A Conceptual Model to Support Security Analysis in the Internet of Things

Orestis Mavropoulos[1], Haralambos Mouratidis[1], Andrew Fish[1],
Emmanouil Panaousis[1], and Christos Kalloniatis[1,2]

[1] School of Computing, Engineering, and Mathematics
University of Brighton, Brighton, UK
{o.mavropoulos,h.mouratidis,andrew.fish,e.panaousis}@brighton.ac.uk

[2] Department of Cultural Technology and Communication
University of the Aegean, Lesvos, Greece
chkallon@aegean.gr

Abstract. This paper proposes a conceptual model to support decision makers during security analysis of Internet of Things (IoT) systems. The world is entering an era of ubiquitous computing with IoT being the main driver. Taking into account the scale of IoT, the number of security issues that are arising are unprecedented. Both academia and industry require methodologies that will enable reasoning about security in IoT system in a concise and holistic manner. The proposed conceptual model addresses a number of challenges in modeling IoT to support security analysis. The model is based on an architecture-oriented approach that incorporates sociotechnical concepts into the security analysis of an IoT system. To demonstrate the usage of the proposed conceptual model, we perform a security analysis on a small scale smart home example.

Keywords: Internet of Things, Conceptual Model, Security.

1. Introduction

An area that has attracted a lot of attention from research and industry alike is the Internet of Things (IoT). The vision of IoT is not new but has existed since the early conception of computer networks. Weiser, in 1991 [43], provides one of the most accurate yet simple visions of IoT by stating that the most profound technologies merge with the environment. He states that technology will be so evident that we will start perceiving it as a natural part of life. IoT along with cloud computing promises to make that statement into a reality.

As IoT becomes more integrated into our daily lives, the concerns surrounding its security aspects are growing at an alarming rate. The research community has already identified a number of security challenges in many areas of IoT [6,14,18]. In terms of security, a prominent concern is Denial of Service (DoS) attacks in embedded devices [40], since such devices lack the resources to withstand repeated requests from malicious attackers. Man-In-The-Middle attacks are another acknowledged issue [9] that take advantage of either weak encryption algorithms of embedded devices or weak authentication mechanisms among the systems [27]. Security researchers have found common vulnerabilities in many IoT devices that could have been prevented if simple security measures were taken into consideration during the development cycle [36]. The practice of implementing security analysis during development ensures that the final product will meet specific security standards. The security standards, in turn, will ensure its robustness when the product is actively deployed in real life scenarios. The method of including secure practices early in the development cycle is advocated by the field of requirements engineering. Requirements engineering analysis is applied by identifying the stakeholders' requirements in the development cycle to produce security requirements [7].

IoT systems allow the network integration of a variety of different devices. Devices such as personal computers, mobile phones, and printers can be considered traditional devices since they have been used in networking scenarios in the past. Traditional devices are used to access the Web, share files or host websites. Devices such as light bulbs, cars or heart monitors are only now gaining networking capabilities and as a result they have significant security flaws [39]. IoT is unique in the sense that it brings together old and robust technology with new and untested technology. The pairing of mature technology with immature technology naturally results in security issues. Given the unique challenges faced by IoT systems, our main question in this paper is how a security engineer can elicit security requirements in IoT systems? The previous question can be expanded to the following research questions:

1. How can IoT systems be modeled in order for an engineer to elicit security requirements of the IoT system?
2. As IoT systems exhibit a high degree of interdependencies, which stakeholders are responsible for satisfying each security requirement?
3. How can we support decision makers to select security controls in IoT systems?

The proposed conceptual model contributes to addressing the first of the above research questions. It is going to be part of a security framework that will address all three research questions. The components of the security framework will be the following:

1. Terminology: used to define the terms that describe the concepts of the proposed security framework. The terminology will facilitate the reasoning about the security of an IoT system by establishing a common language among security engineers.
2. Modeling language: provides components to create IoT system model that will capture the information needed by a security engineer in order to perform security analysis for an IoT system. Part of the modeling language will be the conceptual model presented in this paper. Other parts of the language will be the language semantics and language notation. The models created using the conceptual model will be able to be expressed in both graphical and textual notation. The textual notation will be used to facilitate computer aided analysis, while the graphical notation will be used to assist in visual analysis.
3. Methodology: used to create model instances of an IoT system for security requirements elicitation by security engineers. The methodology will provide instructions as well as restrictions on how modeling instances are created using the modeling language.
4. Analysis: the process of eliciting information from model instances. A form of analysis is the modeling of threats that impact model instances along with the assets that need to be protected. Other types of analysis will include the verification of the proposed security mechanisms along with the propagation of threats based on the compromised nodes of the IoT system.

The conceptual model enables reasoning about security in IoT systems using information retrieved by studying the hardware architecture of an IoT system. The proposed model is based on our previous work [31], which models the hardware architecture of an IoT system with its core concepts being the *IoT node* and the *network connection*. The architecture of a system offers valuable information for security analysis, such as the supported protocols of network connections between nodes or the flow of data inside a network domain. On the other hand, [31] does not express certain aspects of a system, such as user interaction or authentication mechanisms. Thus, the proposed conceptual model aims to address the limitations of a hardware architectural approach by introducing non-hardware architectural concepts along with hardware architectural concepts. Each concept is grouped into different modules based on their thematic context, to allow a security engineer to only use the modules she needs. Since IoT has computer networking components, concepts from computer networks such as *network connections* and *network domains* as well as concepts from non domain specific modeling languages such as *actor*, are incorporated in the proposed metamodel. The security requirements concepts of the metamodel are based on the Secure Tropos method [33]. Secure Tropos was chosen because it is an established requirements engineering method whose security concepts align with other requirements methods such as [16].

Our work here extends [31], by introducing:

- additional concepts to the metamodel that capture additional information from an IoT system. For example, we introduce the concept of Net to express external networks to an IoT system. Moreover, concepts that are introduced in the present metamodel can be used to represent social constructs, such as users and stakeholders, or security constructs, such as threats and vulnerabilities. Social concepts are used to model users activity or behavior, while security concepts are used to model security issues.
- IoT systems are a network of various devices. Our initial attempt in [31] was to model an IoT system in a similar manner to a computer network. In our present work, we introduce concepts to express users, threat or security policies. Such concepts have little relation to computer networks. For that reason, the proposed metamodel divides its concepts into different modules based on their thematic context. Concepts that are used to express network related constructs constitute the network module, while security related concepts constitute the security module. A security engineer may only need to model the network aspect of an IoT system. If that is the case, she only needs to use concepts of the network module.

1.1. Outline

The structure of the paper is the following: Section 2 describes related work in the fields of IoT, Security Requirements and Security Requirements in IoT. Section 3 presents the proposed conceptual model, explains its concepts and describes its modular approach to IoT security modeling. Section 4 displays a security analysis of a smart home using the proposed metamodel. Section 5 discusses the future plans of this research and concludes the paper.

