e-Informatica Software Engineering Journal, Volume 14, Issue 1, 2020, pages: 77–95, DOI 10.37190/e-Inf200103

# SIoT Framework: Towards an Approach for Early Identification of Security Requirements for Internet-of-things Applications

Ronald Jabangwe*, Anh Nguyen-Duc**

* *The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Software Engineering, Denmark / Software Improvement Group, SIG Nordics.*

** *School of Business, University of South Eastern Norway, Norway, Department of Business and IT*

rja@mmmi.sdu.dk / r.jabangwe@sig.eu, anh.nguyen.duc@usn.no

**Abstract**

Background: Security has become more of a concern with the wide deployment of Internet-of-things (IoT) devices. The importance of addressing security risks early in the development lifecycle before pushing to market cannot be over emphasized. Aim: To this end, we propose a conceptual framework to help with identifying security concerns early in the product development lifecycle for Internet-of-things, that we refer to as SIoT (Security for Internet-of-Things). Method: The framework adopts well known security engineering approaches and best practices, and systematically builds on existing research work on IoT architecture. Results: Practitioners at a Norwegian start-up company evaluated the framework and found it useful as a foundation for addressing critical security concerns for IoT applications early in the development lifecycle. The output from using the framework can be a checklist that can be used as input during security requirements engineering activities for IoT applications. Conclusions: However, security is a multi-faced concept; therefore, users of the SIoT framework should not view the framework as a panacea to all security threats. The framework may need to be refined in the future, particularly to improve its completeness to cover various IoT contexts.

**Keywords:** security requirement; Internet-of-things; Software Engineering; Requirement Engineering; Security Framework

## 1. Introduction

Within the past decade, we have witnessed the rapid growth of commercial systems that deeply integrate software, hardware and the contextual environment. The most notable are Internet-of-Things (IoT), Industry 4.0, cyber-physical systems, and smart wearable devices. The number of (IoT) devices being introduced in the market has been increasing drastically with the number of connected devices approaching 15 billion [1]. This trend is expected to continue, with an estimate of 26 billion network connected devices by the year 2020 [1].

Security has become even more important as the number of "things" connected increases through the vulnerable internet and other networks. The border between software and hardware parts is less visible when it comes to providing customer value. Internet-of-things integrate both sensors, connectivity infrastructure and processors with a software platform. The consideration of security, therefore, needs to be in a holistic view that combines both software and hardware parts of the system. Security issues are not new and have been a concern for years to manufacturers. However, security in software-intensive products is often neglected or treated as an afterthought. Business pressure, time-to-market and reduction of development costs are among factors that drive the treatment of security as an add-on feature.

Submitted: 20 November 2019; Revised: 30 March 2020; Accepted: 31 March 2020; Available online: 28 April 2020

Software Engineering (SE) researchers are looking for a way to address security concerns as early as possible in the development and operation of software-intensive products [2, 3]. In our study, we refer to "security concerns" for a given system as vulnerabilities, risks or threats that can negatively impact the security properties of the system, specifically, confidentiality, integrity and availability. The aim is to promote security-by-design, which leads to having a proactive rather than a reactive approach for addressing security. The goal, which is also the same for threat modeling [4, 5], is to help with identifying security concerns for a given system. These security concerns can potentially be mapped to security requirements, which in-turn can help with designing secure systems.

Security requirements affect all aspects of the design, development, deployment, and maintenance of complex systems that provide customer value. To address security early in the development cycle, security aspects should be considered from the planning and requirements phase, and throughout all the other phases. In response to the urgent need to deal with security in software-intensive product development [6–8], we aim at proposing a comprehensive approach to identify security issues in the context of cyber-physical systems, specifically focusing on Internet-of-things. More importantly, the approach will handle data security issues as an input for both product development and operation. Last but not least, the approach should be lightweight and easy to adopt in various sizes of organizations, particularly start-up companies. This is due to the emerging number of Internet-of-things developed by start-ups.

A plethora of research work on software engineering exists in relation to software security and requirements engineering, and there is also a growing interest in Internet-of-things. Internet-of-things development is challenging due to the multiple cross-cutting concerns, such as connectivity, security and the lack of high-level abstractions to address both the large-scale and heterogeneity [9]. The heterogeneousness of Internet-of-things introduces additional complexity to software layer development, in particularly complex data flow and architectural cross-cutting concerns [6, 10]. Consequently, securing the Internet-of-things application development would require joint knowledge from data security, requirements engineering and Internet-of-things architecture. Security should be addressed early in the development process by ensuring that requirements are clearly defined that when implemented, prevent or mitigate security issues. It is noted that we do not aim at generating specific security requirements through our framework. As a preliminary result from a qualitative survey of quality concerns and practices in Internet-of-things startups, we identified the need for early consideration of security requirements and mapping them into actual implementation. Addressing the issues early avoids costly rework late in the development process. To this end, we take a software engineering approach for addressing data security concerns early through a lightweight framework, SIoT (Security for Internet-of-things Applications). For Internet-of-things end-users, this can reduce safety risks and potentially improves privacy and data protection.

Designing secure systems requires understanding the complex interaction between different parts of architecture and the security threats for those parts. The SIoT framework, which takes a layered view of the architecture of Internet-of-things applications, provides a foundation for promoting that thought process. The aim is not to generate specific security requirements through our framework. However, the output from using the framework can be a checklist that can be used to help with identifying security requirements for IoT applications.

The remainder of the paper is organized as follows: Section 2 introduces basic understanding about Internet-of-things products and security identification approaches. Section 3 describes the need for a lightweight and early-stage framework for Internet-of-things development via a preliminary industrial survey. Section 4 describes our framework. Section 4 presents the case company for which the framework was developed and would be evaluated. The discussion and conclusion are in Section 5.

## 2. Background and related work

### 2.1. Existing IoT frameworks

The framework that is more similar to our framework is work of Meridji et al. [11]. The framework is intended to help developers identifying, specifying and measuring security requirements. The design of the overall framework is based on the use of the interdependency graphs (SIG) and the CIA triad, i.e., confidentiality integrity, availability, confidentiality. Whilst the proposed framework was systematically developed, it is, however based on generic models and generic view of security. As a result, the framework takes a broad and generic view of system engineering. In contrast, our framework is intended for a specific type of system, i.e., IoT systems. Another aspect that differs between our framework and the framework proposed by Meridji et al. [11] is in how security aspects are derived. The framework by Meridji et al. [11] relies on three international standards (ECSS, IEEE and ISO) for deriving security requirements. Whereas our framework emphasizes the need to focus on the architecture design in order to derive relevant security concerns for the specific system. Our motivation for going with this approach is that the architecture may differ from system to system, and how a system is designed is crucial for understanding how best to strengthen the security of the overall system. Because our framework focuses more on the architecture of IoT systems, it allows for more flexibility in terms of adoptability and adaptability to various IoT systems. Ammar et el. [12] report on a survey of existing IoT platforms that offer cloud-based services such as AWS IoT. In the report they make an assessment of eight platforms focusing on the features offered by the platforms for developing IoT applications, including hardware and security features. Our framework takes a software engineering and process approach for developing IoT applications. The overarching aim is to provide developers with an approach that can help them with identifying security concerns of their specific IoT applications, irrespective of the platform that they use.

