2021 6th International Conference on Development in Renewable Energy Technology (ICDRET) | 978-1-6654-7826-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/DSC53577.2021.00020

Identification Domain Fronting Traffic for Revealing Obfuscated C2 Communications

1st Zeyu Li
School of Cyberspace Security
Beijing University of Posts and Telecommunications
Beijing, China
lzreal@bupt.edu.cn

2nd Meiqi Wang
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
wangmeiqi@iie.ac.cn

3rd Xuebin Wang
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
wangxuebin@iie.ac.cn

4th Jinqiao Shi
School of Cyberspace Security
Beijing University of Posts and Telecommunications
Beijing, China
shijinqiao@bupt.edu.cn

5th Kexin Zou
School of Cyberspace Security
Beijing University of Posts and Telecommunications
Beijing, China
zoukexin@bupt.edu.cn

6th Majing Su
The 6th Research Institute of China Electronice Corporation
China Electronics Corporation
Beijing, China
sumj@ncse.com.cn

Abstract—Nowadays, as networks grow in size, the scope of malware and malicious traffic is also increasing quickly. For example, some attacks turn a group of Internet-connected hacked devices into botnets, and a command-and-control(C2) tunnel is built to herd bots for illicit purposes such as massive DDoS. In order to evade Internet malware detection, a variety of techniques are used to obfuscate the C2 communications, of which Tor domain-fronting is one of the most sophisticated techniques. In this paper, a method based on deep learning for domain-fronting traffic identification is proposed. CNN model is adopted which integrates feature learning into the training process so that it can classify traffic based only on packet sequences. We identify the meek-azure traffic and meek-fastly traffic mixed with different types of traffic including tor traffic and non-tor traffic, and the method can achieve rather high precision and accuracy of 99.69%. Furthermore, we identify the domain fronting traffic mixed with the non-domain-fronting traffic with the same Server Name Indication (SNI), and the result shows that our method achieves the accuracy of 97.35% by identifying the domain fronting traffic from the mixed dataset. The results of this work provide a new approach to detect obfuscated C2 communications of the botnet.

Index Terms—Domain Fronting, Command-and-Control, Traffic Identification, Tor, Meek

I. Introduction

Botnets are a new type of attack method that has developed and converged on the Internet based on traditional malicious code forms such as network worms, Trojan horses and backdoor tools. Usually the bot herders use botnets to launch large-scale network attacks, such as distributed denial of service attacks (DDoS), massive spam, etc. At the same time, the files in the computer which is a part of the botnet are not safe. Therefore, botnets are a great threat to both personal information security and network security. For instance, the 2016 Mirai attack used infected IoT devices to create a botnet and launch attacks against the domain resolving Internet infrastructure, causing shutdowns across Europe and North America that resulted in an estimated $110 million in economic damage [3].

Botnets are often spread using secondary injection. For a freshly compromised computer, bot herders use command and control tunnels to cause the compromised host to download and install malware on the C2 server itself, allowing the attacker to have the highest privileges over the computer and preventing other behaviours of modifying the computer's privileges. Public cloud services and Content Delivery Networks (CDNs) are frequently used to host or mask C2 activity. It's also common for hackers to compromise legitimate websites and use them to host command and control servers without the owner's knowledge [44]. In addition, the bot herder remotely control the compromised bots via the command and control infrastructure to launch attacks, gather sensitive information and so on. So the C2 tunnel is one of the most important components for a botnet [4], [5]. Since the detection of the C2 communication can reveal both the C2 server and the infected bots in the network, it is of great value to fight against botnets by detecting the C2 tunnels. However, the battle between the botnet operator

This work is supported by the Fundamental Research program of National Defence(JCKY2019211B001)

978-1-6654-1815-7/21/$31.00 ©2021 IEEE
DOI 10.1109/DSC53577.2021.00020

and security researcher has been escalating because more and more advanced C2 architectures are proposed, of which Tor domain-fronting is one of the most sophisticated techniques.

