Toward Automated Threat Modeling of Edge Computing Systems

Valentina Casola, Alessandra De Benedictis
Department of Electrical Engineering
and Information Technology
University of Naples Federico II
Naples, Italy
Email: {casolav,alessandra.debenedictis}@unina.it

Carlo Mazzocca, Rebecca Montanari
Department of Computer Science
and Engineering
Alma Mater Studiorum University of Bologna
Bologna, Italy
Email: {carlo.mazzocca,rebecca.montanari}@unibo.it

***Abstract*—Edge computing brings processing and storage capabilities closer to the data sources, to reduce network latency, save bandwidth, and preserve data locality. Despite the clear benefits, this paradigm brings unprecedented cyber risks due to the combination of the security issues and challenges typical of cloud and Internet of Things (IoT) worlds. Notwithstanding an increasing interest in edge security by academic and industrial communities, there is still no discernible industry consensus on edge computing security best practices, and activities like threat analysis and countermeasure selection are still not well established and are completely left to security experts.**

In order to cope with the need for a simplified yet effective threat modeling process, which is affordable in presence of limited security skills and economic resources, and viable in modern development approaches, in this paper, we propose an automated threat modeling and countermeasure selection strategy targeting edge computing systems. Our approach leverages a comprehensive system model able to describe the main involved architectural elements and the associated data flow, with a focus on the specific properties that may actually impact on the applicability of threats and of associated countermeasures.

***Index Terms*—Edge Computing Security, Automated Threat Modeling, Edge System Modeling**

I. Introduction

The edge computing paradigm allows computation to be performed at the outskirts of a system, as close as possible to data sources, where edge nodes locally handle computing tasks including processing, storage, caching, and load balancing on data sent to and from the cloud [1]. Cloud services will be only occasionally resorted for heavy burden tasks such as big data analytics, machine learning training, etc. This approach is particularly suited for latency-sensitive applications [2], including streaming services, gaming platforms, manufacturing plant automation, self-driving cars, monitoring of energy platforms, and home automation, where the delay resulting from data traveling back and forth between the devices and the cloud data centers may have a negative impact on the user experience, on system functionality, or even on safety. Moreover, large-scale data transmission requires very high network bandwidth, which is quite costly, and it is also critical from a security and privacy perspective since sensitive or proprietary information is transferred and processed away from its source, which may be an issue in presence of data localization laws.

Despite the big potential of edge computing to maximize the benefits coming from both cloud and edge worlds, it is undeniable that it brings new, unprecedented cyber risks [3]. Typical security concerns related to the loss of control experienced in cloud environments add up with the issues affecting the edge and device layers, characterized by highly heterogeneous hardware platforms, operating systems, and communication protocols, different computation, storage, and communication capabilities, typically coarse-grained access control mechanisms, and a general lack of attack awareness due to the limited interfaces usually offered by devices. To make things worse, edge systems are widely employed in applications that collect and process personal and sensitive data, whose protection should be ensured at any level, independently of the specific capabilities of involved devices.

In this context, identifying the actual threats that affect a given deployment is a challenging task, as risks vary significantly by individual use case and application domain, and heavily depend on involved components and technologies. Unfortunately, despite an increasing interest in the edge security topic by both the academic and industrial communities, there is still no discernible industry consensus on edge computing security best practices, and activities like threat analysis and countermeasure selection are completely left to security experts. Unluckily, today's high-speed agile and DevOps IT environments often involve personnel with limited security expertise, or existing security teams are too small to efficiently cope with established requirements. Moreover, to make things worse, security-related activities including threat modeling must keep pace with fast development and deployment processes, and this inevitably requires their simplification and, when possible, their automation.

In order to cope with the need for a simplified yet effective threat modeling process, in this paper we propose an *automated threat modeling and countermeasure selection methodology* targeting edge computing systems. Our approach leverages a comprehensive system model able to describe the main involved architectural elements (i.e., the assets) and the associated data flow, with a focus on the specific properties that may actually impact on the applicability of threats and of associated countermeasures. To validate our approach, we applied it to a simple smart home case study by using the Microsoft Threat Modelling Tool[1] which was suitably extended to include our system model elements and threats.

The paper is organized as follows. Section II presents an overview of existing approaches aimed to model and assess the security in IoT and edge computing systems. Section III illustrates the system model behind our proposal, while Section IV describes our automated threat modeling and countermeasure selection strategy. Finally, Section V provides an example of application of our approach to a simple edge computing system, to demonstrate its feasibility and effectiveness, and Section VI draws our conclusions.

