Protecting IoT-environments against Traffic Analysis Attacks with Traffic Morphing

Ibbad Hafeez
University of Helsinki
ibbad.hafeez@helsinki.fi

Markku Antikainen
Helsinki Institute of Information Technology
markku.antikainen@hiit, aalto.fi

Sasu Tarkoma
University of Helsinki
sasu.tarkoma@helsinki.fi

***Abstract*—Traffic analysis attacks allow an attacker to infer sensitive information about users by analyzing network traffic of user devices. These attacks are passive in nature and are difficult to detect. In this paper, we demonstrate that an adversary, with access to upstream traffic from a smart home network, can identify the device types and user interactions with IoT devices, with significant confidence. These attacks are practical even when device traffic is encrypted because they only utilize statistical properties, such as traffic rates, for analysis. In order to mitigate the privacy implications of traffic analysis attacks, we propose a traffic morphing technique, which shapes network traffic thus making it more difficult to identify IoT devices and their activities. Our evaluation shows that the proposed technique provides protection against traffic analysis attacks and prevent privacy leakages for smart home users.**

I. Introduction

Internet of Things (IoT) revolutionizes our every day living by bringing automation and connectivity to our immediate surroundings. IoT has recently seen tremendous growth and consumer markets are flooded with Internet-connected "smart-devices", ranging from smart plugs to smart refrigerators.

IoT devices require network connectivity even for their basic operations. Research has shown that Internet-connected devices may fail to perform their core function in the absence of network connectivity [1]. This is because very often IoT devices follow an architecture where a large part of device functionality, including the user interface for device configuration, is provided by a cloud service. Thus, even very trivial interactions with an IoT device, such as switching on a smart-plug, may cause network traffic to be sent to the cloud.

The traffic stream for any IoT device can be divided in two sub-streams, representing *background* and *activity* traffic. The background traffic stream contains the communications exchanged between the IoT device and the cloud service when no activity is happening at the IoT device. This stream generally includes keep-alive messages and similar status probes, with low traffic rates. Activity data stream, on the other hand, contains the communications exchanged between an IoT device and cloud services about some activity. The source of such an activity could be a direct user interaction with the IoT device or some change in the surrounding environments which the IoT device is sensing. The activity stream generally has significantly higher traffic rates compared to the background traffic stream.

Usually, network traffic of an IoT device exhibits unique patterns due to different design choices, implementation details, and back-end solutions. An adversary, with access to upstream traffic from a smart home, can study these patterns to identify the type and state of IoT devices generating given traffic flow [2]–[4]. Any information about the type and state of an IoT device can be used to infer online and offline activities of its users [5]–[7]. For example, information about network traffic generated by smart lights can be used to deduce whether there is anyone at home or not. The attacks performed by analyzing network traffic, to identify IoT devices and their activity, can be referred as *traffic analysis attacks*. These attacks are passive in nature and therefore difficult to detect and mitigate.

In this paper, we consider two commonly observed traffic analysis attacks against IoT devices, i.e. *device identification* and *activity recognition*. We demonstrate that an adversary can perform these types of attacks using only the metadata information extracted from IoT devices' network traffic. This metadata information includes DNS queries, connection information and statistical properties of network traffic flows.

We have observed that majority of commercially available IoT devices use secure communication protocols (e.g. TLS) for data encryption. However, encryption does not hide the statistical properties of network traffic, such as traffic rates. Therefore, it is possible to perform the aforementioned traffic analysis attacks, even when network traffic is encrypted.

In order to limit the feasibility of traffic analysis attacks against IoT devices, this paper presents a traffic morphing technique to hide real device traffic from a passive observer snooping on the upstream link of a smart home network. For this purpose, we mask background traffic from IoT devices to prevent the adversary from identifying device-specific patterns in network traffic, and send dummy traffic during the periods an IoT device is inactive, to hide the actual traffic generated due to device activity. Since the dummy traffic only differs slightly from real network traffic in terms of statistical properties, it prevents the adversary from distinguishing real device traffic to perform any attacks. Our results demonstrate that the proposed traffic morphing technique successfully limits the performance of machine learning techniques employed for conducting traffic analysis attacks.

