Network Traffic Analysis for IoT Device Identification and Classification Using ML

J. David Livingston[1], B. Samyuktha[2], B. Aishwarya[2], CH.Hima Bindhu[2], D.Sneha[2]

[1] Assistant Professor, Department of Information Technology, Mallareddy Engineering College for Women, (UGC-Autonomous), Hyderabad, India, davidjlivingston@gmail.com.

[2] Student, Department of Information Technology, Mallareddy Engineering College for Women, (UGC-Autonomous), Hyderabad, India.

Abstract

The Internet of Things (IoT) is being hailed as the next wave revolutionizing our society, and smart homes, enterprises, and cities are increasingly being equipped with a plethora of IoT devices. Yet, operators of such smart environments may not even be fully aware of their IoT assets, let alone whether each IoT device is functioning properly safe from cyber-attacks. In this paper, we address this challenge by developing a robust framework for IoT device classification using traffic characteristics obtained at the network level. Our contributions are fourfold. First, we instrument a smart environment with 28 different IoT devices spanning cameras, lights, plugs, motion sensors, appliances, and health-monitors. We collect and synthesize traffic traces from this infrastructure for a period of 6 months, a subset of which we release as open data for the community to use. Second, we present insights into the underlying network traffic characteristics using statistical attributes such as activity cycles, port numbers, signalling patterns and cipher suites. Third, we develop a multi-stage machine learning based classification algorithm and demonstrate its ability to identify specific IoT devices with over 99% accuracy based on their network activity. Finally, we discuss the trade-offs between cost, speed, and performance involved in deploying the classification framework in real-time. Our study paves the way for operators of smart environments to monitor their IoT assets for presence, functionality, and cyber-security without requiring any specialized devices or protocols.

1. Introduction

The number of devices connecting to the Internet is ballooning, ushering in the era of the "Internet of Things" (IoT). IoT refers to the tens of billions of low-cost devices that communicate with each other and with remote servers on the Internet autonomously. It comprises everyday objects such as lights, cameras, motion sensors, door locks, thermostats, power switches and household appliances, with shipments projected to reach nearly 20 billion by 2020 [1]. Thousands of IoT devices are expected to find their way in homes, enterprises, campuses and cities of the near future, engendering "smart" environments benefiting our society and our lives. The proliferation of IoT, however, creates an important problem. Operators of smart environments can find it difficult to determine what IoT devices are connected to their network and further to ascertain whether each device is functioning normally. This is mainly attributed to the task of managing assets in an organization, which is typically distributed across different departments. For example, in a local council, lighting sensors may be installed by the facilities team, sewage and garbage sensors by the sanitation department and surveillance cameras by the local police division. Coordinating across various departments to obtain an inventory of IoT assets is time consuming, onerous, and error-prone, making it nearly impossible to know precisely what IoT devices are operating on the network at any point in time. Obtaining "visibility" into IoT devices in a timely manner is of paramount importance to the operator, who is tasked with ensuring that devices are in appropriate network security segments, are provisioned for requisite quality of service, and can be quarantined rapidly when breached. The importance of visibility is emphasized in Cisco's most recent IoT security report [2], and further highlighted by two recent events: sensors of a fishtank that compromised a casino in Jul 2017 [3], and attacks on a university campus network from its own vending machines in Feb 2017 [4]. In both cases, network segmentation could have potentially prevented the attack and better visibility would have allowed rapid quarantining to limit the damage of the cyber-attack on the enterprise network. One would expect that devices can be identified by their MAC address and DHCP negotiation. However, this faces several challenges: (a) IoT