II. Related work

In recent years, there has been a growing interest in exploring the security of IoT environments and great research efforts have been directed toward the design and development of reference secure architectures. On the contrary, at the state of art, edge computing security has not been fully investigated. In this section, we focus on the proposed approaches for modeling security threats and assess the security of IoT networks.

The authors in [4] presented a system capable of automatically identifying the types of devices and subsequently establishing which rules should be enforced to constrain the communications of devices affected by potential security vulnerabilities. Adopting this security system allows minimizing the damage resulting from vulnerable devices. A framework for modeling and assessing the security of IoT was proposed in [5]. It is employed to build a graphical security model aiming at capturing potential attack paths in the network. Furthermore, the authors also provide a security evaluator responsible for automating the security analysis. The results of the assessment of the security level of the IoT network provide a clearer picture of which assets and paths should be protected at first. Then, the defense strategies are compared in order to choose the most effective device-level security strategies. Soteria [6] is a static analysis system proposed to validate whether an IoT application or IoT environment adheres to identified security, safety, and functional properties. It translates platform-specific IoT source code into an intermediate representation and then extracts a state model on which verifies the desired properties. In [7], the authors propose an IoT Security Model (IoTSM) that allows organizations to plan and implement a strategy for developing end-to-end IoT security. This approach also enables analyzing, describing, and measuring the security posture, level, and practise of an IoT organization. Most of the current state-of-art research efforts target security issues, challenges, and frameworks for securing edge computing systems [3], whereas the area of automated threat modeling is still in its infancy. Along this direction, this paper represents an original proposal. The only work that has some similarities with ours is [8], where authors introduced an approach meant to support the security analysis of IoT systems. It is based on an almost completely automated process for threat modeling and risk assessment, which also helps identify the security controls to enforce to mitigate existing security vulnerabilities. Although this work shares some basic concepts with [8], the latter considered a more general system model and did not take explicitly into account the data flow and some relevant attributes that enable to obtain a fine-grained threat characterization, which are the focus of this paper.

III. System Modeling

Ecosystems where IoT, edge and cloud converge towards a computing continuum are made up of several heterogeneous components that have specific security requirements and different compute and storage capabilities. A typical cloud continuum system consists of three layers, namely the cloud service layer, the edge layer, and the (IoT) device layer. The edge layer is powerful enough to manage IoT devices and run containerized applications. Therefore, this distributed computing paradigm favors a strong integration between cloud and IoT. However, a downside is the significant number of connected devices and of interactions with the edge computing layer, which considerably broaden the attack surface.

In order to facilitate the security analysis of an edge computing deployment and support developers in the threat modeling and countermeasure selection phases, we propose a system modeling approach that enables a developer to specify those aspects related to the architecture of the system, in terms of its main hardware and software assets, and to the related dataflow, that actually impact on security. Building the model requires a limited effort from developers to specify the essential characteristics of a system, and enables to automatically obtain a threat model that is as much customized as possible for the specific system deployment. In the following subsections, we will illustrate the classification taken into account to model both assets and data (refer to Figure 1).

A. Asset Modeling

As mentioned previously, an edge computing system involves several types of assets, belonging to different architectural and functional levels. In particular, we consider three main asset types, namely physical/virtual processing nodes, software components/modules, and communication channels, discussed in the following.

1) Physical/Virtual Processing Nodes: The first asset category includes the processing nodes belonging to the different layers of an edge computing system, devoted to running application programs and services that implement the system business logic. At the IoT and edge layers, these nodes are represented by the physical devices, while at the cloud layer they basically correspond to the virtual machines offered according to the IaaS paradigm. Hence, a *Processing node* can be classified into the three categories: *Edge node*, *IoT node* and *Cloud VM*. It is worth reminding that processing nodes offer different storage and computational capabilities, which allow for the implementation of different security

[1] https://www.microsoft.com/en-us/download/details.aspx?id=49168

mechanisms. This is a key aspect that must be taken into account during the countermeasure selection process, in order to identify feasible controls on the target architecture. Generally, cloud-based compute resources and edge nodes provide sufficient capabilities to support traditional security mechanisms, while IoT devices are usually characterized by limited compute and storage capabilities that often only allow for the implementation of simple and lightweight protocols and mechanisms. On the other hand, it is worth noting that some physical characteristics of processing nodes have an impact on applicable threats: for instance, having a battery-powered device opens up to specific threats that are not applicable to AC-powered nodes (e.g., battery exhaustion).

