Network-based Anomaly Detection against IoT Cyberattacks using CNN-LSTM Autoencoder

Ruoyu Li	Ce Wang	Pingdong Wang
2019312695	2019214617	2019214616

Abstract

With the dramatically increasing number of insecure IoT devices deployed in the network, new cyberattack surfaces arise by using IoT botnet. Conventional signature-based intrusion detection system or supervised learning based detection system is not as effective as before since the data of unknown attacks cannot always be included in the database in time. Also, traditional machine learning method that requires highly hand-crafted feature engineering is time-consuming. Our work presents a novel anomaly detection system with a CNN-LSTM autoencoder that directly digests raw bytes from network packets. It only learns from normal data so there is no need to collect a huge amount of malicious data. With an empirical experiment, this report demonstrates that our method is effective to detect unknown malicious network-based behaviors.

1. Introduction

The Internet of Things (IoT) devices are increasingly found in every daily life of the people from different fields of work, such as the TVs, the intelligent sweepers, the intelligent lamps and so on. Despite the increasing conveniences that the IoT brings, there exist more and more risks about the Internet for all the IoT devices could communicate the information about their users to other parties over the Internet. Among these risks, the DDoS attack reaches unprecedented levels, the need of timely detection of IoT botnet attacks is becoming imperative for mitigating the disruptions associated with these attacks. In 2016, an IoT botnet Mirai controlled over 100,000 IoT to conduct one of the greatest DDoS attacks. Therefore, it is necessary to find a detection method of IoT cyberattacks for mitigating risks and propagation. Furthermore, new types of attacks are continuously arising, thus a signature-based intrusion detection or a supervised learning method that requires data from all known categories of attacks is not practical as before.

2. Related Work

Many detection algorithms were surveyed in [1], however, autoencoders were not used at all. What's more, there are few works to do related to the IoT. Although autoencoders were extended for outlier detection in [2], they still required security analysts to actively label data for subsequent supervised learning. Similarly, the writers of [3] apply deep learning to system logs for detecting insider threats. Differently, what they use are DNNs and RNNs , and depend on further manual inspection. All in all, our method differs from previous studies because we learn from benign data by training deep autoencoders for each device, and use them as standalone automatic tools for instantaneous detection of IoT botnet attacks.

Our motivation is, given a large number of heterogeneous IoT devices connected to an organizational network, it is imperative to devise an automated method that is unsupervised, efficient and accurate in detecting compromised IoT devices which are being attacked or used as botnet to launch attacks against other entities. In our work, we will implement a system and test it on the real IoT traffic data collected by another research team.

3. Implementation

3.1 System Overview

Basically, our system consists of three components: capturer, handler and detector, as depicted in Fig.1. The input of our system is either a collection of pcap files, which is a file format for captured network traffic packets, or a network interface for real-time network traffic capture. A network packet is a formatted unit of data carried by a packet-switched network, which consists of control information and user data. Typically, a network packet can be divided into five layers, from bottom to top: physical layer, link layer, network layer, transport layer and application layer. In our work, we only care about the data above the network layer, since the network layer contains the IP address that is an identifier of a certain device in the network, and transport layers and application layer reveal the data transmission behavior of a device. Our system can be deployed on a switch where all network traffic will pass through or a PC which can monitor the network behavior of home-based IoT devices. We will explain each component in detail in the following sections.

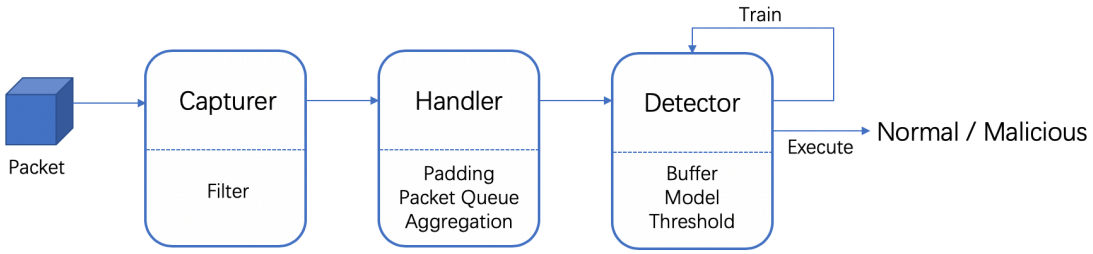

Fig.1 System overview

3.2 Capturer

When a packet from a pcap file is read or from a network interface arrives, the capturer will only capture those packets that come from the IP addresses of certain IoT devices in the network and filter out others. The motivation of doing this step is to reduce the processing overhead. Once a device is compromised by an attacker, the traffic outcoming from the IP address of this device is supposed to reveal abnormality. After filtering, the capturer will hand over the packet to the handler for further processing.

3.3 Handler

The handler is responsible for crafting a sequence of received packets into a sequential data sample that can be digested by an one-dimensional CNN and LSTM model. The handler treats each raw byte in a packet as a feature. It will truncate or pad zeros to the end of packet bytes into a fixed length, denoted as PACKET_LEN, and normalize the bytes so that it transforms a network packet into a feature vector with a length of PACKET_LEN. Then, a Fist-In-First-Out queue is used to store the feature vector from the same IP address and aggregate them into a sequential data sample. As soon as the length of the queue is equal to a certain length, denoted as SEQ_LEN, the handler will emit a sequential data sample with a length of SEQ_LEN to the detector specified for this IP address, and the queue will be updated by popping out the first-in feature vector and waits for the next packet.

