
Martin Höst¹, Jonathan Sönnerup², Martin Hell², Thomas Olsson³
¹Dept. of Computer Science, Lund University, Sweden, martin.host@cs.lth.se
²Dept. of Electrical and Information Technology, Lund University, Sweden, (jonathan.sonnerup, martin.hell)@eit.lth.se
³RISE SICS AB, Lund, Sweden, thomas.olsson@ri.se

Abstract—The area of Internet of Things (IoT) is growing and it affects a large amount of users, which means that security is important. Many parts of IoT systems are built with Open Source Software, for which security vulnerabilities are available. It is important to update the software when vulnerabilities are detected, but it is unclear to what extent this is done in industry today. This study presents an investigation of industrial companies in the area of IoT to understand current procedures and challenges with respect to security updates. The research is conducted as an interview study with qualitative data analysis. It is found that few companies have formalized processes for this type of security updates, and there is a need to support both producers and integrators of IoT components.

Keywords: IoT, security, vulnerability, open source

Contribution type: Regular Research Paper

1. Introduction

The area of Internet of Things (IoT) is becoming a “disruptive innovation that will radically change business processes within and across sectors” [3]. In the current process of transforming traditional product offerings to service offerings, the need for IoT services will also be important. IoT systems are important in a wide spectrum of domains, such as in the medical domain, in “smart homes”, in supervision and control of industrial processes, meaning that many people are, or will be, dependent on the services of IoT systems. These systems and their components often include private data and sensitive information, and can be used to control and monitor critical infrastructure, as well as to protect physical assets, meaning that security is a crucial quality attribute.

Embedded software systems represent an important part of IoT systems. These embedded systems typically include open source software (OSS) components. For OSS components, vulnerabilities are continuously published in publicly accessible databases. For many IoT systems this means that there are known security vulnerabilities, and “exploits”, i.e., descriptions of how to use the vulnerabilities in attacks.

For most products, it is important to update them after they have been taken in use. However, it is unclear to what extent security updates are carried out in a structured way. Since many of the systems are new, there is a risk that too much of the focus is on demonstrating innovative features rather than handling security pro-actively. Motivated by this, the presented research investigates how companies developing IoT system monitor published OSS vulnerabilities, how they decide which updates to make, and what main observations can be made based on this.

2. Background

An IoT system typically consists of a perception layer with physical entities, a network layer together with a middleware layer, an application layer, and a business layer [7]. Solinen and Taivalsaari [12] present an architecture of an IoT system as involving sensors, gateways, cloud solutions, and interfaces. That is, a typical IoT system includes sensors and actuators, gateways to transport data, data in the cloud, and user interfaces e.g. connected through the cloud.

Development of IoT systems typically involves integration of sub-systems into complete systems, as any system development. This involves integration of both components from other companies and OSS components, OSS components can be included as-is, or companies may be part of the OSS community and involved in the development [5]. That is, there is a chain of producers and consumers of components/system, i.e., forming a “value chain” of different company types. In this value chain component developers develop individual components, such as sensors and gateways, and integrators integrate components and develop sub-systems or complete systems that can be used by users. There are also supporting companies, not involved in the development of a specific system, but supporting the development. Examples are consultancy companies with specific competence in IoT and development tool developers.

When securing IoT, there are many aspects that need to be taken into account. Access control to the device must be handled to avoid unauthorized access. If access, or administration through a web interface is offered, web attacks such as SQL injections and cross site scripting (XSS) must
be protected against. If the device itself must access remote services, it must store credentials, private keys, or handle trusted certificates securely. The network in itself must be encrypted and authenticated in order to avoid man-in-the-middle, or replay attacks. A challenge with security is that even with robust security controls in place, and following best practice, newly discovered software vulnerabilities can render a device insecure and easily exploitable. It is thus important to implement methods that efficiently identify and evaluate the need for patching when new vulnerabilities are disclosed. There are documented examples of malware using known software vulnerabilities to infect IoT devices [14].