The key contributions of this work can be summarized as:

- We demonstrate the feasibility of traffic analysis attacks

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

for device identification and activity recognition, using only statistical properties of network traffic.
- We propose a traffic morphing technique, which hides background and activity traffic of IoT devices, such that a network observer can not identify IoT devices and their activities.
- We demonstrate that our traffic morphing technique can prevent traffic analysis attacks against user privacy, without affecting functionality of IoT devices.

II. Related Work

Analyzing encrypted traffic to identify user actions is a well-studied problem [7]–[9]. Traffic analysis attacks have been used for tracking online users, detecting smartphone usage and activities etc. [7]. Recently, a number of techniques have been proposed for IoT device identification. These techniques mainly rely on extracting fingerprints from network traffic generated by IoT devices, and using them to identify the type of IoT devices [3], [10], [11]. Research has shown that it is also possible to identify IoT devices, exhibiting malicious activity, by analyzing their network traffic [4], [12], [13]. Such anomaly detection techniques use fingerprints of normal network behavior for IoT device(s) to detect malicious activities [14].

Traffic shaping has been previously used as a countermeasure against traffic analysis attacks [15]–[17]. Previously, Guan et al. [18] proposed traffic shaping by scheduling payload transmissions such that the security requirements are handled without affecting real time requirements of traffic. Acar et al. [19] proposed the use of spoofed traffic, to prevent identification of IoT devices by analyzing their network traffic. However, they do not discuss overhead costs of sending spoofed traffic and its impact on IoT device functionality. Apthorpe et al. proposed constant rate traffic shaping technique to mitigate traffic analysis attacks [1]. Although constant rate traffic shaping restricts an adversary from inferring device state, it incurs significant bandwidth overhead. Meanwhile, any increase in latency, to limit overhead bandwidth consumption, limits device functionality.

III. Adversary Model

In this work, the victim's network is a typical home network with a star topology, where IoT devices are connected to a gateway that provides Internet access. We consider a passive adversary who is able to monitor up- and downstream traffic of the victim's network, as shown in Fig. 1. The adversary could be, for example, the victim's ISP or anyone who has access to network traffic between the victim's gateway and Internet. The goal of the adversary is to learn user's activities from the traffic that is generated by the IoT devices. We only consider passive attacks – while the adversary might be able to learn additional information by actively probing the victim's network, such attacks are not considered.

Many IoT devices use secure communication protocols to secure network traffic. Therefore, in the given scenario, the adversary can only use traffic metadata for the attacks. This

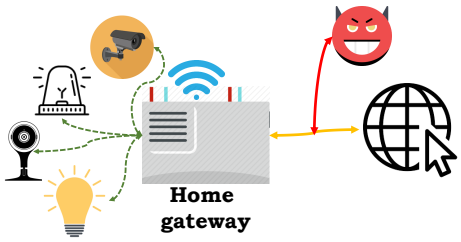

Fig. 1. Smart home testbed network, where an adversary can observe all traffic to- and from smart home network.

work assumes a closed-world setting where the adversary has access to labeled traffic traces of all IoT devices found in the victim's network. The adversary uses these traces to train its machine learning models and there is no upper bound on resources available for this purpose. We also assume that the adversary has full knowledge about traffic morphing and shaping that is being performed at the gateway, and that the adversary can use this information in training the machine learning models.

IV. Testbed

Our testbed setup represents a smart home environment, with commercially available IoT devices such as IoT security cameras, smart plugs, air quality monitors and other sensors.

The testbed uses a Raspberry-PI 3 setup as a wireless access point for connecting user devices. Raspberry-PI also serves as the smart home gateway with upstream connectivity to the Internet. It performs network address translation, runs `DNS` and `DHCP` services for LAN network. All IoT devices are setup from factory-default state and device state changes are triggered by explicit user actions. To mimic an adversary, we record all traffic entering and leaving the smart home network by running `tcpdump` on uplink ethernet interface.

Table I lists the IoT devices used in our testbed, including Google Nest Security Camera (Nest Cam), D-Link WiFi Day/Night Camera (D/N Cam), D-Link Home Siren (Home Siren), Belkin WeMo Insight Smart Plug (Insight Switch), EyePlus Baby Monitor HD camera (Baby Cam), Nokia Video and Air Quality Monitor (VAQM). These devices are representative of the most commonly deployed IoT devices in smart homes [20].

