UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Shapaval Raman

Security Risk Management for the IoT systems

Master's Thesis (30 ECTS)

Supervisor: Raimundas Matulevicius, PhD

Tartu 2017

Security Risk Management for the IoT systems

Abstract:

Since 2012 the number of units in global infrastructure for the information society (The Internet of Things) has grown twice. With this number also has grown the amount of possible threats and risks, which influence security on all levels of the system. As a result a huge amount of users' data was stolen or damaged.

According to Third Quarter, 2016 State of the Internet / Security Report based on data gathered from the Akamai Intelligent Platform (`https://www.akamai.com/us/en/solutions/intelligent-platform/`) the total number of DDoS attacks in Q3 2016 increased in 71% compared to Q3 2015. With 623 Gbps data transfer attack it was largest DDoS ever and this fact will only increase the number of future attack events.

All these facts reveals a problem that a lot of IoT systems are still unsecured and users' data or personal information stay vulnerable for threats.

The paper combines knowledge of Security Risk Management with existing practice in securing in IoT into a framework, which aim is to cover vulnerabilities in IoT systems in order to protect users' data. The framework is then tested on self developed IoT system represented by Raspberry Pi 3 interconnected with sensors and remote data storage.

Keywords:

List of keywords

Contents

Unsolved issues

1 Introduction

Internet of Things (IoT) is a network of connected devices and systems to exchange or accumulate data and information generated by users of and embedded sensors in the physical objects [4]. Among the privacy, energy-awareness, environment, and other concerns, security plays an important role, as the (potentially sensitive) data is sent among the various devices and multiple users. In cases where such a data is intercepted and used for non-intended purposes, it may lead to the severe damages of the valuable system and/or environmental assets [7] [53] [54] [55] [56] [57]. There exist a number of surveys related to the IoT security [9] [10], security of the IoT frameworks [21] [22], or specific components of the IoT systems [13] [14] [17]. In this paper we propose a comprehensive reference model for the security risk management in the IoT systems. We base our proposal on the domain model for the information systems security risk management (ISSRM) [18] [1] – thus, we focus on the security risks to the information and data managed in the IoT system. Since the IoT systems much depend on the cloud and Internet computing we consider how vulnerabilities and their countermeasure considered in the open Web application security project (OWASP) [39] can help when identifying and managing the security risks in the IoT systems.

The rest of the paper is structured as follows: in Sect. 2 we overview the Security Risk Management as a development process along with ISSRM domain model. Sect. 3 based on IoT domain model presents components for managing IoT security risks. This includes discussion on the IoT assets, their vulnerabilities and countermeasures used to mitigate these vulnerabilities. Sect. 4 gives few examples illustrating some reported security risks. Sect. 5 represents an application of the developed framework to the real-word IoT system and test results. Finally, Sect. 6 concludes the paper and provides directions for future work.

2 Related Work

This chapter surveys previous work in Internet of Things security. These days Internet of Things (IoT) popularity, as a research topic, is growing rapidly because of its interconnected components heterogeneity and the availability to communicate without human intervention [5]. The fact that IoT is heavily affecting our lives in different layers, from wearable devices such as smartwatches or fitness trackers till huge manufacturing systems, forced the rapid growth in IoT-based application development and consequently the need of appropriate IoT frameworks [6]. However most of IoT devices are easy to hack and compromise due its hardware limitations of computing power along with lack of storage and network quantity [11]. Therefore, security and the ability to monitor IoT system state, along with study of recent attacks on this sector became a ground vector in developing frameworks for securing IoT systems.

2.1 Taxonomy of the IoT

According to [12] IoT can be classified into four layers: 1) Perception Domain; 2) Access Domain; 3) Network Domain; 4) Application Domain. The general structure is presented on Figure 1.

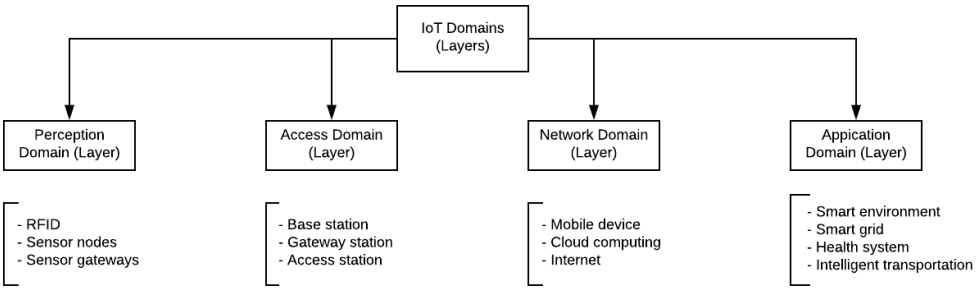

Figure 1. IoT Domains (Layers).

2.1.1 Perception Domain

Current domain (layer) collects information using readers or gateway. This layer can be divided into two sections: (i) *perception node*, which is represented by sensors, controllers etc; (ii) *perception network*, which aim is to interconnect the network domain (layer). Actual reader can extract information from the message received from the RFID tag, which is sticked to the 'thing', while gateway read signals from the sensors, consequently determining the existence of the 'thing' or changes in the environment [12] [5].

RFID can be presented as a revolution technology in the embedded communication model, which provides functionality for microprocessors' wireless communication configuration. There are two types of RFID: (i) *active* - RFID tags with their own power source, which provide signals to readers, creating instant communication; (ii) *passive* - tiny RFID tags with unique identity to support active communication between tags and readers using wireless networks [12].

Sensor nodes take care of gathering and processing of the sensed data, while being interconnected with other nodes in the network. They have next components: (i) *controller* - takes care of data processing and other node's parts performance; (ii) *transceiver* - transmits and receives radio frequencies; (iii) *program memory* - is used for programming device; (iv) *power source* - supplies node with power; (v) *hardware* - senses data from the environment. Sensor itself and actuator can be described as the major one components, as they sense data and activate/deactivate device based on commands from the nodes [12].

If we look at the wireless networks and how the data is collected from various WSN nodes, we will come to the sensor gateways, which provide above mentioned functionality. Each gateway supports 2.4 GHz IEEE 802.15.4 radio and with a communication framework on board it checks and records the conditions of various sensors. Transmitters and receivers create radio channels between two or more devices to provide data-exchange functionality [12].

2.1.2 Access Domain

In the access domain the reader or the head of gateway transfers sensed data to the access point, which is represented by the gateway or base station. Access point then transfers data to the network domain (layer) [5].

2.1.3 Network Domain

When sensed data leaves access domain (layer), it reaches network domain (layer), which provides network transmission along with information security. "The form of network is possible for any existing network, such as wlan, mobile network, Internet or broadcast and television network" [5]. The network layer consists of mobile devices, clouds and the Internet itself [12].

Mobile devices can be characterize as a portable device with its own operating system on board, which supports inbuilt or external applications. These devices are usually equipped with such technologies as Wi-Fi, Bluetooth, Near-Field Communication (NFC), Global Positioning System (GPS) etc. Mobile devices connected to the Internet can provide users with nearly unlimited functionality in digital word. However, they also produce a branch of vulnerabilities, which result into uncounted number of threats to all users' personal data.

Clouds or cloud computing is an Internet-based services, which consist of hardware, systems software, and applications and provides users with resources on-demand. There are three main cloud categories: (i) *Software as a Service* (SaaS) - user can use an application on-demand, but can not control hardware or network. Such model provide users with applications through the network; (ii) *Platform as a Service* (PaaS) - allows user to host environment for their applications, however the operating system, hardware or network infrastructure still can not be controlled; (iii) *Infrastructure as a Service* (IaaS) - gives user an access to 'fundamental computing resources' such as CPU, memory, middleware and storage, but cloud infrastructure remains protected [19].

" The Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony, and file sharing " [50].

2.1.4 Application Domain

The last domain (layer), but not the least, is the application domain, which can implement the logic in interconnection between 'things' and 'things' or between 'things' and person and plays crucial role in the whole IoT structure as a head layer, which is visible to user and wrap all functionality into User Interface (UI) [5] [12]. We can allocate four ground types of applications which represent this layer: (i) *Smart environment* - is represented by intuitively operated devices that help us to organize, structure, and master our everyday life. In other words, smart environments provide functionality, which helps the human to interact with surroundings and reach his goals [23]; (ii) *Smart grid* - intelligent power supply network, which provide communication functionality based on two-way wireless and/or wireline communication technologies. Main aim is to increase the productivity and efficiency of the networks. National Institute of Science and Technology (NIST) considers mobile broadband technologies such as 2G/3G/4G as key enablers for Smart Grid networks, while Electric Power Research Institute (EPRI) examines technologies such as Mobile WiMAX, GPRS and LTE as key enablers for automated metering infrastructure (AMI) [31]; (iii) *Healthcare system* - is a framework based on radio frequency technology to deliver sensed data from fragile and accurate micro-nodes embedded inside or outside humans body; (iv) *Intelligent transportation systems (ITS)* - represents a stack of high technology and improvements in information systems, communication, sensors, and advanced mathematical methods interconnected with the surface transportation infrastructure [24].

Each application should be mindful, active and personalized in order to provide secure and stable communication between devices.