Tor is the most popular anonymous communication system [6], which provide two basic characteristics. The first is untraceable encrypted communication that traffic are encrypted and routed in a multi-hop manner through the Tor network, so that the origination and destination of the path cannot be correlated. The second one is hidden service that a server can be accessed by users in a mutual anonymous manner without revealing its location. Moreover, in order to obfuscate the Tor traffic, more sophisticated pluggable transports protocols such as Meek [7]–[9] that can turn tor traffic into innocent TLS traffic to cloud servers, which makes it indistinguishable from legitimate requests. With these attracting characteristics, botnet can masquerades its traffic as legitimate encrypted traffic to evade C2 traffic identification and build hidden service to protect the localization of its C2 server from taken-over. Tor-based botnet was first discussed in Defcon 2010 [10], and the first practical Tor based botnet Skynet was discovered in 2012. In 2017, FireEye revealed an APT campaign which uses Meek domain fronting to cover their trace [11]. Recently, the emerging IoT botnets also begin to adopt tor to protect their C2 channels. In March 2021, researchers from NetLab 360 revealed a new IoT botnet Gafgyt which used Tor to obfuscate C2 communications [12]. Thus, identity the Tor traffic, especially the tor domain fronting traffic in network is quite important for botnet discovery.

Great effort has been made by researchers to detect malicious traffic in network [13], [14]. However, it is still an open problem to effectively detect the innocent-looking encrypted traffic that are generated by the obfuscate C2 communication of botnets, which adopts the sophisticated techniques such as Tor domain fronting to protect the C2 infrastructure. hackers often imbed C2 commands into legitimate websites or deploy a web-type C2 server for C2 communication, so the detection of C2 traffic using domain fronting technique can be transformed into a problem of detecting domain fronting traffic creating by website. Therefore, in this research work, we propose a novel approach to identify Meek domain fronting network traffic by using deep learning algorithms. The main contributions of our work are as follows:

- We propose a method for identifying domain fronting traffic using a CNN model. We process the traffic trace collected into the length sequence of TLS Application data, and input the length sequence directly into the model, avoiding manual feature extraction.
- We identify the meek traffic among four different types of traffic by using our method. The types of traffic including Meek-Azure, Meek-Fastly, Tor-Direct, Non-Tor. We capture 1500 traces of each type of traffic which the dataset calld E1500, and we achieve a high accuracy of 99.69% for identifying the domain fronting traffic from the mix traffic of two different types of Meek traffic, Tor traffic and non-Tor traffic.
- Further more, We intend to detect meek traffic with the same SNI. This is a much more realistic scenario because meek uses domain fronting to talk to a Tor relay while appearing to talk to another domain. We screened the traffic with the same SNI, including both meek traffic and no meek traffic under the environment of China Unicom and China Telecom. By using our method, we achieve a high accuracy of 97.35% on this larger dataset which all the traffic talk to the Microsoft's CDN and has the same SNI.

The remainders of the paper are organized as follows: Section 2 introduces the background of Tor and domain fronting techniques, and summarizes the prior works done in malware traffic analysis. In Section 3, we give a detailed introduce of the proposed approach for domain fronting traffic identification with deep learning algorithms. Section 4 introduces the dataset used in the experiments and presents experiment results and analysis. Section 5, we discuss the lack of our work and compare the differences with another work. Finally, Section 6 concludes the paper with future work.

II. Background and Related Work

A. Discovery and Identification of Botnet Traffic

Botnets are one of the biggest threats to Internet, therefore, a large amount of research work has been done around how to discover and identify botnets, and the main approach is to detect botnets by analyzing traffic behavior. ALQAHTANI proposed an efficient method for IoT botnet attack detection [19], which used a Fisher score-based feature selection method and a genetic-based extreme gradient boosting (GXGBoost) model to determine the most relevant features and detect IoT botnet attacks. YIN L designed and implemented a lightweight IoT-based botnet detection system that detects IoT-based botnets by fast identification of algorithm-generated domain streams (AGD) [20]. The paper evaluated the approach with real-world DNS traffic collected from two different large ISP networks and showed that it is able to accurately identify devices compromised by unknown botnets. The literature proposed a method about revealing IoT devices which infected malicious code, and the main ideal is to identify the untargeted Darknet traffic [21]. Another literature described a method to target botnet attacks in different scenarios of Bashlite and Mirai [22], which uses multiple machine learning and deep learning models, and found that CNN's model had better results. Also this literature found that the traffic of multiple protocols (e.g. TCP, UDP) can help better detection of IoT botnets. MEIDAN Y proposed a new network-based IoT anomaly detection method [23], N-BaIoT, which extracted a snapshot of the network's behavior and adopted Deep Autoencoders to detect anomalous network traffic from compromised

IoT devices. It also used IoT-based botnets Mirai and BASHLITE in detection with good results. HAFEEZ I‘s research achieves 98% accuracy by semi-supervised learning with 39 features extracted from logs and using a fuzzy C-Mean (FCM) algorithm to distinguish malicious and benign traffic [24]. The literature's research classified malicious and benign traffic by bundling a stacked deep learning approach with five pre-trained residual networks (ResNets) with models in the Power System Dataset and N-BaIoT database with good results [25]. In [26], a machine learning-based approach for IoT botnet attack detection was proposed, which used Pearson Correlation Coefficient (PCC) and Relief-F for Data features were filtered and the model was trained in J48, Random Forest, and Multilayer Perceptron (MLP) neural networks, which not only helped to distinguish normal and malicious traffic, but also detected the type of IoT botnet attacks.

