Identification of IoT User Actions in Encrypted Traffic

Pierre-Marie Junges, Jérôme François, Olivier Festor
Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
firstname.lastname@inria.fr

***Abstract*—The development of Internet of Things (IoT) in the last years provide a tremendous playground for attackers. Large attack campaigns against or using IoT devices highlight the underlying risk they convey. In this paper, we are particularly interested by IoT services that are provided through cloud-based applications.**

Although such services that are accessed by Internet leverages de facto relevant practices for security, like encryption, we propose a technique to demonstrate that private information about the user behavior still leak out. In a nutshell, we aim at decomposing a single user command into atomic actions.

I. Introduction

With the emergence of the Internet of Things (IoT), the use of heterogeneous IoT devices becomes widespread. However, many of them suffer from security issues including the lack of updates or the use of default credentials. As a result, IoT devices are now targets for attackers, and compromised IoT devices can led to the creation of major botnets like Mirai [1] or BrickerBot [2]. In addition to these security concerns, IoT devices in smart homes also present a risk of user privacy leakage [3, 4, 5, 6].

Analyzing the IoT traffic is of paramount importance to evaluate the level of private data a malicious user can infer or to profile malicious actions such as attacks that are now mixed within the IoT traffic. We thus propose a traffic analysis technique dedicated to IoT gateways[1], more precisely by observing the Internet traffic of the IoT gateway, which interacts with a cloud-based web service. Such a case neither assumes to be able to observe IoT device communications themselves, and so to be in their close vicinity, nor supposes to eavesdrop the end-user commands.

Fig. 1 shows an example of the IoT system our work focuses on. The user connects to a mobile application, (1) requests a command c containing the actions a_1 and a_2 to be executed on two IoT devices, (2) the web service sends c to the IoT gateway over an encrypted communication channel and (3) the latter transmits a_1 and a_2 to the intended IoT devices using a wireless protocol (that can be proprietary and/or IoT specific and/or encrypted). Our proposed technique aims at decomposing the encrypted application data, observed during step (2), to deduce information about the IoT devices accessed and related requests during step (3).

[1]This paper summarizes our original paper [7]

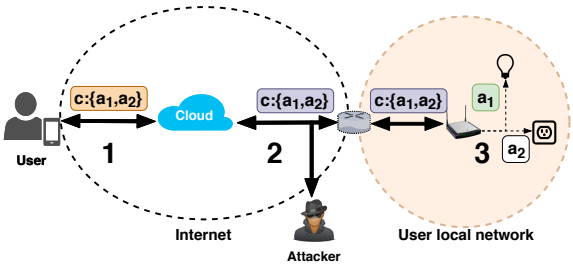

Fig. 1: Attacker model considered in this paper

The paper is structured as follows. Section II introduces the related work on IoT fingerprinting and network traffic analysis. Section III defines the targeted problem. Section IV describes our proposed technique and Section V concludes our paper.

II. Related work

For security purposes, fingerprinting solutions [8, 9] to identify IoT devices in a network using header values (e.g., IP addresses, port numbers, protocol) [8] or period-related features (e.g., period duration, number of periodic flows) [9] were proposed.

IoT devices may also lead to user privacy leakage [3, 4, 6]. In [3], the network traffic generated by a Bluetooth Low Energy (BLE) fitness tracker is analysed and used to identify a person and its current activity. Similar inferences have also been noticed in [4] using the network traffic rate from WiFi devices. In [6], a multi-stage privacy attack able to identify the actions and the states of the IoT devices present in an end-user local network was introduced.

Our work mainly differs by considering exclusively external traffic between the IoT gateway and a web service. However, such traffic is often encrypted which limits the exposure of the IoT devices and their related activities.

From that perspective, a method to identify the web-application related to a SSL traffic was presented in [10] and a fine-grained profiling of user activities of an HTTPS service is possible by reconstructing the sizes of loaded objects [11].

The described techniques used the packet sizes and showed that the use of cryptographic protocols does not guarantee user privacy. In our work, we used the encrypted payload sizes with the aim to decompose it.

III. Problem definition

Considering a vantage point within the end-user IoT network, inferring user activities [6] may be relatively straightforward because the network traffic of each individual IoT

device is observable but forces the attacker to be in a close vicinity which also limits the practicability of the attack.

The presence of IoT gateways in the end-user local network makes the user privacy assessment harder because the gateways receive, from a web service, commands that may concern multiple heterogeneous IoT devices at the same time (see Fig. 1). Thus, once attached to an IoT gateway, these IoT devices are not directly visible or accessible through the Internet.