2.2 Taxonomy of the IoT security

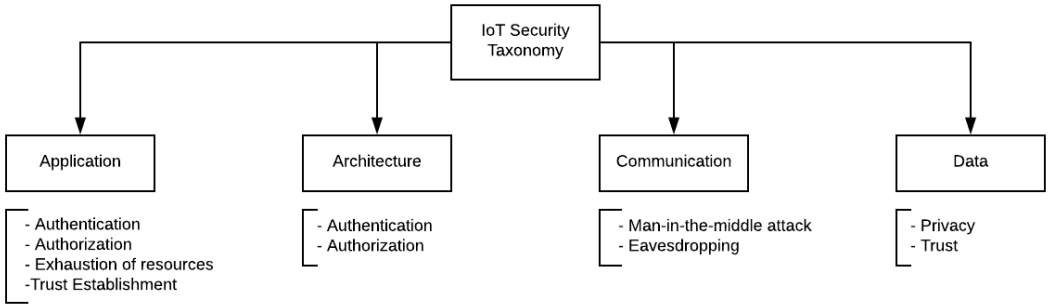

Figure 2. IoT Security Taxonomy.

The existing IoT classification is definitely complicated and heterogeneous, that is why we need to investigate the actual IoT security taxonomy which should be simple and more specifically aimed on characterizing classes of security threats and vulnerabilities in each IoT layer. After studying a branch of existing IoT frameworks [8] [16] [29] [26] [15] we can allocate four main classes in IoT security, see Figure 2: (i) *Application domain* - consists of security techniques such as authentication, authorization, exhaustion of resources, trust establishment etc; (ii) *Architectural domain* - always depends on the scenarios and application domain. According to [5] [8] [16] [29] [26] [15] we can provide a summery of existing IoT architectures, see Table 1; (iii) *Communication* - responsible for information exchange/sharing between IoT devices in the system through its layers or between different IoT systems; (iv) *Data* - one of the main targets for threats, its privacy must be guaranteed.

Table 1. Examples of existing IoT architectures

Architecture	Application Domain	Purpose
SDN Architecture	Smart environment	SDN manages the network state and the resource allocation, influences the whole system through centralization of the network controller. SDN produces the computing results for an optimized network status [8].
NFV Architecture	Smart environment	NFV network functions are implemented with the usage of servers and storage devices of industry standard, which can also be represented by virtual machines. With the reduction of the number of hardware appliances and establishment of the high volume hardware usage, NFV contributes efficiency and flexibility to the system [8].
SEA Architecture	Healthcare	SEA: A Secure and Efficient Authentication and Authorization. SEA accomplishes authentication and authorization of remote end-users using smart e-health gateways. Being relied on the certificate-based DTLS handshake SEA provides secure solution for the medical sensors [16].
Smart City Architecture	Smart City	The main aim is to combine business architecture, information system architecture and technology architecture into metamodel, which supports consistency, completeness, traceability and relationship of components and layers in the enterprise architecture of the smart city [29].
Service-Oriented Architecture (SOA)	Smart transportation	SOA defines secure IoT middleware architecture services, which aggregate data streams within the network in order to reduce the overall network load. Although, such architecture enables the interconnection of system elements with different interaction styles. "IoT-based SOA holds the promise of easing the development of rich applications integrating the physical with the virtual worlds in a multitude of domains" [26].
OSCAR: Object Security Architecture	Smart grid	Introduces modern scalable and efficient architecture for E2E security and access control in IoT along with architecture evaluating in M2M settings.
Conceptual Organizations Framework	Business organizations	Conceptual Organizations framework " adopts a broad perspective based on User Experience design and attempts to bridge design and system engineering constructs with cognitive and socio-cultural theories, in order to provide a common understanding to multi-disciplinary IoT development teams" [15].

It is clear that in the IoT scenario, security solutions cannot be limited to the single layer, it always has to be an end-to-end combination of overcoming challenges for each class in IoT taxonomy. Consequently, we have to understand the possible issues in each layer in detail as the heterogeneity of IoT interacting objects makes it impossible to construct solution from outside the system, we have to go from inside, combining solutions from different security levels into one strong consequent flow of countermeasures.

2.2.1 Security challenges in Application layer

Major security challenges in the Application layer can be presented by next groups [5]:

(i) *Authentication* - allows integration of different IoT devices from different context. During authentication process its required to authenticate routing peers used for data transferring/exchanging along with origin data node authentication. Consequently, we can affirm that key deployment and key management are ground challenges in IoT authentication [11]. Authentication of nodes is always necessary for a good and secure implementation as it provides prevention against illegal node access. Authenticating the end to end communication makes the communication

secure and the attacker will not be able to spoof the messages between those end hosts. One more challenge is to maintain authentication schemes both lightweight and reliable [25].

(ii) *Authorization* - addresses the security issue of illegal node access through specifying the access rights to different resources. The data should always be secure protected and accessible only to authorized users. As each IoT node supports limited techniques for access verification it is always challenging to provide appropriate authorization mechanism which will be beneficial and tolerate to nodes with different capabilities [20].

(iii) *Exhaustion of resources* -

(iv) *Trust establishment* -

2.2.2 Security challenges in Architecture

2.2.3 Security challenges in Communication

2.2.4 Security challenges in Data protection

2.3 The future directions in the IoT

3 Security Risk Management (SRM)

3.1 What is Security Risk Management?

In general, Security Risk Management (SRM) can be described as a part of the secure systems development process, which aim is to understand what assets should be protected, from which risks and how these risks could be allayed while covering discovered vulnerabilities of the system [1]. SRM plays leading role in developing appropriate solutions based on situation and existing security countermeasures. Typically it provides a set of rules to lower the risk level or totally prevent possible attacks on the system along with the hints for successfully and qualitative system monitoring. There are many different methodologies which apply SRM in various domains, that's why deep understanding of the system's architecture is crucial during SRM analytical procedure.

According to [1] we will look at a few of SRM methods as the number of existing approaches always grow and we can not pretend our survey to be complete. However, chosen SRM methods should be enough to understand what king of analysis concepts and principles are used during analytical procedure in security risk management.

CORAS is a risk assessment methodology [34] which provides detailed instructions for the use of UML-oriented modelling for the next three main purposes: (i) "to describe the target of assessment at the right level of abstraction" [34]. (ii) "as a medium for communication and interaction between different groups of stakeholders involved in risk assessment" [34]. (iii) "to document risk assessment results and the assumptions on which these results depend" [34]. There are three ground steps for the risk analysis of the *CORAS* risk management process: (i) identify risks; (ii) analyse risks; (iii) evaluate risks. The *CORAS* risk management process based on using branch of approaches in terms of XML technology for analysing different parts of the system independently. The choice of the method is made by evaluating the viewpoint in which the part to be analysed appears and determining development lifecycle phase.

EBIOS (French: Expression des Besoinset Identification des Objectifs de Securite) is a method for analysing, evaluating and counter-measuring against risks in the system. There are five main steps of the *EBIOS* method [35]: (i) Circumstantial study - determining the context; (ii) Security requirements; (iii) Risk valuation; (iv) Security goals determination; (v) Identification of security requirements. According to [35] *EBIOS* generates a security policy adapted to the system based on its specifications and features. The method was created in 1995 and is now maintained by the ANSSI, a department of the French Prime Minister.

AURUM (Automated Risk and Utility Management) is a framework for information security risk management [36], which was developed to help in decision making according to organisational needs with respect to selection of security measures. It represents a new methodology for supporting the NIST SP 800-30 risk management standard. According to [36] the next benefits were documented: (i) *AURUM* provides risk manager

with an "ontological information security knowledge base" in order to structure the information security knowledge in appropriate way; (ii) Describes the methodology for the consistent modelling of organizational resources; (iii) Provides guidelines for using "widely accepted information security knowledge for threat/vulnerability identification and control recommendations" [36]; (iv) Introduces usage of the Bayesian threat likelihood determination, which gives opportunity for the objective level threat evaluation; (v) Automated threat impacts calculation as a part of system's resources rating; (vi) Automated controls proposals for the risks mitigation; (vii) Provides risk manager with an opportunity to investigate possible scenarios and characterize the problem in details;

MEHARI (MEthod for Harmonized Analysis of RIsk) is a free risk management method for the information systems. This framework combines complex knowledge base with existing tools into the powerful instrument for the information security risk analysis [37]. There are next ground steps: (i) Threat analysis; (ii) Business process analysis; (iii) System assets classification; (iv) Risks ratio processing; (v) Diagnostic questionnaires for evaluation risks' mitigation level of the system; (vi) System Security Requirements evaluation; (vii) Possible scenarios determination; (viii) Developing countermeasures based on possible risks scenarios; Entire focusing on the risks analysing and creating controls makes *MEHARI* a powerful framework for the SRM.

CRAMM (CCTA Risk Analysis and Management Method) is an automated tool based on qualitative risk assessment methodology, which consists of the next stages [38]: (i) Assets identification and evaluation; (ii) Threat and vulnerability assessment; (iii) Countermeasures recommendation; As it mentioned in [38] "CRAMM is a comprehensive and flexible tool especially for justifying prioritized countermeasures at a managerial level, needing, however, qualified and experienced practitioners for efficient results".

