Providing Network-Based Datasets and Multi-dimensional Features for IoT Botnet Detection Research

Jie Yin[1,2], Xianda Wu[1,2(✉)], Junnan Wang[1,2], Kun Jia[1,2], Chaoge Liu[1,2], Yue Shi[4], and Xiang Cui[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
wuxianda@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China
[4] China Academy of Information and Communications Technology, Beijing, China

Abstract. The vulnerabilities found in Internet of Things (IoT) devices have caused a large number of IoT devices being compromised and used as botnet platforms, which imposes a serious threat to the cyber security. In order to mitigate this threat, several network-based intrusion detection methods are proposed. While models and algorithms are important, so are the representative datasets and comprehensive feature vectors. In this paper, we firstly construct a botnet network traffic dataset by automatically monitoring some latest IoT botnet samples in our self-built experimental system. The dataset contains 17.5 GB network traffic which generated by 257 samples from 10 families. We can see the samples' entire lifecycle, including installation, propagation, scanning, DDoS attacks, C&C and other typical botnet behaviors. Then, through an in-depth analysis of the collected dataset, we propose a set of feature vectors for detecting. Since we are from the perspective of samples' entire lifecycle, our feature vectors provide more dimensions and is more expressive than existing works. To evaluate the effect of these feature vectors, we design a classification model based on machine learning, and run it on the constructed dataset and another public dataset. The experiment results demonstrate that the proposed feature vectors perform better on our dataset than on others, showing that the future IoT botnet detection model needs to face a longer botnet lifecycle and adopt more comprehensive feature vectors.

Keywords: Internet of Things · Botnet · Dataset · Feature vectors

1 Introduction

With the rapid development of the Internet, Internet of Things (IoT) plays an increasingly significant role in our daily activities. Unfortunately, IoT devices have traditionally lacked of proper control measures and proactive security management (e.g., usage of default passwords, no firmware updates, no access control policy), featuring them as high vulnerable and prone to be compromised devices. These features have been exploited to

X. Sun et al. (Eds.): ICAIS 2021, CCIS 1424, pp. 713–725, 2021.
https://doi.org/10.1007/978-3-030-78621-2_59

create large and powerful botnets by attackers, such as Mirai [1], Haijme [2], VPNFliter [3]. An IoT botnet [4] is a group of compromised IoT devices that can be controlled remotely by a botmaster to execute coordinated attacks. The lifecycle of a botnet such as Mirai are typically constructed in several distinct operational stages, namely propagation, C&C (Command and Control) communication and attack. By using the power of compromised IoT devices, IoT botnets can perpetrate a wide range of malicious attacks, from massive SPAM and phishing campaigns to distributed denial-of-service (DDoS). Most famously, the Mirai botnet [5] was used to launch a 623 Gbps DDoS attack [4] against krebsonsecurity.com, and a 1.2 Tbps attack [6] against Dyn DNS provider in 2016.

To mitigate the threats caused by IoT botnets, the most common countermeasures for defenders is to deploy network-based intrusion detection systems. Detection systems are built and trained using legitimate and malicious data to establish the normality or abnormality behavior patterns, then using machine learning algorithms to identify potential infected devices. The performance of detection systems based on machine learning algorithms heavily rely on data quality and extracted feature vectors. Therefore, a complete dataset and comprehensive feature vectors are key components for a high-performance effective IoT botnet detection system. While there have been in-depth studies into IoT botnet datasets [7, 11], as well as the features used by the detection models [12–14], there remain some drawbacks. First, most existing detection research only focus on some stage of a botnet lifecycle, making the datasets and the feature vectors incomplete, which not only can not accurately de-scribe the behavior pattern of botnets, but also affect the results of the detection model to some degree. Second, as malwares are constantly being updated, the datasets need to be continuously supplemented and improved.

To overcome limitations of existing studies, we focus on detecting botnets during its entire lifecycle, and present an in-depth analysis of IoT botnet traffic. In the dataset aspect, we construct an experimental environment that automatically captures botnet sample's traffic. By running more than 300 latest botnet samples, we construct a new realistic dataset including 257 botnets from 10 families which can truly reflect the behavior of IoT botnet in the wild. In the feature vectors aspect, we propose a new set of features which depict the traffic characteristics of the entire lifecycle of IoT botnets, including the propagation stage, the C&C stage and the attack stage. Our work is aimed at providing representative dataset and comprehensive features that could be useful for researchers to develop the detection model for IoT botnets. The main contributions of this paper are as follows:

- We collect some latest botnet samples, and construct a new realistic IoT botnet traffic dataset by running samples in a self-built lab environment.
- We propose a set of features including three dimensions that cover the network behavior characteristics of a botnet lifecycle.
- We evaluate the detection effect of the proposed features on our dataset by deploying machine learning algorithms, and it performs better than on some existing datasets, demonstrating that both representative datasets and comprehensive feature vectors are important to botnet detection models.