2. Related work

IoT, due to its magnitude and relatively young age, may be considered the technological field with largest attack surface today [37]. The insecurity of IoT is demonstrated by a large number of surveys that have identified a variety of security issues and challenges found in IoT systems [1, 4, 20, 29]. However, there have been few attempts to address its security issues from a security requirements point of view.

Some academic works identify security challenges in IoT while proposing future directions that security researchers should take. For example, Kumar et al. conceptually breaks down an IoT system into different layers and proceeds to identify the security challenges of each layer and states the options that researchers have to prevent such issues [24].

An attempt was made to provide a framework for security and privacy in IoT systems using requirements engineering by Alqassem [2]. They identify the complexity of analyzing security in IoT systems and states that the key components in IoT are only two: RFID systems and networks of sensors. To reason about security in IoT, they propose the use of the $i*$ framework in order to undertake security analysis in future case studies. In the paper, other technologies and topologies that are common in IoT systems, are not considered. For example, IoT is not restricted to RFID systems but is able to use any communication technology, such Wi-Fi, NFC or Bluetooth. Moreover, architectural topologies are not restricted to networks solely comprised of sensors but may include any type of device capable of using a network protocol.

IoT systems as a whole are composed of a multitude of devices. Many of those devices are embedded devices. An informative paper from Gürgens illustrates a vision in applying security engineering to embedded systems [15]. In their paper, they identify a number of security challenges faced by embedded systems that should be addressed in order to have a secure system. Furthermore, they reason that security requirements tools should be designed and tailored to the needs of embedded systems.

Babar et al. propose a framework aiming to provide security in embedded IoT systems [5]. They propose a basic three-step security framework to elicit requirements in embedded systems, by identifying the building blocks of embedded systems in IoT. Tian et al. design a security framework specific to wireless sensor networks [41]. The framework proposes a system architecture that is broken down into eight modules. Each module has specific functionality to mitigate security issues. In summary, the presented works do not view IoT in a comprehensive manner. They only aim to mitigate security issues in specific domain areas. Accordingly, they cannot be used to offer a universal security analysis to any IoT scenario, but only aim to address specific instances of IoT systems. While it is important to argue about domain specific IoT security issues, there is a need to reason about IoT security in a holistic manner that does not limit a security engineer [35].

A number of issues and open challenges with the integration of IoT and Cloud computing are identified by Díaz [8]. It is argued that IoT is only made possible through a cloud infrastructure. Some IoT systems will use sensors as a service from a third party provider, while other IoT systems may use cloud services to offload heavy processing functions. Díaz states that IoT will function as a middle-ware that will transmit all its data to the cloud for processing. The paper shows that the current trend for IoT application development is based on Cloud computing.

Ikram et al. [17] propose an alternative approach to modeling IoT. They argue that the complexity of IoT can be modeled in a similar manner to chemical computing models. Their work presents a chemical reaction-inspired computational model using the concepts of graphs and reflection, which attempts to address the complexities associated with the visualization, modeling, interaction, analysis and abstraction of information in the IoT. Laghari et al. [26] model a self-adaptive architecture for managing the Carbon footprint in a corporate environment using a heterogeneous multiagent system. They use Cognitive Agent-based Computing (CABC) to create models. Both works propose concepts to model social constructs. They were not designed to perform security analysis and as such do not propose security related concepts.

ThingML is developed as a domain-specific modeling language which includes concepts to describe both software components and communication protocols. The formalism used is a combination of architecture models, state machines, and an imperative action language [42]. ThinkML is supported by a set of open source tools that are built using the Eclipse Modeling Framework. ThingML was developed to model the hardware, software components and communication protocols of IoT systems. It does not have concepts to model social or security components of IoT, such as users, stakeholders, threats or vulnerability.

ASSIST is an agent-based simulator of Social Internet of Thing (SIoT) [21]. The idea behind SIoT is that smart objects will connect with each other to form social networks. ASSIST uses an agent-based approach by defining three types of agents: Device Agents, Human Agents, and Task Agents. While ASSIST can be used to express the social components of IoT systems, it was not designed with security analysis is mind. As such it cannot be used to express security components.

In their work, Ge et al. [11], proposed a framework for modeling and assessing security in an IoT system. The framework has a graphical security model that evaluates the level of security using specific security metrics. The security of an IoT system is assessed in a comprehensive manner and in not limited to a specific IoT scenario, such as embedded systems or RFID systems. Another framework that separates security requirements of IoT systems depending on their architectural layer is made by Rahman et al. [34]. In their work, they propose a four layer approach, with each layer having different security needs. They state that their framework can be used by other researchers to build new security solutions for IoT. In our work, we use a similarly comprehensive approach to security of IoT systems although from a requirements engineering point of view.

3. The proposed conceptual model

In this work, we present a conceptual model for reasoning about security in IoT systems during the implementation phase. The development of the conceptual model draws inspiration and uses similar concepts from a number of requirements engineering frameworks [11, 12, 16, 33]. It is used to model the hardware, software and social components of IoT systems in order to analyze their security. The hardware architectural components of an IoT system are emphasized in the model. Using an architectural approach, an IoT system can be analyzed in a similar manner as a traditional computer network. We use the International Telecommunication Union (ITU) definition of IoT, as “A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [19].

The presented model extends our previous work on the APPARATUS framework [31]. The initial model of APPARATUS consisted of two concepts. The first concept was the *IoT Node* and the second concept was the *Network Connection*. The IoT node was used to represent devices of IoT systems, while the Network Connection was used to represent how the IoT nodes can communicate with each other. Using those two concepts we modeled the hardware architecture of IoT systems in order to analyze their security.

In this work, we extend the model by introducing new concepts to represent social and security components. The *IoT node* has been renamed to *Device* and additional concepts have been introduced in order to represent the hardware components of the IoT system. The metamodel is presented via a UML class diagram. Each class represents a concept that can be used to describe specific objects of IoT systems. Each concept has a number of properties that further describe it in the system. The UML diagram of the model is shown in Fig. 1. The concepts of the presented model have been divided into different modules. The metamodel consists of three modules:

1. **Network module:** is used to model network objects of IoT systems. The Network module is considered the core module of the metamodel. Every other module is designed as an extension to the Network module. This modeling choice was made to give emphasis to the interconnecting nature of IoT systems. The Network module is represented with the color blue in Fig. 1.
2. **Social module:** extends the Network module in an object-oriented manner with social concepts. Social concepts are used to model users and stakeholders. The social module is represented with the color gray in Fig. 1.
3. **Security module:** extends the Network and Social modules with security concepts. The security concepts are used to model threats, assets, security controls and attackers in an IoT system. The concepts used by the Security module are heavily influenced by Secure Tropos security concepts [12]. The security model is represented with the color purple in Fig. 1.