3.2 Domain Model for SRM

One of the most important role in Security Risk Management (SRM) is played by the SRM Domain Model. The main purpose of a domain model is to represent the main concepts of the system with their responsibilities and relationships. Based on main features of the concepts, we can then classify them as different assets [2]. In this way we can determine what assets need to be protected and from which risks. Such strategy helps to generalize understanding of the system's problem between people who are working on possible solutions for it. Only with a common understanding of the main concepts it becomes possible to argue about architectural solutions and to evaluate them.

We have SRM domain model (SRMDM) represented as an UMl diagram on the Figure 1. All elements of the SRMDM could be divided into three major groups [1]: (i) *asset-related concepts* - "describe which of an organisation's assets are important to protect and what criteria guarantee a certain level of asset security" [1]. Typically pure *asset* can be represented by any part of the system which has some value and plays a

role in the work of whole system. Assets can also be divided into sub-groups: *business assets* ("describes the information, processes, capabilities and skills essential to the business and its core mission" [1]) and *organizational assets*, which are represented by the core elements of the system and play crucial role in providing appropriate service to the customers; (ii) *risk-related concepts* - "introduce definitions of risk itself and its immediate components" [1]. According to the Figure 1 we can allocate *risk* itself as a combination of the *threat* which exploits existing *vulnerability* of the system plus *impact* to which such *event* can lead to.; (iii) *risk treatment-related concepts* - "describe the concepts to treat risk" [1]. According to [1] we can choose one of four ground risk treatment decisions while risk mitigating process: *Risk avoidance* - describes approach of modifying system's functionality to fully avoid possible risk; *Risk reduction* - proposes strategy of the minimization risk occurrence possibility, so security requirements are, typically, selected for reducing the risks instead of full avoidance; *Risk transfer* - "defines how risk parties could share the burden of loss from a risk" [1]; *Risk retention* - "constitutes acceptance of the burden of loss from a risk. No design decision is necessary in this case." [1].

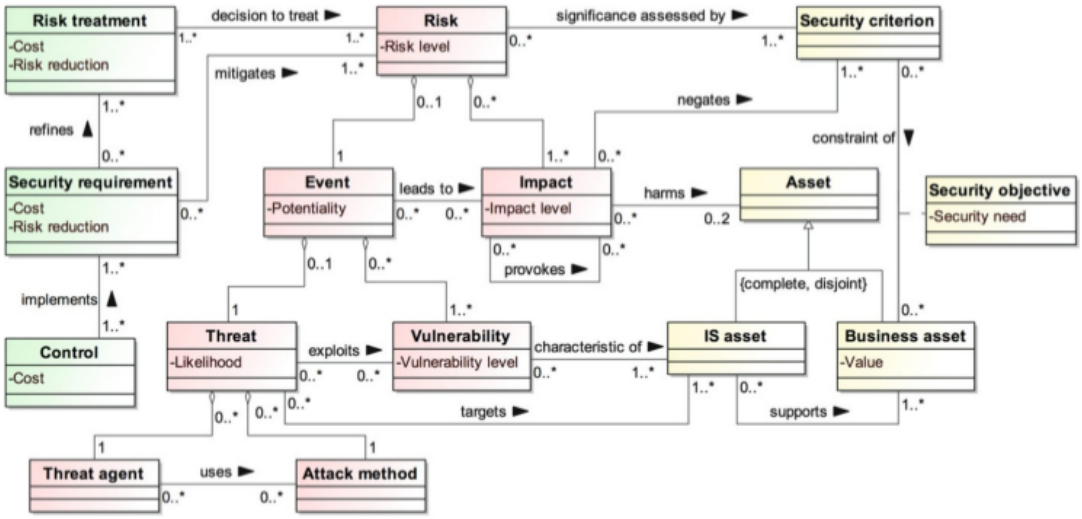

Figure 3. General SRM Domain Model.

4 Content and Assets of IoT

The Internet of things (IoT) is represented by wisely connected devices and systems to exchange or accumulate and process data generated by users or by embedded sensors and actuators in machines and other physical objects. IoT starts playing leading role in improving customers' life quality on different layers including security, health, education and energy efficiency. Industrial business has also experienced advantages of using IoT systems in manufacturing, retail, agriculture and other sectors [32].

As the IoT through last years has gained a significant positive impact on consumers, enterprises and society as a whole it also has bring in a stack of risks connected to beneficial services provided by IoT systems. That's why we need to understand a General Domain Model Architecture of the IoT (GDMA) [33] in order to establish the common grounding definition of IoT system assets and to describe their basic interactions and relationships with each other.

4.1 IoT Domain Model

Internet of Things Domain Model (IoTDM) plays leading role in applying SRM methods to the IoT systems since it provides definition of the main abstract concepts of the system. The Domain Model helps to determine system and business assets of the IoT system and then evaluate their importance in order to apply appropriate countermeasures against existing risks. "The main purpose of a domain model is to generate a common understanding of the target domain in question" [2].

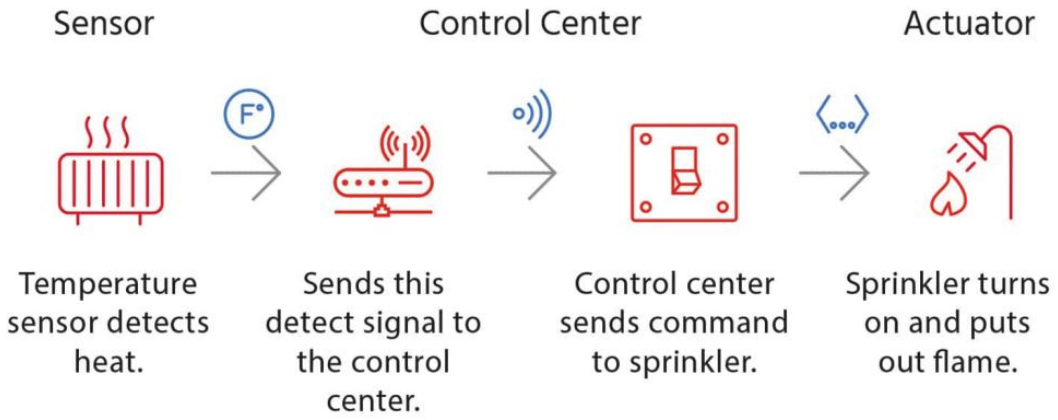

Figure 4. Sensor to Actuator work flow.

Figure 3 shows us general IoTDM which we will use while applying SRM to the existing IoT system. However, before describing how we will secure real system and from which risks we have to describe our IoTDM in terms of SRM and determine relationships between concepts represented in both domain models.

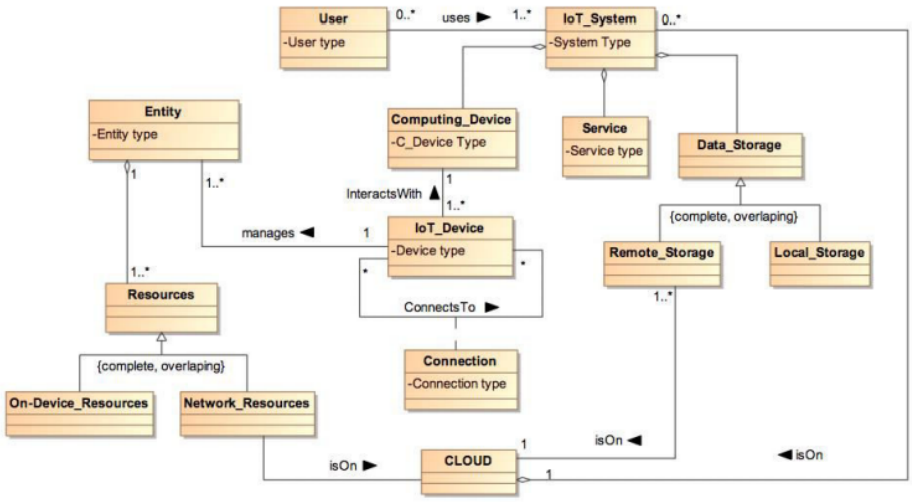

Figure 5. General Domain Model Architecture of the IoT.

Lets represent the initial work flow of the simple IoT system in terms of SRM. If we look at Figure 2 we can determine four IoT elements: (i) *temperature sensor* - detects heat; (ii) *Wi-Fi module* - sends detected temperature from the sensor to the *control center*; (iii) *control center* - makes decision what command should be sent to the sprinkler based on the previously received temperature from the *temperature sensor*; (iv) *actuator* - represented by sprinkler, turns on and puts out flame or turns off depending on the command from the *control center*. Suppose, an attacker, (*Threat agent*) with means to break a normal work flow of our IoT system from Figure 3, exploits the weakness of the Wi-Fi connection between Wi-Fi module and Control center, which represent *System assets*, and get an access to the temperature (*Business asset*) which is going from the sensor. Then an attacker changes the temperature value and this *Event* leads to the *Impact* of overheating and consequently to the *Harm* to sprinkler. Such *Event* and its *Impact* represent *Risk* to the IoT system.

Finally, we can summarize that assets in the IoT represent anything that is valuable for the IoT system or play crucial role in providing appropriate functionality and services

to users. According to [1] *system assets* can be described as parts of the IoT system, which gain their importance through supporting business assets. Typically, system assets in IoT can be represented as ground components of the information technology system such as hardware, software or network. In our simple example from Figure 2 system assets are represented by sensor, Wi-Fi module, Control center and Actuator.