2 Related Work

In recent years, research on detection methods for IoT botnets has gradually studied. Some IoT botnet datasets and detection methods with different feature vectors were proposed.

Koroniotis et al. [7] deployed a testbed relying on some simulated IoT services and network platforms include normal and attacking virtual machines (VMs) with additional network devices such as a firewall, to generate a new realistic Bot-IoT dataset. By implementing different botnet scenarios, the dataset covers various types of attack traffic commonly used by botnets. To evaluate the reliability of the dataset and the performance of detection methods, authors proposed 45 features which indicated an attack flow, the attacks category and subcategory. This work focus on collecting IoT botnet traffic on diverse attack scenarios, and provides the baseline for allowing botnet identification across IoT-specific networks.

Alejandro et al. [8] provided a novel dataset with network data collected from a medium-sized IoT network architecture. To generate legitimate and malicious traffic, some virtual IoT devices were deployed using a Raspberry Pi which allows to emulate the behavior of an IoT device, and three IoT botnet malware were deployed in the IoT devices. The generated dataset focus on the first stages of a botnet deployment, including infection, propagation and C&C communication. Based on this dataset, a total of 100 network traffic statistical features were calculated within different time windows, and used as predictors/input for the machine learning models. Similarly, Bezerra et al. [9] also built an IoT experimental environment using a Raspberry Pi to generate a labelled dataset, but the dataset only covers the infection stage, including the device was not infected, and other ones when multiple botnet malware infected the device.

Meidan et al. [10] proposed and evaluated a novel IoT botnet detection method which extracts behavior snapshots of the network traffic and uses deep autoencoders to detect botnet attacks. The dataset used in this paper gathered from nine commercial IoT devices infected by authentic botnets from two families. To train the detection model, 115 traffic statistics features which capture behavioral snapshots are extracted from the packets' context. Different from [8, 9], this work focus on the attack stage, the propose is to detect compromised IoT devices which have been added to a botnet and have been used to launch attacks.

Different from these studies that only focus on a specific stage of the botnet lifecycle, we propose a set of new features which can cover the nearly all network characteristics of a botnet lifecycle, based on a new generated dataset. Our goal is to address the shortcomings by constructing a new botnet traffic dataset and extracting new detection feature vectors, thus providing a support for developing a high-performance IoT botnet detection model.

3 Experimental Environment and Dataset

This section presents the experimental environment developed to build a dataset, and the detail steps carried out to generate the dataset. The constructed dataset in this paper will be publicly available for the research community, and can be accessed at here [15].

Existing researches usually create a scenario to simulate the IoT botnet activities in local network. For that purpose, it is necessary to set up an isolated network, including the infected devices, C&C servers, real or virtual IoT devices and other network configurations, then capture the traffic generated by the bot. These requirements rely on very time-consuming tasks, multiple device components and complex network structure. Besides, to eliminate the risk of running the botnet sample, the source code is often modified to connect with the specific C&C server which is controlled by re-searchers. However, the C&C server and attack activities are often changed by the botmaster, so the captured traffic in this kind of experimental environment can only represent the behavior of the botnet during a certain period.

To construct datasets that can expose sample's actual behaviors in the wild, we design a real experimental environment allowing botnet samples to connect to real C&C servers. Because the architectures of IoT devices are various, such as MIPS, MIPSEL, ARM, we use QEMU [16] to deploy virtual hosting of several virtual computers on a single computer. QEMU is a free and open-source emulator that performs hardware virtualization, and it can provide a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems [17]. QEMU can also do CPU emulation for user-level processes, allowing applications compiled for one architecture to run on another.