### 2.2. Security requirement identification in software development

Security requirements have traditionally been considered to be non-functional or quality requirements [2, 13]. Like other non-functional requirements, security requirements need to be described in the way that they can be implemented later. Carnegie Mellon University was among the first to propose a methodology (SQUARE) to help organizations build security into the early stages of the production life cycle [14]. The SQUARE approach includes nine steps that require formal participation of requirements engineers and other stakeholders of an IT project. The team starts with outlining security goals, threats identification and risk assessment based on a full understanding of the relevant system. After that, the team decides on the best method for eliciting initial security requirements from stakeholders, and to elicit an initial set of security requirements. In the final step, security requirements are inspected to ensure consistency and accuracy. However, the methodology is at a high level of abstraction and is not specific to a particular domain.

Several researchers have focused on tools and methods for identification of security requirements, for instance, misuse cases [15], goal and anti-goal analysis [16], and patterns of security goals [17]. These approaches are proposed regardless of the context of software development and operation. Security concerns should be considered not only in the early stage of product development, but also as a continuous integral element of product development. Despite the benefits that Agile software development promises, there are security challenges faced within the paradigm that can manifest into vulnerable software products [18]. In turn, this can significantly impact the longevity of the software product on the market. There are studies that adopt and adapt agile approaches in order to ensure that security initiatives are addressed, e.g., Beznosov's work [19] and Ghani's work [20]. However, it is also critical to have a framework that is specific to Internet-of-things contexts that not only helps address security concerns but also can be adopted into agile software development processes.

### 2.3. Non-functional requirement modelling

Modelling and documentation techniques that can also be used when implementing approaches for collecting, categorizing and prioritizing security requirements are attack trees, abuse cases, abuser stories, misuse cases and fault trees [21]. An attack tree is a tree-like representation of the different ways that an identified asset can be attacked based on attack goals. Abuse cases are descriptions of how a user of a system or the system can be attacked or abused. Abuser stories help capture and describe likely goals of an attacker. Unlike user stories that are written from the perspective of a user of a system, abuser stories are written from the perspective of an attacker. Misuse cases are based on use cases, but they describe, using, for example, UML use case diagrams, and how malicious activities can be carried out on the system. A fault tree is a deduction approach for analyzing system failures and security concerns using graphical Boolean logic. These approaches can also be used to support the SQUARE method or any similar approaches. Nevertheless, modelling of security requirements is out of the scope of this paper. We only focus on providing a framework to help with the process of identifying security requirements in Internet-of-things applications.

### 2.4. IoT product development

From a technological perspective, the implementation of Internet-of-things typically requires the combination of hardware, software and middleware components collaborating with each other [22]. Hardware used for Internet-of-things include sensors, actuators, and processors that can be added to existing core hardware components, and integrated to manage and operate the functionality of connected things. Communication protocols such as MQTT, AMQP, XMPP, and Zigbee enable the communication between the sensor devices and the cloud [23]. A typical Internet-of-things product will have a "cloud" part including an application platform that provides fundamental operating environments for Internet-of-things applications. Internet-of-things applications, which employs web or mobile interfaces, provide functionalities to store, process and analyze a vast amount of time series-based machine data. There exist various architectural views on Internet-of-things systems depending on research goals. Based on existing classifications [22, 24], we adopted a 4-layer view on Internet-of-things systems with the purpose of differentiating security concerns and resolution techniques among layers. The layers are Application tier, Network tier, Sensor tier, and Data processing tier, which will be described in the SIoT framework (Section 4).

Internet-of-things development is challenging due to the multiple cross-cutting concerns, such as connectivity, security and the lack of high-level abstractions to address both the large-scale and heterogeneity [9]. Patel et al. proposed a development framework that separates Internet-of-things into four concerns: architecture, domain, platform and deployment concerns [9]. However, the authors do not explicitly explore the elicitation and implementation of security concerns.

### 2.5. Identification and modelling security requirements in IoT development

There are other research articles that address security requirements for IoT applications [25–29]. Babar et al. proposed a framework that separates security concerns for software and hardware parts of embedded systems [25]. Although the need for built-in security framework is emphasized, the model was not validated. Jacobsson et al. proposed a risk analysis for smart home systems based on architectural views [26]. Gan et al. suggested several security requirements for Internet-of-things in their analysis [27]. Ahmad et al. proposed a model to capture security and privacy properties in Internet-of-things [28]. They point out common security challenges, but they are not categorized into system architectural dimensions. Kim et al. discussed the security concerns according to system tiers and system development phases [29]. The proposal is however subjective without validation.

Apart from the industry evaluation, our framework differs in that it adopts the well known

SQUARE methodology [14], and best practices for security engineering (e.g., [5, 32]) and adapts them for Internet-of-things contexts

## 3. Industrial demand on a security modelling framework

In order to motivate the need of a security modelling framework for industry, in this section, we provide preliminary results of a qualitative survey that we are currently performing on Internet-of--things startups. Early results support the notion as they suggest that startups need assistance with a framework for identifying security concerns early in the software development process. The preliminary results are shown in Table 1. In the on-going survey, we are surveying the state-of-practice of Internet-of-things application development, focusing specifically on exploring Internet-of-things development practices among IT startups. We aim at collecting as many participants as possible. There is no limitation on the type of companies in our survey as we would like to have a variety of the sample. The survey was designed in 2015 and is an on-going effort. Participants are being searched from three channels (1) our professional network, (2) regional incubators and accelerator programs, (3) and startup portal, i.e., Startup Norway and Crunchbase. Participants who accept our invitation are also invited to participate in a one-hour interview.

We used semi-structured interviews to enable open-end answers from participants. Our interview process has four parts (1) background information about business and product, (2) prototyping and production development practices and challenges (3) quality concerns and testing, and (4) final reflection. The full interview question list is shown in Appendix A. We found eleven Internet-of-things startups that are relevant to the scope of this study. This is a subset from our IoT survey, from which we can extract information about security requirements. It is possible that the other companies also have similar concerns, but this is not explicitly mentioned in the survey.

Our previous work reveals some of the prototyping and development challenges of such startups, i.e., insufficient testing, technical debt, balancing agility and quality, etc. [33, 34]. Table 1 summarizes how security concerns are considered and managed in the eleven Internet-of--things startups from our survey. In the table, "foundation year" is the year the company was officially formed. Regardless of the startups' active time, a startup can be in an idealization, a prototyping or in a production state [35]. Agile approaches are clearly common across the cases. Many companies report that they adopt certain ways of Agile in developing (part of) their products. Some report waterfall and adhoc approaches as their preferred approaches when dealing with the production of the hardware parts. In the right-most column in Table 1 we also reveal as a part of the product quality assurance practices, how security concerns were considered and managed.

