺	☹	☺

Table 1 provides an overview of our evaluation results. The first column states the framework, whereas the following (by row) contain the evaluation criteria, i.e. the challenges (C) 1 to 7, as introduced in the previous section. To this end, we use three result categories as rating schemata – a happy smiley, if the method is able to handle and support the aspect, a neutral one, if it is partly able to handle it, and a frowning smiley, if this aspect is not addressed.

Consequently, we have evaluated the aspects of applicability, effectiveness and accuracy for each challenge to determine the overall score for the corresponding framework. As the results in Table 1 illustrate, none of the evaluated framework is capable to sufficiently address all of the proposed challenges.

They either focus on aspects related to the cyber-space attributes or focus on security derived from safety-related engineering, whereas a concept to simultaneously address both domains is missing.

4 SEMANTIC SECURITY MODEL

In essence, we set our goal on elaborating a security analysis model, which considers all key challenges and attributes proposed by CPS to provide the most comprehensive view on security in heterogeneous SoS-like environments.

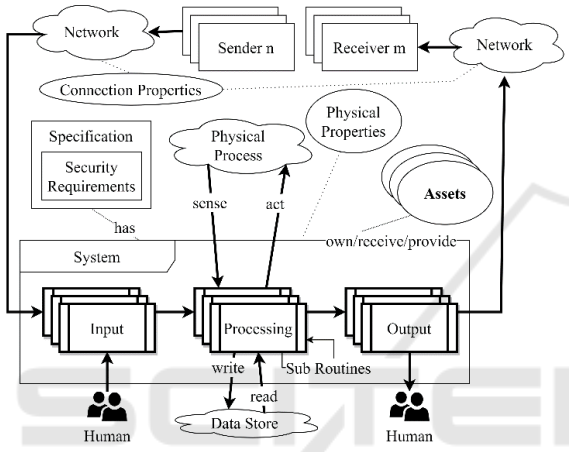

Figure 2: A System Model for CPS.

4.1 General Philosophy

Based on the elements illustrated in Figure 1, we first generate a system model (Figure 2) for one particular system operating in a CPS. Ultimately, this system model allows us to identify the most critical aspects regarding security in CPS, while also representing the baseline for a security model for CPS. Essentially, each system consists of one or multiple input-, processing and output components, whereas these components have direct relationships towards the surrounding environment or additional CPS-related resources. In detail, the input- and output components realize the communication with other systems, whereas the processing unit access physical-, e.g. sensors, or cyber-space, e.g. data storage, resources.

Furthermore, any component may interact with assets or human actors while being in operation. It is important to note, that this system model is structured to being used for either system within CPS, meaning physical-, cyber-physical- and cyber-space systems. Consequently, one particular system may not show all of the listed characteristics. Based on the identified elements and interfaces illustrated by the system model for CPS (Figure 2), we are now able to generate a security model, as outlined in Figure 3, for one particular system within the CPS. The currently analysed system is called Target of Evaluation (ToE). This model represents an abstract view on the relationships between the key security objects and business entities from a SoS perspective. Overall, we identify four critical aspects – the executability-, the exploitability- the effect- and the scalability of one particular attack towards a given system within the current (semantic) operational context, i.e. the characteristics of the system and its ecosystem.

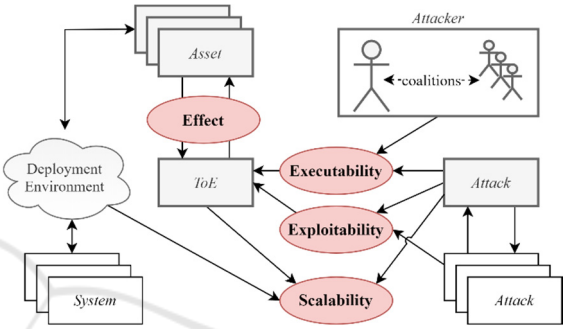

Figure 3: A semantic Security Model for CPS.

