END-TO-END SECURITY IN THE INTERNET OF THINGS

Hunor SÁNDOR,
PhD Candidate, Technical University of Cluj-Napoca
Piroska HALLER, Associate Professor PhD,
„Petru Maior" University of Tîrgu Mureş
Gheorghe SEBESÉTY-PÁL, PhD,
Technical University of Cluj-Napoca

Abstract: *Internet of Things (IoT) is believed to play a major role in the future of Information and Communication Technology. Following this vision, IoT systems will be integrated parts of people's living environments and will provide comfort, safety and entertainment. Due to the proximity of deployment to the everyday life of people, security and privacy are particularly important factors in these systems. In this respect, IoT systems and applications have to be properly designed to guarantee secure and reliable services. In this work we propose a semi-automated threat and risk modeling method which provides end-to-end threat evaluation for IoT systems. This method is dedicated to be used in the design process of secure IoT applications or in the security assessment of IoT systems.*

Keywords: *Internet of Things, security, threat modeling, threat risk modeling, risk aggregation*

1. Introduction

Internet of Things (IoT) is one of the most discussed subjects in modern Computer Science. It initiated a new wave of innovations, which reorganized the core of traditional Information and Communication Technology (ICT) by introducing novel features, infrastructures and architectures for the Internet oriented services. The IoT paves the way for the implementation of complex cyber-physical systems, and it provides direct applications to improve the quality of everyday life. In this respect, we can find different IoT application domains [1], e.g., industrial, smart city and healthcare, with various implementations starting from smart homes [2] and smart cities [3], to smart hospitals [4] or smart grids [5].

A considerable group of IoT systems is able to control the Things connected to them, e.g., building automation systems. These features can provide various advantages, e.g., autonomous environments, intelligent

buildings or smart hospitals, but in the same time these smart features also involve *increased risks*, because in case of these applications the virtual activities from the cyber world are directly connected to the physical world. In this respect, an attacker from the cyber world can remotely reach the physical world and can perform malicious control operations on the physical Things. The impact of these attacks can be more severe than a traditional cyber attack, therefore, these systems have to be properly designed to avoid these unfortunate activities.

The National Institute of Standards and Technology (NIST) specifies that the basic security requirements for Industrial Control Systems (ICS) are *availability*, *integrity* and *confidentiality*, known as AIC model, and these typically have priority in this order [6]. This is the ability of the system to maintain its performance parameters under conditions of stress, i.e., disruptive cyber attacks. Nowadays ICSs tend to include a large variety of different embedded hardware devices in their infrastructure, ranging from sensors, actuators, through industrial equipments to traditional ICT devices. It can be observed that the basic properties of these new heterogeneous infrastructures perfectly match the conditions from the definition of IoT. In this respect, it can be considered that the IoT infrastructures also inherit the AIC requirements, but in case of different IoT applications the priority orders may be reorganized.

In the majority of cases IoT systems are built around the environment of people and a large variety of IoT devices provide information about the movement, interactions or behavior of people. In addition, an emerging technology is the development of wearable embedded devices which are believed to gain high popularity in the IoT technology, but unfortunately this can bring the cyber world and implicitly cyber attacks closer to humans. In this respect, the *privacy* of IoT users has to be treated on the same level with the system security [7][8].

Considering the aforementioned security properties and threats, the IoT hardware, software and the applied infrastructures have to be properly designed and configured to keep these large-scale systems secure. To make a system secure this perspective has to be taken in consideration starting from the design phase of the system. Therefore, the security assessment methods are important components in the design process. In this respect, dedicated methods and tools are required for the design and maintenance of secure IoT systems. In this work we propose a novel *semi-automated threat and risk modeling method* by bringing together and

extending wide-spread threat models, analysis methods and tools. We demonstrate the applicability of this method by applying it on a real-world professional distributed IoT platform.
The remaining of this paper is organized as follows. Section 2 provides a brief overview of related work. Then, Section 3 proposes an IoT data model, while Section 4 describes in details the proposed threat and risk modeling method. Finally, the conclusions are drawn in Section 5.

