Adaptation of Architecture Analyses: An IoT Safety and Security Flaw Assessment Approach

Julia Rauscher and Bernhard Bauer

Software Methodologies for Distributed Systems, University of Augsburg, Universitätsstraße 6a, 86159 Augsburg, Germany

Keywords: Safety, Security, Internet of Things, Medical, Wellbeing, By Design, Safety and Security Critical, Architecture Analysis, Flaw Assessment, Decision Support, Impact Analysis.

Abstract: Almost everything is connected nowadays or will be in the near future. This trend, called Internet of Things (IoT) or Cyber Physical Systems (CPS), is able to enhance multiple areas e.g. an individual's life, complex industrial processes or common medical treatments. Though, these improvements frequently affect safety and security critical topics. While this development has many advantages to bring, many challenges arise as well. Most approaches focus on safety and security analyses or monitoring tools determined to apply during run time. These approaches do not consider that plenty of the most dangerous vulnerabilities have to be addressed in the design phase already. Hence, we present an approach to adapt architecture analyses of IoT related areas to provide a holistic tool to assess flaws and possible countermeasures to design a save and secure CPS system.

1 INTRODUCTION

The Internet of Things (IoT) respectively Cyber Physical Systems (CPS) are one of the fastest and most unstoppable developments of our time. This year's expected installed 31 billion IoT devices will be more than doubled in 5 years. (SecurityToday, 2020) This development could arise positive progress in many fields. But: The IoT is a security nightmare. (iscoop, 2017) This is on a major part caused by unencrypted traffic of IoT devices. Critical data in IoT networks are sent unencrypted in 98% of traffic which enables attackers to listen and exploit the information. (Paloalto Networks, 2020) Security is often considered accordingly, but IoT networks are also fullgrown safety nightmares. The usage of IoT in application fields including human beings produces danger of life-threatening attacks or accidents. As (Paloalto Networks, 2020) has shown especially the medical field faces danger through IoT usage. Regarding this study most of the connected medical image devices are operating on OSs without update support leading to threats and hazards that impact the quality of care or privacy of patients. In addition, malware is spreading easily caused by the combination of IoT and IT assets which can lead to dangerous situations for patients as their data are exposed or interpreted falsely. Despite these dangers IoT offers plenty of benefits for health care and wellbeing processes, e.g. medical smart homes for patient monitoring or Ambient Assisted Living (AAL), connected insulin pumps respectively glucose monitoring or the support of workflows in hospitals. As discussed, complex safety and security activities are highly needed. The lack of update mechanisms is one of the big challenges since many IoT devices do not have the possibility to be updated. Reasons range from software design decisions to hardware issues, e.g. implanted devices have to be flawless before the implantation. Therefore, safety or security activities at runtime are highly needed but are not able to catch flaws early enough. Hence, an architectural approach is required which faces the vulnerabilities early as possible at design time. There are existing approaches like (Rauscher and Bauer, 2020) to identify flaws in design time. However, after flaws were identified they have to be assessed to be prevented. We have developed an architectural IoT analysis approach to enable the evaluation of design flaws followed by a design decision support for diverse countermeasure scenarios. Our approach is based on existing architectural approaches which will be adapted for IoT specific needs to cover the special IoT design challenges. As the medical respectively wellbeing area is one of the most vulnerable fields examples featuring safety and security issues of this topic demonstrate and evaluate our tool.

Rauscher, J. and Bauer, B.
Adaptation of Architecture Analyses: An IoT Safety and Security Flaw Assessment Approach.
DOI: 10.5220/0010206303200327
In *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021) - Volume 5: HEALTHINF*, pages 320-327
ISBN: 978-989-758-490-9

2 BASICS AND RELATED WORK

Flaw assessment is based on several basics depending on the used evaluation method. The needed basics and concepts are presented before our IoT assessment approach is described.

2.1 Architecture Analyses

Architecture analyses apply in several application fields like evaluation or validation possibilities, quality checks or graphical design reviews. As all of these usage opportunities are located in the design phase to detect, mitigate, prevent or assess vulnerabilities these methods come into operation especially in safety or security critical areas as in the automotive, avionics and railway industry.

In current state one of the most known and used method to evaluate failures is FMEA (Failure Mode and Effects Analysis). It is used as reliability and impact analysis to estimate the consequences of failures. The usage and combination of FMEA with a continuative impact analysis was presented by (Lohmüller et al., 2019). They included FMEA in an analysis cycle to identify potential failures following the determination of effects of failure events and the need of countermeasures.