Based on the above considerations, IoT devices can be *labeled* according to their processing/storage capabilities and power supply by means of a `capability` attribute. In particular, following the classification proposed in [9], devices can be distinguished in *Constrained*, *Limited*, *Restricted* and *Normal*. Constrained devices (battery-powered, up to 10KB RAM, and up to 128KB ROM) are the weakest and do not support any security mechanisms. Limited devices (battery-powered, 10-32KB RAM, and 128-512KB ROM) can support some symmetric key-based protocol. Restricted devices (battery/AC powered, 32-128KB RAM, and 512KB-10MB ROM) are more powerful devices able to implement symmetric protocols and lightweight asymmetric key-based protocols. Finally, normal devices (AC-powered, 128KB and above RAM, 10MB ROM and above) are powerful devices able to implement any traditional security protocols.

Besides processing/storage capabilities and power supply characteristics, in an edge computing scenario also the location where a node is physically deployed impacts on its security, as it may lead to specific threats. In order to take this aspect into account, it is possible to identify another attribute of interest to label both edge nodes and IoT devices, namely `location`, which can assume two values: *Protected* and *Open*. The former refers to assets that are placed in an area that can be only accessed by authorized personnel, while the latter is related to assets that can be accessed without any restriction, and that therefore are more exposed to potential attackers.

2) Communication Channel: The second asset category includes the communication channels established among the nodes. Currently, there are many communication protocols employed in edge computing (Zigbee, Bluetooth, Wi-Fi, etc) that can be exploited by malicious attackers to compromise the system. To take into account specific threats associated with the communication channel, we assigned the `protocol` attribute to this asset, which corresponds to the actual protocol used for communication.

3) Software Component: The third asset category includes the software components, modules, and services that help implement the business logic of the system. Involved software components can be very heterogeneous both from the technology and from the complexity point of view. In order to simplify the characterization of software components, we identified a `service type` attribute that can assume one of three possible values, namely *web-based service*, if the component is primarily devoted to processing and exposes its services by means of a web (HTTP) interface (it is the case of cloud-based services and of web interfaces exposed by edge devices to communicate with the cloud layer), *storage service*, if the component is devoted to storing structured or unstructured information (e.g., an on-premise DBMS, a cloud-based storage service, a key-value store, etc.) and can be accessed remotely, and *IoT service*, in case of services/applications running on IoT or edge nodes and accessible via non-HTTP protocols.

With regard to the software component technologies in an edge computing context, it is worth mentioning that container-based virtualization is spreading out, not only at the cloud service layer, but also at the edge layer, due to the fact that containers are lightweight and portable, start in few seconds, and can be deployed, migrated, and upgraded faster on distributed edge infrastructure compared to virtual machine applications. Despite the undeniable benefits brought by containerization, it introduces a unique set of security challenges and risks that must be addressed properly given its central role in edge computing systems. Hence, we considered a second attribute, namely `containerized`, which can assume a boolean value, to track whether or not the software component uses container technology.

B. Data Modeling

For the purpose of providing an effective modeling of the data involved in an edge computing deployment from the security point of view, we considered a data classification based on the subject/element to which the data are related. Before illustrating such classification, it is worth outlining that, independently of the source/responsible of data, they can be classified based on their sensitivity. In fact, according to the ISO27001 standard, each organization should contemplate an information classification process in order to assess the data managed and the level of protection deemed. To address this fundamental aspect, we introduced a `sensitivity` attribute, to be assigned to any type of data, which can assume one of the following three values: *Public*, *Internal*, and *Confidential*. Public data (e.g., temperatures of public places, traffic condition data, etc.) are available to everyone, internal data (e.g., user profile data, configuration settings, etc.) can be only accessed by certain entities of the system and, finally, confidential data (e.g., credentials, biometrics, financial data, etc.) are only available to the owner. Based on the sensitivity level, the unauthorized disclosure, alteration, or destruction of data would result in a low, moderate or significant level of risk, respectively. Let us now illustrate the considered data classification.