In contrast to existing solutions, we take a SoS perspective while analysing security in order to catch all security-relevant side effects and dependencies. Although we use commonly used security-relevant entities within our analysis, i.e. attackers, attacks and mitigation concepts, we focus on their semantic interpretation according based on the actual business logic and context they are analysed in. This shift of paradigm towards a semantic approach allows us to handle the challenges of CPS.

In essence, we are going to express *security* as product (1) of these four factors. In the following, we dive into detail about how the factors are determined. The overall goal is to minimize the result of this multiplication; meaning a value of zero represents the best score in terms of security-related weaknesses within the analysed CPS.

$$\text{Security} = [\ \text{Executability} * \text{Exploitability} * (1 + \text{Effect}) * (1 + \text{Scalability})] \quad (1)$$

4.2 The Executability of Attacks

The first critical dependency illustrated in Figure 3 focuses on the fundamental question, whether a particular attack is theoretically executable within the current context or not. This question can be answered by analyzing the relationship between attackers,

attacks and the ToE within its ecosystem. In general, a number of attackers, e.g. Attacker A, B and C in Fig. 1, is capable to execute a set of different attacks within the domain. In a highly constrained SoS-like environment, multiple attacker entities might be present. Nevertheless, different attackers may own different characteristics, capabilities or skill sets. On the other side, attacks require certain capabilities, skills or resources in order to being executed by a concrete attacker entity. In Figure 1, Attacker A might be able to execute a Spoofing attack, as it has access to a certain part of the network, while Attacker B does not fulfil this requirement an can therefore not perform a Spoofing attempt.

Moreover, the attribute of executability also depends on the characteristics of the analysed ToE. In detail, an attack may require fundamental properties, i.e. a certain condition, provided by the ToE in order to potentially being executable towards it, like physical- or network access. Furthermore, these elements must also be connected towards the capabilities and resources of the attacker objects. Exemplary, not only the system may offer physical access, but the attacker must also be capable to establish physical access.

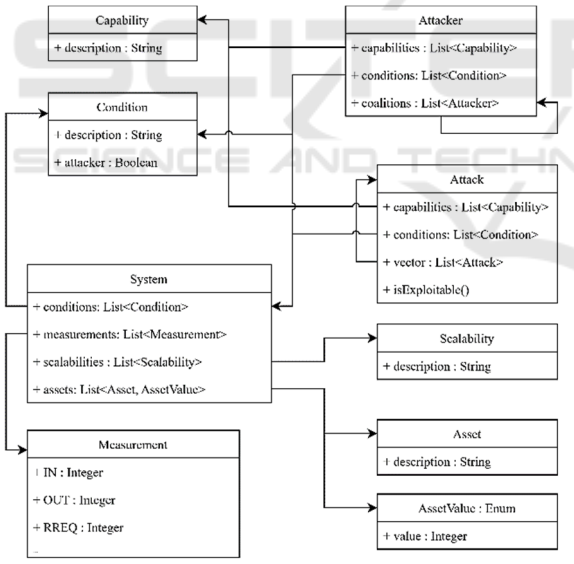

Figure 4: Data Model to store to link Security-relevant Information and the Specification.

The data model (Figure 4) documents the relationship between Attacker, Attack and ToE objects, i.e. the executability. To describe the semantic dependencies between these three objects, we introduce two additional objects – the Capability and Condition attributes. The content of these objects can be customized according to the actual use case. In general, the defined statements should be in the form of yes-/no sentences in order to allow a verification, i.e. a matching of attributes, between them.

The overall executability factor, which holds a value of zero or 1, is determined by cross-checking the capability and condition attributes of the system, attack and attacker objects of the currently used data model. Basically, the analysis verifies for each attack-/system combination, whether the data model contains a valid combination of these attributes or not; meaning it contains an attacker object, which is capable to perform the attack towards this type of system, while also all condition attributes are met, i.e. contained in both objects, as well. The attack and attacker part of the data model can be populated by using publicly available sources.