However, by communicating with a web service through the Internet, the network traffic of the IoT gateway, often encrypted using secure protocols, can be observed. In this work, we evaluate the level of private user information (mainly user actions requested) exposed by an IoT gateway on the Internet.

A. Challenges and assumptions

Considering our point of observation, our approach raises some challenges:

- **C1 - No individual IoT device signature.** IoT devices may be requested by the users to perform multiple actions and the number of IoT devices might be large. Indeed, assuming the user has only 5 IoT devices with 7 possible actions then, it leads to 19607 combinations. As a consequence, it is not possible to learn every combination.
- **C2 - Gateway abstraction.** The IoT gateway receives and processes generic actions. Indeed, even though IoT devices might be completely different (e.g., protocols, brands, models), the IoT gateway receives actions from a single control channel by the web service.
- **C3 - Encryption.** IoT gateway network traffic is encrypted (often using SSL/TLS), so extracting original content from application data is impossible.

Based on preliminary studies and related work described in section II, we make the following assumptions:

- **A1 - Sending actions to the IoT devices.** When the user performs multiple actions on multiple IoT devices *in one command*, we assume that these actions are merged into one actions list *c*.
- **A2 - Incidence of the actions on the packet size.** The larger the list *c* sent from the user to the web service, the larger the corresponding application data sent from the web service to the IoT gateway. So, the actions performed have an incidence on the application data observed and the cryptographic protocol used does not include padding.
- **A3 - Command size stability.** When the same action is performed multiple times, its payload size does not change significantly.
- **A4 - Data structures similarity.** In Fig. 1, we assume some similarities, in their format, between the data sent by the web service to the IoT gateway during step (2) and the original data sent by the user during step (1).

IV. Indirect knowledge extraction

Identifying IoT gateway in a network is not the focus of this paper but the reader can refer to techniques from related works [8, 9]. Here, we consider the IP address of the IoT gateway as known.

Our approach follows three main steps to learn the signatures of individual actions:

- From known user actions, extract relevant features from the corresponding network packets sent by the web service to the IoT gateway.
- Signature construction using the features previously extracted.
- Learning of possible variations between our signatures and the observed encrypted application data sizes.

Once learning achieved, user actions can be identified (testing).

Each of these steps is described in details in next subsections.

A. Features extraction

With respect to our assumptions and challenges, the main feature used is the encrypted application data size.

To derived the size of each possible action $a_1, ..., a_n$, we performed the following steps: (1) perform the action a_i on one IoT device with the user application, (2) extract the data d_i sent to the web service, (3) find the corresponding packets s_i sent by the cloud-based web service to the IoT gateway and (4) repeat the operation by sending a_i on two IoT devices to get another data structure d_{2i} with its corresponding packet s_{2i}. The rationale behind this process is the existence of additional content (such as timestamps), ac, in the message that is not dependent on the actions or their number.

Assuming $|s_i|$ (resp. $|s_{2i}|$), the encrypted application data size of s_i (resp. s_{2i}). Then, the size of action a_i ($|a_i|$) can be computed by subtracting $|s_i|$ to $|s_{2i}|$. Similarly, the size of ac is computed using $|a_i|$ and $|s_i|$.

Our final set of features is composed of $|ac|$ and $|a_i|_{1 \leq i \leq n}$.

B. Learning of the signatures

Once all $|a_i|$ are computed, any size $|s|$ from encrypted application data sent by the web service to an IoT gateway can be rewritten as in equation (1), with $|ac|$ the additional content size, $|a_i|$ the size of the action a_i, $nb\_a_i \in \mathbb{N}$ the number of occurrences of a_i in *s* and $\epsilon \in \mathbb{Z}$, a variation value.

$$|s| = |ac| + \epsilon + \sum_{i=1}^{n} |a_i| \times nb\_a_i \tag{1}$$

C. Learning the variations between the theoretical and observed sizes

We introduced ϵ in (1) because we consider that the encrypted application data size observed may not be exactly equal to the one we can compute using previously inferred $|a_c|$ and $|a_i|_{1 \leq i \leq n}$. The ϵ embeds so both variations of the actions or the additional content.

Hence, ϵ is simply the expected difference between these two sizes (observed and computed by composing $|a_c|$ and $|a_i|_{1 \leq i \leq n}$).

To automatically learn this value, different combinations of actions $A_j = \{nb\_a_1, ..., nb\_a_n\}$ with $j = 1...m$ were

performed and their corresponding encrypted payload size $|s_j|$ retrieved. Then, using (1), each ϵ_j can be derived and a learning dataset containing m tuples $< |s_j|, \epsilon_j >$ is built. Finally, for each new observed encrypted size $|s_j|$ (when we do not control the actions), we search for the closest size in this dataset to deduce the related ϵ_j, *i.e.,* k-Nearest Neighbors classifier (kNN) in one dimension.