The presented metamodel is used to model an IoT during at the implementation phase. During the implementation phase, a security engineer has more detailed knowledge of an IoT system. For example, during the implementation, the security engineer knows the type of network protocols that are used by the system, the type, and versions of the devices that provide services to the system. All the concepts of the metamodel, unless otherwise stated, have the property of *description* which describes the concept in the system. The modules of the implementation phase metamodel along with their concepts are the following:

Network Module

1. **Device**: the concept of *Device* was initially named IoT node in [31]. It is defined as a hardware component of an IoT system. A restriction on the model is that Devices can only have a single functionality. If a Device has more than one function, it has to be represented as different Devices. For example, a laptop running a server (1st function) and client (2nd function), has to be expressed as two separate *virtual* Devices that belong to a parent *physical* Device that is the laptop. The properties of the Device are:

(1) aspect: declares whether the Device is a single node, or composed of sub-nodes. The aspect *physical* means that a Device is a parent Device and may be composed by more *virtual* Devices; *(2) layer*: the conceptual layer of the IoT architecture to which the Device belongs. APPARATUS uses a three-layer architecture that consists of the Application Layer, Network Layer and the Perception Layer [32, 44]. Other works identify other architectures that provide more levels of abstraction. For example, a SOA-based approach identifies five layers, *application, service composition, service management, object abstraction, objects* [4]. Another approach by Lu, identifies other layers, that are *application, middleware, coordination, backbone network, existed alone network, access layer, edge technology* [28]. The proposed architectures for Internet of Things have yet to fuse into a single reference model [23], for that reason we chose the three-layer approach. The three-layer approach provides the necessary properties for reasoning about security while allowing to be extended if more levels of abstraction are introduced into the final reference model of IoT. The layers of IoT architecture should not be confused with the OSI model [25] since the two models try to conceptualize different constructs and concepts. The *layer* concept takes an enumerated value as input that is one of *application, gateway, perception*; *(3) type*: defines the kind of the Device. A Device type may be a sensor, a mobile phone or a server; *(4) service*: is the type of role or operation that the Device performs for the network. This value may include network services such as *ssh*, *ftp*, data processing filtering and relaying of data; *(5) input*: what is required in order for the node to perform its role or operation. It takes an enumerated value as an input that is drawn from *dataEnvironmental, dataDigital, command, action, notification, trigger*; *(6) output*: is the result of the Device operation or role. It may take the same values as the *input* property. *(7) update:* represents the process of how the software aspect of the Device is being updated. The *update* can be *automatic*, require a specific *action* or *false*.

2. **Network Connection**: the type of network communication protocol used between the Devices. The properties of the network connection are: *(1) description*: the type of connection, it can either be *wireless*, signifying a connection using a wireless protocol or *cable*, signifying a connection using a wired medium. It takes an enumerated value as an input; *(2) listOfProtocols*: is a list of the network protocols that are supported by the network connection. It takes an array of string values as an input, each value in the array represents a supported network protocol.

3. **Micronet**: represents environments that a security engineer can configure in terms of their security. A Micronet is a managed environment that constitutes a collection of Things necessary for an IoT system to perform a function. Examples of Micronets are a smart home, an agricultural network of sensors or company's internal network. The properties of the Micronet are: *(1) state*: represents if the Devices in the Micronet remain in the same location. The *state* can either be *dynamic*, meaning that the Devices in the network change network domains during their usage or *static* meaning that the Devices in the system do not change network domains. Examples of *dynamic* IoT systems are networks of vehicular fleets, drones, and other mobile devices since devices in such networks move distances geographically. Examples of *static* IoT are smart homes and industrial IoT systems since devices in such systems are stationary during their usage.

4. **Net**: represents external networks to the IoT system. While Micronet represents environments that have their security configured by a security engineer, Net represents environments that their security configuration is not known. Examples of a Net are external networks to the IoT system that a security engineer has little knowledge of, such as a third party cloud infrastructure or hostile deployment environments. During security analysis, the Net is considered a hostile environment. Communication between a Micronet and a Net is under the assumption that the Net is compromised.
5. **Unidentified Node**: is a Device that is not directly connected to a Micronet and a security engineer has little knowledge of. It may be a malicious device or a legitimate device that is not known by the system. For example, it can be an unauthenticated laptop from a legitimate user trying to connect to an office network or it can be a laptop operated by a malicious attacker trying to compromise the system.
6. **Data**: information that is produced or stored by a Device. Examples of Data is information stored in a database or user passwords. The property of Data is: *(1) location*: corresponds to the geographical location of the Data stored in the device. It can be used to represent if Data are physically stored inside a network or are hosted by a third party service. Moreover different regions have different laws regarding digital information that ultimately affect the overall security of a system.

Social Module

1. **Actor**: is used to represent people or groups of people that interact with an IoT system [12]. An Actor may be a stakeholder of the system. The concept of Actor can be used to represent groups of people with different privileges, such as root users or the administration personnel of a University. An Actor may never be malicious. To represent malicious Actor, the concept of the Security Module, *Malicious Actor* is used. The property of the Actor is the following: *(1) intent*: describes what an Actor wants to achieve or gain by interacting with the IoT system.

Security Module

1. **Malicious Actor**: is a person with malicious intent. Malicious Actors are used to representing attackers or insider threats. The concept of the malicious actor is a generalization of the concept Actor.
2. **Asset**: any Actor, Device or Data of the system that either (1) is considered valuable by the stakeholders and needs to be protected; of (2) a malicious actor wants; or (3) acts as a stepping stone for further attacks. Examples of Assets are the access credentials known by an actor, sensitive user information stored in a database or a sensor that has read/write privileges to a server.
3. **Threat**: a malicious function, or system that has the means to exploit a vulnerability of a legitimate system. A Threat can only target an Asset of the IoT system. The property of the Threat is: *(1) threatType:* represents the classification of the Threat according to the STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Elevation of Privilege) [38]. It takes an enumerated value.
4. **Vulnerability**: a software, hardware or usage policy weakness that can be exploited by a *Malicious Actor* towards compromising an IoT system. Hardware and software Vulnerabilities can be identified using techniques such as penetration testing.

5. **Constraint**: a restriction related to security issues, such as privacy, integrity and availability, which can influence the analysis and design of an IoT system under development by restricting some alternative design solutions, by conflicting with some of the requirements of the system, or by refining some of the system's objectives [12]. Constraint has the following property: *(1) propertyType:* how the Constraint is classified according to the extended CIA (Confidentiality, Integrity, Authentication, Authorization, Non-repudiation, Availability) [3]. It takes an enumerated value.
6. **Mechanism**: a Mechanism, when implemented, protects against one or more Vulnerabilities.