The features supported by IoT cameras include HD video streaming, night vision monitoring, motion and sound detection, and two-way audio streaming. Meanwhile, Home Siren and Insight Switch offer basic features such as on/off and changing device settings. None of these devices have a physical user interface, except for Insight Switch which has a power button. All devices are controlled by a smartphone application provided by the manufacturer.

For data collection, a set of scenarios was drafted based on interactions available for each of the devices and data collection was performed three times for each scenario. Due to limited set of actions supported by these IoT devices, only few scenarios can capture majority of possible user interactions.

TABLE I
IOT DEVICES USED IN SMART HOME TESTBED ENVIRONMENT. MD: MOTION DETECTION, NV: NIGHT VISION, A/V STREAMING: AUDIO/VIDEO STREAMING

Device	Type	Manufacturer	Activity	Firmware
Nest Cam	Camera	Nest Labs	A/V streaming, MD, NV	217-610040
Home Siren	Sensor	DLink	On/Off, Settings	1.22
Baby Cam	Camera	Uniojo	A/V streaming, MD, NV	3.1.1.0908
Insight Switch	Smart plug	Wemo	On/Off, Reporting	
D/N Cam	Camera	DLink	A/V streaming, MD, NV	v2.12-7221
VAQM	Camera, Sensor	Nokia	A/V streaming, MD, NV, sensing	712

TABLE II
DNS QUERIES MADE BY IOT DEVICES

Device	DNS queries
Nest Cam	nexus.dropcam.com, pool.ntp.org, nexus-eu1.dropcam.com, oculus255-eu1.dropcam.com
Home Siren	wrpd.dlink.com, api.dch.dlink.com, ntp1.dlink.com, tzinfo.dch.dlink.com
Baby Cam	esd.icloseli.com, xmpp.icloseli.com, stun.arcsoftcloud.com, relayeu.arcsoftcloud.com, relayjp.arcsoftcloud.com, relayus-w.arcsoftcloud.com
Insight Switch	time.stdtime.gov.tw, nat.xbcs.net, api.xbcs.net
D/N Cam	signal.auto.mydlink.com, ntp1.dlink.com, mp-eu-signal.auto.mydlink.com
VAQM	scalews.withings.net, xmpp.withings.net, xmpp.withings.net, prod-ireland-timeline-2-days.s3.amazonaws.com

Every time the IoT device was individually set up, all incoming and outgoing traffic was collected at the edge gateway. User interactions were performed by a researcher mimicking normal user interactions, in real time. Background traffic was collected by setting up the IoT device and leaving it in operation for several hours. No user interaction took place during this time. After every data collection exercise, the whole testbed including all devices was reset to default state.

We use two commonly used supervised machine learning algorithms, *Random Forest* (RF) and *k-Nearest Neighbors* (kNN), for device identification and activity recognition problems. These algorithms were implemented using Python. We use four fold cross-validation to study the classification performance in terms of precision, recall and F1 score. Four fold cross-validation prevents over-fitting by performing four iterations of training and testing with 25% of data used for hold-out validation in each iteration. For traffic morphing, dummy traffic is generated using `dpkt` and other python libraries and traffic rate limiting is performed using Linux kernel traffic control utility.

V. TRAFFIC ANALYSIS ATTACKS

This section discusses how an adversary can infer information about the type and activity of IoT devices, by analyzing their network traffic metadata only. This work does not aim to develop an extensive device identification or activity recognition technique. Instead, we present proof of concept work to demonstrate the feasibility of traffic analysis attacks, even when network traffic is encrypted.

A. DNS Queries

During normal operation, IoT devices typically communicate with a handful of cloud services only. These cloud services can be associated to device manufacturers and device types. An adversary can use *organizationally unique identifier* (OUI) in device MAC address and the DNS queries, extracted from network traffic, to successfully identify the type and manufacturer of IoT devices deployed in victims' smart home. The additional information about the type and manufacturer of an IoT device can significantly help the adversary to identify user interactions with the device.

We analyzed the DNS queries made by the IoT devices used in our testbed. Table II shows that, in most cases, these queries can be mapped to specific device types and manufacturers. It can also be observed that all devices query more than one domains and the set of domains is unique for each device and manufacturer, for example, D-Link Camera and Home Siren both make DNS queries to ntp1.dlink.com, however, the complete set of domains contacted by each device is unique for the particular device.