device manufacturers typically use NICs supplied by third-party vendors, and hence the Organizationally Unique Identifier (OUI) prefix of the MAC address may not convey any information about the IoT device; (b) MAC addresses can be spoofed by malicious devices; (c) many IoT devices do not set the Host Name option in their DHCP requests [5]; (d) even when the IoT device exposes its host name it may not always be meaningful; and lastly (e) these host names can be changed by the user (e.g. the HP printer can be given an arbitrary host name). For these reasons, relying on DHCP infrastructure is not a viable solution to correctly identify devices at scale. In this project, we address the above problem by developing a robust framework that classifies each IoT device separately in addition to one class of non-IoT devices with high accuracy using statistical attributes derived from network traffic characteristics. Qualitatively, most IoT devices are expected to send short bursts of data sporadically. Quantitatively, our preliminary work in [6] was one of the first attempts to study how much traffic IoT devices send in a burst and how long they idle between activities. We also evaluated how much signalling they perform (e.g., domain lookups using DNS or time synchronization using NTP) in comparison to the data traffic they generate. This paper significantly expands on our prior work by employing a more comprehensive set of attributes on trace data captured over a much longer duration (of 6 months) from a testbed comprising different IoT devices.

There is no doubt that it is becoming increasingly important to understand the nature of IoT traffic. Doing so helps contain unnecessary multicast/broadcast traffic, reducing the impact they have on other applications. It also enables operators of smart cities and enterprises to dimension their networks for appropriate performance levels in terms of reliability, loss, and latency needed by environmental, health, or safety applications. However, the most compelling reason for characterizing IoT traffic is to detect and mitigate cybersecurity attacks. It is widely known that IoT devices are by their nature and design easy to infiltrate [7], [8], [9], [10], [11], [12]. New stories are emerging of how IoT devices have been compromised and used to launch large-scale attacks [13]. The large heterogeneity in IoT devices has led researchers to propose network-level security mechanisms that analyse traffic patterns to identify attacks (see [14] and our recent work [15]); success of these approaches relies on a good understanding of what “normal” IoT traffic profile looks like. Our primary focus in this work is to establish a machine learning framework based on various network traffic characteristics to identify and classify the default (i.e., baseline) behavior of IoT devices on a network. Such a framework can potentially be used in the future to detect anomalous behavior of IoT devices (potentially due to cyber-attacks), and such anomaly detection schemes are beyond the scope of this paper. This paper fills an important gap in the literature relating to classification of IoT devices based on their network traffic characteristics.

2. Literature survey

There is a large body of work characterizing general Internet traffic [16], [17], [18], [19]. These prior works largely focus on application detection (e.g., Web browsing, Gaming, Mail, Skype VoIP, Peer-to-Peer, etc.). However, studies focusing on characterizing IoT traffic (also referred to as machine-to-machine or M2M traffic) are still in their infancy.

Analysis of Empirical Traces: The work in [20] is one of the first large-scale studies to delve into the nature of M2M traffic. It is motivated by the need to understand whether M2M traffic imposes new challenges for the design and management of cellular networks. The work uses a traffic trace spanning one week from a tier-1 cellular network operator and compares M2M traffic with traditional smartphone traffic from a number of different perspectives – temporal variations, mobility, network performance, and so on. The study informs network operators to be cognizant of these factors when managing their networks. In [21], the authors note that the amount of traffic generated by a single M2M device is likely to be small, but the total traffic generated by hundreds or thousands of M2M devices would be substantial. These observations are to some extent corroborated by [22], [23], which note that a remote patient monitoring application is expected to generate about 0.35 MB per day and smart meters roughly 0.07 MB per day.

Aggregated Traffic Model: A Coupled Markov Modulated Poisson Processes framework to capture the behavior of a single machine-type communication as well as the collective behavior of tens of thousands of M2M devices is proposed in [24]. The complexity of the CMMPP framework is shown to grow linearly with the number of M2M devices, rendering it effective for large-scale synthesis of M2M traffic. In [25], the authors