Table 1 reveals that 80% of our cases emphasize the importance of security for their business, regardless of the startup stage. Security goals are often established in both startups at prototyping and production phases. Security is considered a significant concern, and in some cases as an essential value proposition in the companies' business models. The consideration occurs at different levels, organizational, managerial and technical levels, for instance:

> Security is the main infrastructure of Internet-of-things applications. As it is everywhere and one cannot think of compromising the everyday equipment they use (C03).

Table 1. Security goals

| Goals | Description |
|---|---|
| Confidentiality | Ensure that data is not revealed to or accessed by unauthorized individuals [30, 31] |
| Availability | Ensure that authorized users can access and use data on demand [30, 31] |
| Integrity | Prevent unauthorized tampering of data when it is being processed, or in transit or when at rest [30, 31] |

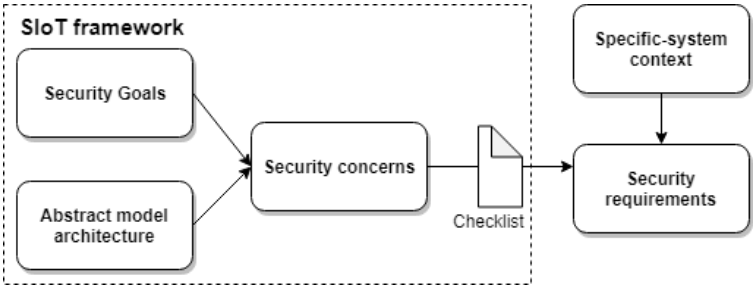


Figure 1. The scope of SIoT framework

As shown in Figure 1, the SIoT framework consists of the following three main components:

1. Security goals.
2. Internet-of-things abstract model architecture.
3. Internet-of-things security concerns.

Devices in Internet-of-things applications communicate through Internet and share their data over the network. So, there are huge chances of vulnerability (C05).

However, there is a limited action on implementing security concerns. 30% of the surveyed companies did not implement any security-related features. 40% of the cases have their security dependent on external vendors or open source components. There are only three cases that implement security as a part of their competitive strategy, as illustrated below:

Our data can be traced and exposed over the network in case of lack of proper security measures. Security features play a key role in Internet-of-things applications (C05).

The two pillars of simple access and security must work in unison. The first provides a simple way to securely connect devices to the Wi-Fi network, while the second ensures only the IoT application can traverse the Wi-Fi network from the IoT device to the server. This can help prevent malicious attacks (C07).

There is a lack of clarity of a systematic approach for mapping security goals to actual actions for addressing security concerns. For instance, it is not clear how startups cooperate security concerns into the product architecture, at different times of consideration (i.e., prototyping or production). We also recognize some startups (40% of the total number of cases) that perceive security as a dependent concern on open source community or third-party providers. All in all, the preliminary observations of the 10 Internet-of-things startups suggest a methodological need of a systematic approach for considering security during Internet-of-things product development life cycle and practices.

Overall, we found only one case in which the company was taking steps to implement a methodological approach for security assurance. The company representatives explained that they do it because of market demands in their domain. Hence it helps them gain a competitive advantage over its competitors. The company was willing to participate in the evaluation of the framework because of their interest in methods for effectively addressing security concerns. More details of the company are presented in Section 5 of the paper.

## 4. The proposed conceptual SIoT framework

The SIoT (Security for Internet-of-things Applications) framework adopts the well known SQUARE methodology [14], and best practices for security engineering (e.g., [5, 32] and adapts them for Internet-of-things contexts. The framework also builds on existing work on Internet-of-things applications [22, 24, 36–38].

### 4.1. Security goals

Maintaining data confidentiality, integrity, availability are the primary goals for data security initiatives [39–41]. The three goals are also re-

Table 2. Internet-of-things application and Security consideration in startups

| ID | Found year | Startup stage | Startup product | Development method | Thoughts | Actions |
|---|---|---|---|---|---|---|
| C01 | 2009 | Prototyping | A underwater camera | Adhoc | Security was not considered at this stage | Quality perceived at open source module |
| C02 | 2013 | Prototyping | A tracking device for shipments | Agile | Quality consideration i.e., robustness and security at software tier | Outsourced: Quality testing was done by subcontractors |
| C03 | 2011 | Production | A mobile muscle trainer | Agile | After thought on security at software tier | Outsourced: Quality testing of the hardware tier was outsourced |
| C04 | 2015 | Production | connected smart home solution | Waterfall and Agile | Importance of security at software and cloud tier | Depending on security of third party modules |
| C05 | 2015 | Production | Home electricity usage management system | Agile for IoTs related development | Security concerns at three components: circuits, mobile apps and cloud | Implementing security features at various tiers |
| C06 | 2016 | Prototyping | A navigating device for visually impaired individuals | Agile | Security is the most prominent feature | No |
| C07 | 2016 | Prototyping | A car remote controller | Agile at start, Waterfall afterward | Security as a main concern | Implementing security features at various tiers |
| C08 | 2014 | Production | A predictive analytic platform for vehicles | Agile | Security is as important as usability. | Limited |
| C09 | 2012 | Prototyping | A body index tracking | Agile | Security concerns at methodological, organizational and technical level | Experimenting at methodological level |
| C10 | 2015 | Prototyping | A water farming management system | Adhoc | Security concerns at organizational level | No |
| C11 | 2013 | Prototyping | Glucose monitoring device | Adhoc | Security concerns at organizational and technical level | No |

ferred to as the CIA triad [40]. The definitions for the security goals are in Table 2. Our proposal is to break down security into the three goals and then identifying security concerns that need to be addressed in order to realize each goal. This approach will help with addressing security from different but critical perspectives for protecting data for Internet-of-things applications.

There are other security attributes, such as authorization, authentication, and non-repudiation that can be perceived as independent categories. However, in line with Bass et al. [41], we also believe these attributes support the security goals outlined in Table 2. For example, authorization is intended to ensure that access to data is based on user privileges. This can be traced to confidentiality. Authentication is about verifying users to ensure confidentiality and integrity. Non repudiation can be traced to confidentiality and integrity as it relates to ensuring that users do not deny accessing, editing or deleting data.

### 4.2. Internet-of-things abstract model architecture

Decomposing the architecture of Internet-of--things applications into distinct layers provides an overview of the idiosyncrasies associated with the systems. This helps to better understand how to tackle data security concerns of such complex systems. Based on existing classification Internet-of-things architecture, for example in [22, 24, 36–38], we have identified the following abstract layers as being the foundation of an Internet-of-things system:

- Application tier [36, 38] This layer provides users access to the Internet-of-things through, for example, a mobile device. The control of the application and intelligent decision-making is performed through this tier. It provides the typical functions of the whole system, including the APIs to consumers, decision-making, task analysis, task schedule and so on. In this tier, a number of services are deployed and interact with each other.
- Network tier [36–38] This layer is responsible for data transmission. The transmission can be through, for example, a local area network or a mobile cellular data network. Hassanalieragh et al. refer to this layer as data transmission [36].
- Sensor tier [36–38]: This layer is responsible for collecting data from an object of interest through sensors. The data can come from a human-being, environment, or any object of interest. Basically, the function of this layer is to provide environment or situational awareness. It is mainly achieved by sensors that may or may not perform a preliminary data pre-processing, which then transmit the data, through the network layer, to the application and eventually to the support layer. WSN (Wireless Sensor Network) is one of the basic techniques of this sensor tier. This layer can also be referred to as the data acquisition layer [36], perceptual layer [38], and sensation layer [37].
- Data processing tier [36]: This layer consists of the computational devices and storage devices, that provide heterogeneous data processing such as normalization, noise reduction, data storage and other similar functions. This tier is the bridge between Producer and Service. Hassanalieragh et al. refer to this layer as the data cloud processing layer [36].