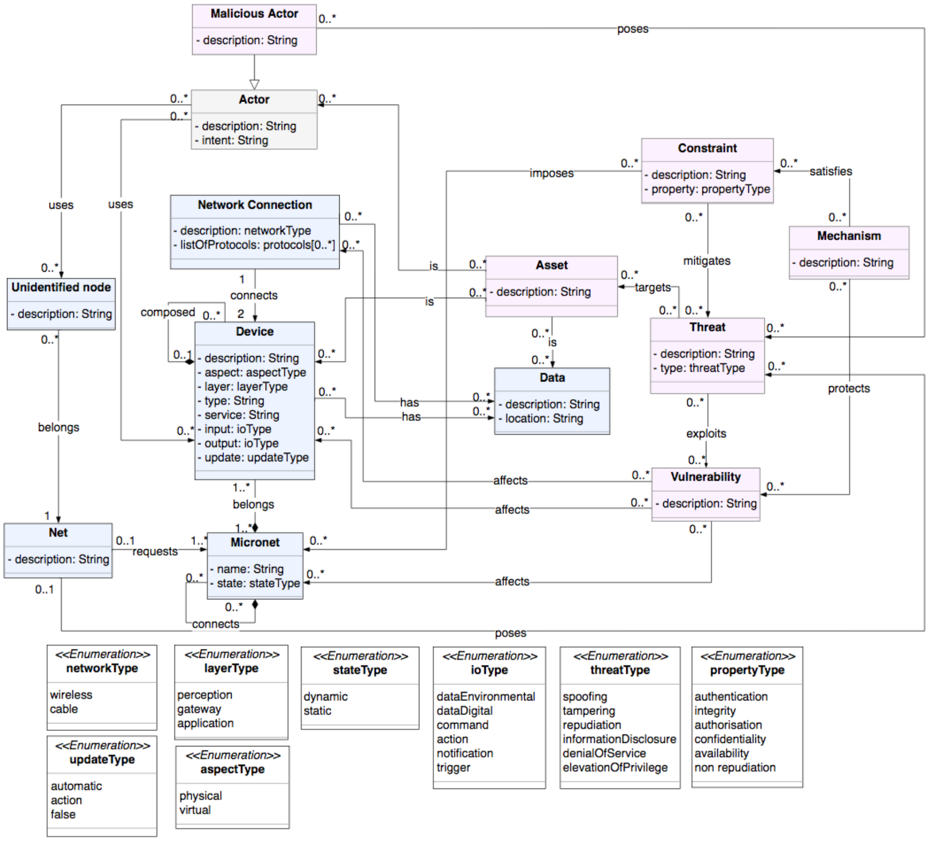

Fig. 1. Metamodel of APPARATUS

Even though the metamodel can be used to produce valid modeling instances, there are some design decisions that act as constraints. The design decisions are not visible in the UML diagram. Hence, they are considered *restrictions* on the model instances. The restrictions in the model instances are:

Constraints on the Metamodel

1. If a Device has a more than one function, it has to be represented as a cluster of Devices, with each Device in the cluster having a single function. This choice adds more nodes in the model. For example, a Device with one hundred functions has to be represented as 101 nodes. One node is used to represent the Device and the remaining 100 nodes are used to represent the functions of the Device. In terms of security, each function is a potential vulnerability that has to be addressed. By having each function represented as an individual node, other concepts can act upon the Device function nodes to represent Threats, Mechanisms that are present in the system due to the function of the Device. Another option to represent the different functions of a Device would be to add multiple functions as attributes inside a single node that constitutes the Device. Although it is a cleaner approach given the fact that it does not introduce more nodes, other concepts cannot directly interact with individual functions but with the node as a whole. For example, a vulnerability may result from a specific function of a Device. The proposed mitigation mechanisms should only affect that specific vulnerability. If the vulnerable function from the Device is removed, the same would apply to the mitigation mechanism.
2. As demonstrated by the metamodel in Fig. 1 the concept of *Threat* can only target the concept of *Asset*. Security analysis is only made on concepts that are considered Assets by the stakeholders or the security engineer.
3. The properties that have enumerated values must only take one of the enumerated values.
4. An *Actor* may never be malicious, but must only have legitimate intentions on using an IoT system. Attackers or insider threats have to be represented by the concept of *Malicious Actor*.
5. We consider IoT systems as systems deployed in hostile environments. The concept of Net is used to represented systems that we cannot configure in terms of security. As such they are considered compromised and malicious. On the other hand, Micronets represents systems whose components can be configured in terms of their security. Since Micronets security can be configured, Devices communicating inside Micronets share a level of trust. When a Device from a Micronet communicates with an Unidentified Node from the Net it does so in the lowest level of the trust and always verifies the exchange. For security purposes, Micronets and Nets have a Zero Trust Network architecture [22].

4. Apparatus concepts based on IoT system modeling

To illustrate how the proposed metamodel can be used, we will perform an illustrative example of security analysis in an IoT system. In the interest of space, the example is designed to showcase all of the features of the metamodel with the least components possible. It is important to note that the security analysis presented is neither exhaustive nor is a realistic security analysis of a network system. The allowed space does not permit an in-depth security analysis. Instead of providing with a monolithic example of an IoT system modeled using the metamodel's concepts, specific instances of the system will be shown. Each instance will be part of the IoT system with the aim of exhibiting specific functionality of the metamodel. We have not yet developed a front end visual language for use by engineers. The analysis of the system will be done by creating UML model instances of the IoT system shown in Fig. 2.

The scenario follows a young couple with a newborn baby. To monitor the baby's activity the couple installs a baby monitor that can be accessed from the Internet. The system is composed of a baby monitor that functions as a camera, the Philips in Sight B120/37 and Macbook laptop. The two devices have access to the Internet through a NETGEAR R8500 router. The baby monitor can be accessed outside of the local network through a third party service. The scenario is similar to commonly deployed IoT systems found in domestic environments. A user buys the IoT equipment, installs it in his home and controls it using a third party application. The processing or authentication is being performed by a third party application on its servers, so in reality, it is a decentralized network. Since the processing is taking place on the Cloud, the IoT system requires an Internet connection in order to function properly.

The hardware components of the network are the following: (1) baby monitor, (2) router, (3) laptop, (4) web server. The laptop can be used to view the camera feed from inside the local network. To access the camera feed from outside the network a user must access the camera through the third party web server. The web server is outside the private network of the application and has to be accessed using an Internet connection, while the baby monitor, router and laptop are connected to the same Local network of the user. In Fig. 2 the network layout of the IoT system is shown.