show that it is possible to split the (traffic) state of an M2M device into three generic categories, namely periodic update, event driven, and payload exchange, and a number of modelling strategies that use these states are developed. An illustration of model fitting is shown via a use-case in fleet management comprising 1000 trucks run by a transportation company. The fitting is based on measured M2M traffic from a 2G/3G network. A simple model to estimate the volume of M2M traffic generated in a wireless sensor network enabled connected home is constructed in [26]. Since behavior of sensors is very application specific, the work identifies certain common communication patterns that can be attributed to any sensor device. Using these attributes, four generalized equations are proposed to estimate the volume of traffic generated by a sensor network enabled connected apartment/home. Use of Machine Learning: Various machine-learning based analytical methods have been proposed in the literature to classify traffic application or identify malwares/botnets for typical computer networks. The work in [27] uses deep learning to classify flow types such as HTTP, SMTP, Telnet, QUIC, Office365, and YouTube by considering six features namely source/destination port number, payload volume, TCP window size, inter-arrival time and direction of traffic that are extracted from the first 20 packets of a flow. The work carried out in [28] suggests that botnets exhibit identifiable traffic patterns that can be classified by considering features such as average time between successive flows, flow duration, inbound/outbound traffic volume, and Fourier transformation over the flow start times. Detection of malicious activity on the network was enhanced in [29] and [30] by combining these flow level features with packet-level attributes including packet size, byte distribution of payload, inter arrival times of packets and TLS handshake metadata (i.e., cipher suite codes). Further, authors have released an open source libpcap-based tool called Joy [31] to extract these features from the passive capture of network traffic. In the context of IoT, [32] uses machine learning to classify a single TCP flow from authorized devices on the network. It employs over 300 attributes (packet-level and flow-level), though the most influential ones are minimum, median, and average of packets Time-To-Live (TTL), the ratio of total bytes transmitted and received, total number packets with reset (RST) flag, and the Alexa rank of server. While all the above works make important contributions, they do not undertake fine-grained characterization and classification of IoT devices in a smart environment such as a home, city, campus or enterprise. Furthermore, statistical models are not developed that enable IoT device classification based on their network traffic characteristics. Most importantly, prior works do not make any data set publicly available for the research community to use and build upon. Our work overcomes these shortcomings.

3. Proposed System Model

Naïve Bayes Classifier Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.

Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier: There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result. The predictions from each tree must have very low correlations.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.

- It can also maintain accuracy when a large proportion of data is missing.

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

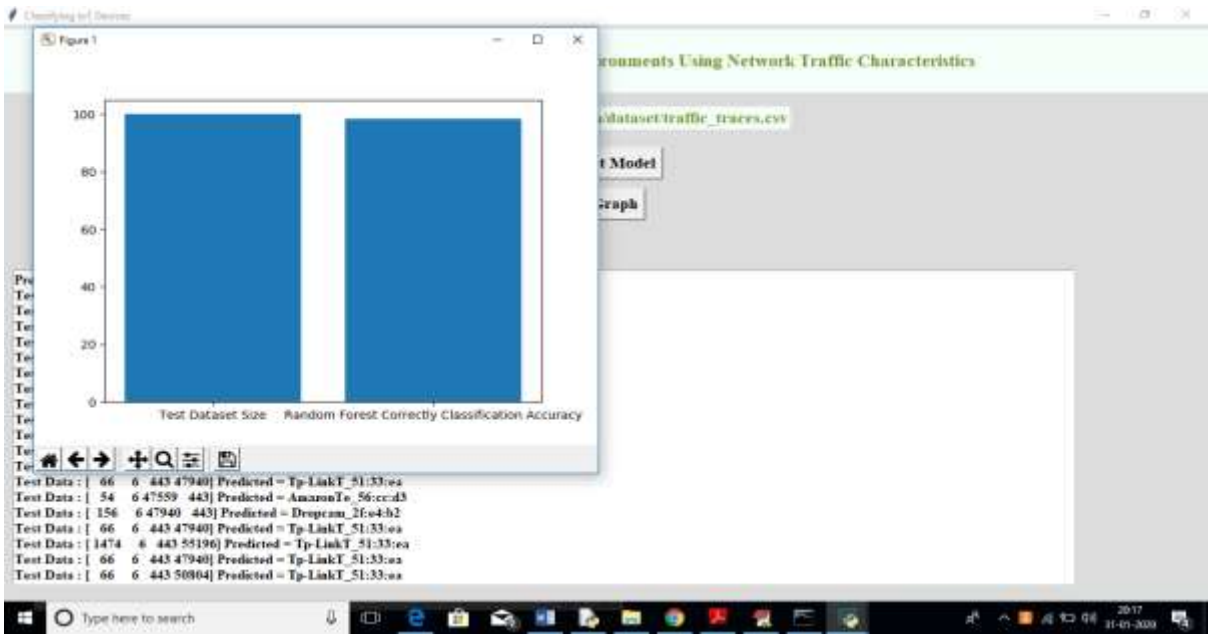
The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random Forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image.