Identification of malicious traffic in Botnets based on machine learning and deep learning methods is a hot research topic in recent years. A new feature selection algorithm called CorrAUC was proposed in [15], which metrics and filters features by area under the curve (AUC), and then applies integrated TOPSIS and Shannon entropy based on bijective soft sets to validate the selected features for malicious traffic identification in IoT networks. CorrAUC achieved good results in different machine learning. An improved version of the most popular feature selection algorithm was proposed, which was named Fast-Correlation Feature (FCBF) [16]. It optimized machine learning models by partitioning the feature space into equal-sized segments to enhance correlation. In recent years, with the rise of deep learning techniques, deep learning methods have been introduced to IoT malicious traffic detection. BENDIAB proposed a new approach for IoT malware traffic analysis that used deep learning and Visual Representation to detect and classify new malware (zero-day malware) faster [17], achieving 94.50% accuracy in classification of Trojans, botnets, IoT-based attacks (DDoS, key loggers, OS scans, spyware). LOPEZ-MARTIN [18] proposed a new NTC technique based on deep learning models. By combining a reductive neural network (RNN) with a convolutional neural network (CNN), they can provide better traffic classification detection results, and this method achieved 96% accuracy on the RedIRIS dataset.

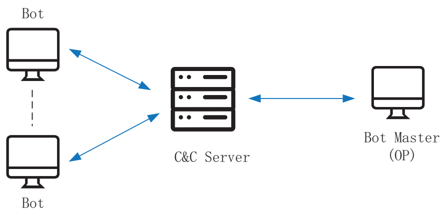

Fig. 1. centralized botnet

B. Discovery and Identification of Traffic with Pluggable Transports

In recent years, as anonymous network technologies such as Tor are increasingly used in botnet construction [4], [10], [27]–[29], traditional traffic-based techniques for identifying botnet command and control channels has been greatly challenged [13]. The literature used machine learning to analyze and fingerprint the timing and frequency of identified Tor network circuit data [30], and created a detection mechanism that can identify infected hosts at the boundaries of the Tor network in real time, while preserving privacy. Experimental data shows 99% accuracy and a very low false alarm rate. However, in recent years, a number of Plugin Transports techniques have been proposed to avoid traffic identification, for example, obfs4, Meek, Snowflake [31], where Meek uses Domain Fronting technology, which exploits the way Google and other Internet content distribution networks (CDNs) route traffic to hide TOR traffic in legitimate HTTPS connections to well-known services that are difficult to detect [11]. It is a difficult problem to detect such traffic obfuscation techniques. For example, in order to detect the traffic of pluggable transports (Obfs3, Obfs4 and ScrambleSuit), SOLEIMANI adopted SVM, AdaBoost, Random Forest and other machine learning algorithms [32], and the method can identify pluggable transports in real time with high determinism by detecting the first 10-50 packets of the ongoing flow. The literature constructs an evaluation framework that is able to detect obfuscation mechanisms including Obfsproxy [33], FTE and Meek as a set of types by building machine learning models with entropy-based features, time-based features and packet header-based features and has a sufficiently low false alarm rate. The mostly related literature is [7], which proposed a hybrid Gaussian Hidden Markov Model to describe the inter-packet time distribution and the packet size density distribution, and a Hidden Markov Model (HMM) to calculate the traffic flow probability of the observed sequence to identify the Meek traffic.

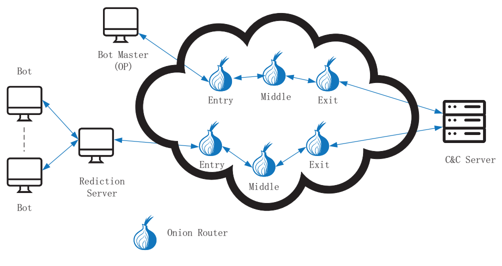

Fig. 2. botnet over tor

In the methods of traditional botnet traffic detection, packets are often detected by TCP and UDP protocols, and botnet traffic is identified by using machine learning and deep learning through extracting features from the packets. When using Tor to transfer C2 communications on C&C server, the features of botnet traffic is hidden by Tor. At the same time, the TLS protocol is used to transfer traffic data instead of using TCP and UDP protocol. Using domain fronting technique can hide the Tor traffic from the general traffic, thus further increasing the difficulty of discovering and identifying botnet traffic.

