CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

Towards a Distributed Learning Architecture for Securing ISP Home Customers

Pedro Miguel Santos*
Joana Sousa
Ricardo Morla
Nuno Martins
João Tagaio
João Serra
Carlos Silva
Mário Sousa*
Pedro Souto*
Luis Lino Ferreira*
Joao M.Ferreira
Luis Almeida*

*CISTER Research Centre

CISTER-TR-210501

2021/06/25

Towards a Distributed Learning Architecture for Securing ISP Home Customers

Pedro Miguel Santos*, Joana Sousa, Ricardo Morla, Nuno Martins, João Tagaio, João Serra, Carlos Silva, Mário Sousa*, Pedro Souto*, Luis Lino Ferreira*, Joao M.Ferreira, Luís Almeida*

*CISTER Research Centre

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: pss@isep.ipp.pt, joana.sousa@parceiros.nos.pt, rmorla@fe.up.pt, nuno.mmartins@parceiros.nos.pt, joao.tagaio@parceiros.nos.pt, joao.mserra@nos.pt, carlos.a.silva@nos.pt, msousa@fe.up.pt, pfs@fe.up.pt, llf@isep.ipp.pt, joao.MFerreira@nos.pt, lda@fe.up.pt

https://www.cister-labs.pt

Abstract

Networking equipment that connects households to an operator network, such as home gateways and routers, are major victims of cyber-attacks, being exposed to a number of threats, from misappropriation of user accounts by malicious agents to access to personal information and data, threatening users' privacy and security. The exposure surface to threats is even wider when the growing ecosystem of Internet-of-Things devices is considered. Thus, it is beneficial for the operator and customer that a security service is provided to protect this ecosystem. The service should be tailored to the particular needs and Internet usage profile of the customer network. For this purpose, Machine Learning methods can be explored to learn typical behaviours and identify anomalies. In this paper, we present preliminary insights into the architecture and mechanisms of a security service offered by an Internet Service Provider. We focus on Distributed Denial-of-Service kind of attacks and define the system requirements. Finally, we analyse the trade-offs of distributing the service between operator equipment deployed at the customer premises and cloud-hosted servers.

Towards a Distributed Learning Architecture for Securing ISP Home Customers

Pedro M. Santos[1,4][0000−0002−7162−0560], Joana Sousa[2,5][0000−0002−6418−2312], Ricardo Morla[3][0000−0002−5162−3019], Nuno Martins[2,6], João Tagaio[2,7], João Serra[2], Carlos Silva[2], Mário Sousa[3,4][0000−0001−7200−1705], Pedro Souto[3,4][0000−0002−0822−3423], Luís Lino Ferreira[1,4][0000−0002−5976−8853], João Ferreira[2], and Luís Almeida[3,4][0000−0002−9544−3028]

[1] Instituto Superior de Engenharia do Porto, Porto, Portugal
{pss,llf}@isep.ipp.pt
[2] NOS Inovação, Portugal
{joana.sousa,nuno.mmartins,joao.tagaio}@parceiros.nos.pt
{joao.mserra,carlos.a.silva,joao.MFerreira}@nos.pt
[3] Universidade do Porto, Faculdade de Engenharia, Portugal
{rmorla,msousa,pfs,lda}@fe.up.pt
[4] CISTER Research Center in Real-Time & Embedded Computing Systems, Portugal
[5] Bold International, Portugal
[6] Caixa Mágica, Portugal
[7] KCSIT, Portugal

Abstract. Networking equipment that connects households to an operator network, such as home gateways and routers, are major victims of cyber-attacks, being exposed to a number of threats, from misappropriation of user accounts by malicious agents to access to personal information and data, threatening users' privacy and security. The exposure surface to threats is even wider when the growing ecosystem of Internet-of-Things devices is considered. Thus, it is beneficial for the operator and customer that a security service is provided to protect this ecosystem. The service should be tailored to the particular needs and Internet usage profile of the customer network. For this purpose, Machine Learning methods can be explored to learn typical behaviours and identify anomalies. In this paper, we present preliminary insights into the architecture and mechanisms of a security service offered by an Internet Service Provider. We focus on Distributed Denial-of-Service kind of attacks and define the system requirements. Finally, we analyse the trade-offs of distributing the service between operator equipment deployed at the customer premises and cloud-hosted servers.