4. Results description

Traffic Traces Dataset

All the traffic on the LAN side was collected using the tcpdump tool running on OpenWrt .It is important to have a one-to-one mapping between a physical device and a known MAC address (by virtue of being in the same LAN) or IP address (i.e. without NAT) in the traffic trace. Capturing traffic on the LAN allowed us to use MAC address as the identifier for a device to isolate its traffic from the traffic mix comprising many other devices in the network. We developed a script to automate the process of data collection and storage. The resulting traces were stored as pcap files on an external USB hard drive of 1 TB storage attached to the gateway.



This setup permitted continuous logging of the traffic across several months. We started logging the network traffic in our smart environment from 1-Oct-2016 to 13-Apr-2017, i.e. over a period of 26 weeks. The raw trace data contains packet headers and payload information. The process of data collection and storage begins at midnight local time each day using the Cron job on OpenWrt. We wrote a monitoring script on the OpenWrt to ensure that data collection/storage was proceeding smoothly. The script checks the processes running on the gateway at 5 second intervals. If the logging process is not running, then the script immediately restarts it, thereby limiting any data loss event to only 5 seconds. To make the trace data publicly available, we set up an Apache server on a virtual machine (VM) in our university data center and wrote a script to periodically transfer the trace data from the previous day, stored on the hard drive, onto the VM. The trace data from two weeks is openly available for download at: http://iotanalytics.unsw.edu.au/. The size of the daily logs varies between 61 MB and 2 GB, with an average of 365 MB. In above graph x-axis represents test data and correctly classified data and y-axis correctly represents classification accuracy. From 100% test data random forest able to predict up to 98%.

5. Conclusion

Our contributions are fourfold. First, we instrument a smart environment with 28 different IoT devices spanning cameras, lights, plugs, motion sensors, appliances, and health-monitors. We collect and synthesize traffic traces from this infrastructure for a period of 6 months, a subset of which we release as open data for the community to use. Second, we present insights into the underlying network traffic characteristics using statistical attributes such as activity cycles, port numbers, signalling patterns and cipher suites. Third, we develop a multi-stage machine learning based classification algorithm and demonstrate its ability to identify specific IoT devices with over 99% accuracy based on their network activity. Finally, we discuss the trade-offs between cost, speed, and performance involved in deploying the classification framework in real-time. Our study paves the way for operators of smart environments to monitor their IoT assets for presence, functionality, and cyber-security without requiring any specialized devices or protocols.

References

[1] I. Spectrum. (Last accessed July 2017.) Popular Internet of Things forecast of 50 billion devices by 2020 Is outdated. https://goo.gl/ 6wSUkk.

[2] Cisco, “Cisco 2017 Midyear Cybersecurity Report,” Tech. Rep., 2017.

[3] A. Schiffer. (2017) How a fish tank helped hack a casino. https: //goo.gl/SAHxCX.

[4] Ms. Smith. (2017) University attacked by its own vending machines, smart light bulbs & 5,000 IoT devices. https://goo.gl/ cdNJnE.

[5] S. Alexander and R. Droms, “DHCP Options and BOOTP Vendor Extensions,” Internet Requests for Comments, RFC Editor, RFC 2132, March 1997. [Online]. Available: https: //tools.ietf.org/rfc/rfc2132.txt

[6] A. Sivanathan et al., “Characterizing and Classifying IoT Traffic in Smart Cities and Campuses,” in Proc. IEEE Infocom Workshop on Smart Cities and Urban Computing, Atlanta, USA, May 2017.