In this paper, a deep-learning based CNN neural network is proposed for the identification of Meek-based command and control channels in IoT networks, which achieves high-precision identification of obfuscated command and control channels by analysing TLS traffic sequences. In addition, we constructed a large-scale dataset for classification of domain fronting and other traffic which come from the same SDN, this large-scale dataset is more realistic of the constituent parts of the traffic and can help us achieve domain fronting obfuscated C2 communications detection.

III. Approach for Domain fronting Traffic Identification

In this section, we provide a detailed introduce of the proposed approach for domain fronting traffic identification with deep learning algorithms. Deep learning provides a series of powerful machine learning techniques with deep architecture. The deep neural network has a hierarchical architecture composed of multiple layers of complex nonlinear mathematical transformation, exploiting which to perform automatic hierarchical feature extraction and selection. Deep neural networks show excellent feature learning capabilities and can solve a variety of tasks. In this study, we apply a Convolutional Neural Network (CNN) to identify domain fronting traffic.

In our proposed method, we follow related work and classify domain fronting traffic identification as a binary classification problem. That is, we perform supervised classification, where we train a classifier on a set of labeled instances and test the classifier by assigning a label to each unlabeled instance in the test set. In the traditional traffic classification method, the captured traffic trace t needs to be processed into (f_t, c_t) instances to input into the classifier, where f_t is the feature vector extracted from the traffic trace t, and c_t is the corresponding category label. In the domain fronting traffic identification problem, the label c_t belongs to the set $0, 1$ respectively corresponding to whether it is domain fronting traffic or not. Our proposed deep neural network-based classifier integrates feature learning into the training process so that it can classify traffic based only on the initial representation of the traffic trace. Therefore, for a deep learning classifier, the input instance has the form (r_t, c_t), where r_t is the raw representation of the traffic trace t that can be interpreted by the neural network. Therefore, we express the traffic trace as a continuous packet sequence, namely $r_t =< l_1, l_2, , l_n >$, where l_i represents the length of the i-th packet, and the sign is used to indicate the direction of the packet. A positive sign indicates that the packet is sent from the client to the domain fronting server, and vice versa, a negative sign indicates that the packet is sent from the domain fronting server to the client.

We use CNN to build our classifier, which is a feed-forward network trained by backpropagation, designed to perform minimal preprocessing [36]. The main component of CNN is the convolutional layer, which performs linear convolution operations instead of regular matrix multiplication. The research of Schuster et al. Roei Schuster [37], which applys CNN on encrypted video streams, shows that the encrypted stream can be uniquely characterized by its burst mode with high precision. This indicates that CNN could also achieve good performance in domain fronting traffic identification. The structure of CNN can be roughly divided into two parts: feature extraction and classification.

In feature extraction, the input is first fed to the convolutional layer, which consists of a set of filters. The filter is used to convolve with each region of the input data to create a feature map, essentially taking the dot product of two vectors to get a set of values, as shown in the Figure 1. The filter is designed to learn the various parts of the basic feature set. Compared with the fully connected layer, the convolutional layer reduces the number of connections and also reduces the parameters that need to be learned. After the convolutional layer is a non-linear activation function, which converts multiple linear inputs into a non-linear relationship and enhances the representation ability of the neural network. Next, the output of the activation function is fed to the pooling layer. The pooling layer can reduce the space size of the representation from the feature map to reduce the number of parameters and the amount of calculation. We use the max pooling layer, which is the most commonly used method in pooling. Only the maximum value in the spatial neighborhood in a specific area of the feature map is selected as the data representation. Since the largest signal in each neighborhood is retained, max pooling helps the representation to remain unchanged for small changes in the input. For example, even though the order of the packets changes slightly, the neural network can still find areas that contain specific features within the packet sequence. In order to extract more abstract features, we use two sets of convolutional layer and max pooling layer.

Next, CNN inputs the feature maps, which represent the input high-level features, into the classification component. In order to limit the number of learnable parameters, we use two fully-connected hidden layers to flatten and conclude the feature map. In addition, we use dropout to prevent overfitting. Finally, CNN outputs the predicted results in the last layer.