Keywords: Safe Home · Cyber-security · Anomaly detection · Distributed systems · Hybrid environment · Machine Learning.

1 Introduction

As the number of Internet-of-Things (IoT) devices increases in households, the exposure surface to cyber-attacks grows in proportion. IoT devices are often

victims of poor security configurations and subject to zero-day exploits, and then enlisted to carry out Distributed Denial-of-Service (DDoS) attacks on other victims. Internet Service Providers (ISP), in their continuous improvement of the service offered to customers, are keen on extending their security services to protect the client devices from such attacks. To provision the Internet service, the ISP (or simply operator) deploys a set of equipment at the customer premises, called the Customer Premises Equipment (CPE), that typically provides routing to the Internet and local wired and wireless networking. Due to its strategic position, this equipment can play an pivotal role in increasing the security of the IoT ecosystem and of other devices that the customer may have at home.

Traditional rule-based approaches at securing the CPE and the customer network – such as TCP port blocking, IP and TLS certificate blacklisting, and traffic signatures – fail to capture the dynamics of attacks and of legitimate traffic. On the one hand, rules for blacklisting zero-day exploits take some time to be created by the security community and to be deployed at the CPE or at upstream network security devices. On the other hand, the diversity of customer traffic including IoT devices makes it extremely hard and cumbersome to define rules for whitelisting the legitimate traffic at each customer. These limitations call for machine learning (ML) techniques to be deployed on top of traditional approaches. With machine learning, legitimate traffic can be profiled and outliers more easily detected. Outliers may result from legitimate yet infrequent behavior, legitimate failure-related anomalous behavior, or attacks – the latter being of interest to security. Machine learning can also be used to profile malicious traffic, learning to predict attacks that do not reuse blacklisted IPs or certificates.

In this paper we discuss and outline a conceptual architecture for an ISP-supported cyber-security system based on machine learning techniques to provide secure Internet services to customers, particularly to the ecosystem of IoT devices. ML techniques will be explored as a flexible mechanism for studying and modelling the traffic observed by the CPE and the CPE's behaviour itself, e.g., traffic profiling, anomalous traffic detection, number of requests to CPE, number of times CPE is resetting in a certain period of time, among others. The IoT security system will feature a distributed architecture, where software components running at the edge node can be complemented by cloud-hosted components. For example, the ML algorithm can be trained at a cloud server, due to its larger computing capacity, and the edge node may host the trained version of the algorithm (to be updated regularly) to carry out the inference task of identifying potential attacks.

To achieve the conceptual ISP-supported architecture for secure Internet provision to customers' homes, we will follow an agile approach:

- define security models for the attacks of interest and the requirements of the IoT security system (functional, technical, and design);
- design an hybrid architecture (edge and cloud) to host a distributed ML mechanism towards providing security to domestic IoT devices;
- outline the steps of development and testing and concerns associated with the development of such system, e.g., data privacy-related (GRDP);

– understand how can this type of architecture be easily and horizontally explored by other use cases with minor customisation.

The remainder of this document is as follows. We present a review of the related literature in Section 2, focusing on DDoS attacks. Our use-case, attack models and requirements elicited are described in Section 3. Section 4 presents considerations about the architecture and development of the proposed system. Final remarks are drawn in Section 5. This work is being carried out within the Eureka ITEA3 *MIRAI*[1] project.

2 Related Work

A taxonomy of types of attacks can be found in [3] and a review of Machine Learning (ML) solutions for IoT security can be found in [11]. A review of ML techniques specifically applied to anomaly detection is presented in [7]. In the following discussion, we review mostly works proposing Intrusion Detection Systems (IDS) that aim specifically at detecting Distributed Denial-of-Service (DDoS) attacks through ML techniques (and, to a lesser extent, works that aim at undifferentiated detection of abnormal traffic). We observe that much of these efforts rely on supervised learning techniques (leveraging existing public datasets), and that a number of works target networks managed by software-defined networking (SDN) techniques, as the SDN paradigm allows to gather a comprehensive view of the network traffic. Table 1 summarizes the most relevant features from all references.