2. Related Work

Many studies discuss the vision, architectural considerations and challenging problems related to IoT [1][9]. One of the most challenging subjects in IoT is to guarantee security over heterogeneous infrastructures. Various surveys summarize the properties, unsolved challenges and/or applied technologies regarding IoT security [7][8][10][11][12].
In the scientific literature several guidelines and tools are described which are dedicated to security and risk assessment.
The STRIDE threat modeling approach and the DREAD threat risk model, both proposed by Microsoft, are the most popular methods for threat analysis. These will be presented in details in section 4. Trike is an open source threat modeling methodology and tool [13] similar to STRIDE/DREAD, with a risk based approach at its core. AS/NZS 4360, proposed by the Australian/New Zealand Standard, is a formal standard for documenting and managing risks [14] which follows the steps: establish context, identify the risk, analyze the risk, evaluate the risk, treat the risk. The guide [15] discusses the process of cyber-security assessment in detail and describes the Common Vulnerability Scoring System (CVSS). This guide states that the traditional cyber-security assessment steps are: assessment team establishment, test plan creation, attack vector identification, testing and reporting. Furthermore, Cyber Security Evaluation Tool (CSET) [16] is a question-answer based assessment tool which basically aims to assist with the correct configuration of systems by following standards and best practices.
Many approaches have been proposed which extend these methods with advanced features [17][18] and/or customize them to become applicable to specific fields of technology [19]. Nonetheless, only few works target the IoT domain and none of them provide an overall threat model or threat modeling method. The work [20] focuses on the threat modeling of

the perception layer of IoT, while in [21] a risk based security model is presented for IoT in eHealth. In this respect the main novelty of our work is that it provides a generic threat and risk modeling method which is applicable for IoT systems independently of architecture and application domain.

3. IoT Data Model

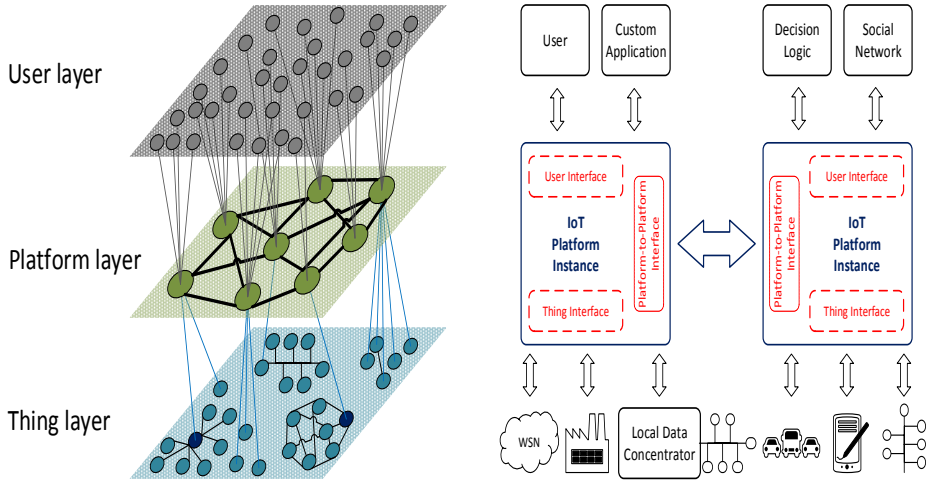

Fig. 1: a) layered architecture; b) communication model of an IoT system

The majority of IoT services and features are in a tight relation with the distribution and delivery of the information, i.e., 'Thing' data. In this respect the understanding and modeling of the data handling in IoT systems is a basic requirement for overall system security, reliability and performance analysis.

IoT platforms can be completely different by using various technologies, hardware and software to accomplish dedicated tasks. The communication within an IoT platform can be implemented using a wide variety of technologies, starting from Web services to hardware buses, e.g., RS-485, CAN, Modbus, and different wireless links, e.g., ZigBee. Nonetheless, the data flows inside these systems tend to be organized based on a similar schema (see Fig. 1a): the users (User layer) can access the resources exposed by the Things (Thing layer) through a middleware layer, referred to as Platform layer in the followings.

Since IoT platforms interconnect Things and users, each of them has to provide at least two types of communication interfaces for external

interactions: (i) one to connect the Things; and (ii) another for user connections. Using these, the IoT platform is able to provide access to the resources exposed by the Things for the users and other Things.
The tiny and cost-aware nature of the 'Things' generally limits their communication capabilities. Therefore in the majority of cases these are able to communicate only on short wired or wireless distances. Thereby, the physical placement of the IoT platform is imposed and limited by the geographical position of the Things connected to them. Sometimes it is feasible to place the IoT platform in the close vicinity of the connected Things, e.g., home automation, but in a significant group of application domains, e.g., Geographic Information Systems (GIS), the connected Things are physically distributed.
We model the problem of the physical distribution using two different approaches (see Fig. 1b): (i) IoT platforms can be distributed in more equivalent instances and interconnected through platform-to-platform communication edges; or (ii) Local Data Concentrator (LDC) entities can be defined, which are able to synchronize a group of local Things with a remote IoT platform instance. The former case requires the definition of a third type of communication interface for platform-to-platform communication used for sharing the resources between the platform instances. The latter case defines a new type of vertical communication.
Furthermore, we believe that the support for storing and retrieving certain information from the IoT platforms is a basic requirement for modern IoT platforms which may enable advanced features ranging from advanced surveillance capabilities to forecasts. In this respect, we consider that each IoT platform instance may use a local data storage to memorize information about the connected Things. Following this architecture, the totality of the interconnected IoT platform instances can be considered as a *fog platform* handling IoT specific data.