D. User action identification

Assuming now the user performs a new command A with $A = \{nb\_a_1, ..., nb\_a_n\}$. The objective is so to automatically infer the value of each nb_a_i with $i = 1...n$, knowing only the global encrypted payload size $|s_c|$ received by the IoT gateway.

Firstly, using the classifier previously trained (kNN), ϵ_c is assigned from $|s_c|$. Then, we subtract $|ac|$ and ϵ_c from $|s|$. Our problem is now similar to the change-making problem [12] and its dynamic programming algorithm can be used to find the commands $A_j = \{nb\_a_{j1}, ..., nb\_a_{jn}\}$ with $j = 1...m$ satisfying equation (1). However, each command A_j is a candidate result, so our technique does not guarantee to return a unique command.

Hence, combining our method with an analysis of the network traffic of the investigated IoT gateway may be useful. Indeed, if we are able to detect or bound the number of actions performed, then it would be possible to remove the commands A_i with $i \in [1, m]$ that do not satisfy this new requirement. Therefore, by removing these commands, we not only reduce the set of possible commands but also keep the commands that best suit the observed situation.

We tested our method on one IoT gateway controlling 12 smart plugs and four smart lamp holders. The smart plugs have four distinct actions whereas the smart holders have seven so, for each encrypted payload size captured, we have to infer the values of seven nb_a_i with $i = 1...7$. To measure the performance of our technique, we performed 307 different commands $A_j = \{nb\_a_{j1}, ..., nb\_a_{j7}\}$ with $j = 1...307$ and our method successfully retrieved the corresponding A_j with a precision of 98.4%. However, for some encrypted payload sizes, multiple commands were found. So, we combined our technique with a network traffic analysis and reduced the number of commands returned but due to the number of actions incorrectly detected, the precision decreased to 91.2%.

V. Conclusion

In this paper, we introduced a method to automatically infer actions requested by a user by solely observing the resulting traffic sent from the web service to the IoT gateway. Our approach consists in finding a correlation between the user inputs (actions performed) and the observed encrypted application data sizes received by the IoT gateways. Thus, to improve the performance of our method it is interesting to combine it with a network traffic analysis of the investigated IoT gateway.

Hence, even though the network traffic is encrypted and the IoT end-devices not directly observable, the presence of an IoT gateway does not prevent an intermediate entity to retrieve fine-grained information about the user activities.

In future work, such an information will be leveraged to create normal activity profiles and detect deviations afterwards.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," vol. 50, pp. 80–84, 01 2017.

[2] Radware, "Brickerbot results in pdos attack," 2017, available at https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service.

[3] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in ble network traffic of wearable fitness trackers," in *17th International Workshop on Mobile Computing Systems and Applications (HotMobile)*. ACM, 2016.

[4] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic," *CoRR*, vol. abs/1708.05044, 2017. [Online]. Available: http://arxiv.org/abs/1708.05044

[5] V. Srinivasan, J. Stankovic, and K. Whitehouse, "Protecting your daily in-home activity information from a wireless snooping attack," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 202–211.

[6] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and A. S. Uluagac, "Peek-a-boo: I see your smart home activities, even encrypted!" 2018.

[7] P.-M. Junges, J. Francois, and O. Festor, "Passive inference of user actions through iot gateway encrypted traffic analysis," in *IFIP/IEEE Workshop on Security for Emerging Distributed Network Technologies (DISSECT), Co-located with IEEE/IFIP IM*, apr 2019.

[8] 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 *37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017.

[9] T. D. Nguyen, S. Marchal, M. Miettinen, M. H. Dang, N. Asokan, and A. Sadeghi, "Dïot: A crowdsourced self-learning approach for detecting compromised iot devices," *CoRR*, vol. abs/1804.07474, 2018. [Online]. Available: http://arxiv.org/abs/1804.07474

[10] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Passive and Active Network Measurement (PAM)*. Springer, 2007.

[11] P.-O. Brissaud, J. Francois, I. Chrisment, T. Cholez, and O. Bettan, "Passive Monitoring of HTTPS Service Use," in *14th International Conference on Network and Service Management (CNSM)*, Rome, Italy, 2018.

[12] J. W. Wright, "The change-making problem," *J. ACM*, vol. 22, no. 1, pp. 125–128, Jan. 1975. [Online]. Available: http://doi.acm.org/10.1145/321864.321874