### 4.3. Identification of security concerns

#### 4.3.1. General Internet-of-things security considerations

Techniques to compromise data security keep evolving just as fast as the countermeasures to address them do. Thus ensuring data confidentiality, integrity and availability is challenging for Internet-of-things systems. The basic security needs that should be taken into consideration in each of the layers are listed in Tables 3–6. The checklist provided in the tables, which includes particular vulnerable areas for each tier, will help to ensure that security requirements are formulated to address security from different angles using the CIA triad. It also helps with capturing well known issues that can compromise data security, for example, eavesdropping and

unauthorized gathering of data, as well as causing data availability issues through distributed denial-of-service attacks [22, 42].

#### 4.3.2. Domain-specific Internet-of-things security needs

The checklist listed in Tables 3–6 are measures that should be taken into consideration to ensure data security. However, it is important to note that the checklist is not a comprehensive list. This is because depending on the configuration of the Internet-of-things the architecture tier, and the types of security risks can differ across contexts. For example, distributed denial-of-service is a common security issue across networks. But because not all networks are based on the same communication protocol, it is important to assess each type of network that is used in the Internet--of-things application for any additional relevant threats. For this reason, a context-specific security modeling approach is needed. The application tier involves integrated or individual specific application business, such as smart grid, intelligent transportation, smart security, smart home, wearable devices, and smart city. There are certain security concerns that cannot be solved in other tiers of Internet-of-things, such as privacy protection issue, which does not occur in sensor layer and network layer but can become a concern

Table 3. Security Concerns for the Application tier

| Goals Checklist | Description |
|---|---|
| Confidentiality | Access control for authorized users (i.e., user authentication) |
| | Authorized user roles types |
| | Least privilege (least functionality) for each user role/type |
| | Verification of authorized users |
| | Third party data and service integration |
| Availability | Access to the system on demand by authorized user |
| | Access to data on demand by authorized users |
| | Data input validation |
| Integrity | Verification of data source |
| | Audit trail of user access into the system |
| | Audit trail of user access to data |
| | Audit trail of changes to data |
| | Principle of separation of duties |
| | Attribution of user access to application |

Table 4. Security Concerns for the Network tier

| Goals Checklist | Description |
|---|---|
| Confidentiality/ | Access control for authorized users (i.e., user authentication) |
| Integrity | Authorized user roles/types |
| | Least privilege (least functionality) for each user role/type |
| | Verification of authorized users |
| | Third party data and service integration |
| Availability | Prevent distributed denial of service attack |
| | Maintain connectivity |
| | Detection and prevention of common network attacks (e.g., denial of service) |
| Integrity | Verification of data source |
| | Audit trail of user access into the system |
| | Audit trail of user access to data |
| | Audit trail of changes to data |
| | Principle of separation of duties |
| | Attribution of user access to application |

Table 5. Security Concerns for the Sensor tier

| Goals Checklist | Description |
|---|---|
| Confidentiality | Data anonymization |
| | Encryption algorithm and protocol when data is in transit from sensors |
| | Access control (user and device authentication) |
| | Encryption key management |
| | Node and device authentication RFID protocol security |
| Availability | Access to data on demand by authorized users and devices |
| | Detection and prevention of common sensor attacks (e.g., denial of service) |
| | Audit of data collected |
| | Audit of data sources |

Table 6. Security Concerns for the Data tier

| Goals Checklist | Description |
|---|---|
| Confidentiality | Encryption method when data is in transit from sensors |
| | Access control (user and device authentication) |
| | Encryption key management |
| Availability | Access to data on demand by authorized users and devices Malware and virus detection |
| | Data recovery mechanism (in case of disaster or failure) |
| | Mitigation strategy for disaster and data recovery |
| Integrity | Verification of data source |
| | Audit trail of user access to data |
| | Audit trail of changes to data |

in certain contexts of the application layer. Moreover, different applications might have a different priority on security requirements. For example, data privacy would be of great importance for Transportation and Healthcare sector, but, on the other hand, data authenticity may be more important for a Smart city.

In addition, in regulated domains, there are security requirements that need to be implemented for data protection. A good example is medical device software, which needs to comply with specific data protection guidelines. In the United States of America, for example, medical device software needs to implement security measures outlined in the Security Rule that is in the Health Insurance Portability and Accountability Act (HIPAA) [43]. Therefore, it is also essential to consider the domain in which the Internet-of-things application will be used and understand the unique data security demands and regulations.

#### 4.3.3. Assessment of security concerns

Identifying and assessing security threats can be performed by following a threat modeling approach [32]. We propose following the approach shown in Figure 2, which adopts well known threat modeling techniques in security engineering [5] that are used as a basis for deriving security requirements [4, 32].

The approach shown in Figure 2 considers the context of the system, the likelihood of threats occurring and the impact that they have on the system. This is essential for an effective threat modeling approach. The decomposition of the Internet-of-things application in Section 4.3.2 can be used as Step 1 in Figure 3. Step 2 is a critical step because in order to identify relevant security concerns it is important to understand the complexities and mechanisms used to collect, transmit and store data. Table 7 shows information that should be defined during Step 2