A literature review has shown that there are plenty of analysis which can be categorized in functional and technical analysis methods respectively goals. (Rauscher et al., 2016) The analyses are able to evaluate whole systems but also to analyze single elements, attributes, dependencies or requirements. This stack of goals is realized with technical methods range from Bayesian Belief Networks (BBN), ontology approaches, weak point analysis to structural methods with matrices and more. This review has shown that almost every aspect, including safety and security issues, can be checked during the design phase with a matching technique depending on the application field, goal and available data. All these approaches have in common that they are applied after flaws are identified or are performed with potential threats. However, mitigation or prevention of vulnerabilities can be performed best if there is a possibility to identify specific threats and accidents as early as possible. Architectural pattern recognition represents one technique to identify design flaws. Events can be prevented in the future if architectural patterns are created and queried on models. (Rauscher and Bauer, 2020) present this method to recognize patterns for safety and security incidents. However, this approach does not cover the continuative assessment of flaws to allow the performance of countermeasures.

As discussed, architecture analyses are able to improve systems or single networks in several ways depending on their needs. Therefore, if suited adaptations are performed, an analyses transition from traditional application fields to IoT is highly suggested to enable the handling of safety and security flaws early. There are already some approaches trying to perform architectural methods in IoT. The work of (Wortman et al., 2017) propose to use AADL to depict security information in IoT models to avoid poor security design. However, this approach does also not include assessment of flaws. Another approach to design secure IoT offer (Lee and Law, 2017). They define IoT-based pattern to apply design check, but they focus on software-based design decisions. Hence, existing approaches to connect IoT and architectural approaches either don't focus on design decisions or missing out the assessment of identified flaws and the following identification of to-be design scenarios. In addition, safety is always only considered in the margin.

2.2 Probability Networks

Many architectural approaches which perform analyses use BBN. BBN are networks that represent the probabilistic conditional dependencies of network elements respectively variables. These networks are probabilistic, graphical models with directed and acyclic relations. The included variables have discrete states and dependencies to ancestor and descendant elements. The specific conditional probability distribution of model elements are calculated through Conditional Probability Tables (CPT) using mainly following formula from (Neapolitan et al., 2004) which have to be selected dependent on the analysis goal:

$$P(B|A) = P(A|B) * P(B)/P(A) \tag{1}$$

$$P(AB) = P(A) * P(B) \tag{2}$$

$$P(X) = \prod_{i=1}^{n} P(X_i|ancestor(X_i)) \tag{3}$$

Common BBN approaches often are not able to cover the design requirements of architectural approaches. Therefore, (Johnson et al., 2006) have developed a graphical and mathematical representation, called Extended Influence Diagrams (EID), to perform probabilistic inferences for decision and interoperability analyses. The approach covers nodes for utility, chances and decision makers alternatives to enable the evaluation of current as-is situations and possible to-be alternatives. This method can be used to review safety and security design situations and support the countermeasure decision.

3 FLAW ASSESSMENT CYCLE

As discussed, there are approaches to identify bad design decisions respectively flaws during design phase with the aid of design pattern. Our IoT assessment tool comes into use after this kind of vulnerability recognition. However, safety and security management does not stop after identification as the main challenge starts with the prevention or mitigation. Flaws have to be assessed with diverse quality attributes depending on requirements of the system or guidelines for critical applications. Additionally, multiple design alternatives are available with sometimes unpredictable impacts. Therefore, our tool consists of two coordinated and consecutive parts: impact assessment (Failure Impact Analysis (FIA) and Quantitative Impact Analysis (QIA)) and decision support (Countermeasure Decision Support Analysis (CDSA) and Service Interoperability Analysis (SIA)).

Our approach offers a holistic assessment cycle embedded in the design environment of IoT models. After identifying a flawed element the cycle of assessment analyses starts with FIA to identify impacts and provides the basis for QIA. QIA reviews the flawed path and estimates the financial impacts. Afterwards the assessment cycle switches to countermeasure decisions. CDSA provides an approach to model possible to-be alternatives and offers a comparison of these. Our assessment comes to a final design decision with SIA which checks interoperability of new services.