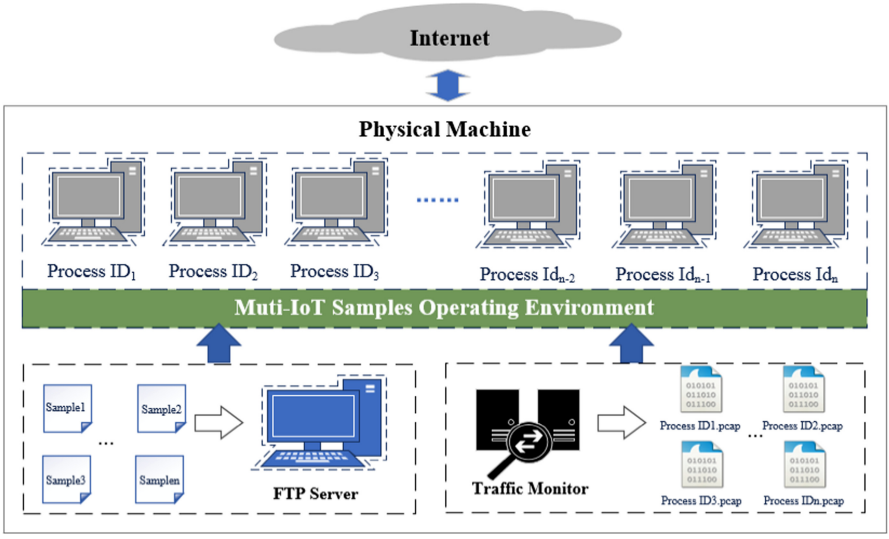

Fig. 1. The architecture of experimental environment.

The architecture of experimental environment is shown in Fig. 1. It consists of three components, namely multi-IoT samples operating environment, traffic monitor and FTP server. The multi-IoT samples operating environment is responsible for running IoT botnet samples. It can support the emulation of various architectures, including MIPS, PowerPC, ARM, RISC-V. The traffic monitor is responsible for capturing the traffic generated by the samples. In this paper, we use the QPA [18] tool as the traffic monitor.

QPA is an open source, real-time traffic analysis software which could capture network traffic generate by specific process. Based on the advantages of process capture, it can accurately determine the process of each package in real time. The FTP server is used to store malware samples. To improve the efficiency of the experiment, we develop a script automatically launch QEMU virtual machine, run botnet samples and capture network traffic.

We collected a set of fresh botnet samples which are captured from honeypots on March 2020. The total number of samples is more than 300, and shown as Table 1, 257 samples from 10 families exhibit network behavior. We executed these samples in the experimental environment and generated 17.5 GB pcap files containing only malicious traffic generated by them. This traffic became an important source for us to analyze IoT botnet network behaviors. Considering that training effective IoT anomaly detection methods rely on the acquisition of both malicious and normal behavioral traffic, we mixed the normal traffic comes from several popular public datasets, including Bot-IoT dataset [7], CTU-IoT-23 dataset [19].

Table 1. The malware families.

Family	Number of samples	Family	Number of samples
Agent	2	Generic	8
CoinMiner	1	Mayday	2
Ddostf	1	Mirai	131
Gafgyt	109	Xarcen	1
Ganiw	1	Xorddos	1

4 Feature Extraction

This section present the feature vectors extracted from the collected network traffic. Bots within a botnet typically exhibit the uniformity of traffic behavior and unique communication patterns. These botnet characteristics are well known and have been exploited by various researchers towards the development of detection systems. Based on the observation, most researches focus on analyzing the botnet network behavior at a certain stage. For example, [13] focus on detecting malicious activities, in particular attacks against DNS, HTTP, and MQTT protocols utilized in IoT networks, so the proposed statistical features are established from aggregating network flows with the potential analysis of MQTT, HTTP, and DNS protocols. In this case, the features are difficult to accurately describe the network behavior characteristics of the complete life cycle of a botnet, the proposed detection model can only be applied to datasets and scenarios containing some specific attacks.

To comprehensively depict the characteristics of botnet networks behavior, we propose a set of features from an in-depth analysis of the raw traffic, which covers the C&C

communication and attack activities of a botnet. The proposed features are based on the flows which are extracted from the captured pcap files, namely our constructed dataset. Flows contain header information about network connections between two endpoint devices. Typically, all transmitted network packets with the same source IP address, source port, destination IP address, destination port and transport protocol within a time window are aggregated into one flow. We divide these features into three groups: basic feature, time-based feature, and service-based feature.