We also noticed that most manufacturers used third party cloud hosting services for deploying their IoT cloud services. In our analysis, three out of six devices used Amazon AWS cloud hosting platform, one used Alibaba cloud. Only one (Google Nest Cam) out of six devices used first party cloud services. We expect similar trend for other IoT devices as well, with most manufacturers using third party cloud hosting platforms for deploying cloud services.

B. Statistical Features

In this work, we restrict our traffic analysis techniques to only use statistical properties of network traffic, because in real world scenario, an adversary only has access to these features, due to encrypted communications. We study the network traffic flows of IoT devices as an ordered sequence of packets exchanged between IoT devices and their respective cloud services and isolate the traffic flows corresponding to each device as a time series of packets exchanged during that flow. We record packet sizes, inter-arrival delay, and direction (incoming/outgoing) for each of the packets in given traffic flow. The resulting vectors are divided into n windows, with each window of size w. The feature set extracted from each window includes average and mean absolute standard deviation for packet size, packet rate, inter-arrival delay and traffic rate and this feature is used for both device type identification and activity recognition problems.

The feature set discussed here is protocol agnostic, therefore, it can be used to analyze ZigBee and BLE communications. The optimal value for w varies with problem scenario,

TABLE III
DEVICE TYPE RECOGNITION

Device Name	Random Forest			k Nearest Neighbors		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Nest Cam	0.87	0.86	0.87	0.89	0.87	0.88
Home Siren	0.88	0.9	0.89	0.88	0.93	0.9
Baby Cam	0.91	0.86	0.88	0.88	0.84	0.86
VAQM	0.84	0.85	0.85	0.83	0.77	0.8
Insight Switch	0.94	0.88	0.91	0.89	0.88	0.89
D/N Cam	0.86	0.78	0.82	0.87	0.81	0.84

TABLE IV
DEVICE ACTIVITY RECOGNITION

Device Name	Random Forest			k Nearest Neighbors		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Nest Cam	0.82	0.85	0.83	0.85	0.89	0.87
Home Siren	0.9	0.96	0.93	0.93	0.96	0.95
Baby Cam	0.82	0.85	0.83	0.88	0.85	0.87
VAQM	0.85	0.92	0.88	0.89	0.92	0.91
Insight switch	0.9	0.93	0.92	0.9	0.9	0.9
D/N Cam	0.93	0.89	0.92	0.90	0.93	0.92

for example, activity recognition problem uses smaller w in comparison of device identification problem because network footprint of IoT device activities is short-lived and a larger w will smoothen any spikes in traffic rates indicating device activity. Meanwhile, it should be noted that smaller w improve the performance of classification techniques upto a certain limit only. Beyond this limit, reducing w will negatively affect the performance of classifier in use. Our analysis showed that optimal window size for device identification and activity recognition problem is $w = 10sec$ and $w = 2sec$, respectively.

C. Device-Type Identification

Table III shows that RF and kNN classifiers achieved an average accuracy of 87% and 86%, respectively, for the IoT device identification problem. These results show that it is feasible for an adversary to identify IoT devices, with high accuracy, using only the statistical properties of network traffic. We observed that IoT cameras had different network traffic footprint due to difference in their implementation details such as video compression, media encoding techniques, which helped in achieving higher accuracy for differentiating between different IoT cameras. Meanwhile, Home Siren and Insight Switch support limited functionality (turned on or off) and infrequent variations in traffic rates resulting in a few false positives for these two devices.

Most low power IoT devices use protocols such as ZigBee, BLE to communicate with IoT hubs. These IoT hubs communicate with cloud services using traditional wired or wireless network and the adversary can identify the communications between IoT hubs and cloud services. Given that most low power devices use manufacturer specific IoT hubs, the adversary can infer the type of IoT devices using these IoT hubs.

D. Device Activity Recognition

Due to lack of physical user interfaces, most user interactions with an IoT device typically happen via a smart-phone application. When the user interacts with the smart-phone application, it sends the commands to the IoT device either directly or via cloud service. In either case, the interaction results in additional network communication observable by the adversary, who can identify the interaction by analyzing the statistics of given traffic stream.

Table IV shows that the best performance for activity recognition was achieved for Home Siren and Insight Switch, which can be attributed to the limited set of actions (on/off) available for these two devices. In case of IoT cameras, the best performance is achieved for D/N camera because this camera does not send video feed to cloud service when the user is not streaming video feed to their mobile or desktop client. Meanwhile, other cameras continuously send video feed to cloud services, irrespective of whether or not user is streaming video to their client devices. We also observed that the difference in traffic rates, due to different encoding techniques used by IoT cameras and variation in traffic rates when motion or sound is detected, significantly helps in identifying user interaction with connected cameras.