IV. Evaluation

In this section, we introduce our experiments setup and evaluate our approach for domain fronting traffic identification using our datasets.

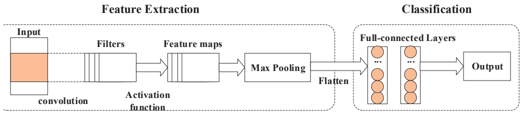

Fig. 3. Basic architecture of convolutional neural network

A. Dataset Collection and Data Processing

We reviewed the current work on botnets and the datasets they provide, for example, GUERRA-MANZANARES offers a dataset MedBIoT [34], they deployed three well-known botnet malware and collected data from botnet infections, propagation and communication with C&C phases (Mirai, BashLite and Torii). They provided the original pcap packages and extracted feature data in the dataset and we expected to use their pcap packages, however the number of these pcap packages was not sufficient for our experimental requirements. On the other hand, KORONIOTIS [35], which presents a new dataset Bot-IoT which contains both legitimate and simulated botnet network traffic, as well as various types of attacks. A realistic testbed environment is also provided to address the shortcomings of existing datasets in obtaining complete network information, accurate labeling, and diversity of recent complex attacks. But they only provide the feature dataset but not the raw data. Based on the consideration above, we decided to collect our own domain front traffic as our dataset.

Between September 2020 and March 2021, we captured traffic under the three ISPs of Educational Network, China Telecom, and China Unicom. Due to lack of real IoT botnet C2 communication traffic, here we generate the traffic dataset by visiting the homepages of the Alexa Top 1k website list through Tor browser (Version 9.0.7) or Firefox browser (Version 68.6.0esr) [38], and use tcpdump to collect the traffic at the client [39]. These traffic are used for classification among Meek traffic and Tor and non-Tor traffic. Because currently most deployed C2 server are Web type server, we think the traffic we collected are representative traffic samples for C2 domain fronting traffic, no matter that they are generated by web surfing or botnet C2 behaviors. When using the Tor browser to access the page, we modify the configuration file torrc to decide whether to use the Meek pluggable transport, which is one of the tools that use domain fronting technology to avoid censorship. The captured traffic is parsed into flow according to the four-tuple (srcIP, srcPort, dstIP, dstPort). We use tshark (version 1.12.1) to parse each flow [40], extract the length sequence of the TLS application data packet, and use the sign to indicate the direction of the packet as mentioned above. The length sequence is used as the input of the classifier. In total, we evaluate our deep learning approach on two different datasets:

1) E1500: We visited the homepages of Alexa Top 1K websites in the Educational Network environment and collected traffic. We use Tor browser to collect Tor-related traffic, and Firefox browser to collect non-Tor traffic. We used three torrc configurations: direct connection without Meek (tor-direct), Meek that relies on the CDN provided by Microsoft Azure (meek-azure), and Meek that uses CDN provided by Fastly (meek-fastly), which are two typical CDN providers for Meek service hosting. After analyzing the traffic into flows and extracting the length sequence according to the steps mentioned above, we got 1500 of each of these four types of traffic. Details are shown in Table 1.

TABLE I
Details of Dataset E1500

Type	Amount	Label
meek-azure	1500	0
meek-fastly	1500	0
tor-direct	1500	1
non-tor	1500	1

2) TU4000: We have used Tor browser and Firefox browser to visit the Alexa Top 1K website several times under the environment of China Unicom and China Telecom. The difference between TU4000 and E1500 is that the scale of TU4000 is larger, with each category has 4000 traces of data, the ISPs in TU4000 are diverse, and the most difference is that the data traces in TU4000 have the same SNI: we additionally screened the data based on Server Name Indication (SNI) value—ajax.aspnetcdn.com, which is a trick meek used that when it talks to a Tor relay,it will appears to talk to Microsoft Azure SDN. The purpose of this is that we want to finely identify the Meek traffic in a dataset mixed with other non-Meek traffic (including Tor traffic and non-tor traffic) which comes from the same CDN, for a better way to revealing obfuscated C2 communications in a cluttered network environment. Details are shown in Table 2.