Business assets are extremely valuable for each IoT system as they represent essential to the business things such as information, processes, capabilities and skills [1]. Besides official definitions, business assets can be commonly represented by the data, which is transferred, stored or manipulated by the IoT system during working process. As a result business assets security is definitely worth attention. According to Figure 3 business assets are represented by the temperature which goes from the Sensor to the Control center and commands from the Control center to the Actuator.

4.2 Vulnerability

According to the cyber-security terms we can determine vulnerability as a " term that refers to a flaw in a system that can leave it open to attack. A vulnerability may also refer to any type of weakness in a computer system itself, in a set of procedures, or in anything that leaves information security exposed to a threat " [51] . In terms of Security Risk Management applied to the IoT vulnerability is a characteristic of the system asset or group of IS assets that exposes a weakness or flaw in terms of security [1].

Have to be mentioned that if we talk about vulnerability of the application, then it could be presented as a weakness in design flaw or an implementation bug, which can allow an attacker to harm application owner, application users, and other entities that rely on the application. According to the Open Web Application Security Project [39] we can allocate next most common IoT vulnerabilities:

1. **Insecure Web Interface, V#1:**
   Threat agent - an attacker with access to the web interface. Could be presented by external or internal user of the system.
   Attack method - an attacker uses weak credentials, gains access to plain-text credentials or enumerates accounts to access the web interface.
   Vulnerability - account enumeration, lack of account lockout or weak credentials are the most common vulnerabilities of the insecure Web Interface.
   How to discover - issues of the Web Interface could be find out during manual work with the system along with using testing tools for cross-site scripting identification.
   Impact to system assets - insecure web interface could result in harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
   Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
   Possible attack examples:

   (i) The Forgot Password functionality provides responses on entering invalid users' credentials. As soon as valid credentials identified an attacker could calculate appropriate password using brood force methods.

   (ii) Web interface maintains vulnerable to cross site scripting. For example, the HTML snippet:

2. **Insufficient Authentication/Authorization, V#2:**
   Threat agent - an attacker with access to the web interface. Could be presented by external or internal user of the system.
   Attack method - an attacker uses weak passwords, insecure password recovery mechanisms, poorly protected credentials or lack of granular access control to access a particular interface.

Vulnerability - weak passwords or credentials with poor protection can lead to insufficient authentication.
How to discover - common issues with authentication/authorization can be discovered while examining the interface of the system with automated testing and require more stable passwords.
Impact to system assets - insufficient authentication/authorization can result into harm to system's data confidentiality, integrity and availability or denial of access.
Impact to business assets - all users could be compromised as a result of their data being stolen.
Possible attack examples:

(i) The interface requires simple passwords.

(ii) Users' credentials are not appropriate protected while transmission to the DB or not encrypted before sending.

3. **Insecure Network Services, V#3:**
Threat agent - an attacker with a network access to the IoT device. Could be internal or external user.
Attack method - an attacker uses vulnerabilities in network services to launch an attack on IoT device itself or bounce attacks off the device.
Vulnerability - not enough controls on open ports and traffic monitoring in the system.
How to discover - insecure network services can often be detected with the help of port scanners tools and fuzzers. " Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. The program is then monitored for exceptions such as crashes, or failing built-in code assertions or for finding potential memory leaks. Typically, fuzzers are used to test programs that take structured inputs." [`https://en.wikipedia.org/wiki/Fuzzing`].
Impact to system assets - insecure network can result into harm to system's data confidentiality, integrity and availability or denial of service. Can also be a reason for launching attacks on other systems' devices.
Impact to business assets - denial of service attack could lead to users' data loss of availability along with data loss.
Possible attack examples:

(i) Fuzzing attacks. Lets consider an example of fuzzing attack with a Burp Suite Intruder and an OWASP WebGoat application. An attacker here with means to log into the app as Admin user without the password.

(ii) Some ports are open and are not monitored by the system, so could be accessed via UPnP.

4. **Lack of Transport Encryption, V#4:**
   Threat agent - an attacker with an access to the network the IoT device is connected to.
   Attack method - attacker uses the lack of transport encryption to view data being passed over the network.
   Vulnerability - lack of transport encryption is a common vulnerability for local networks as it is assumed that in the local and wireless networks the traffic will not be widely visible and accessible. However it makes all the data transferred through such networks visible for possible attacker in range of the wireless network support or with external access to the local network.
   How to discover - many issues with transport encryption are easy to discover simply by launching testing attack on the system with viewing network traffic and searching for readable data in it. An automated tools can also look for proper implementation of common transport encryption such as SSL and TLS.
   Impact to system assets - insecure web interface could result in harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
   Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
   Possible attack examples:

   (i) The cloud interface uses only HTTP.

   (ii) User-name and password are transmitted in the clear over the network.

5. **Privacy Concerns (Confidentiality), V#5:**
   Threat agent - an attacker with an access to the IoT device itself, the network the device is connected to, the mobile application and the cloud connection.
   Attack method - an attacker uses insufficient authentication, lack of transport encryption or insecure network services to break systems' data confidentiality.
   Vulnerability - all vulnerabilities of authentication/authorization, low protected transport protocols and not appropriate secure network services.
   How to discover - privacy concerns are easy to discover by simply reviewing the data that is being collected as the user sets up and activates the device. Automated tools can also look for specific patterns of data that may indicate collection of personal data or other sensitive data.
   Impact to system assets - harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
   Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
   Possible attack examples:

   (i) Collection of personal data.

(ii) Collection of financial and/or health information.

6. **Insecure Cloud Interface, V#6:**
Threat agent - an attacker with an access to the internet.
Attack method - an attacker uses insufficient authentication, lack of transport encryption and account enumeration to access data or controls via the cloud interface (website).
Vulnerability - low level security of the cloud access credentials or account enumeration is possible.
How to discover - insecure cloud interfaces are easy to discover by simply reviewing the connection to the cloud interface and identifying if SSL is in use or by using the password reset mechanism to identify valid accounts which can lead to account enumeration.
Impact to system assets - harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
Possible attack examples:

   (i) Password reset indicates whether account is valid.

   (ii) User-name and password are poorly protected when transmitted over the network. In such cases, an attacker is able to either determine a valid user account or capture the credentials as they cross the network and decode them since the credentials are only protected using Base64 Encoding.

7. **Insecure Mobile Interface, V#7:**
Threat agent - an attacker with access to the mobile application.
Attack method - an attacker uses insufficient authentication, lack of transport encryption and account enumeration to access data or controls via the mobile interface (application).
Vulnerability - low level security of the app access credentials or account enumeration is possible.
How to discover - insecure mobile interfaces are easy to discover by simply reviewing connection to the wireless networks and identifying if SSL is in use or by using the password reset mechanism to identify valid accounts which can lead to account enumeration.
Impact to system assets - harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
Possible attack examples:

(i) Password reset indicates whether account exist or not.

(ii) User-name and password are poorly protected when transmitted over the network. In such cases, an attacker is able to either determine a valid user account or capture the credentials as they cross the network and decode them since the credentials are only protected using Base64 Encoding.

8. **Insufficient Security Configurability, V#8:**
Threat agent - an attacker with an access to the IoT device.
Attack method - an attacker uses the lack of granular permissions to access data or controls on the device, low level transport encryption and the use of low level passwords can also be the doors for the attack on the system.
Vulnerability - insufficient security configurability is present when users of the device have limited or no ability to alter its security controls. If the web interface does not support creating granular user permissions and does not force user for creating strong password.
How to discover - detailed review of the web interface along with provided options can reveal existing vulnerability.
Impact to system assets - harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
Possible attack examples:

(i) Strong passwords are not forced to be used, then an attacker is able to gain access to existing accounts using brood force methods.

(ii) Valuable data which is stored on the device is not encrypted, then an attacker is able to access data at rest which has just simple protection such as asking for administrator's credentials to view data.

9. **Insecure Software/Firmware, V#9:**
Threat agent - an attacker with an access to the IoT device and/or network the device connects to and/or the server which provides updates to the IoT devices Software.
Attack method - an attacker captures files with updates from unencrypted connection between server and ioT devices. The files with updates are not encrypted itself. An attacker performs unauthorized update of the system software via DNS hijacking.
Vulnerability - low system states monitoring, low level encryption. Devices should have the ability to be updated when vulnerabilities are discovered and software/-firmware updates can be insecure when the updated files themselves and the network connection they are delivered on are not protected. Software/Firmware

can also be insecure if they contain hardcoded sensitive data such as credentials.
How to discover - security issues with software/firmware are relatively easy to discover by simply inspecting the network traffic during the update to check for encryption or using a hex editor to inspect the update file itself for interesting information.
Impact to system assets - harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover.
Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
Possible attack examples:

(i) Update file is transmitted via HTTP then an attacker is able to change them on their way from server to device.

(ii) Update file is unencrypted and human readable data can be viewed.

In the cases above, the attacker is able to either capture the update file or capture the file and view it's contents.

10. **Poor Physical Security, V#10:**
Threat agent - an attacker who has physical access to any physical system asset.
Attack method - an attacker uses vectors such as USB ports, SD cards or other storage means to access the Operating System and potentially any data stored on the device.
Vulnerability - Physical security weaknesses are present when an attacker can disassemble a device to easily access the storage medium and any data stored on that medium. Weaknesses are also present when USB ports or other external ports can be used to access the device using features intended for configuration or maintenance.
How to discover - detailed review of the hardware components of the IoT system.
Impact to system assets - harm to service's data confidentiality, integrity and availability which can then lead to complete device takeover
Impact to business assets - the actual device or whole IoT System could be compromised along with the users.
Possible attack examples:

(i) The device can be easily disassembled and storage medium is an unencrypted SD card.