1) User-related data: User-related data represent the information belonging to the end-users, and include both the information used to interact with the system (credentials, profiling information, etc.), referred to as *Service user data*, and so-called personally identifiable information (*PII*), which enable to uniquely identify an individual. Since 2016, the European General Data Protection Regulation (GDPR) [10]

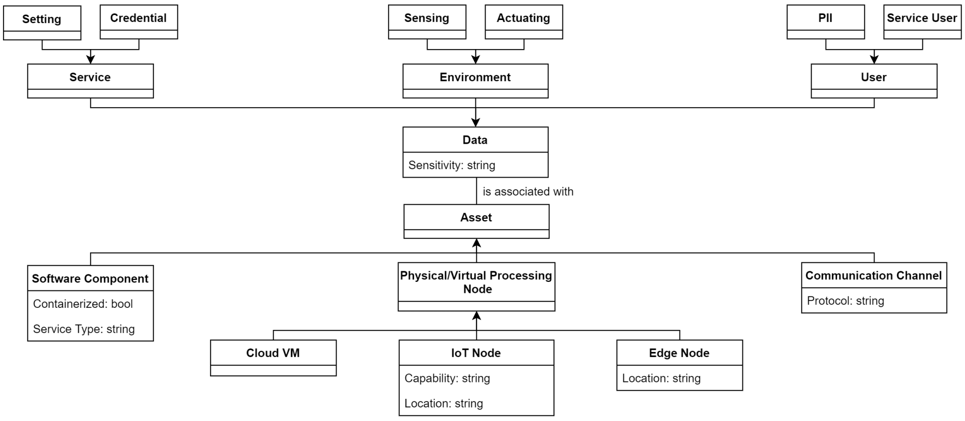

Fig. 1. Asset and data modeling.

established a set of rules aiming at defining what must be protected in order to preserve the privacy of individuals, so companies have been forced to keep PII information in a safe and secure manner. Article 9[2] of the European GDPR outlines the personal data that cannot be processed without the explicit consent of the interested party. To the personal data category belongs all the information revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership. Furthermore, are also included other information such as biometric data aiming at uniquely identifying a person, data concerning health or sexual orientation, judicial information, electronic communication (IP addresses), geographic location, etc.

2) Environmental data: Data exchanged/stored in an edge system may not only concern users, but they may also refer to the environment where the system is deployed. Environment data, in particular, may be either generated by sensors or used to control actuators. Hence, we consider two types of environmental data, namely *Sensing* and *Actuating*. Depending on the specific application domain, these data may have different sensitivity features.

3) Service data: Service data include any other data not directly related to an end-user or to the environment, such as service credentials and configuration parameters (e.g., deployment information), and any further information (critical or not) used by services and applications.

IV. Approach

The approach proposed in this paper enables to perform the threat modeling of edge computing systems in a simple and automated manner, without requiring particular security skills. A developer is only required to describe the system under analysis by identifying and labeling, with the appropriate attributes, the involved assets and data, according to the system model introduced previously. After this modeling step, the developer will be automatically provisioned with a comprehensive threat model of the system and with a set of countermeasures to apply in terms of security controls. The automation relies upon a complex *security data model*, depicted in Figure 2, which suitably links together the concepts related to assets and data classification and characterization, and to threats and security controls. The security data model is instantiated by means of a *Threat Catalogue*, which includes a great number of threats classified based on the STRIDE Threat Model [11] (spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege).

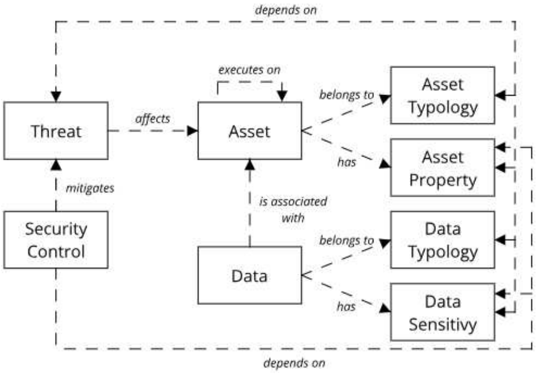

Fig. 2. Security data model.

As said, besides providing a fine-grained threat modeling of a specific edge-deployment, we also aim at automatically binding threats with proper mitigation actions. To accomplish this goal, for each threat, we identified a set of security controls belonging to the NIST Security Control Framework [12] to be applied at each affected asset as a mitigation measure. The NIST framework contains over 900 unique security controls that encompass 18 control families, including both base controls and *control enhancements*, which strengthen the fundamental security capability of a base control. According to our approach, only a subset of the applicable security controls is selected, based on the actual capabilities (in terms of computational power and storage capacity) available on each node. This information is explicit for processing nodes thanks to the `capability` attribute, while for software components

[2] https://www.privacy-regulation.eu/en/article-9-processing-of-special-categories-of-personal-data-GDPR.htm

TABLE I
EXTRACT OF THE THREAT CATALOGUE.