[7] S. Notra et al., “An Experimental Study of Security and Privacy Risks with Emerging Household Appliances,” in Proc. M2MSec, Oct 2014.

[8] F. Loi et al., “Systematically Evaluating Security and Privacy for Consumer IoT Devices,” in Proc. ACM CCS workshop on IoT Security and Privacy (IoT S&P), Texas, USA, Nov 2017.

[9] I. Andrea et al., “Internet of Things: Security vulnerabilities and challenges,” in 2015 IEEE Symposium on Computers and Communication (ISCC), July 2015.

[10] K. Moskvitch, “Securing IoT: In your Smart Home and your Connected Enterprise,” Engineering Technology, vol. 12, April 2017.

[11] N. Dhanjani, Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts. O’Reilly Media, 2015.

[12] E. Fernandes et al., "Security Analysis of Emerging Smart Home Applications," in 2016 IEEE Symposium on Security and Privacy (SP). IEEE, may 2016.

[13] T. guardian. (2016) Why the internet of things is the new magic ingredient for cyber criminals. https://goo.gl/MuH8XS.

[14] T. Yu et al., "Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things," in Proc. ACM HotNets, Nov 2015.

[15] A. Sivanathan et al., "Low-Cost Flow-Based Security Solutions for Smart-Home IoT Devices," in Proc. IEEE ANTS, Nov 2016.

[16] A. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," SIGMETRICS Perform. Eval. Rev., vol. 33, no. 1, pp. 50–60, Jun. 2005.

[17] M. Iliofotou et al., "Exploiting Dynamicity in Graph-based Traffic Analysis: Techniques and Applications," in Proc. ACM CoNEXT, Rome, Italy, Dec 2009.

[18] D. Bonfiglio et al., "Revealing Skype Traffic: When Randomness Plays with You," SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 37–48, Aug. 2007.

[19] R. Ferdous et al., "On the Use of SVMs to Detect Anomalies in a Stream of SIP Messages," in Proc. IEEE ICMLA, Boca Raton, Florida, USA, Dec 2012.

[20] M. Z. Shafiq et al., "A First Look at Cellular Machine-to-Machine Traffic: Large Scale Measurement and Characterization," in Proc. ACM Sigmetrics, England, Jun 2012.

[21] N. Nikaein et al., "Simple Traffic Modeling Framework for Machine Type Communication," in Proc. ISWCS, Germany, Aug 2013.

[22] M. Jadoul. The IoT: The Network Can Make It or Break It. https: //insight.nokia.com/iot-network-can-make-it-or-break-it.

[23] M. Simon and Alcatel-Lucent. Architecting Networks: Supporting IoT.

[24] M. Laner et al., "Traffic Models for Machine Type Communications," in Proc. ISWCS, Germany, Aug 2013.

[25] L. Markus et al., Traffic models for machine-to-machine (M2M) communications: types and applications, 12 2015.

[26] A. Orrevad. M2M Traffic Characteristics: When Machines Participate in Communication.

[27] M. Lopez-Martin et al., "Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things," IEEE Access, vol. 5, 2017.

[28] D. Tegeler et al., "BotFinder: Finding Bots in Network Traffic Without Deep Packet Inspection," in Proc. ACM CoNEXT, Nice, France, Dec 2012.

[29] D. McGrew and B. Anderson, "Enhanced Telemetry for Encrypted Threat Analytics," in Proc. IEEE ICNP, Singapore, Nov 2016.

[30] B. Anderson and D. McGrew, "Identifying Encrypted Malware Traffic with Contextual Flow Data," in Proc. ACM AISec, Vienna, Austria, Oct 2016.

[31] Cisco. (2017) joy. https://github.com/cisco/joy.

[32] Y. Meidan et al., "Detection of Unauthorized IoT Devices Using Machine Learning Techniques," arXiv, 2017. [Online]. Available: http://arxiv.org/abs/1709.04647