The authors of [9] propose an intrusion detection system tailored to identify low rate (LR) DDoS attacks in SDN settings. Six ML models are compared (see Table 1), using the Canadian Institute of Cybersecurity (CIC) DoS dataset. A Multi-Layer Perceptron (MLP), a type of neural network, obtained the best accuracy, around 95%. In [10], also four ML techniques are applied to the identification of DDoS attacks, over datasets collected by the authors. Decision trees offered the best detection performance. The work described in [2] also addresses detection of DDoS attacks through multiple ML techniques. The authors explore "IoT-specific network behaviors (e.g., limited number of endpoints and regular time intervals between packets)" to improve accuracy of DDoS detection. The techniques explored are K-nearest neighhours, linear-kernel support vector machine (SVM), decision trees, random forests, and a neural network, over a purposefully-collected dataset. The authors of [12] propose a deep learning approach to DDoS detection, specifically through the use of various types of recurrent neural networks (RNN). It is reported that the error rate is reduced from 7.517% to 2.103% with respect to conventional machine learning models, specifically Random Forest. In [5], the authors also leverage a Long Short-Term Memory (LSTM) neural network to detect malicious traffic (not exclusively DDoS attacks) at packet-level. The system is evaluated using literature datasets and the authors' own dataset on the MIRAI botnet; an accuracy

[1] https://itea3.org/project/mirai.html

Table 1. Review of relevant ML-based techniques for security.

	Title	Focus/Scope	Techniques	Datasets
[9]	A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning	DDoS attack detection in SDN; focus on low-rate attacks	J48, Random Tree, REP Tree, Random Forest, Multi-Layer Perceptron (MLP), Support Vector Machines (SVM)	DDoSSim, GoldenEye, H.U.L.K., R.U.D.Y., Slow Body, Slow Headers, Slowloris, Slow Read
[10]	Machine learning algorithms to detect DDoS attacks in SDN	DDoS attack detection in SDN	MLP, SVM, Decision Tree, Random Forest	own
[2]	Machine Learning DDoS Detection for Consumer Internet of Things Devices	DDoS attack detection in SDN, focus on IoT devices	K-nearest neighbors "KDTree" algorithm, support vector machine with linear kernel, decision tree and random forest, 4-layer feed-forward NN	own
[1]	The DDoS attacks detection through machine learning and statistical methods in SDN	DDoS attack detection in SDN; low- and high-volume attacks	Combination of entropy-based method and classification algorithm	UNB-ISCX, CTU-13, ISOT-normal traffic
[12]	DeepDefense: Identifying DDoS Attack via Deep Learning	DDoS attack detection	Various types of neural networks: Convolutional (CNN), Recurrent (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GTU)	ISCX2012
[5]	An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level	Anomaly detection, including but not limited to DDoS attacks	Long Short-Term Memory (LSTM) neural network	USTC-TFC2016, Mirai-RGU, Mirai-CCU
[4]	Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection	Anomaly detection	Convolutional Neural Network (CNN) and Autoencoder (a type of NN) used in unsupervised mode	ISCX2012, USTC-TFC2016, Mirai-RGU, Mirai-CCU

of 97.22% is reported. Finally, an IDS leveraging an unsupervised technique is proposed in [4]: the authors use a convolutional neural network (CNN) and an unsupervised Auto-Encoder (a type of NN) for network traffic anomaly detection.

There are several AI/ML approaches to perform of anomaly detection and thus thwart DDoS attacks. Neural networks (NN) take the center stage due to their attractive trade-off between computation cost, performance and flexibility (e.g., typically inferior computing cost than SVM; more flexibility than threshold rule-based decision trees), and due to the range of scope-specific architectures (e.g., recurrent NN, LSTM, convolutional NN) that are tailored to particular applications. In spite of this, considerable datasets and computation power are required, which most CPEs have in limited supply. Thus, the importance of exploring hybrid environments (edge and cloud) to implement complex AI algorithms is emerging in order to finally achieve the balance among data, computation and accuracy.