Basic Feature: The basic feature contains flow statistics, packet statistics and the header information of raw packets. The group 'srcip', 'sport', 'dstip', 'dsport', 'proto' are considered network flow identifiers, as this information is capable of uniquely identifying a flow and assisting in the labeling process. The packet statistics assist the examination of the payload beside the headers of the packets. As a supplement of network behavior, we also extract some features of the TCP protocol, mainly containing TCP flags (e.g. SYN flag, ACK flag, etc.).

Time-based Feature: The time-based feature characterizes the regularity of flow behavior over time. The features contain a timestamp attribute like record start time, record last time, a continuous attribute like record total duration and numeric attributes like bytes per second.

Service-based Feature: The service-based feature comprises intrinsic information of the DNS and HTTP protocols which are acquired from the application layer of the TCP/IP model. These two protocols are often used in the C&C communication. Traditionally, malware finds the C&C server through querying the domain, and after establishing contact with C&C server, it begins to download the payload through the HTTP protocol. Therefore, we extract the DNS features from the common DNS queries and responses. The statistical features include the length and mean of the query and answer attributes. Similarly, the HTTP features are created from analyzing the HTTP requests and responses, and generating statistical features.

The detailed feature description is shown in Table 2, Table 3 and Table 4. These features are proposed based on the dataset which are generated in our experimental environment. They depict the botnet network behavior from multiple dimensions, and can be applied to detection models based on machine learning at various scenarios.

Table 2. Basic features.

#	Feature	Description
1	srcip	Source IP address
2	sport	Source port number
3	dstip	Destination IP address
4	dsport	Destination port number
5	proto	Transaction protocol
6	pktcount	Total count of packets
7	spkts	Source to destination packet count
8	dpkts	Destination to source packet count
9	sloss	Source packets retransmitted or dropped
10	dloss	Destination packets retransmitted or dropped
11–16	spktlength	Statistics on the packet size transmitted from source to destination packet (sum, max, min, avg, std and var)
17–22	dpktlength	Statistics on the packet size transmitted from destination to source packet (sum, max, min, mean, avg, std and var)
23–28	pktlength	Statistics on the packet size (sum, max, min, avg, std and var)
29–34	pktheader_length	Statistics on the packet header size (sum, max, min, avg, std and var)
35–40	spktheader_length	Statistics on the packet header size transmitted from source to destination packet (sum, max, min, avg, std and var)
41–46	dpktheader_length	Statistics on the packet header size transmitted from destination to source packet (sum, max, min, avg, std and var)
47	pktratio	The value of spkts divided by dpkts
48	pktlength_entropy	Entropy of packet length
49	spktlength_entropy	Entropy of source to destination packet length
50	dpktlength_entropy	Entropy of destination to source packet length
51	pktheader_entropy	Entropy of packet header length
52	spktheader_entropy	Entropy of the packet header size transmitted from source to destination packet
53	dpktheader_entropy	Entropy of the packet header size transmitted from destination to source packet
54–61	tcp_flag	TCP flags (FIN,SYN,RES,PSH,ACK,URG,ECE,CWR)
62–69	stcp_flag	TCP flags from source to destination (FIN, SYN, RES, PSH, ACK, URG, ECE, CWR)
70–77	dtcp_flag	TCP flags from destination to source (FIN, SYN, RES, PSH, ACK, URG, ECE, CWR)
79–81	swin	Source TCP window advertisement value(max, avg, std)
82–84	dwin	Destination TCP window advertisement value (max, avg, std)

Table 3. Time-based features.

#	Feature	Description
85	duration	Record total duration
86	bytes_second	Bytes per second
87	sbytes_second	Frequency of bytes transmitted from source to destination
88	dbytes_second	Frequency of bytes transmitted from destination to source
89	pkts_second	Packets per second
90	sbytes_second	Frequency of Packets transmitted from source to destination
91	dbytes_second	Frequency of Packets transmitted from destination to source
92–96	pkttimedistance	Statistics on time interval of all packets (sum, max, min, avg, std)
97–101	spkttimedistance	Statistics on time interval of source to destination packets (sum, max, min, avg, std)
102–106	dpkttimedistance	Statistics on time interval of destination to source packets (sum, max, min, avg, std)
107	sttl	The TTL of source to destination packets
108	dttl	The TTL of destination to source packets
109–111	sjitter	Statistics on jitter time from source to destination (max, min, avg)
111–113	djitter	Statistics on jitter time from destination to source (max, min, avg)

Table 4. Service-based features.