Application possibilities of our cycle are numerous, but if IoT comes into use in medical or wellbeing areas the severeness of vulnerabilities increases significantly. Architectural safety and security critical flaws of medical IoT networks are, e.g.:

- IoT devices with limited space and energy, as smart implanted defibrillators, are easy entry points for attacks if encryption is neglected.
- Devices with low trust level connected directly to the internet enable manipulation, e.g. smart pillboxes changing medication or ordering drugs.
- Low cost sensors are prone to errors which lead to wrong calculations of measurements, e.g. on insulin pumps, caused by external circumstances.
- Authentication can be a safety issue if life-saving functions are blocked by these methods. E.g. complicated authentication methods before enabling an ambulance call.

In the following sections we describe each approach in detail. To evaluate and discuss the usage in a consecutive and consistent way, we use the first named security flaw as a running example.

3.1 Failure Impact Analysis

The tool cycle starts with an architectural analysis of failure impacts (FIA). It aims in the recognition of safety and security impacts including their severeness and highlights required countermeasure points. FIA starts with vulnerable elements, received by flaw identification with patterns and anti-patterns. FIA reviews whole models to enable recognition of issues in even non-obvious parts. Later analyses only uses model parts labeled as endangered. (Holschke et al., 2008) monitored effects through usage of enterprise architecture models with linkage to BBN. Since BBN is mostly used to identify causes of events they focused on possible reasons of accidents or attacks through a top-down approach. We adapt the basics of this approach to change it to a bottom-up approach aiming in calculating likelihood and severeness of impacts of known events. The adaptation process included new layers, analysis components, steps and calculation basics to cover the new desired goals.

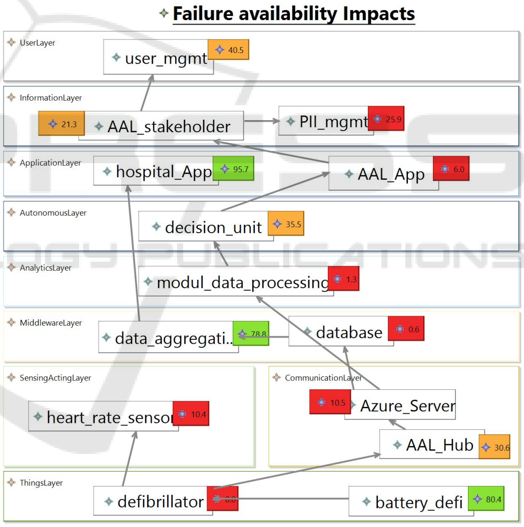

Figure 1: FIA Example.

1) Graphical Representation of IoT Systems: The representation of an IoT system with a model is needed to tag the flaw initially and to provide assessment data. It is based on a layered architecture and is depicted by a modeling editor inspired by ArchiMate Version 3.0.1 including IoT specific elements.

2) Selection of Assessment Attribute: Impacts depend on the chosen assessment attributes. Accordingly, different aspects are reviewed as relations differ dependently. Possible attributes are e.g. availability or integrity. Only quantifiable attributes are usable on QIA afterwards, e.g. availability can be quantified through numbers of Mean Time Between Failures.

3) DAG Mapping: To enable the calculations, the model has to be mapped into a Directed Acyclic Graph (DAG). DAGs have the same characteristics as a BBN, but are not limited to BBN typical goals. Hence, our tool creates a decreased model, only containing relations and elements concerning the assessment attribute. Directed relations present the assessment attribute dependencies. In addition, to make the DAG mapping suitable for IoT models special uncertainty nodes are introduced which face the possibility of new unknown devices. These have to be taken into consideration as they bring new impacts in the model.

4) Discretization of Nodes and Determination of Probability Tables: Every node in the DAG represents a variable with value of the assessment attribute. To enable the usage of probability tables, as required in BBNs, the values have to be discrete, e.g., discrete values to assess availability ”Up” and ”Down” time can be used, i.e. ”Up = 0.3” and ”Down = 0.7”. After discrete values are set a Probability Table (PT) and a Joint Probability Table (JPT) per each node are created by the tool. A PT represents the current state of the assessment value, whereas a JPT contains the calculated assessment value of the associated node and all ancestors. A final joint probability (FJP) of all nodes is calculated through multiplication of independent probabilities. Since the values are independent without an accident or attack happened, which cause impacts, the tables contain not conditional values.