The results presented in Tab. III and Tab. IV can be further improved by adjusting w, updating classification techniques and using additional features. However, based on current results, it can be established that traffic encryption does not protect user privacy against traffic analysis attacks.

VI. TRAFFIC MORPHING

In order to limit the ability of an adversary to compromise user privacy using traffic analysis attacks, we propose a traffic morphing technique. Our proposed technique masks real traffic from IoT devices in such a way that an adversary is not able to distinguish between real and dummy traffic, thereby, limiting its ability to perform aforementioned traffic analysis attacks and compromise user privacy. We assume that the adversary knows about traffic morphing being performed at the home gateway, therefore, the statistical properties of dummy traffic should be almost identical to real IoT traffic, so that the adversary can not distinguish between real and dummy traffic. To address this requirement, we generate dummy traffic using real traffic captured from IoT devices.

To generate the dummy traffic, we collect information about packet sizes, packet rates, and inter-arrival delay from real traffic of IoT devices. We also capture the IP-five-tuple connection information that is used to replay the traffic. We use real traffic data and cubic-spline interpolation [21], to generate packet sizes and inter-arrival delays for dummy traffic. Table V shows that statistical features of dummy traffic are similar (not exactly identical) to those seen in real background and activity traffic generated by IoT devices. It can also be observed that the inter-arrival delay is significantly different for different IoT devices, as well as it is different for the background (BG_r)

and activity (A_r) traffic. It shows that statistical properties of traffic reveal significant information about the IoT devices.

In order to mask the background traffic, we send traffic on upstream link at a constant rate, irrespective of real background traffic rate of an IoT device. Meanwhile, when an IoT device is inactive, we send dummy traffic representing device activity to upstream link, so that an adversary can not identify real activity of the IoT device. In current design, dummy traffic follows the same path as real traffic and upon reaching the destination, it is silently dropped by the destination service. Any concerns of dummy traffic becoming a denial of service attack against destination service are discussed in section VII.

For traffic management, we specify traffic rate and queueing strategy over the uplink ethernet interface of Raspberry PI using Linux traffic utility. We maintain two separate queues Q_r and Q_d for real and dummy traffic, respectively. The traffic for real and dummy IoT device activity is added to Q_r. As soon as a new packet is available in Q_r, it is immediately forwarded to network driver for transmission. High prioritization of packets in Q_r ensures that there is no additional delay experienced by real traffic from IoT devices. Meanwhile, background dummy traffic is stored in Q_d. Traffic from this queue is sent to upstream link only to maintain a constant traffic rate in scenarios where IoT device is not generating any traffic. It should be noted that the rate of generation of background dummy traffic is high enough that there are always enough packets available in Q_d to be sent on upstream link.

To evaluate the proposed traffic morphing technique, we studied the accuracy achieved for device type identification and activity recognition problems. We observed that the performance of both RF and kNN classification techniques significantly degraded when traffic morphing was applied. For device identification, we were able to achieve maximum true positive rate of 21% and 24% for RF and kNN, respectively. We observed that the best performance for device identification was achieved for the devices with low traffic rates (i.e. Home Siren and Insight Switch). We attribute this performance to small diversity in device set used in testbed setup, that is, if there are other devices with similar traffic rates, the performance will further degrade. Similarly, we only achieved true positive rate of 12% and 11% using RF and kNN, respectively, for activity recognition problem.

In summary, when traffic morphing was applied, the performance achieved for device identification and activity recognition was worse than random guessing. Therefore, it can be assumed that traffic morphing limits the ability of an adversary to perform traffic analysis attacks against IoT devices.

VII. Discussion

Our preliminary results show that the proposed traffic morphing technique provides sufficient protection against traffic analysis attacks. Any traffic morphing technique sends additional traffic to hide real traffic, therefore, it consumes some additional bandwidth. However, the bandwidth consumed by IoT traffic is fairly small due to limited functionality of IoT devices. Consequently, the amount of dummy traffic needed to mask IoT traffic is also very small. In case of IoT devices with high bandwidth consumption, such as IoT cameras, the bandwidth consumed by device activities is only a fraction of total bandwidth consumed. Since we only replay these device activities in traffic morphing, the bandwidth overhead for the proposed traffic morphing technique is fairly limited and does not incur additional costs even if there are data caps on bandwidth available to user.