3 Use-case and Requirements

3.1 Use-Case Description

The proposed IoT security system will ensure coverage of the customers' devices as a flexible protection solution that the customer can configure and adapt to the needs of the household. The system will be integrated directly in home gateways, supported by intelligence in the cloud, and supplemented by mobile applications that ensure services such as household member profiling to adapt browsing protection and parental control as well as privacy management.

3.2 Actors

For convenience, we identify the main actors in the use-case: **Customer**; **Attacker** and **Operator**. The setting is the Customer network, enabled by the Operator equipment and infrastructure. We identify the following relevant infrastructure/equipment/agents:

- **Home Gateway & Router:** equipment deployed by the Operator that: (i) enables the Customer network; and (ii) connects the Customer network to Operator Infrastructure. Typically, both network nodes (gateway and router) are provided by the same physical equipment.
- **Customer-Premises Equipment (CPE):** any type of Operator-provided equipment deployed in the Customer premises, including but not limited to the home gateway and router (e.g., WiFi range extenders). In practice, and unless otherwise noted, the term *CPE* is used to identify the home gateway & router.
- **Customer Network:** network (wireless and wired) enabled by the Operator equipment at the Customer's house to which only Customer devices are connected and that is, ideally, secured against unauthorized associations or attacks.

- **Customer Devices:** the set of devices connected to the Customer network, with particular focus on IoT devices.
- **Attacker Agent (Malware):** an agent (typically software) that can: (i) grant control of a Customer Device to an Attacker; (ii) participate in (D)DoS attacks.
- **Operator Infrastructure:** service and networking infrastructure that connects the Customer network to Internet.
- **IoT Security System (or System):** system aiming to consolidate the security of the Customer network against external threats and attacks.

3.3 Requirements Elicitation

In the context of the setting described in the previous section, we list the requirements of the proposed system.

Functional Requirements

- **Req. #1 - Always-on Monitoring:** The System shall monitor traffic patterns 24/7 to find indicators of security threats and attacks.
- **Req. #2 - Smart and Customized Protection:** The System shall learn the patterns of traffic of applications concerning IoT devices, and use the protection profile that best suits the identified patterns.
- **Req. #3 - Mitigation of Scale Attacks:** The System shall provide mitigation mechanisms against large scale attacks.
- **Req. #4 - Automated Zero-Day Attack Mitigation:** The System shall provide detection and mitigation mechanisms against vulnerabilities of the IoT devices that are unknown *a priori*.
- **Req. #5 - Analysis:** If an attack is ongoing, the System shall provide a detailed report in real-time or near real-time during the attack, and complete resume after it. The System shall provide a global report with information about all installed CPE to have knowledge about how many CPE were exposed to attacks, and which was the attack profile/type.
- **Req. #6 - Metrics:** The System shall provide metrics related to vulnerabilities and attacks that have been identified and thwarted (or not).
- **Req. #7 - Alerts:** The System shall provide the possibility for the Operator and/or Customer to be informed whenever a vulnerability and/or attack is detected.

Design Requirements

- **Req. #1 - Operation in Home Gateways:** The System shall be integrated directly and operate in the Home Gateways.
- **Req. #2 - Edge/cloud architecture:** The System may be complemented by components deployed outside the Home Gateway, namely mechanisms to support the operation of the System deployed in edge and/or cloud platforms.

Technical Requirements

- **Req. #1 - RDK:** The System shall use the operating system RDK at the Home Gateways.

3.4 Attack and Threat Models

To support the functional requirements identified in the previous section, specific attack and threat models need to be defined. By *attack*, we refer to malicious activity that disrupts regular service, targeting customer devices, operator equipment, and/or third-party nodes.

The IoT security system shall provide the following services:

- **Service #1:** detect and mitigate involvement of customer devices in attack attempts (e.g., DoS) to other parties (to other devices in the customer network; to infrastructure of the operator; to external parties);
- **Service #2:** detect and mitigate DoS attacks to the Customer Devices.

Figure 1 represents generically the attack and threat model over the target scenario.