5) Simulation of Event: After we defined the current state with our tool, we are able to simulate the identified possible accident or attack in the model. As the model elements are influenced from now on the presented BBN formula to calculate the new dependent values of nodes depending on its new ancestors values are used. Therefore, CPTs are calculated for every node of the DAG and JPTs respectively FJP are updated. The final sub-step is the identification of impacted nodes. Hence, our tool create differential matrices regarding the changes of assessment values and highlights the severeness of impacts. Nodes highlighted in red are endangered and needs further assessment respectively countermeasures.

To evaluate this part of our IoT tool we used the introduced case study. Figure 1 shows FIA results of a wellbeing IoT model. A flaw identification has recognized a vulnerable *heart_rate_sensor* in sensing and acting layer caused by lack of encryption possibilities. Hence, a DAG was created with all dependencies of *defibrillator* which the sensor is part of. The most important feature of a defibrillator is guaranteed functionality. Therefore, we chose availability as assessment attribute. After the discretization of all nodes, the event was simulated and all tables were calculated or updated. The differential matrices of each element show their current conditional availability highlighted depending on the scale. Elements with a low availability were highlighted red, whereas unharmed elements with still a high availability stayed green. E.g. *heart_rate_sensor* went down to 10.4%, whereas *battery_defi* remained high as it was not impacted. As multiple nodes are endangered and all layers were impacted further assessment is necessary.

3.2 Quantitative Impact Analysis

The financial aspect, mostly of security attacks, have to be analyzed as often countermeasure decisions depend on possible loss of attacks. QIA aims in quantitative assessment of the impacted path discovered in a previous conducted FIA. Therefore, QIA calculates costs of services or processes of the impacted DAG. Through quantity of events, costs per event and likelihood of occurrence a financial assessment will be calculated. QIA helps to estimate appropriate efforts to mitigate the impacts, e.g if the occurrence is low or the financial impacts are weak countermeasures would cause disproportionate expenses. Our approach is leaned on (Breu et al., 2008) and (Innerhofer-Oberperfler and Breu, 2006). They created a security enterprise model with a high-level business security goal and calculate the looses if the security goal fails. We adapted this approach not even to make it suitable for IoT but also to enable the assessment of financial impacts of safety events. To calculate our QIA goals requirements and weights of nodes or relations are transmitted from FIA results.

1) Impact Graph Generation: Elements which are not impacted of a flaw don’t have to be checked for financial effects. To minimize the elements that will be reviewed, the tool automatically create an impact graph containing only elements with a negatively differential matrix in FIA results. Hence, a flawed path through the IoT model is created which can be analyzed for possible costs of events.

2) Setting of Weights and Requirements: To enable calculation of costs safety or security requirements must be set to identify corresponding risk and severity. In addition, every leaf node has to be assigned a rate of occurrence (RO) depending on its risk. This can vary from past attacks, happened accidents to average downtime. Leaf nodes are nodes with direct known safety or security events which cause further issues. To weigh the graph the contained edges get assigned values through calculated probabilities of FIA’s CPTs. This represents the probability relation of two dependent nodes respectively risks.

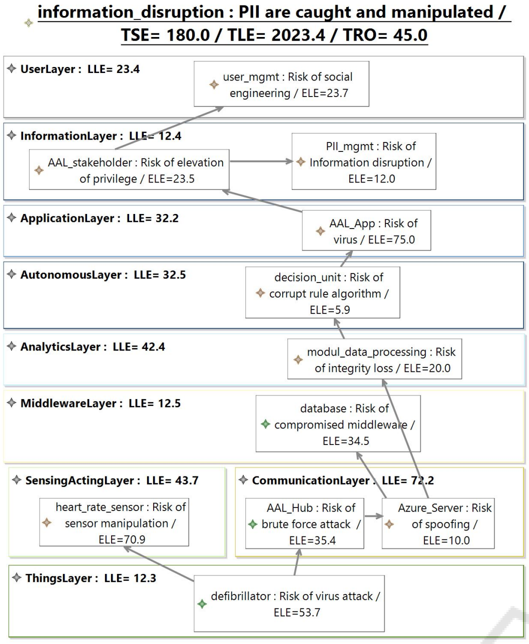
information_disruption : PII are caught and manipulated /
TSE= 180.0 / TLE= 2023.4 / TRO= 45.0
UserLayer : LLE= 23.4
user_mgmt : Risk of social engineering / ELE=23.7
InformationLayer : LLE= 12.4
AAL_stakeholder : Risk of elevation of privilege / ELE=23.5
PII_mgmt : Risk of Information disruption / ELE=12.0
ApplicationLayer : LLE= 32.2
AAL_App : Risk of virus / ELE=75.0
AutonomousLayer : LLE= 32.5
decision_unit : Risk of corrupt rule algorithm / ELE=5.9
AnalyticsLayer : LLE= 42.4
modul_data_processing : Risk of integrity loss / ELE=20.0
MiddlewareLayer : LLE= 12.5
database : Risk of compromised middleware / ELE=34.5
SensingActingLayer : LLE= 43.7
heart_rate_sensor : Risk of sensor manipulation / ELE=70.9
CommunicationLayer : LLE= 72.2
AAL_Hub : Risk of brute force attack / ELE=35.4
Azure_Server : Risk of spoofing / ELE=10.0
ThingsLayer : LLE= 12.3
defibrillator : Risk of virus attack / ELE=53.7