TABLE V
Mean, IQR and standard deviation for inter arrival delay (in ms) of real ($*_r$) and dummy ($*_d$) traffic for IoT device activity (A_*) and background traffic (BG_*).

	Traffic type	IAD (BG_r)	IAD (BG_d)	IAD (A_r)	IAD (A_d)
Nest Cam	Mean	143.2	143.9	8.7	9.1
	IQR	14.2	13.9	3.65	3.4
	StDev	199.9	202.8	99.1	97.9
Siren	Mean	627.9	628.6	218.3	225.2
	IQR	998.2	1001.9	44.07	45.79
	StDev	1720.4	1775.4	670.0	657.4
Baby Cam	Mean	7.2	6.9	2.4	3.3
	IQR	3.64	3.79	0.25	0.27
	StDev	280.5	279.1	230	228.4
VAQM	Mean	5.7	5.6	3.9	3.8
	IQR	1.58	1.6	1.9	1.9
	StDev	149.1	148.0	31.8	32.2
Insight Switch	Mean	359.9	357.5	119.6	120.0
	IQR	1.25	1.2	3.6	3.7
	StDev	3854.1	3791.3	1268	1267.8
D/N Cam	Mean	560.2	561.7	2	2.5
	IQR	4.8	4.73	1.4	1.3
	StDev	567.1	595.0	16	16.3

It should also be noted that dummy traffic needed to achieve constant traffic rate on upstream network is not proportional to the number of devices in the network. For example, if there are multiple IoT devices in the network, the volume of dummy traffic needed to achieve constant traffic rate will be less than the total volume of dummy traffic needed to mask background traffic of each device individually. This is because dummy traffic is used to smoothen any variations in network traffic which can be attributed to specific devices. In case of more than one device, we only have to smoothen the variations on final network flow exiting home gateway.

In extended deployments, the location of deploying traffic morphing can vary with the location of adversary, for example, in enterprise environments, traffic morphing can be done at the network perimeter where other network middleboxes are deployed. Traffic morphing technique can be employed to mask traffic from specific devices or specific intervals to limit the bandwidth overhead, while maximizing privacy of users.

In case an IoT device itself performs traffic morphing, it will protect the device against any adversaries within the same network as IoT devices. Meanwhile, if every device performs traffic morphing individually, it will increase the overall bandwidth and energy consumption of IoT devices in smart home network. It will also require significant effort to integrate traffic morphing logic in IoT devices due to limited features and support available for IoT device firmware.

Existing constant rate traffic shaping techniques limit the

additional bandwidth consumption at the cost of increased latency. However, high latency negatively affects IoT device functionality because an additional delay in connectivity to cloud services can render IoT devices non-functional in some cases. In order to prevent such scenarios, we prioritize real traffic over dummy traffic to ensure that there is no additional latency and IoT devices can perform their normal functioning, as it can be critical e.g. health IoT devices.

In order to prevent dummy traffic from becoming a denial of service attack against cloud service, dummy traffic can be sent to a sink hole address instead of real destination service. However, a clever adversary can use this information to isolate real traffic and perform traffic analysis attacks. Alternatively, we can use a special flag in dummy traffic packets to direct firewall at destination service to drop these packets at network layer, without any processing. It is also possible to transmit all real and dummy traffic flows via an encrypted tunnel or virtual private network (VPN). In such case, the VPN server at the exit node will be configured to drop all dummy traffic and only send real traffic to destination cloud services. However, this approach incurs additional costs of VPN deployment and maintenance. If only the real traffic from IoT devices is transmitted via VPN tunnel, the additional layer of encryption does not provide significant advantage as the adversary can extract all statistical properties from traffic flows to perform traffic analysis attacks.

VIII. Conclusion and Future Work

In this paper, we have demonstrated that a passive network observer can successfully perform traffic analysis attacks, even when network traffic is fully encrypted. Our results show that it is possible to identify IoT devices and their activity with high accuracy, using only the statistical features obtained from network traffic meta-data information, available to any passive network observer. Such attacks are a serious threat to users' privacy. In order to limit the feasibility of these attacks, we present a traffic morphing technique. Our technique uses dummy traffic to mask the real traffic generated by IoT devices, where dummy traffic is statistically similar to real IoT traffic, therefore, an adversary is not able to distinguish between real and dummy traffic. Traffic morphing makes it difficult for network observer to effectively perform traffic analysis attacks. This phenomenon has been validated by our evaluation, which shows significant drop in performance of supervised machine learning algorithms for device identification and activity recognition.