#	Asset Typology	Threat	STRIDE Category	Asset Properties	Data Typology	Data Sensitivity	Security Controls
1	Edge Node	Camouflage	Spoofing	Location:*Open*	-	-	IA-3, IA-3(1, 3), IA-5, ...
2	Edge Node	Hardware Trojan	Tampering	Location:*Open*	-	-	SI-3(3), SI-16, ...
3	Edge Node	Unauthorized Access Control	Elevation of Privilege	-	PII, User Service, Credential, Setting	Internal, Confidential	AC-17, AC-17(1, 2, 4, 5, 6), ...
4	IoT Node	Battery Draining	Denial of Service	Location:*Open*, Capability:*Constrained, Limited, Restricted*	-	-	PE-2, PE-3, ...
5	IoT Node	Denial of Service	Denial of Service	-	-	-	SC-5
6	IoT Node	Exhaustion of Power	Denial of Service, Spoofing	Capability:*Constrained, Limited, Restricted*	-	-	PE-11
7	Communication Channel	Jamming	Denial of Service, Spoofing	-	-	-	IA-3, SC-5, ...
8	Communication Channel	Network Key Sniffing	Information Disclosure	Protocol:*Zigbee*	Credential	Confidential	SC-8, SC-13, ...
9	Communication Channel	Message Elimination	Information Disclosure, Spoofing, Tempering	-	-	Internal, Confidential	AC-17, SA-18, ...

it depends on the capability specified for the processing node used for their execution (this relationship must be specified by the developer during system modeling).

In Table I we report an extract of the Threat Catalogue. It is worth outlining that, in the table, an asset property that does not influence a threat is simply omitted, while if a threat is independent of a field of the catalogue, that field is filled with ”-”. With regards to security controls, the last column of the table reports some of the NIST controls that represent valid countermeasures to thwart or mitigate each threat. As anticipated, during the countermeasure selection step a subset of these controls will be actually selected based on the capabilities of involved assets.

As mentioned, the catalogue has been built by collecting threats from multiple sources. So far, we have collected more than 150 threats specific to cloud services, web-based application, storage services, IoT and edge devices, and network protocols, derived from existing standards and scientific studies [13], [14], [15], [3], [16], [17], [18]. While the effectiveness of our approach clearly depends on the completeness of the catalogue, it can be easily extended to include new threats and cope with new issues.

V. EXAMPLE

In this section, we provide a concrete example of application of the proposed approach. The deployment under study, sketched in Figure 3, is a smart home environment that involves different IoT devices connected to an edge node capable of locally managing them and running software services. As mentioned in the Introduction, for our case study we adopted the Microsoft’s Threat Modeling Tool, which was enriched with the assets and data flows described in Section III, as well as with their related threats.

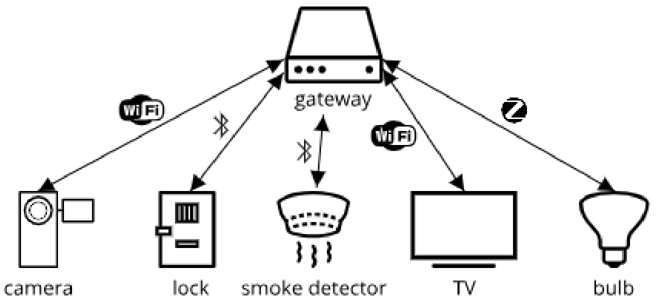

Fig. 3. Smart home system under study.

Following our approach, for each asset that composes the system, we have to determine its properties and typology, as well as the typology and sensitivity of data stored and/or transmitted. For the sake of brevity, we will focus on edge nodes, IoT devices and their communication channels, without considering other component types. The edge node is a gateway deployed inside the habitation, therefore its location is assumed as protected since only the owners can physically access it. With reference to the end devices involved, we considered a system made of smart bulbs, smart locks, smoke detectors, smart TVs, and smart cameras. In Table II, we report the specifications of the gateway, IoT devices, and an instance of communication channels. All the IoT devices are connected with the gateway by means of the communication channel asset. We assumed that smart bulbs support ZigBee protocol, smart locks and smoke detectors use BLE, while smart TVs and cameras support Wi-Fi protocol.

After having modeled the system, we proceed with the threat identification and countermeasure selection. The threats and their mitigation are automatically retrieved through the Threat

TABLE II
ASSET SPECIFICATIONS.