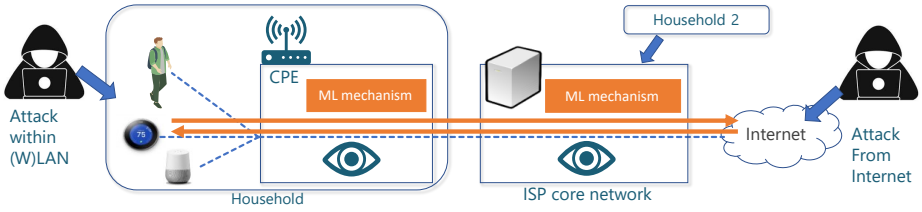

Fig. 1. Attack and threat model to the customer premises.

4 Design Considerations and Development Process

We propose to leverage artificial intelligence and machine learning (AI/ML) and an edge/cloud distributed architecture to implement the IoT security system (or simply system). DoS attacks are best addressed by traffic profiling, as learning the traffic patterns of legitimate customer devices enables the system to identify anomalous behaviours. Profiles are very user-dependent and, as such, an approach based on AI/ML offers the flexibility to extract meaningful characterizing features from a wide variety of traffic profiles. AI/ML techniques typically involve two stages: the learning stage, in which the model is trained, and the inference stage, in which the trained model merely classifies input samples. The two stages need not to be performed in the same physical/logical component, adding to flexibility in the design of the architecture discussed next.

4.1 Development Process

The development of the IoT security system will follow an Agile development (develop, integrate, test, demo, feedback, improve). The ML development lifecycle involves additional steps, such as collecting or identifying datasets to train and test the developed ML models. The following points can already be identified (and are discussed further in subsequent sections):

- **Architecture Design** (Section 4.2): we will evaluate existing ML models and how they can be deployed in such distributed edge/cloud computation architecture, taking into consideration: (i) the resource-constrained nature of CPEs vs. higher computing power of cloud nodes; (ii) data availability on either edge and cloud components; and (iii) possibility of using distributed algorithms and/or of the model training and inference stages being performed in different components.
- **Training AI/ML models** (Section 4.3): we will produce or identify a dataset of network traffic to build and train ML models in a fashion that complies with the EU Ethics Guidelines for trustworthy AI. The ML model will be refined over multiple iterations to achieve an adequate performance and it will be validated over the distributed architecture.
- **Evaluation and Customer Perception** (Section 4.4): When the performance of ML models achieve a good accuracy, the proposed system will be tested in a real-environment using real customers as pilots.

4.2 Architecture Design: Distributed Learning Approaches

The higher computing power of cloud servers can complement the security mechanisms at the edge equipment, the CPE. The distribution of the IoT security system over edge and cloud components can be influenced by aspects such as: (i) adopted ML strategy: some ML approaches may lend themselves better to distributed training than others; (ii) data protection: different strategies may require different exchanges of sensitive data; (iii) required data exchange & bandwidth: moving large (network traffic) datasets between edge and cloud is not ideal. We discuss some of the possible strategies for the distribution of the tools for traffic profiling and anomalous traffic detection.

1. Train and inference at CPE: AI/ML techniques are typically computationally expensive, particularly at the training stage, and CPEs (e.g., home gateway and router) are often limited in computational capacity. To perform training or inference at the resource-constrained CPE, strategies such as trading off classification accuracy for resource usage (e.g., using fixed point instead of floating point variables) can be explored. By keeping all user data at the CPE, there is less exposure of user data while traffic data or profiles are being reported back to cloud, albeit measures are necessary to ensure the protection of locally stored data.

2. Train at Cloud: All traffic between the customer network and the Internet passes through the operator's core network, where the necessary computing

power to train the models can be assumed to exist. In this scenario, the model can be trained at the cloud and then either the inference stage is carried out at the cloud server or the trained model can be transferred to the CPE and updated regularly. However, servers at the core network do not observe traffic occurring within the customer network (i.e., device-to-device) unless intentionally mirrored (incurring a penalty in edge-to-cloud traffic). In addition, the observation of customer traffic at the core network may miss information to detect some types of attacks, such as those leveraging covert timing channels.