#	Feature	Description
114	Domain_count	Number of domain names requested
115	q_type	DNS Query type
116–118	len_qry	Length of DNS query packets (max, avg, std)
119–121	len_asn	Length of DNS response packets (max, avg, std)
122–123	ttls	Statistics on DNS query packet buffer time (max, std)
124	host_count	Total number of HTTP request hosts
125	len_host	Average length of the HTTP host header
126	url_count	The total number of HTTP request URLs
127	len_url	Average length of URL length

5 Evaluation

In this section, we present the design and results of several experiments conducted to evaluate the proposed features and generated dataset.

5.1 Detection Model

To distinguish botnet and benign traffic effectively, we design a detection model based on the proposed features, as shown in Fig. 2. The detection model consists of three classifiers, and each classifier is used to process the network flow data of the TCP/UDP, HTTP, DNS protocol, respectively. We first extract all flow information from the given pcap files, then parse the flows into relevant attribute vectors based on different protocols, and use corresponding classifier to identify malicious and benign traffic. As shown in Table 5, there are a total of 106,827 malicious flows in our test set. Of these flows, 76,315 are DNS flow, 512 are HTTP flow, and 30,000 are TCP/UDP flow. A total of 8,963 non-malicious flows are extracted from Bot-IoT dataset and CTU-IoT-23 dataset, including 4,509 DNS flows, 336 HTTP flows and 4,118 TCP/UDP flows.

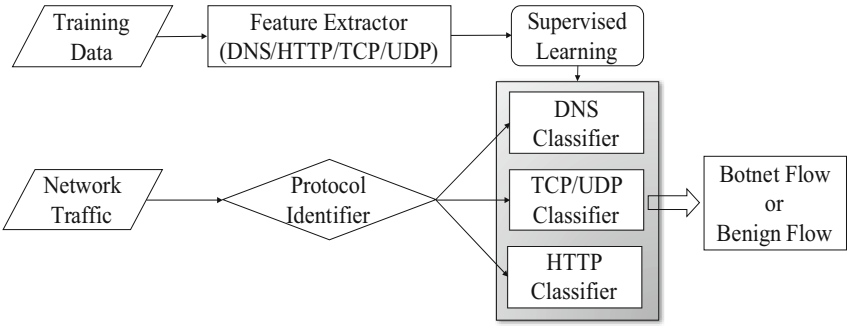

Fig. 2. Detection model.

We investigated different machine learning based classification techniques, finally selected the random forest classification algorithm for our classifiers. For each classifier, two performance metrics are reported: Precision and Recall. Precision (Pre) refer to the fraction of positive instances correctly classified among all the positive classified instances. Recall refer to the fraction of positive instances correctly classified among all the actual positive instances. All the performance metrics are bounded on the interval [0, 1]. If the classifiers show a performance close to 1 in both metrics it may be inferred that the data is suitable for machine learning-based IoT botnet detection, and can effectively distinguish between malicious traffic and legitimate traffic. To evaluate detection accuracy, we used the 5-fold cross validation. The results are listed in Table 6. As we can see, the detection model produce a very high precision and recall, 0.9996 and 0.9998 respectively in malicious flows, 0.9978 and 0.9949 respectively in non-malicious flows. We also observe that the results of the TCP/UDP classifier and the HTTP classifier both achieve a perfect value 1. On the one hand, it may be because the amount of HTTP traffic

is too small, the difference between benign traffic and malicious traffic is obvious, so that it is easy to be detected; on the other hand, for the TCP/UDP classifier, it shows that the basic feature can well distinguish malicious traffic patterns from benign ones. Overall, these results indicate that the proposed feature vectors are effective in detecting IoT botnets.

Table 5. The number of flows in dataset.

Categories	DNS	HTTP	TCP/UDP	Total
Malicious flow	76,315	512	30,000	106,827
Non-malicious flow	4,509	336	4,118	4,509

Table 6. Classification results.