4.3 The Exploitability of the ToE

The second dependency in Figure 3, refers to the exploitability of the ToE. Besides looking at the theoretical aspect, i.e. the executability of attacks, we want to elaborate a verification schema, which provides an (measureable) indicator, whether the ToE can actually be exploited by an attack or not.

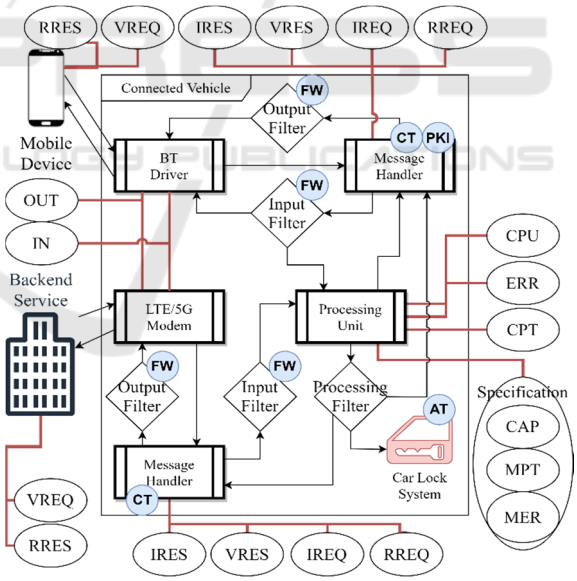

Figure 6: Abstract Specification of the Connected Vehicle.

First, certain type of attacks, e.g. Elevation of Privileges in Figure 1, require that other attacks, e.g. Spoofing, have successfully been performed in advance in order to being executable themselves. Therefore, one type of attack enables one or multiple other types of attacks. We call this sequential execution of attacks to exploit a system *attack vector*. Especially in a highly constrained, unpredictable and

dynamic ecosystem like CPS, attackers may perform complete attack vectors in order to exploit the ToE.

Second, we integrate an evaluation criterion, which indicates, whether a ToE can be exploited by an attack, if all proceeding requirements are met to perform the attack towards the ToE based on its behavior security-wise. In order to construct such evaluation criteria based on semantic data, we define a formula for each type of attack, which utilizes actual (quantitative) measurements. To this end, we refine the system model (Figure 2) by identifying potential points of measurements according to the processed assets, i.e. requests, function calls and response.

In essence, the business process offered by CPS usually implements a request-response mechanism, which processing steps within. However, some system might only provide response or formulate requests. Based on these fundamental and highly abstracted entities, i.e. request and response, we are able to derive additional parameters, which have been integrated in Figure 5. Initially, we measure the raw data concerting incoming (IN) and outgoing (OUT) data, followed by the number of real requests (RREQ) and responses (VRES), alongside the number of invalid- (IREQ) and processed requests (CPU), as well as the number of produced errors (ERR).

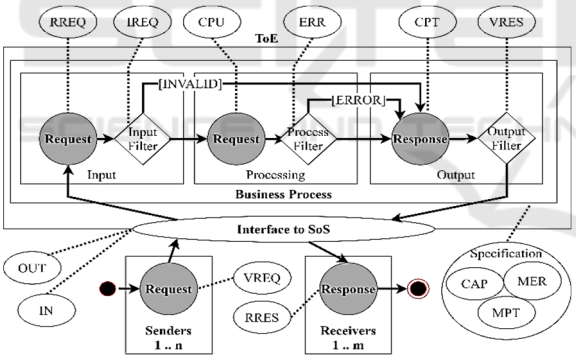

Figure 5: Measurements to determine Exploitability.