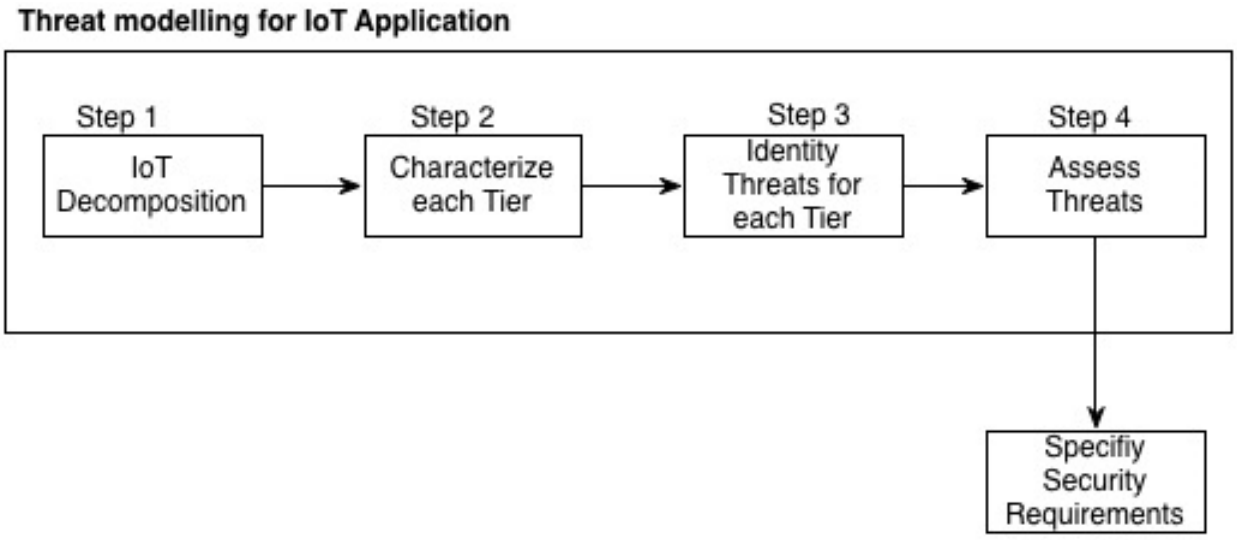


Figure 2. Threat modelling for Internet-of-things application

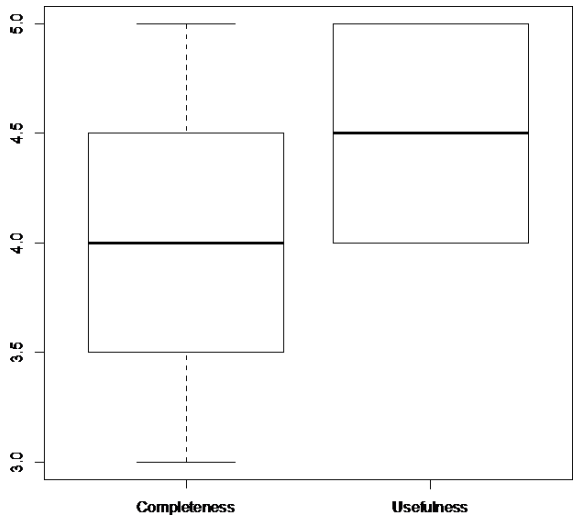


Figure 3. The evaluation of the SIoT Framework at ABC company

in order to characterize each tier and describe contextual factors that are unique to a particular Internet-of-things application. The aim is to help identify the contextual setting of product development. Data flow diagrams can also help model and understand how data flows through each tier, which is useful information for understanding data security concerns in the Internet-of-things application.

The characterization done in Step 2 should then be used as input in Step 3, which involves activities of identifying threats within each tier. Assessment of the relevance to the Internet-of-things application being analyzed will then be done in Step 4. The threats that are found to be relevant can also be assessed on their likelihood of occurrence and severity or extent of negative impact on data security. The assessment can be used for prioritization during the implementation of security requirements.

## 5. Evaluation of the SIoT framework

### 5.1. The company context

To evaluate the SIoT framework, we apply it in a company that we will refer to as ABC to preserve its anonymity. ABC is a spin-off startup from an international enterprise that provides real-time industrial IT integration, automation and manufacturing solutions. ABC has approximately 20 employees, developing a system for estimating glucose (blood sugar) based on the combination of several non-invasive measurement principles. The company adopts several engineering methodologies:

– Process-driven development: the company was approved according to ISO 13485 (quality management system for the development and manufacturing of medical devices). Product development involves a significant amount of documentation for internal use and external communication. The company adopts

Table 7. Security Concerns for the Data tier

| Goals Checklist | Description |
|---|---|
| Application Tier | Types/roles of authorized users |
| | Number of authorized users |
| | Criticality of data that can be accessed (e.g., private and sensitive data) |
| | Devices used (e.g., hand-held mobile devices and laptops) |
| Sensor tier | Number of authorized users to connect to sensors |
| | Number of authorized devices to connect to sensors |
| | Criticality of data that can collected (e.g., private and sensitive data) |
| | Data transmission method from the sensor layer to the application layer (wireless personal networks [44], e.g., Bluetooth and Zigbee [45] |
| Network tier | Data transmission method from the application layer to the data processing layer, e.g., mobile cellular network, wireless local area networks [44] |
| | Data transmission method from the data processing layer to the application layer, e.g., mobile cellular network, Wireless Fidelity (i.e., WiFi). |
| | Communication protocols |
| Data processing tier | Use of local device storage system |
| | Cloud storage model (e.g., the use of either private, public, community or hybrid cloud [46] |

a tailored version of Agile with long-term iterations. A lot of physical tests are performed on hardware components, e.g., strap test, temperature test and load test. Automated testing and continuous integration are done for software components.

– Quality-driven development: the developed product is classified as a Class IIa Medical Device product, which highlights the criticality of several quality attributes, such as performance, safety and most importantly, security.
– Software platform development: product development involves the implementation of an embedded platform using C++/ RTOS, Java, noSQL and secured REST-API.

ABC was used to evaluate the SIoT framework as a follow-up research activity after the industrial survey (case C11 in the survey in Section 3). Security has been recognized as a vital quality attribute at ABC. The company also expressed the need for a framework for addressing the security concerns of their product.

With the permission of the CTO we formed a focus group consisting of developers from ABC to evaluate the SIoT framework. The focus group aimed at evaluating the SIoT framework on its usefulness in practice, and to assess if SIoT can help identify any additional security requirements apart from those that they already knew and had documented. During the evaluation, the focus group used the security requirements for the current glucose estimator prototype. Some of the security concerns that the company was keen on addressing are data confidentiality and integrity.

The glucose estimator device will be body mounted in the form of a wearable device that communicates with a mobile app for displaying and monitoring data. The system is similar to the Internet-of-things health monitoring system presented by Hassanalieragh et al. that is also based on WBAN and cloud-based processing [36]. The prototyping development was finished in the winter of 2017 and the product is currently under European regulator evaluation. The development team includes five people with competences in electronic engineering, software engineering and medical expertise. The prototyping process started in Spring 2015. The

development approach is research-based with long iteration. All R & D activities occurred in-house.

### 5.2. Focus group and the evaluation process

A focus group is a popular research method in social science, that involves carefully planned discussions to obtain the perceptions of group members on a defined topic [47]. Typically, there are 3 to 10 participants in a focus group, that are facilitated by a moderator, who guides a structured discussion on a specific topic. The approach has been used in requirements engineering to elicit and to analyze requirements [48]. In our case, we aim at discovering the security concerns of the current prototypes by using the SIoT framework as a proxy object. We followed the focus group meeting guidelines proposed by Edmunds [49], specifically:

- Defining the research problem: the group aims at evaluating the current security concerns for the prototype.
- Selecting participants: we include all stakeholders who are involved in the development of the prototype. In total four engineers participated in the focus group
- Planning and conducting the focus group: we hold a 120-minute discussion with participants. Each session started with an overview of the objectives of the study and with a discussion on how participants should discuss and act during the session. The first topic was to discuss the current security level of the prototype. A researcher, who acted as the moderator, went through the four checklists of security requirements. The second topic was to discuss the usefulness and completeness of our SIoT. Each participant would give evaluation scores for the completeness and usefulness of SIoT at the end of the activity.
- Analysis: the discussion was noted and summarized into points. Each point was mapped to (1) requirements that are already implemented in the current prototype and (2) requirements should be implemented in the next version of the prototype.