New software vulnerabilities are reported in a large variety of databases. Large multi-vendor databases for vulnerabilities include, e.g., Bugtraq and NVD. Bugtraq is a computer security mailing list, while NVD, maintained by NIST, provides a more structured way of describing a certain vulnerability. This includes, e.g., criticality given as a CVSS score, which software and versions are affected, and links to further information. In addition to the multi-vendor databases, some software vendors maintain their own vulnerability information. While the vulnerabilities often overlap between databases, different information is also published in different sources [8], [10]. This makes it difficult to accurately assess the impact of a certain vulnerability. Using responsible disclosure, the vulnerability is typically assigned a new CVE identifier (Common Vulnerabilities and Exposures), but the vendor is given some time to develop and issue a patch for the vulnerability before it is made public. There are efforts that aim to standardize different aspects of identifying and evaluating vulnerabilities. NIST maintains SCAP [16], which is a protocol for automatically managing vulnerabilities, and measure and evaluate systems and their configuration. Using SCAP will help monitoring systems and their configuration in order to quickly respond to new vulnerabilities. It consists of several components, e.g., the Common Platform Enumeration (CPE) for assigning names to IT products, Common Configuration Enumeration (CCE) for describing system configuration and Open Vulnerability and Assessment Language (OVAL), which is a language for assessing the state of a machine and report results of assessment. Also, CVE and CVSS as described above are components of SCAP. In addition to vulnerability information, detailed information on exploits can be found in ExploitDB, while Metasploit provides a framework for easily using developed exploits. The presence of exploits is a factor to consider when evaluating vulnerabilities.

3. Related research

The degree to which security is seen as important by practitioners is different in different domains, as described by [2]. They found that legacy systems enhanced by IoT solutions were often highly critical for society, which could slow down the process of transforming them to an IoT architecture, and they also found that system availability in general is more important than confidentiality of data. They also saw that practitioners value security differently in different domains. For example, the electricity sector is more “conservative” and there is a tradition of maintaining safety. However, based on the study, it can be concluded that security is important for many IoT services and products.

A recent systematic mapping study by Mohammed et al. [9] indicates that there is an overweight in research on how to handle security issues in the development phase as opposed to other parts of the process, especially the operational phase. To some extent, the research presented in this paper addresses this by investigating the state of practice for IoT companies, and by studying several development phases.

Telang and Wattal [15] analyzed the impact of security announcements on the firm stock price. They suggest that even though the direct liability for a pure software company is lower than that of, for example, a medical equipment company that typically have clearer liability clauses in their contracts, the software companies suffer more in terms of stock price. Furthermore, they state that there is a relationship between the announcement of a vulnerability and a patch to address it. The longer it takes, the more negative is the impact on the stock price. In this study, we interview companies in the IoT domain to understand their challenges in identifying vulnerabilities and addressing them. Telang and Wattel point out that smaller companies have a larger negative impact on stock price. In IoT in general and for the companies in this study, many are small and hence this emphasizes the need for fast rapid in IoT.

Software companies are getting better and faster in addressing security vulnerabilities. However, hackers are also getting faster [13]. On a similar study, Arora et al. [1] observe that the patch time is faster once a vulnerability is announced. Furthermore, open source providers are quicker in addressing vulnerabilities and having an authority such as CERT increases the likelihood that software providers address the vulnerabilities. We complement these studies by analyzing a number of case companies in detail and how they use e.g., vulnerability databases such as CVE.

Hafiz and Fang [4] report that there is a preference to public announcement of vulnerabilities rather than reporting them privately to the software providers, at least when it comes to experienced reporters. However, they also find that software providers are more likely to address a vulnerability if it is reported directly to them. This indicates that the software providers struggle to analyze public vulnerability databases and need strategies and tools for this. In our study, we focus on lead-times in general for the case companies and based on internal and private information, as opposed to Hafiz and Fang who rely on publicly available information [4].
4. Research Methodology

The following research questions have been investigated in the research

1) How are vulnerabilities for OSS identified and evaluated in companies developing IoT systems?
2) When a company has decided to update to address a vulnerability, how is the update work planned and synchronized with other development and maintenance work for the product?

The research was carried out as an interview-based survey with companies in the area of IoT, i.e., a qualitative survey (e.g. [6], [17]). This type of survey is, as opposed to a traditional quantitative survey, based on qualitative analysis and not on statistical significance. That is, the sampling in the survey is based on diversity, purpose, and saturation, instead of probability-based sampling. The analysis relies on coding and finding patterns instead of counting frequencies and determining statistical tests. Qualitative surveys are appropriate for studying diversity, opinions, relations, etc., which was the reason it was chosen for the research objective of this paper.

The research was conducted in three main phases:
1) In the first phase of the research, the overall goal was decided, see above.
2) In the second phase, empirical data was collected through interviews in two rounds.
3) When data had been collected, an analysis phase could start.