Throughout the processing sequence, we measure the actual time (CPT) from receiving a request to generating a response. The information about objects marked as invalid directly reflects on the capabilities of the ToE to detect potential threats by itself, by aborting the processing of such request. In order to set this security enhancing capabilities in contrast, we utilize additional specification parameters of the ToE within the formulas, namely the processing capabilities (CAP), the median error rate (MER) and the median processing time (MPT). Based on these data points, we are now able to quantitatively abbreviate the effects a successful attack would lead to. This is done by defining a set of binary expressions, based on the comparison of various measurements, as required by the associated type of attack. Consequently, we must identify such expression for each type of attack we want to include into our evaluation.

In this work, we have prepared the formulas (2) to (7), which contain the evaluation rules for the attacks used in Figure 1. We built these formulas by considering the impact or outcome a successful attack would have on the exchanged assets, i.e. requests and response. For example, a ToE can be exploited via Spoofing, if a modified request is not detected by the ToE through the IREQ or ERR parameter within the input interface or processing unit.

Nevertheless, the generic model (Figure 5) enables engineers to easily define formulas for additional attacks. However, the STRDIE classification should cover a variety of different potential attacks. In future work, we want to further shape and refine these evaluation formulas based on knowledge gained through applying the methodology in real-life CPS deployments.

The evaluation formulas verify, whether the security-enhancing capabilities, i.e. the Input-, Processing- and Output-Filter of the ToE are strong or sufficient enough to counter the effect of this particular attack. The values of the measurement, can be obtained by running a simulation, e.g. by using an executable UML activity model of the business process. After running the simulation, the evaluation formulas, i.e. if-statements, are used to indicate, whether the Filters have been able to neglect the effect to the attack based on the conducted measurements. Alternatively, the actually deployed systems can be extended with the ability to conduct the necessary measurements and forward the obtained information for further analysis. The defined parameters should be collectable without any user interaction or dependencies towards other systems.

$$\text{Spoofing} = [(\text{IREQ} == 0) \;\&\&\; (\text{ERR} == 0)] \quad (2)$$

$$\text{Tampering} = [(\text{IREQ} == 0) \;\&\&\; (\text{RREQ} == \text{CPU}) \;\&\&\; (\text{CPU} == \text{ERR} * 2=] \quad (3)$$

$$\text{Repudiation} = [\text{OUT} > \text{CPU}] \quad (4)$$

$$\text{Information Disclosure} = [(\text{OUT} > \text{CPU}) \,\|\, (\text{VRES} != (\text{CPU} - \text{ERR})) \,\|\, ((\text{VRES} != (\text{RREQ} - \text{IRES}) \;\&\&\; (\text{ERR} == 0))] \quad (5)$$

$$\text{Denial-of-Service} = [(\text{CPU} < \text{RRES}) \,\|\, (\text{CPT} > \text{MER}) \,\|\, (\text{ERR} > \text{MER}) \,\|\, (\text{RRES} > \text{CPU})] \quad (6)$$

$$\text{Elevation-of-Privileges} = [((\text{RREQ} - \text{IREQ}) != \text{CPU}) \;\&\&\; (\text{VRES} != \text{OUT})] \quad (7)$$

Concluding, the exploitability factor of an attack-/system combination is defined as produced of two factors. The first factor is determined by analysing the attack vector, if applicable. Here, all attacks, which are required by the currently analysed attack must have an executability- and exploitability factor of 1. If the attacks contained in the vectors do also required other attacks to be performed in advanced, their factors must also be 1 throughout. If one attack within the chain holds a value of 0, the vector is not applicable for this attack-/system combination. Overall, this can be realized as iterative verification process. The second parameter of the exploitability product is determined by the result of the evaluation formula, which leads to a value of true (=1) or false.

The data model (Figure 4) contains the necessary attributes to evaluate the relationship between the system- and attack-objects regarding the exploitability potential. However, this attribute can also be empty, which indicates, that this attack does not require any other attack to being performed in advance. Additionally, the attack object is extended with a function (*isExploitable*), which takes a set of measurements provided by the system object and returns *true* or *false* by running through the formula.

To illustrate the integration of measurements, we have refined the system type *Vehicle* of the car sharing domain accordingly, as shown in Figure. 6.