Figure 2: QIA Example.

3) Cost Calculations: The evaluation of costs can refer to different aspects in the IoT model. Our tool calculates costs to assess the total model, single layers and elements. In addition, a severity assessment of the whole model is estimated. The significance of financial loss depends on the use case and interpretation of results is decided individually. Following formula are used for the quantitative assessment:

$$ELE = RO * ASLE \tag{4}$$

$$LLE = LRO * ASLE \tag{5}$$

$$TLE = TRO * ASLE \tag{6}$$

$$TSE = Severity * TRO \tag{7}$$

First, the Element Lost Expectancy (ELE) of a single element is calculated by Average Single Loss Expectancy (ASLE) and RO of an element. Non leaf nodes calculate their RO through ROs and dependencies of ancestor nodes. ASLE can be defined through past events, hardware costs, costs per support unit or other quantitative loss rates. ELE compares impacts if single components are replaced. Second, our tool provides assessment of whole layers. Therefore, the Layered Rate of Occurrence (LRO), is used to define the Layered Loss Expectancy (LLE) which offers an evaluation of severely affected layers. Equation 6 provides the formula to calculate the Total Loss Expectancy (TLE) to assess total costs of the IoT networks. This is enabled through a Total Rate of Occurrence (TRO) which is determined through multiplication of single ROs and relations of nodes respectively propagated flaws. As a last option of QIA the Total Severity Expectancy (TSE) is provided. Hence, the severity factor of the system and TRO are used to offer a review of severity in case of an accident or attack. This option is especially important for safety hazards as the severity for human beings is crucial.

A subsequent QIA is shown in figure 2 which analyzed the graph generated from figure 1. Every impacted element presents its risk and calculated ELE value, e.g. *modul_data_processing* faces integrity loss with an ELE value of 12.0 caused by usage of compromised data. On the top, results of TSE (= 180.0), TLE (= 2023.4) and TRO (= 45.0) can be seen. The analysis result identified a possible information disruption which can lead to manipulated personal identifiable information (PII). This risk can lead to misuse of an implanted device or transmission of wrong data. For example, *heart_rate_sensor* which was the main flaw root, faces a sensor data manipulation through lack of encryption. This causes a lost expectancy of 70.9 and more costs in related elements like *AAL_Hub* or *database* since the corrupted sensors have to be replaced or a software update has to be enrolled.

3.3 Countermeasure Decision Support Analysis

After assessing diverse attributes our tool cycle offers two countermeasure analyses. First, a generic Countermeasure Decision Support Analysis (CDSA) conducts a comparison of current as-is models and possible to-be model design alternatives. EID is used for decision support including utility and chance nodes to evaluate the alternatives. (Sommestad et al., 2008) and (Johnson et al., 2007) are only a few of plenty approaches which already successfully used EID for security analysis. Therefore, we use EID to discover the best suited countermeasure scenario through calculation and comparison of utility nodes and regarding requirement compliances. Depending on available information or mitigation goals our tool offers a bottom-up and a top-down type of CDSA. If there is no predefined target value for utility or requirement compliance multiple countermeasure models are set up to determine the utility from bottom up. Calculated utilities can be used to compare alternatives. However, if there is a pre-defined target value which has to be reached a top-down approach is conducted. Since this case requires backwards calculation of countermeasures' CPTs the Bayesian Theorem is used to check if the scenario can reach the target values of a future model.