4. Semi-Automated End-to-End Threat and Risk Modeling

In this section we present several methods and tools which are widely used for security analysis and we propose a novel method for end-to-end threat and risk modeling in IoT systems. The proposed method is created by combining and extending the presented security analysis mechanisms.
According to the definition of the Open Web Application Security Project (OWASP) *threat modeling* is a method for analyzing the security of an

application (or system) [22]. This process consists of the identification and evaluation of security risks, and based on this knowledge, the determination of the mitigation techniques, methods or algorithms.

According to traditional threat analysis techniques the first step is to identify the main assets that need to be protected. For this, the first requirement is the adequate understanding of the analyzed system. This includes the identification of the interaction points with the external entities and the detailed mapping of the interactions between the different components of the system. The usage of the Data Flow Diagrams (DFD) in the process of the threat modeling is a common technique, because this provides a structured representation about the interactions inside the represented system.

The next step in the threat modeling process is the identification and ranking of the possible threats based on information gained in the previous step. To organize these activities into a structured form Microsoft has proposed the STRIDE approach [23][24] and the DREAD model [23][25]. The former is dedicated for threat categorization while the latter for threat risk evaluation.

The last step consists of the determination of the countermeasures and mitigation techniques. These can be software architectural decisions, code quality improvements, but also infrastructural or configuration improvements.

4.1. The STRIDE and DREAD Model

The STRIDE model [23][24] has been proposed as a basic component of the Microsoft Security Development Lifecycle (SDL) [26] for general threat categorization. STRIDE is an acronym of the defined threat groups, which are: *Spoofing*, *Tampering*, *Repudiation*, *Information Disclosure*, *Denial of Service*, and *Elevation of Privileges* groups.

Spoofing allows users to masquerade and to impersonate other users in order to gain access to possibly critical resources. *Tampering* is an action aimed to illegally modify the data handled (transported or stored) by the system. *Repudiation* is a malicious action with the goal to hide or change the authoring information of other prohibited operations. *Information disclosure* means reading data without granted access, *Denial of Service* is a malicious action with the goal to deny access to valid users, while *Elevation of Privilege* is a threat aimed to maliciously gain unauthorized access to resources.

DREAD is a classification scheme for threat rating [23]. Similarly with STRIDE, it is an acronym: each letter represents a component which has to be considered at the threat evaluation. These factors are: *Damage*, *Reproducibility*, *Exploitability*, *Affected Users*, and *Discoverability*. In this respect, the final risk score assigned to a thread has to be calculated as the sum of the ratings from each of the aforementioned categories [25].
Damage indicates the amount of damage caused if the threat is exploited. *Reproducibility* represents the level of ease to exploit the threat. *Exploitability* indicates the required skills and tools to exploit the threat. *Affected users* shows the amount of users which will be affected if the threat is exploited, while *Discoverability* expresses how easy it is to discover the threat.
According to the specification, ratings do not have to use a large scale because it highly complicates the rating of threats, therefore it is recommended to use a simple scheme such as *Low (1)*, *Medium (2)*, *High (3)*. Following this scoring system threats with a final score between 5 and 7 can be treated as Low risk, scores between 8-11 as Medium risk, while threats which have received a score between 12 and 15 as High risk. Based on the numerical values assigned to different threats, these can be compared and prioritized.

4.2. Semi-automated Threat Modeling

Microsoft has created the SDL Threat Modeling Tool [27] with the purpose of making threat modeling easier and to partially automatize it. This tool operates with the extended DFD representation of the analyzed system/software created with the embedded editor of the tool. These DFDs are extended with trust boundaries, which are the representations of the locations where the level of trust changes inside the represented software/system. In addition, a set of attributes are assigned to each component of the DFD, which represent several properties of the system/software. Based on this representation the tool is able to automatically detect the potential threads targeting the different components and the interaction lines of the represented system. In addition, it automatically categorizes the detected threats using the STRIDE approach. The automatic threat detection is based on a matchmaking mechanism which uses grammar for threat definition in Backus-Naur Form (BNF). These definitions are stored in an embedded database of the tool which is also extensible. In the process of the

automatic discovery the tool iterates over the given DFD and analyses it per interactions between the specified components.