SIoT helped the focus group identify security requirements that they had missed when developing the prototype. The results of the focus group were summarized in Table 8.

Table 8. Security requirements in ABC: "had" vs "should have" lists

| Goals | Requirements Implemented | Requirements to be implemented |
|---|---|---|
| Application Tier | | |
| Having: Confidentiality, Availability<br>Should have: Confidentiality | Access control for authorized users (i.e., user authentication)<br>Access to the system on demand by authorized user<br>Data input validation | Authorized user roles/types<br>Least privilege (least functionality) for each user role/type<br>Verification of authorized users<br>Third party data and service integration |
| Sensor Tier | | |
| Having: Confidentiality<br>Should have: Confidentiality, Availability | Encryption algorithm and protocol when data is in transit from sensors | Node and device authentication<br>RFID protocol security<br>Access to data on demand by authorized users and devices |
| Network Tier | | |
| Having: Confidentiality<br>Should have: Confidentiality, Availability | Identity authentication | N/A |
| Data processing tiers | | |
| Having: Confidentiality<br>Should have: Integrity | Access control (user and device authentication) | Verification of data source |

### 5.3. Evaluation results and lessons learned

Figure 3 presents the boxplot (mean, min, max and outliners) of the evaluation scores from focus group participants. As mentioned previously, each participant gave a score for the completeness of the framework (if SioT covers all relevant aspects to them) and the usefulness of the framework (if SIoT helps them in identifying security requirements). The scores range from one to five, with five being the highest degree. Figure 3 shows that participants perceived SIoT positively and very useful. They appreciate the framework in facilitating the discussion on security concerns for the product as well as helping with identifying security requirements for future releases.

Nevertheless, the participants pointed out three main issues when using the framework:

- Requirements of multiple expertise: it is remarked that the framework goes beyond a singular tier of Internet-of-things system, hence, the adoption of the framework needs to involve software developers, hardware engineers, business analysts, etc. to reflect the comprehensive set of requirements.
- Abstraction of some security concerns: for instance, common attacking approaches or detection of a virus are elements that locate in a high abstract level. These elements are not directly transferred into implementable requirements. Further discussions would be required to identify specific security requirements.
- Domain-specific focus: the evaluation was done on a prototype from the healthcare domain. In such an application domain with a lot of regulations, there are specific demands on adhering to security standards and laws. While the framework is useful as a starting point to help address critical security concerns, domain-specific concerns linked to regulations and standards are not straightforward.

## 6. Discussions

In this section, we discuss and summarize the SIoT framework (Section 6.1), and also discuss the application of SIoT in iterative development (Section 6.2) and opportunities for improvement (Section 6.3).

### 6.1. Comprehensive vs. domain specific security consideration

Our SIoT framework looks at security concerns from the product architecture perspective. We emphasize the security concerns of the entire system, rather than just the security of a single piece of software or a single Internet-of-things layer. This incorporates the fact that Internet-of-things application includes multiple tiers of software, middleware and hardware. A multi-perspective view has also been considered in existing models [25, 27]. Our model has been revised and validated by practitioners. Nevertheless, the real-life scenario of Internet-of-things applications could become an ecosystem, in which security is not only considered at a technical level, but also at organizational level and ecosystem level. Therefore, one might consider in future work, a comprehensive framework that captures security concerns at both technical, organizational and ecosystem levels.

The SIoT framework consists of three main components (See Section 4). We propose that the first step in applying the framework should be the "security goals", which results in the characterization of data security for the system. This is necessary for performing actions within the other two components, i.e., "IoT abstract model" and "IoT security considerations". These other two components can be performed in parallel, and their output is a list of security needs that are then translated into security requirements. The security requirements can be documented, for example, using natural language [50].

The consensus within the focus group during the evaluation in the industry was that SIoT is useful and helps address security from different key angles of Internet-of-things systems. As the focus group participants pointed out the framework goes beyond a singular tier of Internet-of--things system, hence promoting the notion of having a holistic view of security for the system. Developing and marking secure Internet-of--things systems requires that not only software engineers are security-aware but all stakeholders

within the company. There are hardware concerns and market considerations. Therefore, in order to get a comprehensive set of security requirements, we encourage that the adoption of the SIoT framework should involve software and hardware engineers, testers, business analysts, architects, and any other personnel that is involved directly or indirectly in the development of the Internet-of-things system.

Overall SIoT addresses data security from multiple perspectives in order to ensure that essential security requirements are identified. This includes, taking into account the idiosyncrasies of the Internet-of-things application as well as the regulatory requirements of the domain in which it will be used. This provides a more comprehensive view of security concerns for the Internet-of-things application. However, there will be some overlap in terms of security requirements in particular during the activities for the Internet-of-things security considerations. Thus, the security requirements should be aggregated, analysed, and duplicates should be removed.

### 6.2. Fitting SIoT into iterative development

The proposed SIoT framework can be used in an iteration/ sprint plan to ensure security concerns are addressed and captured by the security requirements [51]. The output from the framework can then be used as a starting point to develop metaphors or user stories which then could be added to the backlog. Schwaber provides a list of questions to guide a retrospective meeting [51]. We propose to add the following to the list in order to bring the topic of security into the meetings: (1) which security concerns have been addressed, (2) which security concerns might have been missed in the backlog, and (3) which security concerns need to be added to the backlog. In addition, an agile approach to continuous changes and integration may introduce security vulnerabilities, which results in the development of insecure software [18]. Hence, we suggest adopting an agile and continuous assessment of security requirements to ensure that they are appropriate and in line with the best practices for addressing the most current and sophisticated security risks that are relevant to the Internet-of-things system. This can be facilitated by adopting threat modeling within the development process, specifically during the requirements and design phases [18]. Furthermore, having a role within the process dedicated to overseeing software security related aspects should also be considered, for example, Ghani et al. suggest adding a "security master" role in Extreme Programming (XP) [20].

Security risks and threats will keep evolving as mitigation strategies and new technologies emerge. Therefore, it is crucial to keep monitoring and managing mechanisms that are intended to protect data. Therefore, a continuous approach of updating security requirements is paramount for ensuring that appropriate risks are taken into account as the software evolves. This can be done by adopting an iterative continuous process for addressing security requirements as captured in Figure 4. Security goals need to be identified at the beginning of the project, i.e., aligning with business strategy. The first iteration (Sprint 1) will identify the architectural model at the abstract level, following by the identification of general security concerns. Consequent iterations refine the architectural model and re-evaluate the list of security concerns. At some point in time (Sprint N), the domain-specific security concerns are identified and reflected in the architectural model.