Acknowledgement

This work was in part supported by Academy of Finland grant number 314008 and Doctoral Programme in Computer Sciences (DoCS) at University of Helsinki.

References

[1] 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

[2] Y. Zhu, Z. Xiao, Y. Chen, Z. Li, M. Liu, B. Y. Zhao, and H. Zheng, "Adversarial WiFi Sensing," *CoRR*, vol. abs/1810.10109, 2018. [Online]. Available: http://arxiv.org/abs/1810.10109

[3] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. R. 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)*, June 2017.

[4] I. Hafeez, A. Y. Ding, M. Antikainen, and S. Tarkoma, "Toward Secure Edge Networks: Taming Device-to-Device (D2D) Communication in IoT," *CoRR*, vol. abs/1712.05958, 2017. [Online]. Available: http://arxiv.org/abs/1712.05958

[5] F. Möllers, S. Seitz, A. Hellmann, and C. Sorge, "Short Paper: Extrapolation and Prediction of User Behaviour from Wireless Home Automation Communication," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*, ser. WiSec '14. New York, NY, USA: ACM, 2014, pp. 195–200.

[6] B. Copos, K. Levitt, M. Bishop, and J. Rowe, "Is Anybody Home? Inferring Activity From Smart Home Network Traffic," in *2016 IEEE Security and Privacy Workshops (SPW)*, May 2016, pp. 245–251.

[7] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing Android Encrypted Network Traffic to Identify User Actions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 114–125, Jan 2016.

[8] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A Survey of Methods for Encrypted Traffic Classification and Analysis," *Netw.*, vol. 25, no. 5, pp. 355–374, Sep. 2015.

[9] S. E. Coull and K. P. Dyer, "Traffic Analysis of Encrypted Messaging Services: Apple iMessage and Beyond," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 5–11, Oct. 2014.

[10] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: ACM, 2017, pp. 506–509.

[11] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "IoTSense: Behavioral Fingerprinting of IoT Devices," *CoRR*, vol. abs/1804.03852, 2018. [Online]. Available: http://arxiv.org/abs/1804.03852

[12] I. Hafeez, M. Antikainen, A. Y. Ding, and S. Tarkoma, "IoTKEEPER: Securing IoT Communications in Edge Networks," *CoRR*, vol. abs/1808.08415, 2018. [Online]. Available: https://arxiv.org/abs/1808.08415

[13] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," *CoRR*, vol. abs/1802.09089, 2018. [Online]. Available: http://arxiv.org/abs/1802.09089

[14] 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

[15] Y. Zhao, B. Zhang, C. Li, and C. Chen, "ON/OFF Traffic Shaping in the Internet: Motivation, Challenges, and Solutions," *IEEE Network*, vol. 31, no. 2, pp. 48–57, March 2017.

[16] W. M. Shbair, A. R. Bashandy, and S. I. Shaheen, "A New Security Mechanism to Perform Traffic Anonymity with Dummy Traffic Synthesis," in *2009 International Conference on Computational Science and Engineering*, vol. 1, Aug 2009, pp. 405–411.

[17] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis," in *In Proceedings of the 16th Network and Distributed Security Symposium*. IEEE, 2009, pp. 237–250.

[18] Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao, "NetCamo: camouflaging network traffic for QoS-guaranteed mission critical applications," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 4, pp. 253–265, July 2001.

[19] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, A.-R. S. Mauro Conti, and A. S. Uluagac, "Peek-a-Boo: I see your smart home activities, even encrypted!" *CoRR*, vol. abs/1808.02741, 2018. [Online]. Available: https://arxiv.org/abs/1808.02741

[20] IoTLineup, "Most popular smart home deviecs," http://iotlineup.com/, IoTLineup, 2018, [Accessed: 2019-01-01].

[21] S. A. Dyer and J. S. Dyer, "Cubic-spline interpolation. 1," *IEEE Instrumentation Measurement Magazine*, vol. 4, no. 1, pp. 44–46, March 2001.