4.3. End-to-End Threat and Risk Modeling Method

In the case of the above mentioned methods the threat detection, categorization, ranking and evaluation is always performed separately on the different components and interactions of the software/system. The IoT paradigm is built over a schema in which the thing-to-user and user-to-thing interactions are placed into the center. In addition, because of the complexity and heterogeneity of these infrastructures, the results of the different threat modeling and ranking activities may be highly varying in the different local segments of the end-to-end communication paths. Even if these local threat evaluation results hold important information about the overall security these do not express the state of the end-to-end security in a unified form. Therefore, an aggregation method is required to generate a unified expression which reflects the overall security state of the end-to-end data paths inside these systems.

In this respect, we propose the End-to-End Threat and Risk Modeling Method (EETRMM) which is able to calculate the unified risk indicators per STRIDE threat groups separately for all the data paths of a system represented by a DFD, i.e., end-to-end interactions. This method is organized in the following steps:

Step 1. Representation of the system in DFD form

Step 2. Identification of the threats

Step 3. Grouping of threats from **Step 2.** by the interaction sections where these were detected

Step 4. Categorization of threats detected in **Step 2.** in STRIDE groups

Step 5. Calculating the DREAD score for the threats identified in **Step 2.**

Step 6. Aggregation of the DREAD scores from **Step 5.** per STRIDE groups of interaction sections

Step 7. Detection of the end-to-end interactions in the system and assigning the threat groups from **Step 3.** and the aggregated DREAD scores from **Step 6.** to them

Step 8. Aggregation of the scores from **Step 6.** per end-to-end interaction paths determined in **Step 7.**

To perform **Step 1.**, **Step 2.**, **Step 3.** and **Step 4.** we employed the Microsoft Threat Modeling Tool. After the generation of the categorized threat list the tool stores all the information in a single file using the XML

description language. To extract the information needed in the next steps we parsed this XML.

Step 5. has to be performed manually by considering the information from the former steps and analyzing in depth the properties and configurations of the system at the hand.

Let us consider T the set of the identified threats, n the number of the interactions in the analyzed system, and $I_i \subset T$ the set of threats on the i^{th} interaction, where $1 \leq i < n$, $I_p \cap I_q = \emptyset, \forall\, p,q \in [1,n], p \neq q$, and $T = \bigcup_{i=1}^{n} I_i$. Let us also consider $S_{i,j} \subset I_i$ the set of the threats categorized in the j^{th} STRIDE group from the i^{th} interaction group, where $1 \leq j \leq 6$, 6 the total number of the STRIDE threat groups, $S_{i,v} \cap S_{i,w} = \emptyset, \forall\, v,w \in [1,6]$, $v \neq w$, and $T = \bigcup_{i=1}^{n} \bigcup_{j=1}^{6} S_{i,j}$. Let us denote with $r(i,j,k)$ the risk score of an $s_k \in S_{i,j}$ threat, where $k \in [1, |S_{i,j}|]$, and with $R(i,j)$ the aggregated risk score from a $S_{i,j}$ threat set.

In this respect, in **Step 6.** we applied the (1) function:

$$R(i,j) = \bigcup_{k=1}^{|S_{i,j}|} r(i,j,k)$$

where $\cup$ is the aggregation function (AF).

To automatically perform **Step 7.** we employed the graph theory. We constructed the graph model of the analyzed system as a directed graph based on the information parsed from the XML provided by the SDL Threat Modeling Tool. We considered as *data path endpoints (DPE)* the vertices whose in- and/or out-degree was one. We applied the depth-first search to find all end-to-end data paths (non-cyclical paths) between all combinations of the DPEs. This method might return several paths which are not real end-to-end data paths, therefore, these have to be manually excluded.

Finally, to implement the aggregation function for the **Step 8.** let us consider $R(j)$ the end-to-end risk score of a data path for the j^{th} STRIDE group. To calculate this we applied the (2) function:

$$R(j) = \bigcup_{i=1}^{n} R(i,j)$$

where $\cup$ is the AF.

In both of the (1) and (2) equations the AF can be implemented in various forms. Traditional AFs, e.g., *maximum*, *average*, *sum*, *weighted sum*, are suitable in the majority of the situations [28][29], but more complex approaches are also known which employ advanced statistical tools, e.g., distribution functions [30].

5. Conclusions

While IoT is believed to occupy an important role in the future life of people, the security of these kind of systems plays a critical role. The security requirement of IoT services can only be met by taking it into consideration from the very first steps of system design. In this respect, dedicated tools are required for threat and risk modeling for IoT systems. In this work we proposed a semi-automated end-to-end threat and risk modeling method dedicated for IoT systems which is aimed to be used for the design of secure IoT applications and for the security assessment of IoT systems.