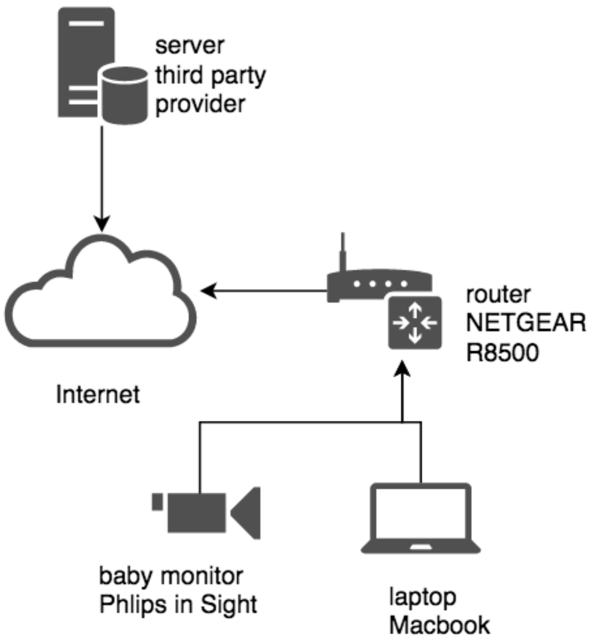

Fig. 2. Hardware components of the IoT system

For our example, the stakeholders of the IoT system provide us with the following security requirements:

SR1 Camera feed from the baby monitor should only be accessible by authorized users.
SR2 Devices of the smart home should be physically protected.

In order to group together Devices with specific functionality the concept of Micronet is used. The smart home under analysis has one Micronet. The Devices of the Micronet are the baby monitor, the laptop, and the router. During our security analysis, we make the assumption that the Devices in the Micronet will remain connected to the same network domain. To represent that assumption the Micronet has the property of *state: static*. The web server that enables users to access the baby from outside the smart home's local network is part of the Net. Because we cannot configure the security of the web server and have limited information about it, we represent the web server as an Unknown Node. The Unknown Node is connected to the Net. In Fig. 3 we show the view of the high-level components of the IoT system.

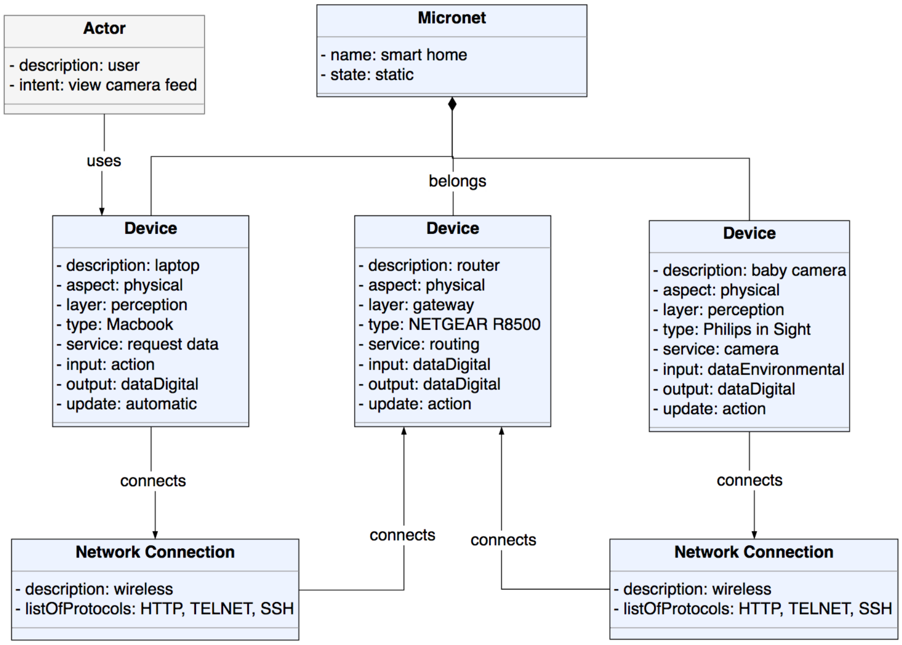

Fig. 3. Network and Social constructs of the IoT system

We identify the following Assets of IoT system:

A1 camera feed of the baby monitor.
A2 credentials to remotely access the smarts home's router.
A3 physical aspect of the smart home's Devices.

The security analysis will be performed on the Micronet of the smart home. We express the components of the smart home using the Network and Social modules of the metamodel as shown in Fig. 3. The smart home Micronet is composed of three Devices. The Devices are a laptop, a baby monitor, and a router. The router provides the Micronet with wireless connectivity, so each Device can communicate using a wireless medium. For the purposes of the example, the only supported network protocols would be HTTP, Telnet, and SSH. We represent a legitimate user with the intent of viewing the camera feed as an Actor.

Due to the paper's size limitations, instead of using a single model for our analysis, we will create a separate model for each Asset. Normally, the analysis is performed in the same model.

A1: camera feed of the baby monitor. To satisfy SR1 we specify an authorization Constraint on the users. The baby monitor is a Philips in Sight B120/37. The Asset that we want to protect is the Data that is being transmitted by the baby monitor. By performing a vulnerability search of that model we identify a direct browsing vulnerability, CVE-2015-2884 [1]. The remote viewing stream is created by a proxy connection to the camera's internal web service via the cloud provider and is bound to a public hostname and port number without credentials. The attacker can locate the hostname and port number and access the camera feed. To mitigate the vulnerability we could create a rule to whitelist the allowed users. The intent of the Malicious Actor is to *view the camera feed*. The model of the security analysis is shown in Fig. 4.

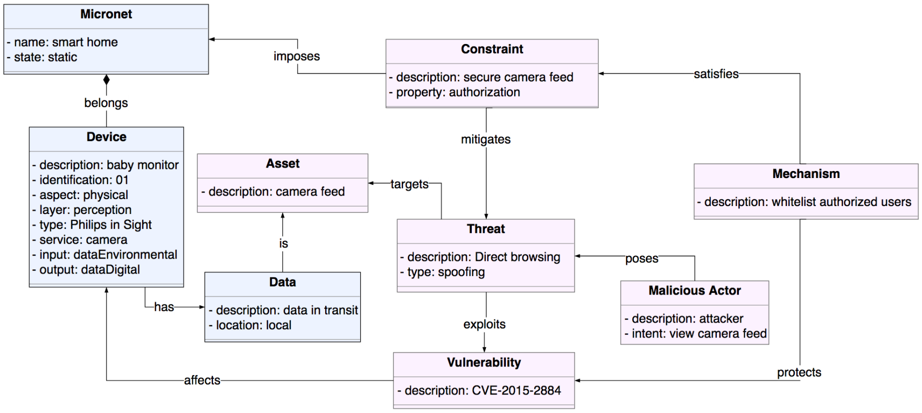

Fig. 4. Threat modeling of A1

A2: credentials to remotely access the smarts home's router. The router is a Device that belongs to the gateway IoT layer. It acts as the bridge between the local network of the smart home and the remaining of the Internet. If the router is compromised then an

[1] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2884

attacker can access the camera feed of the baby monitor. To satisfy SR1 the router's credential must be protected. The router's model is NETGEAR R8500. Those models have a password disclosure Vulnerability, the CVE-2017-5521 [2]. When trying to access the web panel, a user is asked to authenticate; if the authentication is canceled and password recovery is not enabled, the user is redirected to a page that exposes a password recovery token. To mitigate the CVE-2017-5521, the proposed Mechanism is to update the firmware of the Device. The router has the property *update: action*, meaning that a user has to manually apply the update as shown in Fig. 5. The intent of the Malicious Actor, in this case, is to *access the router*.