(ii) USB ports are present on the device then an attacker is able to change device software or stole valuable data from it.

4.3 IoT Security Controls

During SRM process in IoT determining possible risks is not the only step in securing system. It is always important to know how to cover existing vulnerabilities in order to mitigate possible risks. In the IoT, securing process requires a holistic understanding of security controls for all assets of the system. Security mechanism should be unique for each valuable element of the system as it has to follow elements' lifecycle and functionality provided to the system. In this section we allocate and describe ground keys in securing IoT systems and propose appropriate contermasures for covering different types of systems' vulnerabilities [28].

4.3.1 Protocol and network security

Protocol and network security often requires optimizations in cryptography algorithms and appropriate key management systems, as cryptography plays leading role in developing security protocols for securing network infrastructure. However, applying standard Internet security mechanisms to the functional assets of the IoT system could be inefficient due to the lack of the assets' resources. Therefore, security protocols should be adapted based on the assets' architecture taking into account their performance and capabilities. According to the Open Web Application Security Project [39] we can propose next security countermeasure to apply protocol and network security on appropriate level:

1. **Secure Network Services, Cm#1:**
    - Only necessary ports which are crucial for the appropriate system functionality should be exposed and available.
    - Service should be protected from buffer overflow (overrun) [43] and fuzzing attacks [44].
    - Ensure that the service is prepared and not vulnerable to DoS attacks [45] or at least monitoring system is provided to determine such type of an attack.
    - Ensure that network ports and services exposed to the internet via UPnP for example.
    - Monitoring system should also detect abnormal service request traffic and block gateways if necessary.
2. **Transport Encryption, Cm#2:**
    - Always use transport protocols such as SSL [46] and TLS [46] while data transferring through network to be sure that your data is encrypted.
    - If SSL or TLS are not available do not hesitate to use other standard encryption methods to protect data.

- Pay attention what encryption techniques are used. Avoid using unaccepted encryption standards.
- Use MQTT payload encryption [47] to protect system's specific data on the application level.
- Ensure to use secure encryption key handshaking.
- Always verify incoming data for integrity.

Countermeasure on secure network services (Cm#1) mitigate risks with vulnerabilities of insecure network services (V#3), and communication encryption (Cm#2) – vulnerabilities related the lack of communication encryption (V#4).

4.3.2 Data and privacy

Data and privacy key often represented by the most sensitive and valuable assets of the IoT system. Each user wants their personal data be protected and managed by themselves, but there are a lot of systems which based on managing users' data. In such cases a huge issue is to constrain and monitor the list of accepted data managers. Consequently, using cryptographic algorithms and different protocols for secure data transferring could be not enough. In other words it should be developed special management policy based on the data type managed by the system. We can propose next countermeasures [39]:

1. **Privacy Concerns, Cm#3:**
    - While collecting data be sure that: (i) only critical to the functionality of the system data is collected; (ii) avoid collecting sensitive data; (iii) collected data is de-identified or anonymized; (iv) retention limits are set for the collected data; (v) protect collected data with encryption.
    - Ensure that personal information is properly protected on all levels of the system.
    - Give access to the collected data only for authorized users.
    - Do not collected more data that actually needed for appropriate functionality of the system.
    - Always de-identify data before analyzing.
2. **Secure Software/Firmware, Cm#4:**
    - Ensure that the system uses secure update mechanism and all files transferring is based on accepted encryption methods.
    - The update file should not expose sensitive data.

- Each pack of incoming files with updates should be signed and verified before even saving in system memory storage.
- Use only trusted and secure servers for updates.
- If possible, its recommended to use secure boot ("Secure Boot is a technology where the system firmware checks that the system boot loader is signed with a cryptographic key authorized by a database contained in the firmware." [49]).

3. **Physical Security, Cm#5:**

   - Use only trusted and protected data storage services that can not be easily removed.
   - All sensitive system's data should be encrypted at rest.
   - Always check that USB ports or other external ports are protected from uploading malicious software through.
   - It is recommended to minimize the number of external ports such as USB used by the system.
   - "Ensuring the product has the ability to limit administrative capabilities" [39].

Countermeasures regarding the privacy concerns (Cm#3) help to mitigate security risks with vulnerabilities related to privacy concerns (V#5); secure software and/or firmware (Cm#4) – vulnerabilities related to insecure software and/or firmware (V#9). Countermeasure of physical security (Cm#5) address risks with vulnerabilities of poor physical security (V#10).

4.3.3 Identity management

Identity management represents a non-trivial process of verifying a staggering variety of identity and connection types. We can allocate a key rules which should be followed during IoT system identity management:

1. An object's identity should always be unique compared to the other objects from its family.
2. Unique identity can be called core identity, as an object can also have several temporary identities.
3. Self-identification is one of the ground features of an object.
4. An object knows the identity of its owner if it exists.

One more important mechanism in identity management is to give an object opportunity to cover its identity if needed. As IoT systems cover different backgrounds of the humans' everyday life, there are situations when it is not safe to reveal identities only based on incoming requests. If look more precise at Identity management we can propose to follow next recommendations:

1. **Secure Authentication/Authorization, Cm#6:**
    - Ensure that strong usernames and passwords are always required.
    - Two factor authentication and users' credentials encryption should be implemented if possible.
    - Be aware of insecure password recovery mechanisms and be sure that "re-authentication is required for sensitive features" [39].
    - Revoking mechanism should be developed for the system's credentials.
    - Application, device and server authentication are always required.
    - "Manage authenicated user id(credential info.) and the user's device id, the user's app id mapping table in the authentication server" [39].
    - Authentication token/session should always be unique to each user along with user id, app id and device id.
2. **Secure Web Interface, Cm#7:**
    - Default username and password should be non trivial and its recommended to change them during initial setup.
    - "Forgot password" functionality should be secure and sturdy. Be sure that you do not provide user with information indicating valid account.
    - Always check if your Web Interface is protected from XSS (Cross-site Scripting) [40], SQLi (SQL injections) [41] or CSRF (Cross-site request forgery) [42];
    - Do not hesitate to use only encrypted transport protocols while transferring system's credentials.
    - Do not allow usage of low level passwords.
    - Set appropriate number of attempts that allowed while logging in to the system. If the number was exceeded it is recommended temporary block the user for further authentication procedures if needed.
3. **Secure Mobile Interface, Cm#8:**
    - Default usernames and passwords should always be changed during initial setup.

- Ensure you system is protected from account enumeration through password reset mechanisms
- Set a limit for unsuccessful login attempts, so if it was reached the user is temporary blocked.
- Try not to transfer system's credentials over the internet.
- Two factor factor authentication should be implemented.
- Its recommended to use obfuscation technique [48] applied to mobile app of the system.
- Restrict the mobile app's execution on tempered OS environment" [39].

Countermeasures to secure authentication and/or authorization (Cm#6) mitigate risks with vulnerabilities of insufficient authentication and/or validation (V#2); to secure Web interface (Cm#7) – vulnerabilities of insecure Web interface (V#1); and to secure mobile interface (Cm#8) – vulnerabilities of insecure mobile interface (V#7).

4.3.4 Trust and governance, Cm#9:

Trust and governance the fundamentals in each IoT system. Represented by mechanism which dynamically evaluates objects it helps to control users' services based on the interaction process. In pair with governance trust play role of a framework which key role is to support cohesion and stable work of the security protocols in the system. This key plays crucial role for the systems which services rely on Cloud interface. According to [39] we can recommend next countermeasures:

- Default usernames and passwords should always be changed during initial setup.
- Ensure you system is protected from account enumeration through password reset mechanisms.
- Set a limit for unsuccessful login attempts, so if it was reached the user is temporary blocked.
- Cloud-based interface should be always protected from XSS (Cross-site Scripting) [40], SQLi (SQL injections) [41] or CSRF (Cross-site request forgery) [42];
- Try not to transfer system's credentials over the internet.
- Two factor factor authentication should be implemented.
- Monitoring system should detect abnormal service request traffic and block gateways if necessary.

Countermeasures regarding the trust and governance (Cm#9) deal with the security risks with vulnerabilities of insecure cloud interface (V#6).

4.3.5 Fault tolerance, Cm#10:

Clearly, no IoT system can pretend to be entirely secure, since the number of possible threads rising significantly faster then the number of solutions for covering new-appeared vulnerabilities in the system. Accomplishing acceptable fault tolerance in IoT requires next interdependent efforts:

1. All objects have to be secure by default, it means that beside secure protocols and algorithms the comprehensive software structure should always be improved.

2. The state of the network and its services should be shared among all IoT objects. This will provide an opportunity to maintain states' monitoring management on appropriate level as each object would be able to associate its own state changes with the network state changes. Constituently, the system monitoring quality will grow up.

3. Based on the second effort, each object should be able to protect themselves if the network state said about network collapse or attack. To arrange this functionality intrusion-detection system should be introduced as well as other defensive structures.