The phases were not simply conducted in sequence but instead iterated. For example, the formulated objective was somewhat adjusted based on the findings in the interviews, and the interview questions were adjusted based on the findings from analysis, especially between the two interview rounds. Below the phases are further described.

Interview questions were formulated based on the research questions. There were questions about the company and their business, about how they identify and evaluate vulnerabilities, and about how they update and distribute new software versions.

The interviews were conducted in a semi-structured way (e.g. [11]), in two interview rounds, round 1 and round 2. This means that the researcher adhered to the list of interview questions during the interview, but the questions were not given to the interviewee literally and exactly in the order as they were listed. Every interview lasted for about one hour. In some cases it was obvious that some questions were not applicable, and in those cases the questions were left out. The interview approach allowed the researcher to follow-up answers with new questions, which in some cases dealt with areas that were not included in the prepared questions. This allowed for learning, and a part in formulating questions for the second interview round. All interviews in the first interview round were carried out by at least two researchers.

One of the researchers took the lead and was responsible for the communication with the interviewee, and the other researcher asked follow-up questions. All interviews were audio-recorded and transcribed. The findings from interview-round 1 were summarized informally by the authors at a meeting with the objective to get a first understanding of the interview data. In this way it was possible to update the interview questions before the remaining interviews in round 2 were conducted. The interview questions were refined based on the results and the discussions during the interviews. Mostly this was done by adding detailed follow-up questions. Interview round 2 was carried out in a similar way as in interview round 1. However, some interviews here were carried out by only one researcher.

In the analysis, first a set of codes were derived, then all transcripts were coded. After that, a number of subject areas were decided and for each area a number of relevant codes were identified, and after that the subject areas were summarized. The first set of codes were defined before the coding began, and then the coding list was iteratively increased during the coding. Examples of codes are “identification method” and “information in identification”. In total 31 codes were defined. Defining the codes is an illustrating example of a research step that was iterated in the research. Coding was first carried out by one researcher per interview. After that a few interviews were coded by a second researcher, and it was found that it was hard to get a perfect match of how each text segment was matched to a specific code. It was therefore decided that each interview should be coded by two researchers.

The following subject areas were defined:
- Identification of vulnerabilities
- Evaluation of vulnerabilities
- How to decide and deploy
- Challenges and improvements

For each area a set of relevant codes were defined, which served as a guide in summarizing the area. The resulting texts are presented in Section 5.2. When quotes are given they have, in most cases, been translated from Swedish to English.

5. Results

5.1 Involved Companies

In this section, the involved companies are presented based on the interview results. We use descriptive names instead of their real names in order to preserve anonymity.

In Table 1, all interviewed companies are listed and summarized. The value chain area denotes the company’s products in the value chain according to Section 2. The third column (“# part.”) lists the number of interviewees that participated in the interview. The age is an assessment of the total age of the company, regardless the time they have been working with IoT. The term “startup” means that the
A senior technical expert. At the interview, a main architect with strategic responsibilities at the company and a developer participated.

**IoT Platform Developer** develops a platform for connecting different IoT devices to each other, and they also develop specific sensors for IoT systems. The company is rather new, and can be characterized as a small company and a start up. They are currently selling a component (at sensor level) that they have developed, and they are now developing the platform. They are both a component developer and an integrator. At the interview, a main architect with strategic responsibilities at the company and a developer participated.

**Gateway Consultant** has a long history of developing embedded systems for customers covering a full range of devices. Lately, a new focus has been on so-called industrial-grade devices with high quality and a life-time of more than 10 years. They help customers with security-related questions, as this becomes more and more important. They both give support in specific areas, like security, and take on development assignments. A senior expert with regional responsibility for customers was interviewed.

**Consumer IoT Developer** is new to developing IoT products. The company is rather old and they are used to providing services to other companies and selling products with a low amount of software that are not connected to the Internet. They are now approaching a new market segment, the consumer segment, with a new product in their domain, which is a typical IoT product. The lead developer, one more developer, and their main strategically responsible person, were interviewed.

**IoT Device Developer** is a large company developing IoT products. It is an old company with significant experience of developing wireless communication systems. They provide products with a base unit on which customers may apply their own applications and integrate in their own systems. The interview was conducted with the CTO.

**Software Tool Developer** are developing a software tool to integrate and visualize information from other tools, where security is one aspect. Their tool integrates a wide range of tools and services and visualizes available information from the tools, updating the information in real-time. The interview was conducted with the CEO.