TABLE II
Details of Dataset TU4000

Type	ISP	Amount	Label
meek-azure	China Telecom	4000	0
non-meek	China Telecom	4000	1
meek-azure	China Unicom	4000	0
non-meek	China Unicom	4000	1

For both datasets, we pad and truncate sequence into 500 length. For training, we use categorical cross-entropy as loss function and Adam [41] as optimizer, with 0.001 learning rate. We implement our model with Keras [42], which using Tensorflow as backend [12]. We practice deep learning experiments on a platform that has two Nvidia Tesla P100 GPUs with 24 GB GPU memory, which can make high parallelism performance of CNNs due to cuDNN primitives [43]. We use the default parameter settings and perform each experiment three times and present the results as an average.

B. Experiments Results and Analysis

In this section, we evaluate the proposed model, present the final results and analyze the results.

1) The results of dataset E1500:

We intercept different lengths of the sequence data, and train our CNN classifier on the E1500 dataset for experiments. We use 70% of the data as the training set and 30% as the test set. We use four commonly used metrics for binary classification problems to evaluate our model, namely: precision, recall, F1-value, and accuracy. We record the number of true positives in the classification result as TP, the number of false positives as FP, the number of true negatives as TN, and the number of false negatives as FN. The calculation methods of the four metrics are as follows:

$$\text{precision } = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{1}$$

$$\text{recall } = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{2}$$

$$\mathrm{F1} = \frac{\text{precision } * \text{recall} * 2}{\text{precision } + \text{recall}} \tag{3}$$

$$\text{accuracy } = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{4}$$

The experimental results are shown in Table 3:

TABLE III
Experiment results on E1500

sequence length	acc	precision	recall	F1
20	99.33%	98.86%	99.81%	99.34%
50	99.58%	99.29%	99.90%	99.59%
100	99.69%	99.48%	99.90%	99.69%
200	98.90%	99.13%	98.66%	98.90%
500	98.88%	98.58%	99.19%	98.88%

It can be seen from the experimental results that our CNN model has a very good identification effect on domain fronting traffic, no matter which CDN service provider is, with an accuracy rate of about 99%. When the intercepted sequence length is 100, the precision rate is 99.48%, the recall rate is 99.9%, and the classification effect is the best. If the interception length is too short, the classifier cannot learn the features of the following sequence. If the interception length is too long, important features may be "diluted", which will affect the performance of the classifier.

2) The results of dataset TU4000:

Further more, we prefer to demonstrate whether our model can identify domain fronting traffic over other non-meek traffic (including Tor traffic and non-tor traffic) comes from the same CDN, we use TU4000 to conduct experiments. Based on the experimental results of the E1500 dataset, we cut the length of each flow sequence to 100, and add 0 to the back if the length is less than 100. Then input the data to the classifier for training. We use accuracy as the metric to evaluate our CNN model. The experimental results are shown in Table 4:

TABLE IV
Experiment results on TU4000

ISP	accuracy
China Unicom	99.96%
China Telecom	96.60%
China Unicom & China Telecom	97.35%

The results show that even in a different ISP environment, our CNN model can also perform well at classifying domain fronting traffic and non-domain-fronting traffic with the same SNI which provided by Microsoft Azure. The experimental results also show that it is feasible to train a classifier that is universally applicable to various ISPs.

V. Discussion

In this section, we enumerate the limitations of this work, discuss the feasibility of the dataset we proposed and some of the outstanding challenges.

Nowadays, C2 communication in botnets are typically hidden by using Tor and domain fronting technique to resist traditional methods of botnet traffic detection. However, the C2 server is essentially a web type service, we consider that an ordinary website could also become a C2 server for a botnet, which stores malicious files and malicious code as the website's files, besides that a legitimate website can become as a C2 server compromised by hackers. We admit that ordinary websites are less correlated with C2 servers in botnets. Our aim is to propose a method to detect domain fronting traffic and thus improve the detection of C2 traffic hidden in domain fronting technology. Based on the above considerations, we captured the traffic data with our method, and constructed the corresponding dataset.

The mostly related work comparing with us is [7]. As there is no corresponding code in YAO Z's approach for reference, and no specific dataset for comparison, we can only compare qualitatively with it. The size of our dataset is larger than YAO Z's, including the fact that our background set uses non-Tor traffic as well as Tor traffic. Also, our dataset contains two types of meek traffic including meek-azure and meek-fastly. Secondly, the approach proposed by YAO Z requires more processing, the steps of their feature extraction and classification methods are complex. Our method don't need manual feature extraction, only needs to parse the packets and extract the TLS sequences, and then put the TLS sequences into model. We believe that our approach is more feasible as well as closer to real-life scenarios. In terms of results, YAO Z's proposed method has 99.98% accuracy in identifying

Meek and ordinary traffic; our method's accuracy is 0.3% lower than theirs, but we have put Tor traffic into our background set, increasing the variaties of background set. So we consider this result acceptable. In addition, considering that hackers often use different ISPs to operate C2 servers to transmit control commands, we perform the identification of domain fronting traffic from different ISPs. From the results, with the same SNI value, our model is able to distinguish well between domain fronting traffic and other traffic in different ISP scenarios.