4. Force system to separate normal users from administrative users.

5. Provide functionality to encrypt data data at rest or in transit.

6. Force users to use only strong passwords during authorization/authentication processes.

7. "Ensuring the ability to enable logging of security events and to notify end users of security events"

Countermeasures regarding the fault tolerance (Cm#10) mitigate security risks with vulnerabilities of insufficient security configurability (V#8).

5 Analysis of Security Risks in IoT Systems

In this section we discuss few reported examples. The purpose is to illustrate the IoT assets, risks, and their vulnerabilities; potentially the countermeasures discussed in Section 3 should be applied to mitigate these risks.

	Example 1 [52]	Example 2 [53]	Example 3 [54]
Threat agent	An attacker with means to break normal work flow of the popular web services.	An attacker with means to interrupt heating process of the buildings.	An attacker with means to slow down the work flow of the university servers.
Attack method	1) Infection IoT devices using Mirai botnet; 2) Overload servers using infected devices.	1) DDoS attack on the heating network.	1) Botnet which makes system do DNS lookups every 15 min; 2) Brute force method to hack weak passwords.
Vulnerability	1) Devices' Linux kernel version ran out of date; 2) Users did not change the default usernames/passwords on their devices.	1) Network was not under monitoring.	1) Weak passwords; 2) Low level network activity monitoring.
Impact	1) Major websites servers' overload; 2) Loss of routers and IP cameras confidentiality; 3) Loss of websites' services reliability.	1) Two buildings were left unheated in the freezing temperature; 2) Loss of heating controllers' reliability; 3) Loss of network confidentiality.	1) Harm to the system's network; 2) Harm to servers; 3) Negation of integrity of the system's data.

	Example 4 [55]	Example 5 [56]	Example 6 [57]
Threat agent	An attacker with means to publish the victim's data or perpetually block access to it unless a ransom is paid uses malicious software from cryptovirology.	An attacker with means to control another's car remotely.	An attacker with means to stole personal data or change stored data in the doll's DB.
Attack method	Malicious software from cryptovirology.	Uploading malicious software to the control module.	Bluetooth connection attack (Bluebugging).
Vulnerability	Was not mentioned but allegedly - low level monitoring of the system's incoming data (Ransomware attacks are typically carried out using a Trojan that is disguised as a legitimate file that the user is tricked into downloading or opening when it arrives as an email attachment).	Insecure transport protocol (UConnect (module installed on Chryslers) uses the GSM network to access Internet but it was unsecured).	An insecure Bluetooth connection.
Impact	1)Harm to 70% of the city's CCTV systems.	1) Harm to transport protocol of the UConnect system; 2) Negation of integrity of the car control commands; 3) Loss of vehicle control.	1) Loss of users' data confidentiality; 2) Negation of integrity of the data stored on the doll's database .

Based on the tables above we can determine next security risks:

Risk 1: An attacker with means to break normal work flow of the popular web services uses Mirai botnet exploits that devices' Linux kernel version ran out of date and users did not change the default usernames/passwords on their devices which leads to major websites servers' overload, loss of routers and IP cameras confidentiality and loss of websites' services reliability.

Here the *business asset* is the cameras' IP supported by the Web services, Linux kernels, and other devices. The risk is possible because of the vulnerabilities V#2 (in Web service) and V#7 (in IoT device). It could be mitigated by a set of countermeasures selected from Cm#6 and Cm#8.

Risk 2: An attacker with means to interrupt heating process of the buildings uses DDoS attack on the heating network exploits that system's network was not under monitoring which leads to the fact that two buildings were left unheated in the freezing temperature, loss of heating controllers' reliability and loss of network confidentiality.

In this examples the *business asset* is the data sent over the network. This data is supported by (i.e., sent over) the heating network. The risk becomes possible because of the vulnerability V#3 in the network. This risk could be mitigated by countermeasures selected from Cm#1.

Risk 3: An attacker with means to slow down the work flow of the university servers uses Botnet and brute force password hack method and exploits low level network activity monitoring and weak passwords which leads to harm to the system's network, servers and data.

The *business asset* is the data used in the university workflow. This data is supported by the university and protected using passwords. The risk becomes possible because of the vulnerabilities V#3 in the network (i.e., the University servers) and V#2 in the provided service (i.e., activity/workflow supported by the server). This risk could be mitigated by a set of countermeasures selected from Cm#1 and Cm#6.

Risk 4: An attacker with means to publish the victim's data or perpetually block access to it unless a ransom is paid uses malicious software from cryptovirology and exploits (allegedly) the low level monitoring of the system's incoming data which leads to harm to 70% of the city's CCTV systems.

In this example the *business asset* is the victim's data supported by the network. The risk becomes possible because of the vulnerabilities V#3 in the remote storage and V#8 in the IoT system. This risk could be mitigated by a set of countermeasures selected from Cm#1 and Cm#10.

Risk 5: Two white hat-hackers with means to control another's car remotely used malicious software uploaded to the UConnect control module through insecure transport protocol (UConnect (module installed on Chryslers) uses the GSM network to access Internet but it was unsecured) and were able stop remotely Jeep Cherokee in the highway.

Here, the *business asset* is the car control commands supported by the UConnect module to sent them over the GSM communication channel. The risk becomes possible because of the vulnerability V#4 in the transport protocol (i.e., UConnect). This risk could be mitigated by countermeasures from Cm#2.

Risk 6: A white-hat attacker with means to stole personal data or change stored data in the “My friend Cayla” doll’s DB, claimed to be the first world interactive doll, uses Bluetooth connection attack to exploit an insecure Bluetooth connection and modified doll’s database. As a result, "My friend Cayla" was banned by German government, qualified as an “espionage device”.

The *business asset* is a date supported by (sent through) the Bluetooth connection and (stored in) the doll’s database. The risk becomes possible because of the vulnerabilities V#3 in the data storage (i.e., doll’s database) and V#4 in the communication (i.e., Bluetooth communication). This risk could be mitigated by a set of countermeasures selected from Cm#1 and Cm#2.

Vulnerability itself represents only characteristic of an asset or a group of assets and describes weakness of the system. However, if we apply threats to different vulnerabilities a risk of a negative impact on system assets occurs. As a result, we can conclude that the combination of threats and vulnerabilities represents a main reason for both the risk event and its harmful consequences for the system [1].

6 Framework application for the IoT system

To illustrate the SRM process we decided to take as an example arguably one of the most known IoT system - Smart Home [27]. Since Smart Home is considered as an essential domain in IoT, we can use its wide architecture to show how such automated IoT system raises a great concern of the privacy and security due to heterogeneity of the interconnected elements and capability to be controlled remotely.

The basic idea about the project is to build a simple smart home prototype using different types of sensor modules controlled by Raspberry Pi, which will then store aggregated data on cloud. Consequently, the user will be able to get data from the cloud, monitor real time system state and send commands to the control center. Then show how such system could be effected through uncovered vulnerabilities of its elements. And finally apply previously developed framework in order to mitigate possible risks.

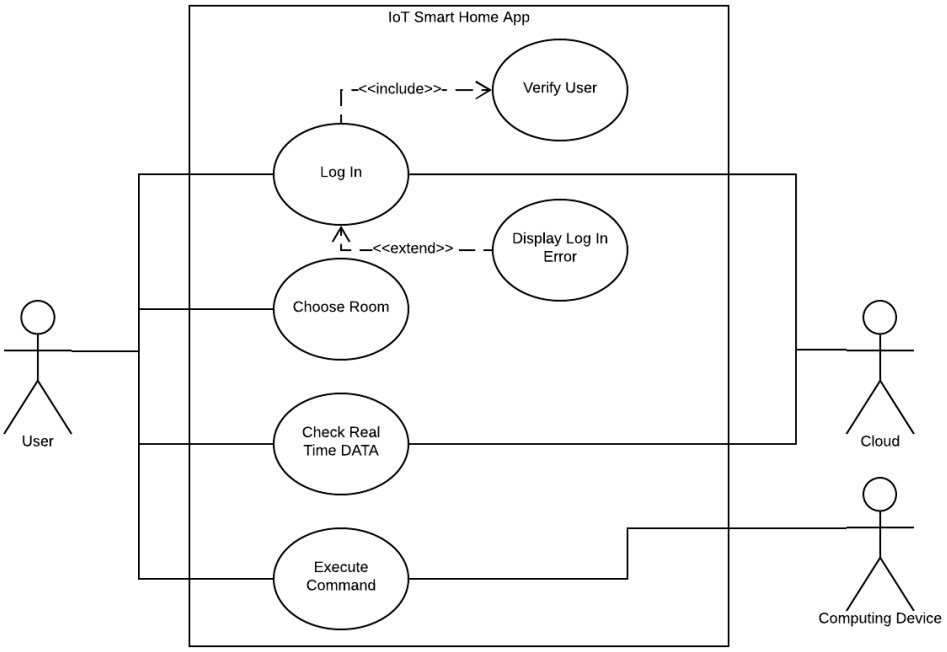

Figure 6. Smart Home App Use Case Diagram.

6.1 IoT system architecture

Lets start with the Figure 4, which represents Smart Home App Use Case Diagram. We can determine three main actors: 1) *User*; 2) *Cloud*; 3) *Computing Device*. *User* can Log In into the App, Choose Room to check DATA from and based on the data Execute Commands. *Cloud* as an actor takes part in Log In and Checking Real Time DATA processes, while *Computing Device* executes commands received from the *User*.