1) Definition of Utility Node: An utility node represent the main goal of the current analysis by checking for what is desired. Since our tool fo-

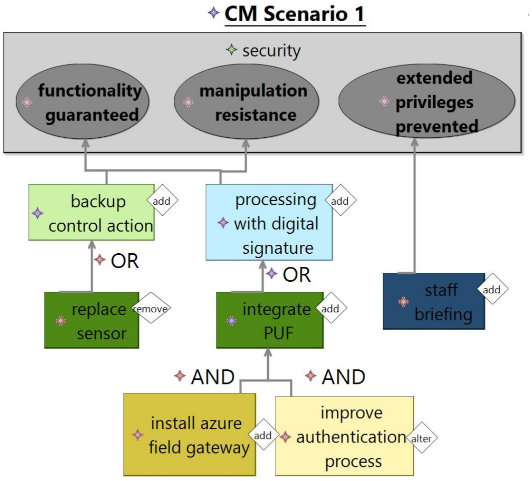

Figure 3: CDSA Example.

cuses on threats and hazards there are utility nodes for safety and security. As described an utility node can be set up with a pre-defined value which has to be reached. Utility nodes are depicted as rectangle located on the top of the model. In addition, a scenario must be set up which defines different domains of control.

2) Scenario Specifications: Chance nodes describe possible countermeasures with type of ”Alter”, ”Add” or ”Remove” and are connected through ”AND” or ”OR” relations depending whether all or just one countermeasure has to be fulfilled. Chance nodes are depicted through squares with colors regarding their concerning layer. Requirement chance nodes are located in the utility node representing direct requirements which lead to fulfillment of defined utility and are calculated through their connected chance nodes’ conditional probabilities.

3) Determine CPTs of Chance Nodes: Every countermeasure influences its associated requirements. To determine the conditional probability of their requirements’ impacts, their connected countermeasure ancestors respectively descendants have to be calculated. Therefore, every chance node has a PT, JPT and CPT for each requirement they influence. The fulfillment of requirements must be defined through discretized values, e.g. ”True” or ”False”. Again formula of BBN are used to calculate the tables depending whether they are dependent or independent specified by their connection type.

4) Calculations by Target Type: Depending whether a top-down or bottom-up approach is required our tool calculates the scenario status. If there is no utility target value JPTs of requirement nodes are determined by conditional probabilities of related nodes to analyze how the planned countermeasures influences the requirements and utility. In case of different importance of requirements they can be weighted to determine utility. In case an utility target value was set our tool analyses with aid of Bayesian Theorem and tables whether the planned countermeasures are able to reach this goal.

5) Design Decision Support: CDSA offers the possibility to compare diverse modeled and calculated countermeasure scenarios. For decision making the scenarios can compare the reached utility respectively which countermeasures come closest to the target value. In addition, countermeasures regarding a specific layer can be checked and single components can be monitored and changed as the tool automatically determine the impacts of changes.

Figure 3 gives an insight of a performing CDSA to mitigate the flaw of the known defibrillator risk. Since this challenge focuses on an attack *security* was set as utility node with three requirements: *functionality guaranteed*, *manipulation resistance* and *extended privileges prevented*. A set of multiple countermeasures in diverse layers are defined since the impacts and risks of FIA and QIA were located in several layers. For example, to guarantee functionality it is planned to replace the sensor and set up a backup control action to install a sensor with encryption possibilities and to ensure the control of captured elements. In addition, to mitigate the risk of attacks caused by extended privileges the staff will be briefed. Furthermore, field gateways, better authentication processes, PUFs and digital signatures are planned against manipulation. The next step in this analysis will be the calculation of scenario utility and comparison of other scenarios to chose the most suited for the use case.

3.4 Service Interoperability Analysis

Once countermeasures are chosen the detailed technical impacts of these have to be checked. Therefore, we developed a tool component for a Service Interoperability Analysis (SIA) to review new services of countermeasures and previous existing services to prevent new vulnerabilities caused by faulty cooperation or mismatching properties. If services are not able to work together flawlessly, especially in critical areas, the safety or security for human beings or systems can not be guaranteed. (Ullberg et al., 2008) have already proven to use EIDs to review service interoperability in models by checking the quality of services. Since they mainly focused on run time interoperability our tool adapted the definitions to make it suitable for IoT design assessment. Design time interoperability consists of three subcategories: Single services, pairs of services and power set of services:

- Single services: Quality Assessment of independent service blocks.
- Pairs of services: Pair-wise comparison of interoperability, e.g. data exchange.
- Power set of services: Weighted assessment of all quality of service attributes must be included.