VI. Conclusions

C2 communication in botnet often uses Tor domain fronting techniques to obfuscate traffic, which is hard to classify between domain fronting traffic and Tor traffic. Therefore, in this paper, we propose a deep learning-based method for identifying domain-front traffic, which focuses on Meek, the most commonly used domain fronting technology in Tor networks. Our method using TLS sequences in domain-front traffic as features and identifying them through CNN networks. Firstly, we build a dataset E1500 to test the ability of identify traffic generated by domain fronting technology among a mixed traffic, and secondly, we build another dataset TU4000 to classify large-scale traffic generated by the same SNI to classify domain-fronting traffic and general Tor traffic. Experiments show that our CNN model can identify the meek traffic among mixed traffic of normal non-tor traffic and Tor traffic. And our approach performs well in classifying the domain fronting traffic with the same SNI traffic.

Considering the ability of C2 communication traffic can be hidden by other technologies besides domain fronting, the more advanced C2 architectures will be proposed. We will focus on other anonymous communication technologies , try to reveal the obfuscated C2 communication traffic and find the way to discovery C2 servers. This will be our future work.

References

[1] Securing the Internet of Things. https://www.dhs.gov/securingtheIoT.

[2] The Internet of Things (IoT) – essential IoT business guide. https://www.i-scoop.eu/internet-of-things-guide/.

[3] BAUER H, BURKACKY O, KNOCHENHAUER C. ”Security in the Internet of Things.” 2017.

[4] ANAGNOSTOPOULOS M, KAMBOURAKIS G, DRAKATOS P, et al. ”Botnet Command and Control Architectures Revisited: Tor Hidden Services and Fluxing,” Cham, F, 2017 [C]. Springer International Publishing.

[5] SU S-C, CHEN Y-R, TSAI S-C, et al. ”Detecting P2P Botnet in Software Defined Networks.” Security and Communication Networks, 2018(4723862 2018.

[6] DINGLEDINE R, MATHEWSON N, SYVERSON P: Naval Research Lab Washington DC, 2004.

[7] YAO Z, GE J, WU Y, et al. Meek-based Tor Traffic Identification with Hidden Markov Model [M]. 2018.

[8] LIN Z, TONG L, ZHIJIE M, et al. ”Research on Cyber Crime Threats and Countermeasures about Tor Anonymous Network Based on Meek Confusion Plug-in,” proceedings of the 2017 International Conference on Robots & Intelligent System (ICRIS), F 15-16 Oct. 2017, 2017 [C].

[9] XIE H, WANG L, YIN S, et al. ”Adaptive Meek Technology for Anti-Traffic Analysis,” proceedings of the 2020 International Conference on Networking and Network Applications (NaNA), F 10-13 Dec. 2020, 2020 [C].

[10] BROWN D J D C. ”Resilient botnet command and control with tor.” 18(105 2010.

[11] Threat Research: APT29 Domain Fronting With TOR. https://www.fireeye.com/blog/threat-research/2017/03/apt29_domain_frontin.html.

[12] https://blog.netlab.360.com/new_threat_zhtrap_botnet_cn/.

[13] GU G, ZHANG J, LEE W. ”BotSniffer: Detecting botnet command and control channels in network traffic.” 2008.

[14] SHAFIQ M, TIAN Z, BASHIR A K, et al. ”IoT malicious traffic identification using wrapper-based feature selection mechanisms.” Computers & Security, 94(101863 2020.

[15] SHAFIQ M, TIAN Z, BASHIR A K, et al. ”CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques.” Ieee Internet of Things Journal, 8(5): 3242-54 2021.

[16] EGEA S, REGO MANEZ A, CARRO B, et al. ”Intelligent IoT Traffic Classification Using Novel Search Strategy for Fast-Based-Correlation Feature Selection in Industrial Environments.” Ieee Internet of Things Journal, 5(3): 1616-24 2018.

[17] BENDIAB G, SHIAELES S, ALRUBAN A, et al. IoT Malware Network Traffic Classification using Visual Representation and Deep Learning [M]. 2020.