3. Transfer Learning [13]: The operator may leverage its access to a larger number of households to create generic models that are then fine-tuned to particular customers. This modular training that some ML algorithms can perform allows to ease the training requirements on the CPE. For example, neural networks can be trained by the operator using large datasets from multiple households with similar traffic patterns. At each household, the generic model can be refined by just retraining the last layer of neurons.

4. Federated Learning [8, 6]: Federated learning is a variation of the previous approach, aiming at the development of a high-quality centralized model by aggregating updates provided by multiple clients. While some learning takes place at the edge node (CPE), the requirement on the quality of training at the CPE is alleviated, allowing to benefit of techniques for operation in resource-constrained platforms. In turn, and leveraging a transfer learning approach, the high-quality centralized mode can be transferred back to the CPE for inference operation. However, this option requires identifying and developing an ML algorithm that can learn a shared model from local updates. Also, home network-specific aspects captured in the edge updates, such as detection of intrusions originating in the local network, may be eroded when computing the global model.

Table 2 condenses some of the trade-offs to be considered in the final decision. We discuss the expected model quality, required flows between edge and cloud, and the exposure of the user data while in transit between components. On the point of user data exposure, we take as implicit the need to store data securely both at CPE and cloud server; our focus is on identifying whether there is the need to transmit logs or profiles of user behaviours between the edge and cloud components, thus increasing the exposure of that sensitive information. While the first approach is not strictly distributed (*Training/inference at CPE*), it is discussed as a baseline to the remaining. In the second approach (*Train at Cloud*), the operator may choose whether monitoring traffic internal to the customer network is relevant for the service or not; if it is, such option entails transmitting some sensitive customer traffic information to the cloud. Finally, Transfer Learning and Federated Learning offer a similar distributed architecture, with the latter requiring only the transfer of trained updates or models.

4.3 Training AI/ML Models & Ethical Pillars for Trustworthy AI

There are several AI/ML approaches to perform anomaly detection and thus thwart DDoS attacks. The method (or methods) selected for the IoT security

	Criteria to inform selection			
Approach	**Model quality**	**Edge-cloud flows**	**Cloud-edge flows**	**In-transit exposure of user data**
Train/ inference at CPE	Limited: only CPE resources to train/infer	None	None	Kept solely at CPE
Train at Cloud	Limited, if traffic within home network is not considered	Intra-home network traffic data (if desired)	Trained model transferred to edge	In-transit to cloud
Transfer Learning	Model training uses traffic data from many customers; edge learning may be limited	Intra-home network traffic data (if desired)	Trained model transferred to edge	In-transit to cloud
Federated Learning	Model training uses traffic data from many customers; customer-specific threats may get overlooked	Only training updates sent to cloud	Trained model transfered to edge	Only training updates or models are exchanged

Table 2. Trade-offs to be considered in selecting a distributed architecture.

system should offer performance that meets the agreed service levels, and be suited to operate over a distributed architecture. From the review performed in Section 2, neural networks offer an appealing trade-off between performance and computation cost. Neural networks also benefit of high level of design and operational flexibility that allows them to be mapped into one of the distributed strategies discussed in the previous subsection.

Developing a system to provide user-tailored security requires the machine learning models to be trained with traffic drawn from actual customer networks. This raises data privacy issues to be addressed within the scope of the General Data Protection Regulation (GDPR). In complement to privacy concerns, the European Commission presented on the April 8th, 2019, the Ethics Guidelines for Trustworthy AI. According to the guidelines, trustworthy AI should be:

1. lawful - respecting all applicable laws and regulations;
2. ethical - respecting ethical principles and values;
3. robust - both from a technical perspective while taking into account its social environment.

Based on these principles, the guidelines provide seven key requirements to use AI assuring its trustworthiness:

1. Human agency and oversight;
2. Technical robustness and safety;
3. Privacy and data governance;

4. Transparency;
5. Diversity, non-discrimination and fairness;
6. Societal and environment well-being;
7. Accountability.