**Security Product Developer** has for about 15 years developed products with a focus on high security applications. The company is currently broadening their focus to offer high security products also to a more general range of potential customers. The interview was conducted with a senior technical expert.

**Advanced IoT Product Developer** develops hardware and software for advanced IoT products. The products are sold to many different domains, where they are integrated into larger plants and installations. The interviews were conducted with a company software security responsible and a system architect.

**Product Integrator** has a long history in proving services to a global market, and has now started to complement their services with services based on IoT. They are new to IoT, but they see the services as important, e.g., since they believe that there may be new competitors with services based on IoT in the future. The interview was conducted with one person responsible for technical decisions.

### 5.2 Results for the Defined Subject Areas

#### 5.2.1 Identification of Vulnerabilities

The interviewed companies have somewhat different approach to identification of vulnerabilities. Small component developing companies with no explicit high requirements on security (Consumer IoT Developer, the Software Tool Developer, and the IoT Platform Developer) focus on keeping everything as updated as possible at all times. They see no possibility to read security reports, but the importance of being aware of potential security problems is recognized. In their experience, there is too much information available to keep track of it. The Gateway Consultant, that has many different customers with different needs, see the same tendency from their customers. Also the Product Integrator who integrates systems spend low effort on identification. To a large extent the problem of identifying new vulnerabilities is left to the product manufacturer. The information is not processed much further but relayed as information to customers to update the product. While they do not attempt to identify vulnerabilities themselves, they see a need to put clearer requirements on the product developers to have well defined vulnerability and security policies in place.

The larger component developing companies do work with identification of vulnerabilities. The Advanced IoT Product Developer, distributes the responsibility for vulnerability identification among the developers. Every software package has an owner within the development organization. Each developer is responsible for at least one package and the owner is responsible for the quality of that software package, which includes security:

<table>
<thead>
<tr>
<th>Company</th>
<th>Value chain</th>
<th># part.</th>
<th>Age</th>
<th>IoT exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT Platform Developer</td>
<td>CD, I</td>
<td>2</td>
<td>Startup</td>
<td>High</td>
</tr>
<tr>
<td>Gateway Consultant</td>
<td>CD, S</td>
<td>1</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Consumer IoT Developer</td>
<td>CD, I</td>
<td>3</td>
<td>Old</td>
<td>High</td>
</tr>
<tr>
<td>IoT Device Developer</td>
<td>CD</td>
<td>1</td>
<td>Old</td>
<td>High</td>
</tr>
<tr>
<td>Software Tool Developer</td>
<td>S</td>
<td>1</td>
<td>Old</td>
<td>Medium</td>
</tr>
<tr>
<td>Security Product Developer</td>
<td>CD, I</td>
<td>1</td>
<td>Old</td>
<td>Medium</td>
</tr>
<tr>
<td>Advanced IoT Product Dev.</td>
<td>CD</td>
<td>2</td>
<td>Old</td>
<td>High</td>
</tr>
<tr>
<td>Product Integrator</td>
<td>I</td>
<td>1</td>
<td>Old</td>
<td>Low</td>
</tr>
</tbody>
</table>
There is however no defined process for how to identify vulnerabilities. The importance of security is well recognized, resulting in many vulnerabilities being identified shortly after disclosure. The company support also has an external entry point for reporting of vulnerabilities. The IoT Device Developer has development teams located in several different countries. Locally, there is no defined process for identifying vulnerabilities. There is a reliance on the fact that some employees are interested in security and the developers responsible for security are assumed to identify new vulnerabilities.

The Security Product Developer has a high focus on security. They monitor both new CVEs and the web pages of the software developers/manufacturer. They also take part of at new results presented at conferences in cryptology and security. Developers have been assigned specific sources to audit and when something comes up, a meeting is arranged to discuss the new vulnerability. This group will then report to the product owner, who in turn decides on how to react.

“We have distributed the responsibility. Every module has a designated owner who is responsible for quality. This includes discovery of vulnerabilities.” (Advanced IoT Product Developer)

They observe that security-aware customers have recently started to make more detailed requirements on security intelligence, resulting in the need for more stringent methods for vulnerability identification. Their experience from these cases are that due to the very broad audit, most of the information is irrelevant. It is estimated that about 5% of the information found in general sources is somewhat relevant, while almost all information found on developers’ own web pages is relevant. The 5% can easily be cut in half, but after that, identifying which issues need to be further scrutinized is much more costly, since it is necessary to take configuration and operation into account.