### 6.3. Limitation of the framework

The framework provides a conceptual approach that can be applied in any company context. The security consideration (described in Table 1) presents the need for such a framework. The completeness and usefulness of the framework is evaluated using quantitative forms. However, the evaluation of the framework is preliminary and only bases on a focus group. We are aware of the limitation and plan for future work with thorough validation of the model. Our main future plan is to continue evaluating the framework more cases industry, particularly in various domains. At the moment we mitigate this issue by carefully ensuring that SIoT is based on existing best practices for security engineering,

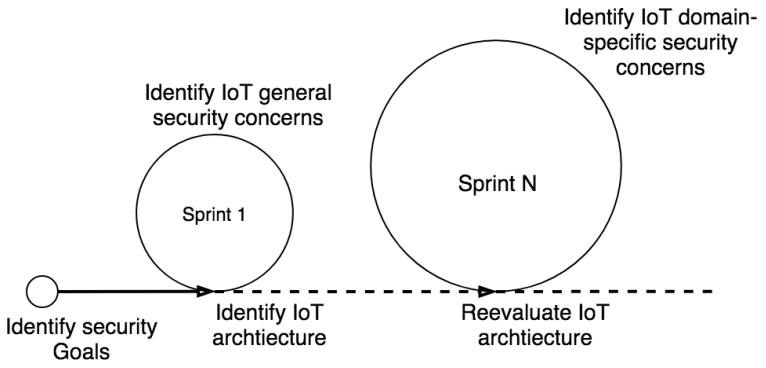


Figure 4. Continuously considering security concerns

e.g., [5, 14, 32] and research on Internet-of-things security, e.g., [16, 22, 24–29, 36–38].

While SIoT was perceived positively with regards to its usefulness, there was an issue regarding its completeness. This was pointed out by practitioners in the focus group meeting. This pertains to addressing security outlined for regulatory environments, demanded by law or specified in technical standards such as those in a certain domain like the medical device industry. This is a critical issue to note. Therefore, users of the framework should also reflect on any other requirements demanded in their own regulated environment. Possible future work is to provide a guideline for mapping the SIoT framework to specific regulatory requirements in a specific domain.

## 7. Conclusions

In this paper, we proposed a systematic engineering approach for identifying security requirements in Internet-of-things systems. The gaps in requirements and system design were found in Internet-of-thing startups. Considering an abstract architecture of an Internet-of-things application, we are able to come up with a SIoT framework, that offers a systematic way to identify, maintain and to evaluate security aspects in Internet-of-thing applications. The framework has been used in a Norwegian startup with initial positive feedback. However, it is worth mentioning that Internet-of-things security issues are application specific, so the approach needs to be adapted in a specific application domain, which might introduce specific architectural elements. Hence, this SIoT framework may need to be refined in the future.

Our case company suggested that the framework provides a good basis to help address critical security concerns for Internet-of-things applications. However, security is a multi-faced concept; therefore users of the framework should not view the framework as a panacea to all security threats. In addition, security threats will keep evolving as technology evolves. Therefore, there is a need to update SIoT accordingly over time, as well as to conduct further validation with practitioners and improving the framework based on feedback. Furthermore, we suggest complementing the framework by adopting supporting security activities, e.g., continuous security considerations (e.g., shown in Figure 4), penetration testing and keeping up-to-date with emerging security threats from resources like OWASP foundation[1]. Designing secure systems requires understanding the complex interaction between different parts of architecture and the security threats for those parts. The SIoT framework, which takes a layered view of the architecture of Internet-of-things applications, provides a foundation for promoting that thought process.

## Acknowledgments

We appreciate Prof. Tor Stalhane from NTNU and Dr. Indira Nurdiani (University of Southern Denmark) for their constructive review and feedback on the SIoT framework.

[1]The OWASP foundation can be found at this link: www.owasp.org

## References


[1] S. Lucero, "IoT platforms: Enabling the Internet of Things," IHS Technology, Whitepaper, 2016. [Online]. https://www.esparkinfo.com/wp-content/uploads/2018/11/enabling-IOT.pdf

[2] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, International Series in Software Engineering. Springer, 2000. [Online]. https://www.springer.com/gp/book/9780792386667

[3] A. Olmsted, "Secure software development through non-functional requirements modeling," in *International Conference on Information Society (i-Society)*, 2016, pp. 22–27.

[4] S. Myagmar, A.J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Proceedings of the IEEE Symposium on Requirements Engineering for Information Security*, 2005.

[5] F. Swiderski and W. Snyder, *Threat Modeling*. Microsoft Press, 2004.

[6] A.N. Duc, R. Jabangwe, P. Paul, and P. Abrahamsson, "Security challenges in IoT development: A software engineering perspective," in *Proceedings of the XP2017 Scientific Workshops*, XP '17. ACM, 2017, pp. 11:1–11:5.

[7] A.S. Sani, D. Yuan, J. Jin, L. Gao, S. Yu, and Z.Y. Dong, "Cyber security framework for internet of things-based energy internet," *Future Generation Computer Systems*, Vol. 93, No. 4, 2019, pp. 849–859.

[8] I. Jacobson, I. Spence, and P.W. Ng, "Is there a single method for the internet of things?" *Queue*, Vol. 60, No. 11, 2017.

[9] P. Patel and D. Cassou, "Enabling high-level application development for the Internet of Things," *Journal of Systems and Software*, Vol. 103, 2015, pp. 62–84.

[10] B. Morin, N. Harrand, and F. Fleurey, "Model-based software engineering to tame the IoT jungle," *IEEE Software*, Vol. 34, No. 1, 2017, pp. 30–36.

[11] K. Meridji, K.T. Al-Sarayreh, A. Abran, and S. Trudel, "System security requirements: A framework for early identification, specification and measurement of related software requirements," *Computer Standards and Interfaces*, Vol. 66, 2019, p. 103346.

[12] M. Ammar, G. Russello, and B. Crispo, "Internet of things: A survey on the security of IoT frameworks," *Journal of Information Security and Applications*, Vol. 38, 2018, pp. 8–27. [Online]. http://www.sciencedirect.com/science/article/pii/S2214212617302934

[13] P. Devanbu and S. Stubblebine, "Software engineering for security: A roadmap," in *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, 2000. [Online]. https://www.researchgate.net/publication/2393383_Software_Engineering_for_Security_a_Roadmap

[14] N. Mead, "Security quality requirements engineering (SQUARE)," Software Engineering Institute, Tech. Rep., 2011.

[15] G. Sindre and A.L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, Vol. 10, No. 1, 2005, pp. 34–44.

[16] A. van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in *Proceedings. 26th International Conference on Software Engineering*, 2004, pp. 148–157.

[17] Y. Yu, H. Kaiya, H. Washizaki, Y. Xiong, Z. Hu, and N. Yoshioka, "Enforcing a security pattern in stakeholder goal models," in *Proceedings of the 4th ACM workshop on Quality of protection*, 2008, pp. 9–14.