3.4 Detector

The detector is a network-based anomaly detection approach by a CNN-LSTM autoencoder that attempts to compress behavioral snapshots of benign IoT traffic. Each device has its own detector, that is, a detector only train and execute on the traffic data from specific IP address. It is because different types of devices might exhibit remarkably different behaviors so that we cannot simply include all of them in one

detector. A detector has a buffer to store the data samples up to a size of MINI_BATCH. Once the buffer is filled up, the training process of the autoencoder will start. In the model, one-dimensional CNN is used to learn high-dimensional features from each sequential data sample and LSTM is used to capture the sequence relevance. The model architecture is illustrated as Fig.2. The output of an autoencoder is the reconstruction of the input data and the training process will try to minimize the loss between the input and output so that the model can learn the snapshot of benign data. By using an autoencoder architecture, all we use to train is the normal data from benign IoT devices so we do not need to label any data, or collect data from all kinds of known cyberattacks.

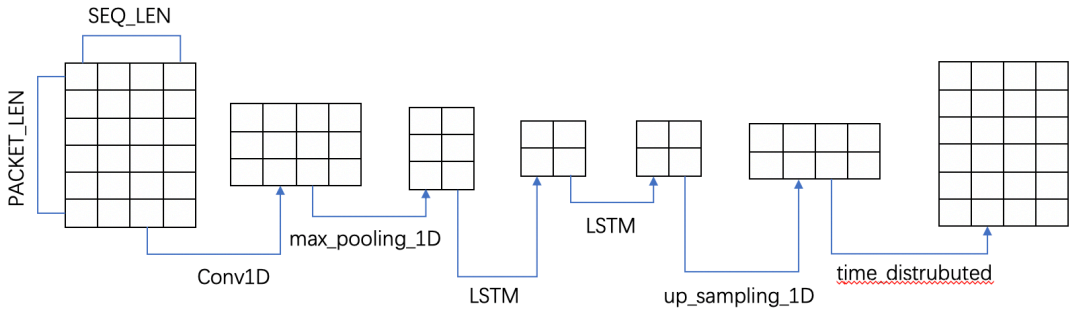

Fig.2 Model architecture

After the training process and when the detector is executing on testing data, a threshold of mean squared error between the input and the output of the autoencoder is decided to differentiate the normal and malicious behavior. We assume that the mean squared errors among benign data is subject to a normal distribution, so we use 2-sigma rule to generate an interval of threshold. If an error is out of this interval, the detector will consider it as abnormality and log some information of the malicious behavior, such as the destination IP address, port, payload, etc.

4. Evaluation

4.1 Dataset

The IoT traffic dataset was asked from the research team of an IMC 2019 paper: "Information Exposure From Consumer IoT Devices: A Multidimensional, Network – Informed Measurement Approach" [4]. This data set is compromised of real network traffic data when various interactions with different IoT devices are conducted in the two labs, one located in US and another in UK. Another dataset that we used is the Mirai traffic provided by the team of paper "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection" [5]. The dataset consists of full infection

stages of Mirai, including scanning, command & control, malware loading and launching attacks.

4.2 Environment

The experiment was conducted on a server with 24-core Intel® Xeon® CPU E5-2630 v2 2.60GHZ, and NVIDIA GeForce RTX 208 GPU. We use Keras backended with Tensorflow 2.0 for model implementation. Scapy is used as a tool for pcap file parsing and network packet capturing.

4.3 Metrics

In this experiment, the data from the US lab are used to train the model as training samples and the data from the UK lab are used to select the proper hyperparameters of PACKET_LEN, SEQ_LEN, MINI_BATCH and epochs. By running a grid search, we set the hyperparameters as follows: PACKET_LEN:=1500, SEQ_LEN:=6, MINI_BATCH:=128, epochs:=10. The data from the UK lab are also used to evaluate the model training by the former one. Moreover, the real Mirai botnet and DDoS attack traffic will be used to test the detection capability of the system.

We use 4 IP-enabled IoT devices in the dataset for the experiment: a TP-Link plug, a Google Home Mini, a Xiaomi cleaner and a Blink camera. The testing dataset is generated by the combination of benign traffic from UK's lab and malicious traffic from bot IoT device infected by Mirai. The detector of each device is supposed to do a binary classification to predict whether the ongoing traffic is benign or malicious.

As Table.1 shows, all of the four devices demonstrate a good performance. The metrics suggest that our system is capable of detecting malicious traffic when a device is controlled or infected by the malware, while holding a very low level of misidentifying some benign traffic as abnormality.

Table.1 Metrics of four devices for binary classification

Device	TNR	FPR	TPR	FNR	Precision	Recall	F1-score
TP-Link plug	0.966	0.0345	1.00	0.00	0.967	1.00	0.983
Google Home Mini	0.923	0.0765	1.00	0.00	0.929	1.00	0.963
Xiaomi cleaner	1.00	0.00	0.999	0.000814	1.00	0.999	0.9996
Blink camera	0.992	0.00822	1.00	0.00	0.992	1.00	0.996

Average	0.970	0.0298	0.9998	0.000203	0.972	0.9998	0.985

5. Conclusion

In this project, we proposed a CNN-LSTM autoencoder to learn from normal traffic data among IoT devices and to detect the abnormal behaviors and predict whether they are compromised by attackers. We evaluate our system by both real IoT network traffic data and Mirai botnet traffic data. The result shows that our system is able to learn the normal behaviors well from different IoT devices with various use and interactions, and precisely detect abnormal traffic generated by infected devices.

6. References

[1] García, S., A. Zunino, and M. Campo, Survey on network-based botnet detection methods. Security and Communication Networks, 2014. 7(5): p. 878-903.

[2] Veeramachaneni, K., et al. AI^ 2: training a big data machine to defend. in 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS). 2016. IEEE.

[3] Tuor, A., et al. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. in Workshops at the Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[4] Ren, J., et al. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. in Proceedings of the Internet Measurement Conference. 2019. ACM.

[5] Mirsky, Y., et al., Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089, 2018.