5.2.2 Evaluation of Vulnerabilities

As with the identification, the organizations have somewhat different approaches. All three small companies, Consumer IoT Developer, Software Tool Developer, and the IoT Platform Developer lack resources and competence to be able to evaluate most of the vulnerabilities. The Gateway Consultant, that has customers, have the possibility to evaluate their customers’ products, set up test environments and examine exploits to simulate events of attacks, but it is not done frequently. Neither the Product Integrator do perform any extensive evaluation. Instead they relies, as with the identification, on the recommendations from the companies developing the products. They usually follow the recommendation. They do not have the required resources or in-depth competence:

“We are unfortunately not mature in the evaluation process to question the manufacturer, how this affects us in both a large scale and small.” (Product Integrator)

Advanced IoT Product Developer does not have a well defined process for evaluating vulnerabilities. Instead, it is up to each maintainer of a specific package to either perform the evaluation or distribute it to another part of the company. It is usually the case that a maintainer discusses the issue with other developers and/or managers. Since the majority of the developers are security conscious, most of the identified vulnerabilities are assumed to be correctly evaluated and, if necessary, patched.

The IoT Device Developer has a dedicated group focusing on security and vulnerabilities. Since most of the software modules are based on OSS, the security group can analyze the source code when evaluating a vulnerability. They use information from previously conducted evaluations when assessing new vulnerabilities, but the process is not formally defined. They are quite confident in the decisions they make due to discussions between the developers, but most of the decisions are based on ‘gut feeling’. The Security Product Developer is currently defining a process for evaluation of vulnerabilities where they perform market analysis for the products based on key components and information regarding the vulnerabilities. When a vulnerability is identified, the evaluation is performed by a developer with experience regarding the vulnerable software. If the vulnerability is considered severe, the issue is raised among the group and project manager. In the gray zones, the developer goes with the ‘gut feeling’.

5.2.3 How to Decide and Deploy

After vulnerabilities have been identified and evaluated the next step is to decide when and how to deploy the changes to products. This is typically carried out in relation to other deployment processes in the company. The large and mature companies in the study, i.e., Advanced IoT Product Developer and Security Product Developer, have experience of working with evaluation of security vulnerabilities and distributing new versions. The companies typically have regular releases with planned dates for when new releases should be distributed. When a security update has been decided, it could either be planned into one of the next releases or planned into a new dedicated release especially for that vulnerability if it is crucial. While the first evaluation of the vulnerability, with the purpose to decide if it is a threat to the actual product, focuses on technical aspects such as configuration and Internet access, there are other aspects of importance in this evaluation, such as the number of components that are installed and in use, and the types and
importance of customers. It is not common to roll out a patch before a planned release. The companies state that they do manage to do an update quickly, but it is not probable that it will be done according to a formalized process. Instead, this is seen as special circumstances, which requires people to take extra decisions in order to solve the problems. When it comes to the less mature companies, it is clear that they rely on non-formal processes for making these kind of changes.

Companies have different roles that take decisions on when and how to roll out changes to customers and users. The larger and mature companies refer to “product owners” with the responsibility of the product. That role can decide how to make changes and contact affected parties. In smaller companies and in less mature companies it is not certain they have that kind of role defined. In those cases, some companies formally or informally relied on the support department for making this kind of decisions. These departments have contact with the customers and their view on the change is seen as important.

It was found for all types of companies, that there to some extent is a tendency to postpone updates to the next planned release instead of making a specific release for a vulnerability. It is a trade-off between risk of waiting and cost and effort to make changes in the field. This is, for example, stated as

“It can take long time to update, and typically they are reluctant to update, they 'need to check'.”

(Gateway Consultant)

The rather complicated structure of the value chain with component developers, integrators, etc., makes it complicated to distribute updates. For example, a component developer, such as the Advanced IoT Product Developer develops components that are distributed to integrators. They may then deliver the system to a customer that then are in charge of the system. This means that it is not only the developing company that decides if an update actually is made in the field.

5.2.4 Challenges and Improvements

The main challenge, regardless of company size and place in the value chain, seems to be that there is too much information regarding vulnerabilities. With around 7,000 new CVEs annually combined with the often large number of software components used in the products, makes it very time consuming to keep track of everything. The larger companies can still do it to some extent since they have the resources, but for the smaller ones it is infeasible. Interestingly, Security Product Developer suggests using the great amount of information to their advantage by rotating the responsibility for the identification part among everyone in the development organization. This can be used for educating the developers and give them insight into vulnerability information and its sources. The same strategy is however not considered appropriate in the evaluation phase, since evaluation requires more in-depth knowledge of the software component or the product in general. In order to improve their possibility to identify and evaluate vulnerabilities, the smaller companies mention automation and automatic filtering of information. This could reduce the amount of information and make it more manageable. Having someone that is explicitly responsible for security is also mentioned as prioritized among the small companies when more resources are available.