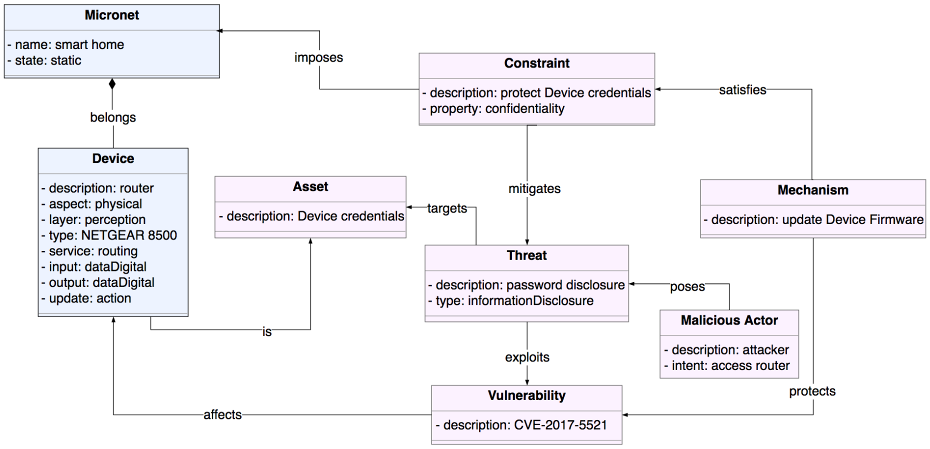

Fig. 5. Threat modeling of A2

A3: physical aspect of the smart home's Devices. In Fig. 6, the Assets are the physical aspect of the Devices of the smart home. Protecting those Assets is the SR2. The security issue is that an attacker can physically access the Devices of the system. Physical access allows an attacker a variety of attacks such as installing rootkits, backdoors, stealing data and physically damaging the Devices. The Threat is the *physical tampering* of the devices and the Vulnerability it exploits is that the devices are *physically accessible* by an attacker. The Security Constraint that mitigates the Threat is that the *devices of the system should be physically protected*. The proposed Security Constraint satisfies the SR2 and it is implemented by the Mechanism of *physically secure access to the devices*. The idea is that the system's devices will be placed in a secure location that an attacker will not have access to. The Threat of *physical tampering* is manifested by the Malicious Actor with the intention of stealing the Devices.

The Constraints from Fig. 4, Fig. 5 and Fig. 6 may be used to develop a security policy that each device in the Micronet must follow. Each component in the smart home's Micronet must adhere by the security policy imposed by the Constraints of the system.

[2] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5521

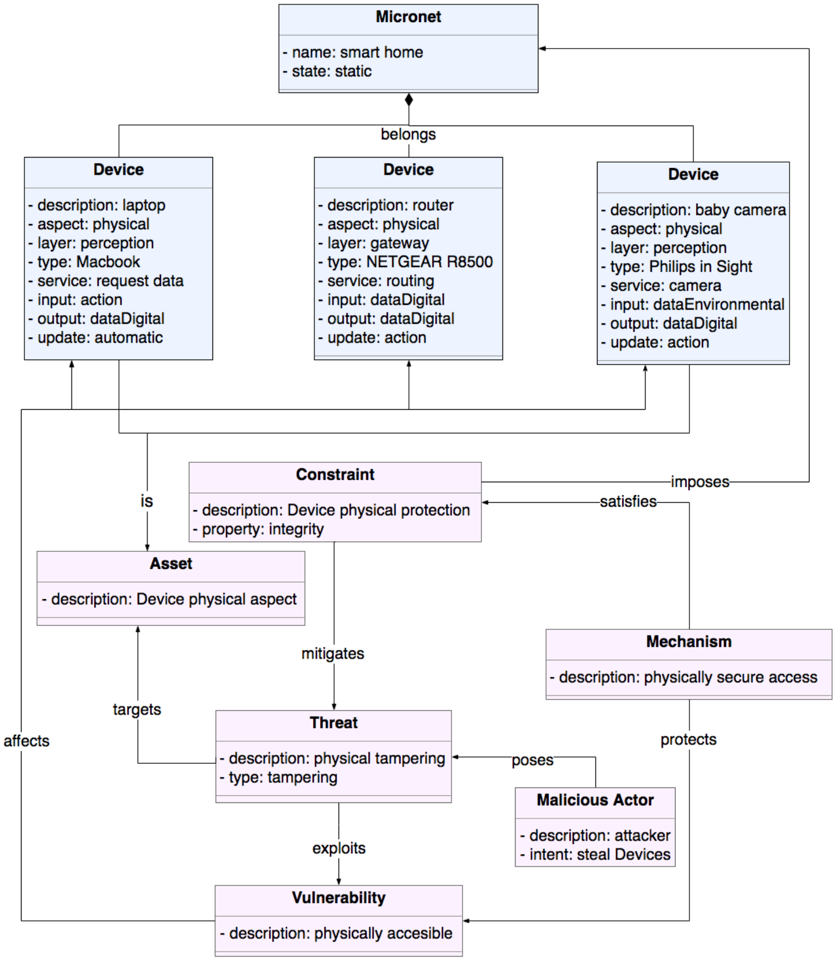

Fig. 6. Threat modeling of A3

If the stakeholders introduce a new Device in the smart home, Constraints in Fig. 7 must be enforced in order for the system to be secure. For example, one of the enforced Constraints is to ensure the protection of the physical aspect of the Devices. When Devices are introduced in the Micronet they must be physically protected.

In Table. 1 a presentation of the security concepts that were used, showing the Constraints, Mechanisms, Threats, Vulnerabilities, and Assets is presented.

Our aim with the provided examples was to only exhibit how models of IoT systems can be used to elicit security requirements and perform threat analysis. For example, the Threat of physical tampering on Devices has more implications that making a device unavailable. An attacker can perform Man-In-The-Middle attacks, physically steal data or deploy backdoors to the network. Similar issues present security challenges that need to be addressed by a security engineer in order for an IoT system to be secure.