Asset	Asset Properties	Data Typology	Data Sensitivity
Gateway	Location:*Protected*	-	-
TV	Location:*Protected*, Capability:*Normal*	Service User	Confidential
Internal Camera	Location:*Protected*, Capability:*Normal*	Sensing, PII	Internal, Confidential
External Camera	Location:*Open*, Capability:*Normal*	Sensing, Actuating	Public
Smoke Detector	Location:*Protected*, Capability:*Restricted*	Sensing, Actuating	Confidential
Internal Lock	Location:*Protected*, Capability:*Constrained*	Sensing, Actuating	Internal
External Lock	Location:*Open*, Capability:*Constrained*	Sensing, Actuating	Confidential
Bulb	Location:*Protected*, Capability:*Limited*	Sensing, Actuating	Public
Comm. Channel Ext Lock-Edge	Protocol:*Zigbee*	Sensing, Actuating, Credential	Confidential

Catalogue. As mentioned previously, each threat depends on the asset it refers to (in terms of its typology and properties) and on the type and sensitivity of the data concerned.

For example, with reference to Table I, the gateway cannot be affected by threats that are applicable to nodes whose location is public, while it will be affected by the *Camouflage* threat. All IoT devices are subject to *Batter Draining* and , except for smart cameras and smart TV, they will be subject as well to the *Exhaustion of Power* threat. Communication channels, independently of the protocol used, can be always subject to *Jamming* attacks. Moreover, communication channels conveying data whose sensitivity level is internal and/or confidential, will be affected by the *Message Elimination* threat. Finally, due to the protocol adopted, the communication channel between smart bulbs and the edge node will be threatened by the *Newtwork Key Sniffing* threat.

VI. CONCLUSIONS AND FUTURE WORK

Taking into account security from the very beginning development stages is fundamental to design secure systems, especially in an edge computing scenario characterized by several heterogeneous components. In this paper, we first provided a comprehensive system model able to describe and characterize the assets that typically make up an edge computing system and the involved data flow. Then, we introduced an approach that, based on the system model, enables to automate the threat modeling and countermeasure selection processes, and validated our approach with a smart home case study. As discussed, our approach heavily depends on the completeness and accuracy of our Threat Catalogue that, however, has been designed to be fully extensible in order to include new threats and cope with new attack scenarios. In our future work, we plan to make available our tool for system modeling and further enrich the catalogue by including threats specific to recent device technologies and communication protocols. Moreover, we plan to refine the countermeasure selection step by considering more specific actions tailored to the different assets instead of generic, technology agnostic security controls.

REFERENCES

[1] W. Shi and S. Dustdar, "The Promise of Edge Computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[3] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A Survey on Security and Privacy Issues in Edge Computing-Assisted Internet of Things," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[4] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2177–2184.

[5] M. Ge, J. B. Hong, W. Guttmann, and D. S. Kim, "A framework for automating security analysis of the internet of things," *Jouranl of Network and Computer Applications*, vol. 83, pp. 12–27, 2017.

[6] Z. B. Celik, P. McDaniel, and G. Tan, "Soteria: Automated IoT Safety and Security Analysis," in *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. Boston, MA: USENIX Association, Jul. 2018, pp. 147–158. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/celik

[7] J. Bugeja, B. Vogel, A. Jacobsson, and R. Varshney, "IoTSM: An End-to-end Security Model for IoT Ecosystems," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019, pp. 267–272.

[8] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "Toward the automation of threat modeling and risk assessment in IoT systems," *Internet of Things*, vol. 7, no. 100056, 2019.

[9] K. Sha, R. Errabelly, W. Wei, T. A. Yang, and Z. Wang, "EdgeSec: Design of an Edge Layer Security Service to Enhance IoT Security," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, 2017, pp. 81–88.

[10] EU, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

[11] Microsoft Corporation, "The STRIDE Threat Model," 2009. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)

[12] National Institute of Standards and Technology, "NIST Special Publication 800-53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations," 2013. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[13] OWASP, "The Ten Most Critical Web Application Security Risks," 2017.

[14] Cloud Security Alliance, "Top Threats to Cloud Computing: Egregious Eleven," 2019. [Online]. Available: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-egregious-eleven

[15] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[16] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.

[17] SSL Labs, "SSL Threat Model," 2018. [Online]. Available: https://www.ssllabs.com/projects/ssl-threat-model/

[18] P. Tedeschi and S. Sciancalepore, "Edge and Fog Computing in Critical Infrastructures: Analysis, Security Threats, and Research Challenges," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2019, pp. 1–10.