The implementation of the proposed architecture involves the development of AI/ML models and, consequently, collecting data several times throughout the AI/ML development life-cycle. During the implementation, the EU Ethics guidelines will be taken into account in order to assure the transparency and trust, but also to prepare the architecture for a production environment following all European recommendations. Thus, we will try to adopt Trustworthy AI when developing, deploying or using AI models, and adapt it to our secure IoT framework use case.

4.4 Evaluation and Perception of Security by Customers

When the outcomes of ML models achieve a good accuracy and performance, the proposed architecture will be tested in a real-environment using real customers as pilots. The pilot tests are important not only for getting information about ML models and system performance, but also for gathering feedback from the customers, particularly:

- if they feel safer knowing that the CPE also provides a service to protect their IoT environment;
- if they still have concerns regarding their privacy (not only related to attacks but also data to be processed by the algorithm);
- if they feel that the new service is affecting the quality of Internet service;
- if they would be willing to keep this type of service and, if yes, under which business models (free of charge, subscription, pay-per-use, others...).

5 Conclusion

This work presented the use-case of secure Internet provision to customers that use IoT devices and are served by an Internet Service Provider. The motivation is to develop a security system capable of detecting Denial-of-Service attacks involving the IoT devices. This is a preliminary study in which we defined an attack model and the security system requirements. This work allowed exposing the main design trade-offs that need to be considered when deciding the distribution of the service components between the edge and cloud. Future work involves the service implementation, training and profiling.

Acknowledgements. This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020), and by the Portuguese National Innovation Agency (ANI) through the Operational Competitiveness Programme and Internationalization (COMPETE 2020) under the PT2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within project(s) grant nr. 69522, POCI-01-0247-FEDER-069522 (MIRAI).

References

1. Banitalebi Dehkordi, A., Soltanaghaei, M., Boroujeni, F.Z.: The DDoS attacks detection through machine learning and statistical methods in SDN. The Journal of Supercomputing (Jun 2020). https://doi.org/10.1007/s11227-020-03323-w
2. Doshi, R., Apthorpe, N., Feamster, N.: Machine Learning DDoS Detection for Consumer Internet of Things Devices. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 29–35. IEEE, San Francisco, CA (May 2018). https://doi.org/10.1109/SPW.2018.00013
3. Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., Sikdar, B.: A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. IEEE Access **7**, 82721–82743 (2019). https://doi.org/10.1109/ACCESS.2019.2924045
4. Hwang, R.H., Lin, P.C., Nguyen, V.L.: An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection **8**, 13 (2020)
5. Hwang, R.H., Peng, M.C., Nguyen, V.L., Chang, Y.L.: An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level. Applied Sciences **9**(16), 3414 (Aug 2019). https://doi.org/10.3390/app9163414
6. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated Learning: Strategies for Improving Communication Efficiency. arXiv:1610.05492 [cs] (Oct 2017)
7. Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I., Kim, K.J.: A survey of deep learning-based network anomaly detection. Cluster Computing **22**(S1), 949–961 (Jan 2019). https://doi.org/10.1007/s10586-017-1117-8
8. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629 [cs] (Feb 2017)
9. Perez-Diaz, J.A., Valdovinos, I.A., Choo, K.K.R., Zhu, D.: A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning. IEEE Access **8**, 155859–155872 (2020). https://doi.org/10.1109/ACCESS.2020.3019330
10. Santos, R., Souza, D., Santo, W., Ribeiro, A., Moreno, E.: Machine learning algorithms to detect DDoS attacks in SDN. Concurrency and Computation: Practice and Experience **32**(16) (Aug 2020). https://doi.org/10.1002/cpe.5402
11. Tahsien, S.M., Karimipour, H., Spachos, P.: Machine learning based solutions for security of Internet of Things (IoT): A survey. Journal of Network and Computer Applications **161**, 102630 (Jul 2020). https://doi.org/10.1016/j.jnca.2020.102630
12. Yuan, X., Li, C., Li, X.: DeepDefense: Identifying DDoS Attack via Deep Learning. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 1–8. IEEE, Hong Kong, China (May 2017). https://doi.org/10.1109/SMARTCOMP.2017.7946998
13. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE **109**(1), 43–76 (Jan 2021). https://doi.org/10.1109/JPROC.2020.3004555