4.4 The Scalability of Attacks

As highlighted in the problem statement, CPS showcase a variety of characteristics, which have the potential to significantly impact the overall level of security in a negative way. We integrate these aspects by extending the attack- and system objects within the data model (Figure 4) with a Scalability attribute.

This attribute integrates events, properties or characteristics of the interacting systems, which increase the likelihood of being exploited by a certain attack or the severity of them. To this end, we provide a top-ten list of scalability factors (SF). This list has been generated based on the previously described key challenges proposed by CPS, but can easily be tailored to the needs according to the ecosystem:

- SF-01: The system is geographically constrained
- SF-02: The system is limited in physical resources
- SF-03: The system implements or requires safety-relevant functionality
- SF-04: The system interacts with humans
- SF-05: The system interacts with other type of systems
- SF-06: The system processes different type of assets
- SF-07: The system interacts with an unknown number of systems or depends on their input
- SF-08: The system uses unsecure, unknown or unstable communication links
- SF-09: The system changes its operational context
- SF-10: The system implements or depends on a physical process

Consequently, the scalability attribute of the system object contains all statements, which are given for this particular system. However, equal to the capability and condition attribute, this property can also be empty. On the other side, these scalability attributes define, whether they increase the likelihood or impact of a given attack object. Therefore, to evaluate this factor, the two scalability attributes of an attack-/system combination must be cross-checked for common entries. For each matching entry, the overall scalability factor for the concluding evaluation of this attack-/system combination is increased by 0.1. We advise to use a top-10 list of scalability statements for each security evaluation. Consequently, the value of this factor ranges from zero to 1.

4.5 The Effect of Attacks on Assets

If an attack is eventually able to exploit a system, we must determine the potential harm this might cause. If an attack might not be able to lead to any harm, the necessity to implement counter measurements for this particular attack might not be given. For this, we evaluate the concrete effect the attack has on all assets owned or accessed by the exploited system by using a set of classification rules. In our definition, an asset is a piece of information or functionality, which holds a certain value from the business process perspective.

Table 2: Asset Classification Schemata.

Value	Description
High (=3)	The asset contains sensitive or critical data or functionality, as the business process cannot be fulfilled as designed.
Medium (=2)	The asset holds valuable information, but the core functionality or data can still be provided.
Low (=1)	The asset is replaceable or sustainable by other assets, while it does also not contain sensitive information or logic.
n/a (=0)	The asset does not hold any value. Its non-existence of the asset does not lead to any limitation.

This classification also allows us to highlight the importance of certain assets, e.g. safety-relevant functionality. We define four classes, ranging from not applicable (n/a), Low, Medium to High, as listed

in Table 2. The system object within the data model (Figure 4) contains a reference towards the asset object, which also holds the value. This structure allows us to display the actual relationship between the system and asset, as one asset might have different values for different systems. The overall effect factor for one system is then calculated by analysing all references it has towards asset objects. For each asset, the effect factor is incremented by the value of the asset. In addition, the effect factor is incremented, if a referenced asset is also accessed or used by another system and this system is exploitable. Especially on a SoS level, this is critical to consider, as assets might be corrupted by systems, which are considered valid partners.

4.6 The Big Picture

The four factors (executability, exploitability, effect and scalability) have now been filled with semantic information by evaluating the dependencies between the CPS specification and common security-related information. In order to utilize our method, two essential task must be performed: (a) populating the data model and (b) conduct the measurements through simulation or from the deployed ToE in order to determine the exploitability.

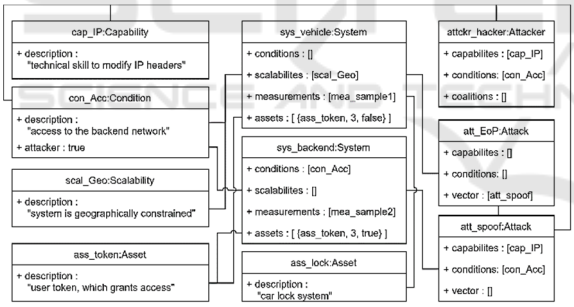

Figure 7: Case Study – Security Data Model (Extract).