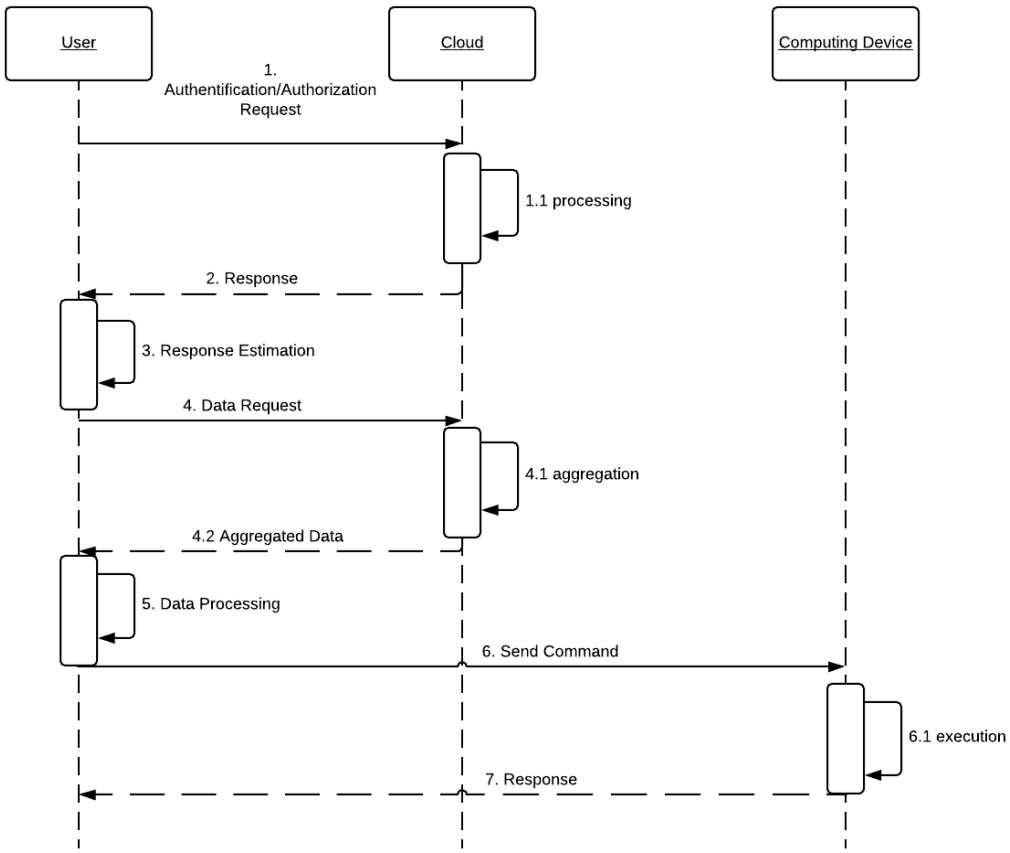

Figure 7. sHome App Sequence Diagram.

If we look at the Figure 5, we can see how the actors from the Figure 4 communicate with each other. *User* sends authentication/authorization request with credentials to the *Cloud*, which processes incoming request and, if credentials are matched, gives positive response to the *User*, otherwise negative. Consequently, based on the *Cloud's* response *User* will be given an access to the Smart Home application functionality or asked to check Log In credentials. If the Log In processes finishes successful *User* can request

Real Time DATA from the *Cloud*, which will be displayed in the App. After evaluating DATA *User* can send command to the *Computing Device*, which will then try to execute received command and provide *User* with appropriate response message.

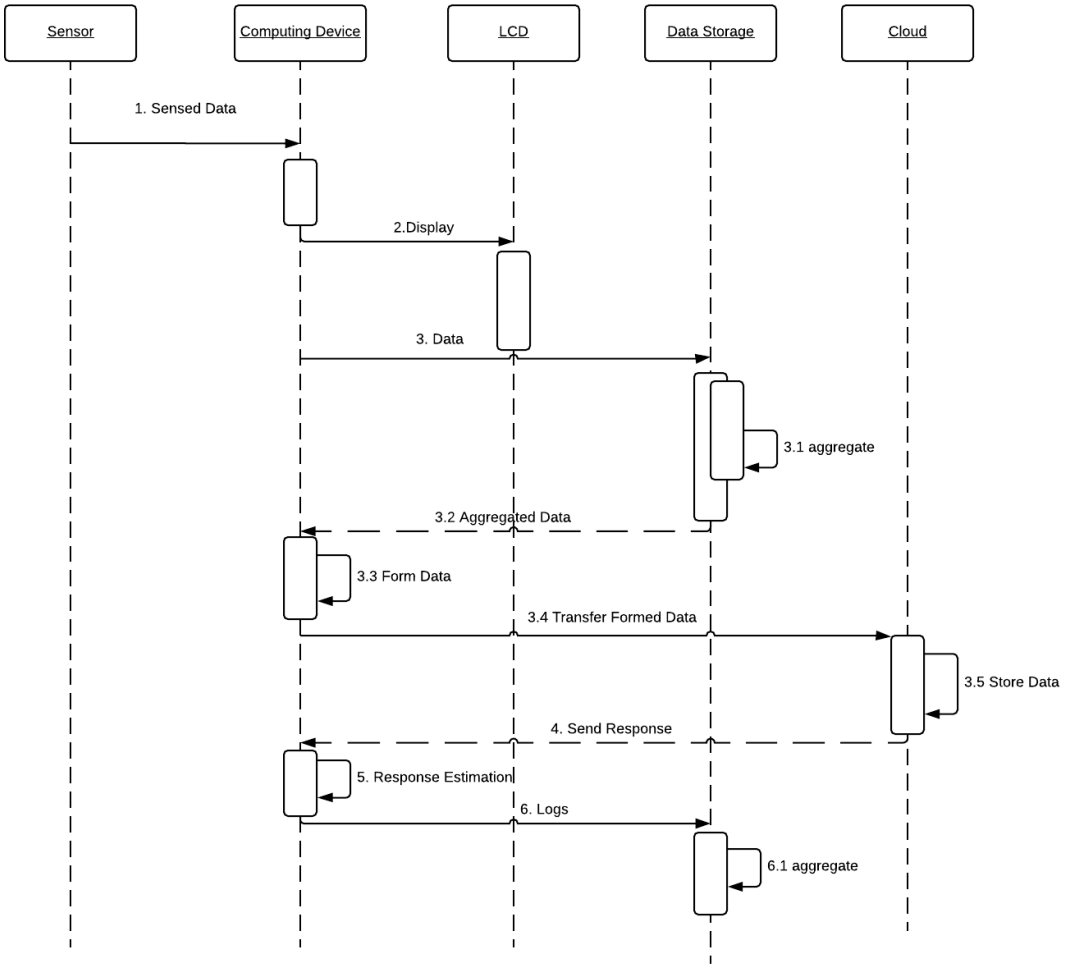

Figure 8. sHome IoT system Sequence Diagram.

Now lets dive dipper into the actual Smart Home IoT system architecture. From the Figure 6 we can see that actual work flow of the system starts from collecting Sensed Data from each available *Sensor* in the system. Then this Data is displayed on *LCD* screen, so you can monitor system state at home without using application. Sensed Data then is stored in local Data Storage and, when data from each sensor is aggregated, it is formed on *Computing Device* and transfered to the *Cloud*. On *Cloud* data is stored in database and *Computing Device* receives response from *Cloud* if the data was stored

successfully or not. Based on received response *Computing Device* create Logs and store them in the local *Data Storage* as a .log file.

6.2 Application security

This subsection will explain how we will try to secure system and on which level. Will be based on previous subsections' info and personal contribution.

6.3 Testing results

This subsection will describe results of influencing system's work before and after applying previously developed security framework.

7 Conclusion

References

[1] Raimundas Matulevičius. FundSecSysMod: Fundamentals of Secure System Modelling. http://www.springer.com/gp/book/9783319617169#aboutBook. (year 2017)

[2] Martin Bauer, Nicola Bui, Jourik De Loof, Carsten Magerkurth, Andreas Nettsträter, Julinda Stefa, Joachim W. Walewski. IoTRefMod: Enabling Things to Talk ("IoT Reference Model" pp 113-162). https://link.springer.com/book/10.1007/978-3-642-40403-0#about. Publisher Name: Springer, Berlin, Heidelberg (year 2013)

[3] Anth 'ea Mayzaud, R 'emi Badonnel, and Isabelle Chrisment. MonSecIoT: Monitoring and Security for the Internet of Things. Universit 'e de Lorraine, LORIA, UMR 7503, France Inria Grand Est - Nancy, France. https://link.springer.com/chapter/10.1007/978-3-642-38998-6_4.

[4] GSMA Connected Living: Understanding the Internet of Things (IoT) (2014)

[5] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, Faiz Alotaibi: Internet of Things security: A survey. Journal of Network and Computer Applications. Volume 88 Issue C, June 2017.

[6] Mahmoud Ammar, Giovanni Russello, Bruno Crispo: Internet of Things: A survey on the security of IoT frameworks. Journal of Information Security and Applications. Year 2017.

[7] The Guardian: DDoS attack that disrupted internet was largest of its kind in history, experts say. https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet (2016)

[8] Na Chen, Bo Rong, Abdel Mouaki, Wei Li: Self-Organizing Scheme Based on NFV and SDN Architecture for Future Heterogeneous Networks. Springer Science+Business Media New York 2015.