Quality of services is received through the assessment attributes *availability*, *correctness*, *verifiability* and *communication compatibility* of all containing services. IoT is mainly based on services of different types regarding autonomous, connected things. Hence, our tool provides a possibility to model special service nodes connected with chance nodes to enable the assessment of IoT specific challenges.

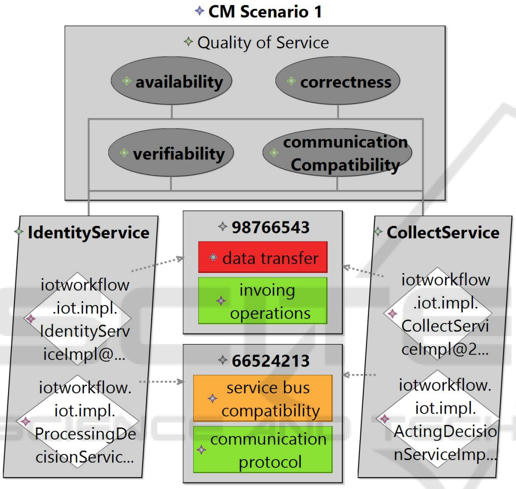
CM Scenario 1
Quality of Service
availability
correctness
verifiability
communication Compatibility
IdentityService
iotworkflow .iot.impl. IdentityServ iceImpl@...
iotworkflow. iot.impl. ProcessingDe cisionServic...
98766543
data transfer
invoing operations
66524213
service bus compatibility
communication protocol
CollectService
iotworkflow .iot.impl. CollectServi ceImpl@2...
iotworkflow. iot.impl. ActingDecisio nServiceImp...

Figure 4: SIA Example.

1) Set Up Original Service Model: Services which are part of the as-is IoT model before applying countermeasures have to be transformed into a SIA model to enable comparison of new services afterwards. As SIA models are based on EIDs they contain an utility node to assess the quality of scenario. Hence, in SIA models the utility node represent the power set of services that depends on the described four assessment attributes which are represented by chance nodes. Under utility and chance nodes the service nodes are located. They are clustered regarding their service type which are connected to chance nodes to determine their probable interoperability. Therefore, each service has four assigned PTs for every assessment attribute with discrete probability values of fulfillment, e.g. "Communication compatibility - True = 0.8".

2) Define Countermeasure Service Model: Countermeasures can add new services, replace previous ones or alter some service details. Hence, the defined original service model has to be adapted into countermeasure scenarios with new service information. In addition, weights of assessment attributes must be updated and new PTs must be specified.

3) Determine Design Time Interoperability: As soon as a service is added, changed or removed the tool automatically determines the new design time interoperability status. First, associated PTs of single services use independent probability formula to calculate the quality values of assessment attributes respectively chance nodes. Second, our tool creates the final utility value and evaluate the power set of services with quality of service weights. As a last step, pairs of service interoperability is determined. Hence, our tool pair up every possible combination of service pairs, depending on their connections, and create automatically interoperability interface matrices to compare multiple features to evaluate their cooperation possibilities. These features can range from used protocols, provided operations to service bus compatibility. The matrices are highlighted with a scale regarding their results.

4) Design Decision Support: After the original service model and possible countermeasure service scenarios are depicted and analyzed the assigned design time interoperability can be evaluated to support the design decision. As described three subcategories have to be considered to compare the flawed original scenario with mitigation opportunities. We compare all SIA models on finale utility values, single service value changes and provides a report of interface matrices to estimate the status of service pairs. If quality of service values have increased by countermeasures the new defined services are suitable. SIA completes the tool cycle to assess and redesign a flawed IoT model to mitigate vulnerabilities in the design phase.

Our running example conducts a SIA to check for interoperability of new planned countermeasure services. Figure 4 views a small excerpt of the corresponding countermeasure scenario of figure 3. Two collect services and two identity services of sensor respectively database access processes are contained and compared in their pairs of service matrices for characteristics like *communication protocol* or *invoking operations*. An incompatibility of *data transfer* was identified between a sensor collect service and a database identity service as different data transfer methods are planned. In addition, the *service bus compatibility* of another service pair must examined more closely. The SIA model calculates next the quality of service of power set which enables the comparison with other SIA models. Afterwards a final countermeasure design decision can be made and the identified flaw can be prevented.