[18] S.H. Adelyar and A. Norta, "Towards a secure agile software development process," in *10th International Conference on the Quality of Information and Communications Technology (QUATIC)*, 2016, pp. 101–106.

[19] K. Beznosov, "eXtreme security engineering: On employing XP practices to achieve "good enough security" without defining it," in *First ACM Workshop on Business Driven Security Engineering (BizSec)*, 2005.

[20] I. Ghani and N.I.A. Firdaus, "Role-based extreme programming (XP) for secure software development," in *Special Issue – Agile Symposium*, 2013.

[21] M.R.R. Ramesh and A. Tadepalligudem, "A survey on security requirement elicitation methods: classification, merits and demerits," *International Journal of Applied Engineering Research*, 2016.

[22] Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: Perspectives and challenges," *Wireless Networks*, 2014.

[23] F. Wortmann and K. Fluchter, "Internet of Things," *Business and Information Systems Engineering*, Vol. 57, No. 3, 2015, pp. 221–224.

[24] H.J. La and S.D. Kim, "A service-based approach to designing cyber physical systems," in *IEEE/ACIS 9th International Conference on Computer and Information Science*, 2010, pp. 895–900.

[25] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, "Proposed embedded security framework for internet of things (IoT)," in *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, 2011, pp. 1–5.

[26] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," *Future Generation Computer Systems*, Vol. 56, 2016, pp. 719–733.

[27] G. Gan, Z. Lu, and J. Jiang, "Internet of things security analysis," in *International Conference on Internet Technology and Applications*, 2011, pp. 1–4.

[28] A.W. Atamli and A. Martin, "Threat-based security analysis for the internet of things," in *International Workshop on Secure Internet of Things*, 2014, pp. 35–43.

[29] D.H. Kim, J.Y. Cho, S. Kim, and J. Lim, *A Study of Developing Security Requirements for Internet of Things (IoT)*, 2015. [Online]. https://www.semanticscholar.org/paper/A-Study-of-Developing-Security-Requirements-for-of-Kim-Cho/

[30] R.L. Kissel, Ed., *Glossary of Key Information Security Terms*. National Institute of Standards and Technology, 2013. [Online]. https://www.nist.gov/publications/glossary-key-information-security-terms-1

[31] G. Stoneburner, "Underlying technical models for information technology security," National Institute of Standards and Technology, Tech. Rep. 800-33, 2001.

[32] A. Shostack, *Threat Modeling: Designing for Security*. Wiley, 2014.

[33] A. Nguyen Duc, K. Khalid, T. Lønnestad, S. Bajwa Shahid, X. Wang, and P. Abrahamsson, "How do startups develop internet-of-things systems – A multiple exploratory case study," in *IEEE/ACM International Conference on Software and System Processes (ICSSP)*, 2019, pp. 74–83.

[34] A. Nguyen-Duc, X. Weng, and P. Abrahamsson, "A preliminary study of agility in business and production: Cases of early-stage hardware startups," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '18. ACM, 2018, pp. 51:1–51:4.

[35] A. Nguyen-Duc, S.M.A. Shah, and P. Ambrahamsson, "Towards an early stage software startups evolution model," in *42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2016, pp. 120–127.

[36] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *IEEE International Conference on Services Computing*, 2015, pp. 285–292.

[37] X. Sun and C. Wang, "The research of security technology in the internet of things," in *Advances in Computer Science, Intelligent System and Environment*, Advances in Intelligent and Soft Computing, D. Jin and S. Lin, Eds. Springer, 2011, pp. 113–119.

[38] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: A review," in *International Conference on Computer Science and Electronics Engineering*, Vol. 3, 2012, pp. 648–651.

[39] National Institute of Standards and Technology, "Standards for security categorization of federal information and information systems," U.S. Department of Commerce, Tech. Rep. Federal Information Processing Standard (FIPS) 199, 2004.

[40] F.Y. Sattarova and T.H. Kim, "IT security review: Privacy, protection, access control, assurance and system security," *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 2, No. 2, 2007, pp. 17–31.

[41] L. Bass, P. Clements, and R. Kazman, *Software architecture in practice*. Addison-Wesley, 2003.

[42] D. Fischer, B. Markscheffel, S. Frosch, and D. Buettner, "A survey of threats and security measures for data transmission over GSM/ UMTS networks," in *International Conference for Internet Technology and Secured Transactions*, 2012, pp. 477–482.

[43] M. Scholl, K. Stine, J. Hash, P. Bowen, L. Johnson, C. Smith, and D. Steinberg, "An introductory resource guide for implementing the health insurance portability and accountability act (HIPAA) security rule," National Institute of Standards and Technology, Tech. Rep. 800-66, 2008. [Online]. https://csrc.nist.gov/publications/detail/sp/800-66/rev-1/final

[44] K. Scarfone, D. Dicoi, M. Sexton, K. Scarfone, D. Dicoi, M. Sexton, C. Tibbs, and C.M. Gutierrez, "Guide to securing legacy IEEE 802.11 wireless networks recommendations of the national," NIST, Tech. Rep. 800-48 Rev 1, 2008.

[45] D. Gislason, *Zigbee Wireless Networking*. Newnes, 2008.

[46] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep. 800-145, 2011.

[47] S. Caplan, "Using focus group methodology for ergonomic design," *Ergonomics*, Vol. 33, No. 5, 1990, pp. 527–533.

[48] K. Garmer, J. Ylven, and M. Karlsson, "User participation in requirements elicitation comparing focus group interviews and usability tests for eliciting usability requirements for medical equipment: A case study," *International Journal of Industrial Ergonomics*, Vol. 33, No. 2, 2004, pp. 85–98. [Online]. http://www.sciencedirect.com/science/article/pii/S0169814103001318

[49] H. Edmunds, *Focus Group Research Handbook*. McGraw-Hill, 2000.

[50] P. Salini and S. Kanmani, "Survey and analysis on security requirements engineering," *Computers and Electricale Engineering*, Vol. 38, No. 6, 2012, pp. 1785–1797. [Online]. http://www.sciencedirect.com/science/article/pii/S0045790612001644

[51] M. Sliger, *Agile project management with Scrum*. Project Management Institute, 2011.

## Appendix A.

- **Part 1: General information**
  Q1a. Describe your product
  Q1b. Describe your company, i.e history, current head count
  Q1c. What are the key software development methods, processes, environments and tools?
- **Part 2: Production development practices**
  Q2a. How did you build the first prototype?
  Q2b. What were the reasons behind the first prototype?
  Q2c. How did you make other prototypes?
  Q2d. What have you learnt from the prototyping process?
  Q2e. When the actual development started?
  Q2f. How does the final product different from prototypes?
  Q2g. Please name three most important challenges during product development
  Q2h. How many significant pivots have you encountered?
- **Part 3: Quality concerns and testing**
  Q3a. What are quality attributes important for your products? Q3b. How do you manage your product quality? Q3c. How do you do testing? Q3d. When did you last refactor your codebase? Q3e. How do you consider Security in your final product?
- **Part 4 – final reflection**
  Q4. Any final interesting comment ?