[9] Abomhara, M., Koien: Security and Privacy in the Internet of Things: Current Status and Open Issues. In: International Conference on Privacy and Security in Mobile Systems (PRISMS), (2014)

[10] Alabaa, F.A., Othma, M., Abaker, I., Hashem, I.A.T., Alotaibib, F.: Internet of Things security: A survey. Journal of Network and Computer Applications 88(15), 10–28 (2017)

[11] Mauro Conti, Ali Dehghantanha, Katrin Franke, Steve Watson: Internet of Things security and forensics: Chalenges and opportunities. Future Generation Computer Systems. Year 2017.

[12] Hui Li, Xin Zhou: Study on Security Architecture for Internet of Things. Haidian, Beijing, China. ICAIC 2011: Applied Informatics and Communication pp 404-411.

[13] Li, H., X., Z.: Study on Security Architecture for Internet of Things. In: ICAIC 2011, Part I. vol. CCIS 224, pp. 404–411 (2011)

[14] Hellaoui, H., Koudil, M., Bouabdallah, A.: Energy-efficient mechanisms in security of the internet of things: A survey. Computer Networks 127, 173–189 (2017)

[15] Irene Mavrommati, George Birbilis, John Darzentas: A conceptual framework for the design of IoT architectures that support end-user development. Networking Science: December 2013, Volume 3, Issue 1–4, pp 71–81.

[16] Sanaz Rahimi Moosavi, Tuan Nguyen Gia, Amir-Mohammad Rahmani, Ethiopia Nigussie, Seppo Virtanen, Jouni Isoaho, Hannu Tenhunen: SEA: A Secure and Efficient Authentication and Authorization Architecture for IoT-Based Healthcare Using Smart Gateways. 6th International Conference on Ambient Systems, Networks and Technologies. Year 2015. https://www.sciencedirect.com/science/article/pii/S1877050915008133#!

[17] Banerjee, M., Lee, J., R., C.K.K.: A blockchain future to Internet of Things security: A position paper. Digital Communications and Networks (2018)

[18] Dubois, E., Heymans, P., Mayer, N., Matulevi˘cius, R.: A Systematic Approach to Define the Domain of Information System Security Risk Management, pp. 289–306. Springer (2010)

[19] Nick Antonopoulos, Lee Gillam: Cloud Computing: Principles, Systems and Applications. Springer-Verlag London Limited 2010.

[20] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, Dechao Qiu: Security of the Internet of Things: perspectives and challenges 17 June 2014. Springer Science+Business Media New York 2014. https://link.springer.com/content/pdf/10.1007%2Fs11276-014-0761-7.pdf

[21] Yang, X., Li, Z., Geng, Z., Zhang, H.: A Multi-layer Security Model for Internet of Things. In: IOT Workshop 2012. vol. CCIS 312, pp. 388–393 (2012)

[22] Ammar, M., Russello, G., B., C.: Internet of Things: A survey on the security of IoT frameworks. Journal of Information Security and Applications 38, 8–27 (2018)

[23] Thomas Kirste: Smart Environments: True Visions. Springer-Verlag Berlin Heidelberg 2006.

[24] Joseph M. Sussman: Perspectives on Intelligent Transportation Systems (ITS). Massachusetts Institute of Technology Cambridge, Massachusetts. Year 2005.

[25] Vipindev Adat, B. B. Gupta: Security in Internet of Things: issues, challenges, taxonomy, and architecture. 13 June 2017. © Springer Science+Business Media, LLC 2017. https://link.springer.com/content/pdf/10.1007%2Fs11235-017-0345-9.pdf

[26] Valérie Issarny, Georgios Bouloukakis, Nikolaos Georgantas, Benjamin Billet: Revisiting Service-Oriented Architecture for the IoT: A Middleware Perspective. ICSOC 2016: Service-Oriented Computing pp 3-17 20 September 2016.

[27] Zahrah A. Almusaylim, Noor Zaman: A review on smart home present state and challenges: linked to context-awareness internet of things (IoT) https://link.springer.com/article/10.1007/s11276-018-1712-5

[28] Rodrigo Roman, Pablo Najera, and Javier Lopez. SecIoT: Securing the Internet of Things. University of Malaga, Spain. https://www.nics.uma.es/pub/papers/1633.pdf

[29] Viviana Bastidas, Marija Bezbradica, Markus Helfert: Cities as Enterprises: A Comparison of Smart City Frameworks Based on Enterprise Architecture Requirements. 26 May 2017.

[30] Ashish Gehani and Gershon Kedem. RheoStat: Real-Time Risk Management. Department of Computer Science, Duke University. https://link.springer.com/chapter/10.1007/978-3-540-30143-1_16

[31] Kwang-Ryul Jung, Aesoon Park, and Sungwon Lee: Machine-Type-Communication (MTC) Device Grouping: Algorithm for Congestion Avoidance of MTC Oriented LTE Network. Springer-Verlag Berlin Heidelberg 2010.

[32] GSM Assosiation. UnderIoT: Understanding the Internet of Things (IoT). July 2014. https://www.gsma.com/iot/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf

[33] Rebika Rai, Chunkey Lepcha, Partha Pratim Ray, Prashant Chettri. GDMA: General Domain Model Architecture of the IoT. October 2013. https://www.researchgate.net/publication/259497112_GDMA_Generalized_Domain_Model_Architecture_of_Internet_of_Things

[34] Theo Dimitrakos, Juan Bicarregui and Ketil Stølen. CORAS: CORAS - A Framework for Risk Analysis of Security Critical Systems. April 2002. https://www.ercim.eu/publication/Ercim_News/enw49/dimitrakos.html

[35] EBIOS: EBIOS - is a method for analysis, evaluation and action on risks relating to information systems. Year 2000. https://en.wikipedia.org/wiki/EBIOS

[36] Andreas Ekelhart, Stefan Fenz and Thomas Neubauer. AURUM: AURUM - A Framework for Information Security Risk Management Year 2009. https://pdfs.semanticscholar.org/f107/871d55daba22e55d0afe594c19339d882cb9.pdf

[37] MEHARI: MEHARI - MEthod for Harmonized Analysis of RIsk. Year 2010. https://en.wikipedia.org/wiki/MEHARI

[38] SANS Institute. CRAMM: CRAMM - A Qualitative Risk Analysis and Management Tool. https://www.sans.org/reading-room/whitepapers/auditing/qualitative-risk-analysis-management-tool-cramm-83

[39] OWASP: OWASP - Open Web Application Security Project is a 501(c)(3) worldwide not-for-profit charitable organization focused on improving the security of software. https://www.owasp.org/index.php/Main_Page

[40] XSS: XSS - Cross-site Scripting. https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

[41] SQLi: SQLi - SQL injection is a code injection technique, used to attack data-driven applications. https://en.wikipedia.org/wiki/SQL_injection

[42] CSRF: CSRF - Cross-site request forgery, also known as one-click attack or session riding. https://en.wikipedia.org/wiki/Cross-site_request_forgery

[43] BufOver: BufOver - Buffer overflow, or buffer overrun, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations. https://en.wikipedia.org/wiki/Buffer_overflow

[44] FuzA: FuzA - Fuzz testing or Fuzzing is a Black Box software testing technique. https://www.owasp.org/index.php/Fuzzing

[45] DoS: DoS - Denial-of-service attack. https://en.wikipedia.org/wiki/Denial-of-service_attack

[46] SSL/TLS: SSL/TLS - Secure Sockets Layer/Transport Layer Security. https://en.wikipedia.org/wiki/Transport_Layer_Security

[47] MQTT: MQTT - Message Queue Telemetry Transport. https://www.hivemq.com/blog/mqtt-security-fundamentals-payload-encryption

[48] CO: CO - Code obfuscation. https://www.ncsc.gov.uk/content/files/protected_files/guidance_files/Code-obfuscation.pdf

[49] SB: SB - Secure Boot. https://docs-old.fedoraproject.org/en-US/Fedora/18/html/UEFI_Secure_Boot_Guide/chap-UEFI_Secure_Boot_Guide-What_is_Secure_Boot.html

[50] The Internet: This article about the worldwide computer network. Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Internet

[51] V: V - Vulnerability in terms of cyber-security. https://www.techopedia.com/definition/13484/vulnerability

[52] Ex1: Ex1 - Mirai botnet. https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

[53] Mathews, L.: Hackers Use DDoS Attack To Cut Heat To Apartments. https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-apartments-without-heat/#45631d451a09

[54] Weagle, S.: IoT-Driven Botnet Attacks US University. https://www.corero.com/blog/798-iot-driven-botnet-attacks-us-university.html

[55] Khandelwal, S.: Two Romanians Charged with Hacking Police CCTV Cameras Before Trump Inauguration. https://thehackernews.com/2017/12/police-camera-hacking.html

[56] Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway - with me in it. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[57] Carolina: Goodbye Spy Toy: Germany Bans My Friend Cayla Doll. https://www.hackread.com/good-bye-spying-toy-germany-bans-my-friend-cayla-doll/

Appendix

I. Glossary

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Shapaval Raman**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   Security Risk Management for the IoT systems

   supervised by Raimundas Matulevicius

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, dd.mm.yyyy