Our case study performed the whole assessment cycle to assess and mitigate a flaw. FIA identified the direct and indirect impacts of a vulnerable defibrillator and the weak points which come along in diverse layers. Since different angles should be observed as well, we calculated the financial expenses of possible attacks which enables the estimation of needed resources. To mitigate the impacts we evaluated security requirements and quality of services to find the most suitable countermeasures which are not causing other issues. The to-be scenario alternative with the highest increase of assessment attribute values will be chosen to be the new flawless as-is model.

4 CONCLUSION AND FURTHER WORK

In this paper we discussed the need of architecture analyses in the medical IoT safety and security management. As not every IoT component can be updated harmful design decision have to be detected in the design phase. However, the detection is not enough. The consequences of a flaw have to be assessed and prevented to be able to chose the required countermeasures. Hence, we presented our holistic IoT tool approach to support assessment of safety or security related design flaws in IoT models and the following evaluation of possible countermeasure options. We explained the connections and workflow of included analyses and their steps in detail. Our approach contains two analyses to identify and assess possible impacts of an accident or attack to estimate technical and quantitative consequences for single components respectively the whole system. After getting an overview of the meaning of a flaw, our approach provides an analysis to design and compare countermeasure scenarios through the calculation of conditional probability of single countermeasures and their impact. As a last component our workflow supports the countermeasure decision with an analysis addressing new services which come along with countermeasures to check the interoperability. Through these approach we were able to develop a holistic safety and security architecture analysis approach for IoT systems to increase the handling of complex systems to create save and secure IoT environments for human beings.

ACKNOWLEDGMENT

Electronic Component and Systems for European Leadership (ECSEL) supported the development of this approach within the project CPS4EU (Grant Agreement Number 826276).

REFERENCES

Breu, R., Innerhofer-Oberperfler, F., and Yautsiukhin, A. (2008). Quantitative Assessment of Enterprise Security System. *2008 3rd ARES*, pages 921–928.

Holschke, O., Närman, P., and Flores, W. R. (2008). Using Enterprise Architecture Models and Bayesian Belief Networks for Failure Impact Analysis. *ICSOC*, pages 339–350.

Innerhofer-Oberperfler, F. and Breu, R. (2006). Using an Enterprise Architecture for IT Risk Management. *ISSA*, pages 1–12.

iscoop (2017). Internet of Things – the Complete Online Guide to the IoT. *https://www.i-scoop.eu/internet-of-things-guide/*. Accessed on: 25.08.2020.

Johnson, P., Lagerström, R., Närman, P., and Simonsson, M. (2006). Extended Influence Diagrams for Enterprise Architecture Analysis. *EDOC*, pages 3–12.

Johnson, P., Lagerström, R., Närman, P., and Simonsson, M. (2007). Enterprise Architecture Analysis with Extended Influence Diagrams. *Information Systems Frontiers*, 9(2-3):163–180.

Lee, W.-T. and Law, P.-J. (2017). A Case Study in applying Security Design Patterns for IoT- Software System. In *ICASI*, pages 1162–1165. IEEE.

Lohmüller, P., Rauscher, J., and Bauer, B. (2019). Failure and Change Impact Analysis for Safety-Critical Systems: Applied on a Medical Use Case. *BMSD*.

Neapolitan, R. E. et al. (2004). *Learning Bayesian Networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ.

Paloalto Networks (2020). IoT Threat Report. https://bit.ly/2H0kecc. Accessed on 25.08.2020.

Rauscher, J. and Bauer, B. (2020). Design Optimization of IoT Models: Structured Safety and Security Flaw Identification. In *BMSD*, pages 84–102. Springer.

Rauscher, J., Langermeier, M., and Bauer, B. (2016). Characteristics of Enterprise Architecture Analyses. *BMSD*, pages 104–113.

SecurityToday (2020). The IoT Rundown For 2020: Stats, Risks, and Solutions. https://bit.ly/33i6rFg. Accessed on 25.08.2020.

Sommestad, T., Ekstedt, M., and Johnson, P. (2008). Combining Defense Graphs and Enterprise Architecture Models for Security Analysis. *EDOC*, pages 349–355.

Ullberg, J., Lagerström, R., and Johnson, P. (2008). A Framework for Service Interoperability Analysis using Enterprise Architecture Models. *SCC*, 2:99–107.

Wortman, P. A., Tehranipoor, F., Karimian, N., and Chandy, J. A. (2017). Proposing a Modeling Framework for Minimizing Security Vulnerabilities in IoT Systems in the Healthcare Domain. *ITAB*, pages 185–188.