Categories	Total		TCP/UDP classifier		HTTP classifier		DNS classifier	
	Pre	Recall	Pre	Recall	Pre	Recall	Pre	Recall
Malicious	0.9996	0.9998	1	1	1	1	0.9994	0.9997
Non-malicious	0.9978	0.9949	1	1	1	1	0.9954	0.9897

5.2 Comparison

To prove that our features are relatively robust against various detection scenarios, we design two comparison experiments. First, we evaluate classification result based on Bot-IoT dataset using Bot-IoT features compare with ours. Bot-IoT dataset incorporates both normal IoT-related and other network traffic, along with various types of attack traffic commonly used by botnets. To enhance the predictive capabilities of classifiers, authors proposed 45 features which describe the patterns of several attacks. Based on the Bot-IoT dataset, we trained two classification models using our features and theirs, Table 7 lists the results of classification. We can see that the precision and recall using our features is higher than using theirs, which shows that the proposed features can more comprehensively represent the botnet network behavior pattern, effectively improve the predictive capabilities of classifiers. Second, we evaluate classification result using the Bot-IoT features on our dataset. As shown in Table 8, the result also achieves a very high precision and recall.

Both comparison results show that the proposed features can be applied to various datasets and detection scenarios, and achieve better performance. Moreover, our dataset is updated and more abundant, so we believe that the detection model trained with our feature vectors and dataset may be more powerful.

Table 7. Classification based on Bot-IoT dataset using different features.

Categories	Our features		Bot-IoT features	
	Pre	Recall	Pre	Recall
Malicious	0.9937	0.9951	0.9936	0.9941
Non-malicious	0.9768	0.9702	0.9718	0.9697

Table 8. Classification based on the proposed dataset using Bot-IoT features.

Categories	Total		TCP/UDP classifier		HTTP classifier		DNS classifier	
	Pre	Recall	Pre	Recall	Pre	Recall	Pre	Recall
Malicious	0.9933	0.9994	1	0.9962	1	0.9701	0.9995	0.9942
Non-malicious	0.9996	0.9950	0.9995	1	0.9806	1	0.9956	0.9920

5.3 Feature Importance

We also evaluated the feature importance on three classifiers based on the Random Forest algorithm. Feature importance gives a score for each feature of the dataset, the higher the score more important or relevant is the feature towards results. The TCP/UDP classifier used the basic feature and time-based feature, except the five tuple features, there are 108 features. The HTTP classifier used the basic feature, time-based feature, and HTTP feature, a total of 112 features, and the DNS classifier used 118 features. The sum of the scores of these features used by each classifier is 1. According to the algorithm, we got the top 10 features in each classifier as shown in Fig. 3, Fig. 4(a) and Fig. 4(b). For TCP/UDP classifier, we identified toe following best 10 features: sum_spktlength, avg_swin, max_swin, sum_pktlength, sum_spktheaderlen, sum_spktlength, pktcount, bytes_second, fwdpkts, slos. We can notice that the basic feature has higher weight than other features, a similar situation occurs in the HTTP classifier and the DNS classifier. In particular, the statistics features of packet header are more important in HTTP classifier. For DNS classifier, there are three features related to DNS protocol in top 10 features: the type of query, the max length of query and the average length of query. It shows that the domain query in IoT botnets is quite different from the normal domain query.

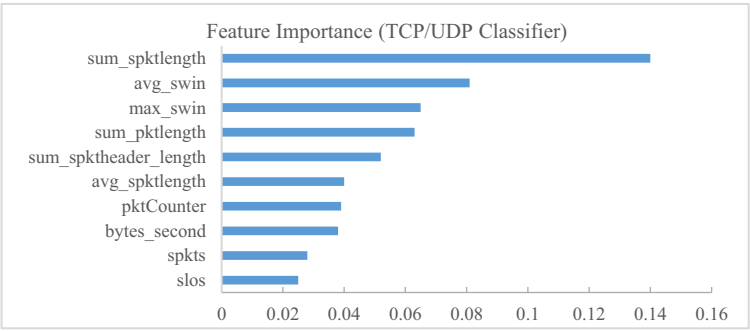

Fig. 3. The feature importance on TCP/UDP classifier.

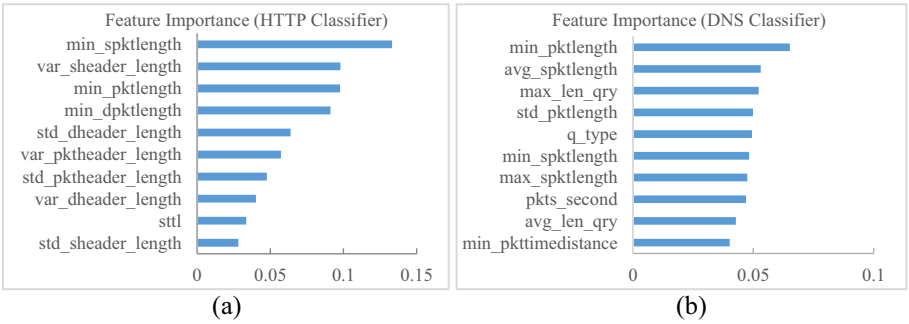

Fig. 4. The feature importance on HTTP classifier (a) and DNS classifier (b).