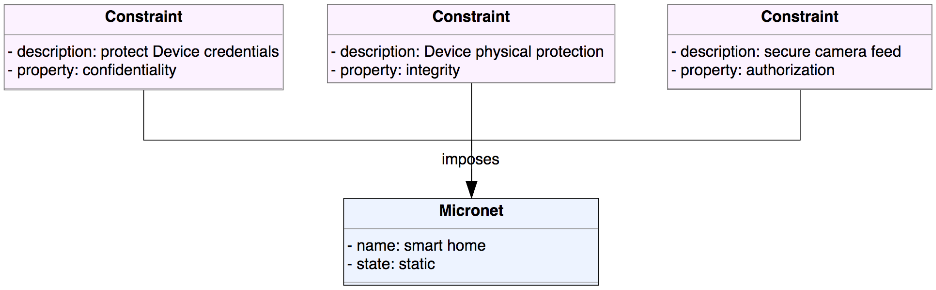
Constraint
- description: protect Device credentials
- property: confidentiality
Constraint
- description: Device physical protection
- property: integrity
Constraint
- description: secure camera feed
- property: authorization
imposes
Micronet
- name: smart home
- state: static

Fig. 7. Security policy of smart home

Table 1. Presentation of security concepts of smart home

Asset		
camera feed	Device credentials	Device physical aspect
Constraint		
secure camera feed	password disclosure	Device physical protection
Threat		
direct browsing	protect Device credentials	physical tampering
Mechanism		
whitelist authorised users	update Device Firmware	physically secure access
Vulnerability		
CVE-2015-2884	CVE-2017-5521	physically secure access

Information encoded in a model can be used to deduce security issues that would not be otherwise apparent to a security engineer. The concept *Data* has the attribute of *location*, that shows the physical location of the stored data. The information can be used to understand the data flow of the network and the legal security requirements of data since countries have different laws regarding digital information. The attribute *state* of the Micronet, shows the movability of the system's devices. For example, IoT systems can be composed of vehicular fleets as well as stationary weather sensors. Moving devices that change gateway layers often, will have different security requirements than stationary devices. The attribute of *type* of the concept *Device* can be used as keywords to search for Vulnerabilities in vulnerability databases. In the presented security analysis example, we queried the CVE database for known vulnerabilities of two Devices in the smart home based on the value of *type*. A security engineer could create a list of the values of the property *type* and use it to query vulnerabilities databases for relevant vulnerabilities. The more information a security engineer has, the more comprehensive her analysis would be. An important issue in IoT is the existence of deployed vulnerable devices. Many of those devices will never receive security updates due to the stakeholder's inability to install them or due to the manufacturers lack of support. The vast number of vulnerable IoT devices is evident by million size botnets, such as the Mirai botnet. To model the ability of how a device can receive security updates, we introduce the *update* attribute in the concept of *Device*. Certain devices will not be able to receive any security update and as such will be vulnerable. If a device has the value of *false* in the *update* attribute it can be considered compromised in the context of security analysis.

The concept of *Network Connection* has the attribute of *type*, which is used to represent whether the connection between two Devices is wireless or wired. In the majority of cases, a wired connection is more secure than a wireless one since it is less prone to spoofing attacks. Wired connections also have a longer transmission range than wireless ones. Depending on the *type* a security engineer can propose different security mechanisms. For example, wireless connections, even short range ones, require encryption mechanisms to protect against replay attacks. Consider the case where an implanted heart defibrillator communicates without any encryption to external devices. Any attacker will be able to control it with fatal consequences. The other attribute of the *Network Connection* is the *supportedProtocols* which holds an array of the supported networks of the connection. In the presented example of the smart home, the supported network protocols were the HTTP, TELNET, and SSH. HTTP and TELNET are unencrypted protocols that are subject to a variety of attacks, most commonly spoofing attacks. On the other hand, SSH is a more secure protocol since it supports encryption by default. The knowledge of which network protocols are used between devices is helpful to a security engineer in a number of ways. For example, the NFC (Near Field Communication) has a range of 10cm. In order for spoofing attacks to be successful, an attacker has to be in close proximity to the target.

A design choice that may be considered a limitation is that security analysis can only be made on components that are considered *assets*. The general consensus in security engineering is that "*a system is as strong as its weakest link*", and in the case of APPARATUS it can be argued that a system can be compromised by a component that is not considered an asset. One of the definitions of *Asset* in APPARATUS is that: *(3) may act as a stepping stone to further attacks*. If a component can be used to compromise an IoT system, it is considered an *Asset* and as such has to be secured.

5. Conclusion

Given the dynamic nature of IoT systems and their vast applications, it is expected that their security specifications will differ from system to system. In the present paper, a conceptual model to express IoT system for security analysis was proposed. The conceptual model is used to create models to reason about security in IoT systems from a system's hardware architectural point of view. It consists of different modules, with each module extending its security analysis capabilities when needed. The core module, called *network* module, is used to map an IoT system in a similar manner to a computer network, by expressing its hardware architecture as a cluster of nodes in a network. In order to model users and stakeholders, the *social* module was proposed, that extends the core network module. Security concepts that allow security analysis were introduced in the *security* module, that extends both the network and social modules. The security module, along with information provided by the network and social modules are used to elicit security requirements.

To demonstrate how the concepts of the proposed metamodel can be used to model an IoT system, we perform a security analysis in a smart home. The smart home was composed of a baby monitor, a laptop, and a router. The smart home was modeled using the metamodel's concepts, with the smart home's network diagram as a basis. Based on the security requirements of the smart home's stakeholders, we identified three threats that impact the assets of the smart home. Based on the threats and the information of the model instance, we identified specific vulnerabilities that exist in the system. We proposed mechanisms to mitigate the vulnerabilities along with security constraints on the system to further secure it. The identified security constraints were used to create a security policy that was imposed on the system.

Future work aims to extend the metamodel with additional modules to enable other types analysis. Currently, we are planning on introducing two modules. One such module is the *event* module, that will introduce concepts more akin to event driven system security to facilitate dynamic security analysis. Whereas another module will be the privacy module, which will introduce privacy related concepts. Privacy in any context is not taken into account in the current metamodel.

The metamodel is supported by a software application [30] named *ASTo* that facilitates the security analysis of IoT systems. The application is hosted on Github under the MIT license. It is based on Electron [13] and Cytoscape.js [10].