[18] LOPEZ-MARTIN M, CARRO B, SANCHEZ-ESGUEVILLAS A, et al. ”Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things.” Ieee Access, 5(18042-50 2017.

[19] ALQAHTANI M, MATHKOUR H, BEN ISMAIL M M. ”IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection.” Sensors, 20(21): 2020.

[20] YIN L, LUO X, ZHU C, et al. ”ConnSpoiler: Disrupting C&C Communication of IoT-Based Botnet Through Fast Detection of Anomalous Domain Queries.” Ieee Transactions on Industrial Informatics, 16(2): 1373-84 2020.

[21] HASHIMOTO N, OZAWA S, BAN T, et al. A Darknet Traffic Analysis for IoT Malwares Using Association Rule Learning [M]//OZAWA S, TAN A H, ANGELOV P P, et al. Inns Conference on Big Data and Deep Learning. 2018: 118-23.

[22] KIM J, SHIM M, HONG S, et al. ”Intelligent Detection of IoT Botnets Using Machine Learning and Deep Learning.” Applied Sciences-Basel, 10(19): 2020.

[23] MEIDAN Y, BOHADANA M, MATHOV Y, et al. ”N-BaIoT-Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders.” Ieee Pervasive Computing, 17(3): 12-22 2018.

[24] HAFEEZ I, DING A Y, ANTIKAINEN M, et al. ”Real-Time IoT Device Activity Detection in Edge Networks,” Cham, F, 2018 [C]. Springer International Publishing.

[25] ALOTAIBI B, ALOTAIBI M. ”A Stacked Deep Learning Approach for IoT Cyberattack Detection.” Journal of Sensors, 2020(8828591 2020.

[26] ALOTHMAN Z, ALKASASSBEH M, AL-HAJ BADDAR S. ”An efficient approach to detect IoT botnet attacks using machine learning.” Journal of High Speed Networks, 26(241-54 2020.

[27] CASENOVE M, MIRAGLIA A. ”Botnet over Tor: The illusion of hiding,” proceedings of the 2014 6th International Conference On Cyber Conflict (CyCon 2014), F 3-6 June 2014, 2014 [C].

[28] KANG L. ”Efficient botnet herding within the Tor network.” Journal of Computer Virology and Hacking Techniques, 11(1): 19-26 2015.

[29] SANATINIA A, NOUBIR G. ”OnionBots: Subverting Privacy Infrastructure for Cyber Attacks,” proceedings of the 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, F 22-25 June 2015, 2015 [C].

[30] FAJANA O, OWENSON G, COCEA M, et al. TorBot Stalker: Detecting Tor Botnets through Intelligent Circuit Data Analysis [M]. 2018.

[31] CIRCUMVENTION. https://tb-manual.torproject.org/circumvention/.

[32] SOLEIMANI M H M, MANSOORIZADEH M, NASSIRI M. ”Real-time identification of three Tor pluggable transports

using machine learning techniques." Journal of Supercomputing, 74(10): 4910-27 2018.

[33] WANG L, DYER K P, AKELLA A, et al. Seeing through Network-Protocol Obfuscation [M]. 2015.

[34] GUERRA-MANZANARES A, MEDINA-GALINDO J, BAHSI H, et al. MedBIoT: Generation of an IoT Botnet Dataset in a Medium-sized IoT Network [M]. 2020.

[35] 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." Future Generation Computer Systems-the International Journal of Escience, 100(779-96 2019.

[36] LECUN Y, BENGIO Y. Convolutional networks for images, speech, and time series [M]. The handbook of brain theory and neural networks. MIT Press. 1998: 255–8.

[37] SCHUSTER R, SHMATIKOV V, TROMER E, et al. Beauty and the Burst: Remote Identification of Encrypted Video Streams [M]. 2017.

[38] Alexa the web information company. http://alexa.com.

[39] tcpdump. http://www.tcpdump.org/.

[40] tshark. https://www.wireshark.org/docs/man-pages/tshark.html.

[41] KINGMA D, BA J J C S. "Adam: A Method for Stochastic Optimization." 2014.

[42] Keras, 2015. https://keras.io.

[43] CHETLUR S, WOOLLEY C, VANDERMERSCH P, et al. "cuDNN: Efficient Primitives for Deep Learning." 2014.

[44] What is C2? Command and Control Infrastructure Explained. https://www.varonis.com/blog/what-is-c2/