6 Conclusion

In this paper, we present an experimental environment based on QUME to automatically capture botnet samples' network traffic. By running 257 samples from 10 families, we construct an IoT botnet network traffic dataset, which covers all samples' network behaviors during their entire lifecycles, including propagate, C&C communication and attack stage. Through in-depth analysis of the botnet network behavior, a set of features are proposed. We divide these features into three groups, basic feature, time-based feature, and service-based feature which related to DNS and HTTP protocol. To measure the effect of these features, we design a detection model based on Random Forest algorithm. The results show that the proposed detection model using our extracted features perform well on both benign flows and malicious flows based on our dataset, with the precision and recall being above 99%. Besides, our features perform also well on both our and the other's dataset, especially in detecting malicious traffic. It is demonstrated that the proposed features have the potential to be applied to various detection scenarios. Finally, we compute feature importance on the generated dataset, and get the top 10 features. Although algorithms are the core of botnet detection model, the updated datasets and comprehensive feature vectors used for training and evaluation are important as well.

We hope that the dataset and the features provided in this paper can benefit researchers to develop and compare their detection methods.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 61902396), the Youth Innovation Promotion Association CAS (No. 2019163), the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02040100), the Key Laboratory of Network Assessment Technology at Chinese Academy of Sciences and Beijing Key Laboratory of Network security and Protection Technology.

References

1. Antonakakis, M., April, T., Bailey, M., et al.: Understanding the mirai botnet. In: Proceedings of USENIX, USA, pp. 093–1110 (2017). Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Herwig, S., Harvey, K., Hughey, G., et al.: Measurement and analysis of Hajime, a peer-to-peer IoT botnet. In: Proceedings of NDSS (2019)
3. Sicato, J.C.S., Sharma, P.K., Loia, V., et al.: VPNFilter malware analysis on cyber threat in smart home Network. Appl. Sci. **9**(13), 2763 (2019)
4. Bertino, E., Islam, N.: Botnets and internet of things security. Computer **50**(2), 76–79 (2017)
5. Akamai: Akamai's State of the Internet/Security, Q3 2016 Report (2016). https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q3-2016-state-of-the-internet-security-report.pdf
6. Dyn: Dyn analysis summary of Friday October 21 attack (2016). https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/
7. Koroniotis, N., Moustafa, N., Sitnikova, E., et al.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-IoT dataset. Futur. Gener. Comput. Syst. **100**, 779–796 (2019)
8. Guerra-Manzanares, A., Medina-Galindo, J., Bahsi, H., et al.: MedBIoT: generation of an IoT botnet dataset in a medium-sized IoT network. In: Proceedings of ICISSP, pp. 207–218 (2020)
9. Bezerra, V.H., da Costa, V.G.T., Martins, R.A., et al.: Providing IoT host-based datasets for in-trusion detection research. In: Proceedings of SBC, pp.15–28 (2018)
10. Meidan, Y., Bohadana, M., Mathov, Y., et al.: N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Comput. **17**(3), 12–22 (2018)
11. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of MilCIS, pp. 1–6 (2015)
12. Sagirlar, G., Carminati, B., Ferrari, E.: AutoBotCatcher: blockchain-based P2P botnet detection for the Internet of things. In: Proceedings of 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pp. 1–8 (2018)
13. Moustafa, N., Turnbull, B., Choo, K.K.R.: An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet Things J. **6**(3), 4815–4830 (2018)
14. McDermot, C.D., Majdani, F., Petrovski, A.V.: Botnet detection in the internet of things using deep learning approaches. In: Proceedings of 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2018)
15. Our Dataset. https://pan.baidu.com/s/19gR-lqT6kGy1iKYqxo0Rrw. code:zr2t
16. QUME. https://www.qemu.org/
17. QUME Wiki. https://en.wikipedia.org/wiki/QEMU
18. QPA. http://www.l7dpi.com/
19. CTU-IoT-23 Dataset. https://www.stratosphereips.org/blog/2020/1/22/aposemat-iot-23-a-labeled-dataset-with-malicious-and-benign-iot-network-traffic