References

[1] E. Borgia, "The Internet of Things vision: Key features, applications and open issues", Computer Communications, vol. 54, pp. 1–31, 2014.

[2] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes", Sensors Journal, IEEE, vol. 13, no. 10, pp. 3846–3853, 2013.

[3] T. Bakıcı, E. Almirall, and J. Wareham, "A smart city initiative: the case of Barcelona", Journal of the Knowledge Economy, vol. 4, no. 2, pp. 135–148, 2013.

[4] Y. J. Fan, Y. H. Yin, L. Da Xu, Y. Zeng, and F. Wu, "IoT-based smart rehabilitation system", Industrial Informatics, IEEE Transactions on, vol. 10, no. 2, pp. 1568–1577, 2014.

[5] B. Genge, P. Haller, A. Gligor, and A. Beres, "An approach for cyber security experimentation supporting Sensei/IoT for Smart Grid", Second International Symposium on Digital Forensics and Security, 2014.

[6] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security", NIST special publication, pp. 800–82, 2011.

[7] R. H. Weber, "Internet of Things–New security and privacy challenges", Computer Law & Security Review, vol. 26, no. 1, pp. 23–30, 2010.
[8] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead", Computer Networks, vol. 76, pp. 146–164, 2015.
[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions", Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.
[10] K. Zhao and L. Ge, "A survey on the Internet of Things security", in Computational Intelligence and Security (CIS), 2013 9th International Conference on. IEEE, 2013, pp. 663–667.
[11] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, "Security challenges in the IP-based Internet of Things", Wireless Personal Communications, vol. 61, no. 3, pp. 527–542, 2011.
[12] H. Sandor, B. Genge, and Z. Gal, "Security assessment of modern data aggregation platforms in the Internet of Things", International Journal of Information Security Science, vol. 4, no. 3, pp. 92–103, 2015.
[13] "Trike Threat Modeling," http://octotrike.org/, [Online; accessed 9-November-2015].
[14] A. Z. Standard, "AS/NZS 4360-2004–Risk management", 2004.
[15] U.S. Department of Homeland Security guide, "Cyber security assessments of industrial control systems," Washington, 2010.
[16] U.S. Department of Homeland Security, Industrial Control Systems Cyber Emergency Response Team, "Cyber security evaluation tool (CSET)", Washington, 2014.
[17] X. Yuan, E. B. Nuakoh, and H. Y. Imano Williams, "Developing abuse cases based on threat modeling and attack patterns", Journal of Software, 2015.
[18] G. Ruiz, E. Heymann, E. C´esar, and B. P. Miller, "Automating threat modeling through the software development life-cycle", XXIII Jornadas de Paralelismo, pp. 21–38, 2012.
[19] S. P. Kadhirvelan and A. S¨oderberg-Rivkin, "Threat modelling and risk assessment within vehicular systems", Chalmers University of Technology, Department of Computer Science and Engineering, Goteborg, Sweden, 2014.

[20] Z. Li and T. Xin, "Threat modeling and countermeasures study for the Internet of Things", Journal of Convergence Information Technology, vol. 8, no. 5, 2013.
[21] H. Abie and I. Balasingham, "Risk-based adaptive security for smart IoT in eHealth", in Proceedings of the 7th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 269–275.
[22] "Open web application security project", https://www.owasp.org, [Online; accessed 9-November-2015].
[23] F. Swiderski and W. Snyder, Threat modeling. Microsoft Press, 2004.
[24] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Threat modeling uncover security design flaws using the STRIDE approach", MSDN Magazine-Louisville, pp. 68–75, 2006.
[25] "Improving web application security: Threats and countermeasures, chapter 3.", https://msdn.microsoft.com/en-us/library/ff649874.aspx, [Online; accessed 9-November-2015].
[26] M. Howard and S. Lipner, The security development lifecycle. O'Reilly Media, Incorporated, 2009.
[27] "SDL Threat Modeling Tool", https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx, [Online; accessed 9-November-2015].
[28] G. Stoneburner, A. Y. Goguen, and A. Feringa, "Sp 800-30. risk management guide for information technology systems", 2002.
[29] H. Chivers, J. A. Clark, and P.-C. Cheng, "Risk profiles and distributed risk assessment", computers & security, vol. 28, no. 7, pp. 521–535, 2009.
[30] A. Lenstra and T. Voss, "Information security risk assessment, aggregation, and mitigation", in Information Security and Privacy. Springer, 2004, pp. 391–401.