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials 17(4), 2347–2376 (2015)
2. Alqassem, I.: Privacy and security requirements framework for the internet of things (iot). Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014 pp. 739–741 (2014)
3. Andress, J.: The basics of information security: Understanding the fundamentals of Infosec in theory and practice. Syngress Media,U.S., United States (2014)
4. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer Networks 54(15), 2787–2805 (2010)
5. Babar, S., Stango, A., Prasad, N., Sen, J., Prasad, R.: Proposed embedded security framework for internet of things (iot). 2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE) pp. 1–5 (2011)
6. Benabdessalem, R., Hamdi, M., Kim, T.H.: A survey on security models, techniques, and tools for the internet of things. 2014 7th International Conference on Advanced Software Engineering and Its Applications (2014)
7. Coles, E.S.: Analyzing and specifying security requirements in early stages of software development life cycle. Journal of Mobile, Embedded and Distributed Systems 7(2), 87–94 (2015)
8. Díaz, M., Martín, C., Rubio, B.: State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing. Journal of Network and Computer Applications 67, 99–117 (2016)
9. Du, J., Chao, S.: A study of information security for m2m of iot. 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE) (2010)
10. Franz, M., Lopes, C.T., Huck, G., Dong, Y., Sumer, O., Bader, G.D.: Cytoscape.js: a graph theory library for visualisation and analysis. Bioinformatics 32(2), 309 (2016)

11. Ge, M., Kim, D.S.: A framework for modeling and assessing security of the internet of things. In: 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS). pp. 776 – 781. Institute of Electrical & Electronics Engineers (IEEE), Melbourne, VIC (2015)
12. Giorgini, P., Mouratidis, H.: Secure tropos: A security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering 17(02), 285–309 (2011)
13. GitHub Inc: electron. `http://electron.atom.io/` (2015)
14. Granjal, J., Monteiro, E., Silva, J.S.: Security for the internet of things: A survey of existing protocols and open research issues. IEEE Communications Surveys & Tutorials 17(3), 1294–1312 (2015)
15. Gürgens, S., Rudolph, C., Maña, A., Nadjm-Tehrani, S.: Security engineering for embedded systems. Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems - S&D4RCES '10 (2010)
16. Haley, C., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering 34(1), 133–153 (2008)
17. Ikram, A., Anjum, A., Hill, R., Antonopoulos, N., Liu, L., Sotiriadis, S.: Approaching the internet of things (iot): a modelling, analysis and abstraction framework. Concurrency and Computation: Practice and Experience 27(8), 1966–1984 (2015)
18. Irshad, M.: A systematic review of information security frameworks in the internet of things (iot). 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (2016)
19. ITU: Global standards initiative on internet of things recommendation itu-t y.2060 (2012), `http://handle.itu.int/11.1002/1000/11559`
20. Jing, Q., Vasilakos, A.V., Wan, J., Lu, J., Qiu, D.: Security of the internet of things: Perspectives and challenges. Wireless Networks 20(8), 2481–2501 (2014)
21. Kasnesis, P., Toumanidis, L., Kogias, D., Patrikakis, C.Z., Venieris, I.S.: Assist: An agent-based siot simulator. In: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on. pp. 353–358. IEEE (2016)
22. Kindervag, J.: No more chewy centers: Introducing the zero trust model of information security. Forrester Research (2010)
23. Krco, S., Pokric, B., Carrez, F.: Designing iot architecture(s): A european perspective. In: 2014 IEEE World Forum on Internet of Things (WF-IoT). pp. 79 – 84. Institute of Electrical & Electronics Engineers (IEEE), Seoul (2014)
24. Kumar, S.A., Vealey, T., Srivastava, H.: Security in internet of things: Challenges, solutions and future directions. 2016 49th Hawaii International Conference on System Sciences (HICSS) pp. 5772–5781 (01 2016)
25. Kurose, J.F., Ross, K.W.: Computer networking: A top-down approach. Addison-Wesley Educational Publishers, Boston, 6 edn. (2012)
26. Laghari, S., Niazi, M.A.: Modeling the internet of things, self-organizing and other complex adaptive communication networks: a cognitive agent-based computing approach. PloS one 11(1), e0146760 (2016)
27. Liu, J., Xiao, Y., Chen, C.P.: Authentication and access control in the internet of things. In: 2012 32nd International Conference on Distributed Computing Systems Workshops. pp. 588–592. Institute of Electrical & Electronics Engineers (IEEE), Macau (2012)
28. Lu, T., Neng, W.: In: Future internet: The Internet of things. vol. 5, pp. 376–5. IEEE, Chengdu (2010)
29. Madhura, Jain, P., Ranjith, Bilurkar, N.: A survey on internet of things: Security and privacy issues. IJITR 3(3), 2069–2074 (2015)
30. Mavropoulos, O.: Apparatus. `https://github.com/Or3stis/apparatus` (2016)

31. Mavropoulos, O., Mouratidis, H., Fish, A., Panaousis, E., Kalloniatis, C.: Apparatus: Reasoning about security requirements in the internet of things. Advanced Information Systems Engineering Workshops 249, 219–230 (2016)
32. Miao, W., Ting-lie, L., Fei-Yang, L., Ling, S., Hui-Ying, D.: Research on the architecture of internet of things. vol. 5, pp. 484–5. IEEE, Chengdu (2010)
33. Mouratidis, H., Argyropoulos, N., Shei, S.: Security requirements engineering for cloud computing: The secure tropos approach. Domain-Specific Conceptual Modeling pp. 357–380 (2016)
34. Rahman, A.F.A., Daud, M., Mohamad, M.Z.: Securing sensor to cloud ecosystem using internet of things (iot) security framework. Proceedings of the International Conference on Internet of things and Cloud Computing - ICC '16 79(79), 1–5 (2016)
35. Roman, R., Najera, P., Lopez, J.: Securing the internet of things. Computer 44(9), 51–58 (09 2011)
36. Roy, S., Manoj, B.S.: Iot enablers and their security and privacy issues. Modeling and Optimization in Science and Technologies 8(8), 449–482 (2016)
37. Sadeghi, A.R., Wachsmann, C., Waidner, M.: Security and privacy challenges in industrial internet of things. Proceedings of the 52nd Annual Design Automation Conference on - DAC '15 54(54), 1–6 (2015)
38. Shostack, A.: Threat modeling: Designing for security. John Wiley & Sons, Indianapolis, IN (2014)
39. Stojmenovic, I., Wen, S., Huang, X., Luan, H.: An overview of fog computing and its security issues. Concurrency and Computation: Practice and Experience 28(10), 2991–3005 (2015)
40. Suo, H., Wan, J., Zou, C., Liu, J.: Security in the internet of things: A review. In: 2012 International Conference on Computer Science and Electronics Engineering. pp. 648 – 651. Institute of Electrical & Electronics Engineers (IEEE), Hangzhou (2012)
41. Tian, B., xian Yang, Y., Li, D., Li, Q., Xin, Y.: A security framework for wireless sensor networks. The Journal of China Universities of Posts and Telecommunications 17(2), 118–122 (2010)
42. Vasilevskiy, A., Morin, B., Haugen, O., Evensen, P.: Agile development of home automation system with thingml. In: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN) (2016)
43. Weiser, M.: The computer for the 21st century. Scientific American 265(3), 94–104 (1991)
44. Yang, Z., Yue, Y., Yang, Y., Peng, Y., Wang, X., Liu, W.: Study and application on the architecture and key technologies for iot. In: 2011 International Conference on Multimedia Technology. pp. 747 – 751. Institute of Electrical & Electronics Engineers (IEEE), Hangzhou (2011)