6. Discussion

Our interviews indicate that when it comes to identification of vulnerabilities the extent to which vulnerabilities are monitored depends on both company size, its place in the value chain, and business segment. We have also identified an increased awareness of the need for security. For example, Product Integrator would like to improve their ability to put requirements on the manufacturers and the small companies Consumer IoT Developer, the Software Tool Developer, and the IoT Platform Developer, rely heavily on suppliers but without clear security requirements on them, i.e., placing a large trust on their suppliers. It is probably natural and a general problem for integrators of products to follow recommendations and assessments of the developers of subsystems without doing thorough investigations themselves. We have also seen that Security Product Developer has experienced these increased requirements from customers. These findings lead to Observation 1: There is an increased demand for improved vulnerability management throughout the value chain. Investing in increased knowledge about vulnerability management will allow organizations to be more efficient and agile when new vulnerabilities emerge.

For the two manufacturers that do have described processes for identifying and evaluating vulnerabilities, i.e. the Advanced IoT Product Developer and the Security Product Developer, we can identify two different approaches. For the Advanced IoT product, the identifier and evaluator is almost always the same person, i.e., the software package maintainer. For the Security Product Developer, the identification process is more general and developers do not have responsibility for specific packages. Instead, when a vulnerability is identified, the most suitable person is given the task to evaluate, typically the software package maintainer. This many-to-many relationship between potential points of vulnerabilities and information sources seems to be of primary concern for the smaller companies. Based on this, we formulate the following observation: Observation 2: The roles and processes for identifying and evaluating new vulnerabilities differ between companies. The responsibility can be focused on both code parts and on information sources.

Concerning how updates are implemented and distributed, it seems like there is a tendency to postpone updates until the
next planned release. When a decision is taken to actually distribute it earlier, then it is given high priority and the tasks can be fulfilled, even if informal processes must be used. In many cases it was found that companies decided to postpone an update until the next planned release, instead of making a specific release for the security patch. This could probably be explained either by a low perceived risk of being attacked or a low perceived effect of being attacked. Based on the interviews, we believe that it is the former case. A complicating factor for decisions to deploy an extra release is the relationships in the value chain. As mentioned by the Advanced IoT Product Developer, it is not always the case that the integrators will always update. Hence, for larger, more complex systems, there seems to be a reluctance to update. However, for the smaller companies who are also integrating supplier software (cf. Table 1), they have a much more optimistic approach and update suppliers’ software without a thorough analysis. This leads to: Observation 3: The decision to update, or apply an update, depends not only on the severity of the vulnerability, but also on organizational factors, which can complicate the process and leave devices vulnerable for an extended amount of time.

7. Conclusions

Though many are aware that they need to quickly adapt to the increased importance of addressing security vulnerabilities in their products, the pace is not in line with the development in terms of features and connectivity. A recurring problem is that there is too much information and it is very difficult to efficiently process it. The companies simply cannot afford to invest enough resources to review the complete flow of published vulnerability information. The pace of change in the IoT domain and the priority on innovation and new features probably mean that the security processes get little attention and are hence informal and ad-hoc. Three observations can be made from organizations developing IoT products. Firstly, there is an increased demand for improved vulnerability management throughout the value chain, and therefore probably a need to invest in knowledge, mainly on how to identify vulnerabilities and their impact, and how to formulate requirements on identifying vulnerabilities if you are integrating components. Secondly, the roles and processes for identifying and evaluating new vulnerabilities differ between companies, either based on the source of information or on the software architecture. Thirdly, the decision to update, or apply an update, depends on more than the severity of the vulnerability, such as organizational factors.

The results that have been identified are based on a limited set of interviews. However, based on the identified underlying complexity with a fast past of innovation in companies, complex product architectures, and a fast pace of published security vulnerabilities, we believe that it is possible to generalize the observations to IoT development as such. In further research it would be possible to study more cases, e.g. in other types of IoT companies, to improve the generalizability of the results.

Acknowledgments

This work was supported by Swedish Governmental Agency for Innovation Systems (Vinnova), grant 2016-00603.

References