For example, Figure 7 shows an extract of the data model used within the car sharing domain to evaluate security. Based on this preliminary work, we are now able to express and evaluate *security* for all attack-/system combination within our ecosystem by using the formula defined in (1). Figure 8 illustrates the calculated results for the system type *Vehicle* for all STRIDE attacks (Figure 1). As our method can effectively be used as objective and quantitative baseline to compare various specifications. To illustrate this aspect, we have generated the results for three specifications of the *Vehicle*: worst-case (not mitigations), AUTOSAR solutions and with our additional enhancements.

Table 3: Case Study – Output of the Security Model.

Vehicle Specification	S	T	R	I	D	E
Worst Case	20	18	0	15	15	17
AUTOSAR	18	9	0	9	6	9
Additional Enhancements	5	6	0	1	2	2

As the scores (Table 3) indicate, the AUTOSAR mitigations are indeed capable to reduce the impact of all attacks, although they are just implemented within the *Vehicle*. This refers to the aspect, that they directly impact the processing of the core assets within the vehicle. However, a significant impact on the overall results cannot be achieved. In contrast, the enhanced solution, i.e. the implementation of additional mitigations (blue background in Figure 1), is capable to further lower the score, as they focus on the way the assets are handled, exchanged and processed between the *Vehicle* and the interacting systems. Moreover, these results should then be used by engineers to determine the most severe attacks within the ecosystem to easily identify suitable or more effective counter measurements.

5 CONCLUSION

In this work, we have introduced a security model for CPS, which analyses the semantic dependencies between interacting systems and their relationship towards key security objects on a SoS level. In detail, we have taken a straight-forward approach using the main challenges proposed by CPS to generate a system- and consequently a security model. Concluding, this model can efficiently and effectively be used to quantitively express security based on the aspects of exploitability, executability, scalability and effect of attacks on systems and assets in order to identify actual threats or compare specifications. Furthermore, we have been able to outline several potential aspects in the form of mitigations, which should be considered when engineering a (physical) system in a connected world.

REFERENCES

Y. Ashibani and Q. H. Mahmoud (2017). *Cyber physical systems security: Analysis, challenges and solutions*. Elsevier Computers & Security 68.

M. O'Neill (2016). *Insecurity by Design: Todays IoT Device Security Problem*. Elsevier Journal on Engineering 2(1).

M. Pendleton, et al. (2017). *A Survey on Systems Security Metrics*. ACM Computing Surveys 49(4).

M. Rudolph and R. Schwarz (2021), *A Critical Survey of Security Indicator Approaches*. In Proceedings of the 7th International Conference of Availability, Reliability and Security.

N. Shevchenko, et al. (2018). *Threat Modeling for Cyber-Physical System-of-Systems: Methods Evaluation.* Technical Report. Carnegie Mellon University.

S. Fürst and M. Bechter (2016). *AUTOSAR for Connected and Autonomous Vehicles: The AUTOSAR Adaptive Platform*. In Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W).

R. Chandramouli, et al. (2006). *Common Vulnerability Scoring System*. Emerging Standards. IEEE Security & Privacy 4(6).

V. Nagaraju, et al. (2017). *A survey of fault and attack tree modeling and analysis for cyber risk management*. In Proceedings of the International Symposium on Technologies for Homeland Security.

P. Manadhata and J. M. Wing (2011). *An Attack Surface Metric*. IEEE Transactions on Software Engineering 37(3).

R. Khan, et al. (2017). *STRIDE-based Threat Modeling for Cyber-Physical Systems*. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference.

N. R. Mead, et al. (2018). *A Hybrid Threat Modeling Method*. Carnegie Mellon University.