Poster: DBDS: A Botnet Detection System Based on Deep Learning

Di Wu[1,2], Binxing Fang[3,4,5], Xiang Cui[3,1]
1 (Institute of Information Engineering, Chinese Academy of Sciences)
2 (School of Cyber Security, University of Chinese Academy of Sciences)
3(Cyberspace Institute of Advanced Technology, Guangzhou University)
4 (Institute of Electronic and Information Engineering in Dongguan UESTC)
5(School of Cyberspace Security, Beijing University of Posts and Telecommunications)
wudi6@iie.ac.cn

***Abstract*—Botnets represent one of the most serious cyber security threats today, which have been used as main vectors to launch large-scale cyber attacks, such as DDoS and spam. Machine learning has wide application in botnet detection, but along with the changes of the botnet control mechanism, selecting features manually becomes increasingly difficult. Attackers can utilize adversarial machine learning strategy to bypass the detection facilities. To change this situation, we studied the feasibility of using deep learning technology to automatically extract traffic features. In this paper, we propose a novel detection approach called the deep botnet detection system (DBDS). The system studies features from temporal and spatial dimension, and establish classifier model by combining deep convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) structure. DBDS does not depend on any prior knowledge which about the protocol and topology, and works without manually selecting features. The experiment results show that the proposed model has good performance in botnet detection.**

I. Introduction

Machine learning is widely used in botnet detection, especially in intrusion detection. Researchers distinguish malicious traffic by utilizing classification algorithm (e.g., SVM [1]) and clustering algorithm (e.g., X-means [2]) to build models. These detection systems generally show a high rate of accuracy in experiments, but a majority of them face the same problem: the models depend on the set of features which should be selected manually.

The traffic features are usually specified by researchers in accordance with the experience, and common ones include the traffic properties (e.g., the number of packets per flow and the average number of bytes per packets), time properties (e.g., the intervals of two adjacent flows) and behavior properties (e.g., whether the hosts access the same server). The outstanding features could effectively improve the model performance, but they also have some disadvantages. On the one hand, selecting appropriate features has a high requirement to the prior knowledge of the designer. On the other hand, the constant features may also be utilized by attackers. Attackers can utilize adversarial machine learning strategy to bypass the detection system by purposefully changing the correlation attributes of the botnet. Cui et al. [3] point out that the botmaster can eliminate the space-time similarities by injecting flow-level noise and random delay into the botnet traffic.

Deep learning is already known as a technique supporting feature learning based on multiple neural networks, which greatly applied in pattern recognition. Therefore, in order to solve the difficulty in feature selection, we discuss the issue about the botnet detection system based on deep learning, and make two main contributions. First, we propose a novel detection system DBDS. The system combines the process of feature extracting and training, and identifies malicious traffic from a global perspective. Second, we present two deep neural network constructions, which use CNN and LSTM to automatically study features from temporal and spatial dimension, respectively.

II. System Design

The purpose of DBDS is training a classification model through the features extracted from network traffic. The structure of DBDS consists of two main components, shown in Figure 1: the spatial feature learning module and the temporal feature learning module.

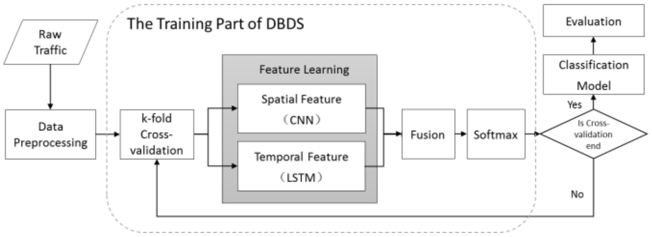

Figure 1. The process of DBDS

The raw traffic files are in pcap format and composed of multiple packets. However, the datasets trained for DBDS are flows. Therefore, it is necessary to preprocess the data before training. The pkt2flow tool has been used to turn the pcap file into flows arranged in a time sequence, and the packets in each flow have the same 5-tuple (i.e., the source IP & port, the destination IP & port, the protocol). We use k-fold cross-validation to improve the generalization ability of the model and assess the performance more accurately. And the fusion process is concatenating the two features.

A. The Spatial Feature Learning Module

CNN is used to extract the spatial features and the module process is shown in Figure 2. Before training, each flow should be trimmed to 1024 bytes, and if it is not long enough, the 0x00 will be added in the end. Then the results are converted to grey images of size of 32*32. Generally, the previous bytes in a flow will contain main connection information and a handful of data of content exchange, so they can better reflect the characteristics of the flow.

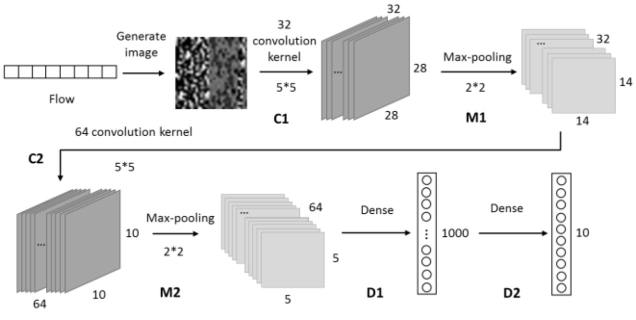

Figure 2. The structure of CNN

The structure of CNN contains 7 layers:

Convolution layer C1: C1 has 32 kernels of size of 5*5, and the outputs are 32 feature maps of size of 28*28.

Max-pooling layer M1: M1 has a 2*2 max-pooling operation, the outputs are 32 feature maps of size of 14*14.

Convolution layer C2: C2 has 64 kernels of size of 5*5, and the outputs are 64 feature maps of size of 10*10.

Max-pooling layer M2: M2 is same as M1, and the outputs are 64 feature maps of size of 5*5.

Dense layer D1 & D2: D1 and D2 have 1000 and 10 neurons, respectively. The output of D2 is a 10-dimension spatial feature vector.

B. The Temporal Feature Learning Moduel

CNN can only deal with the spatial features, but cannot be used to extract the dependency relationships among chain structures. Thus, LSTM is used to learning the temporal features of traffic flows, shown in Figure 3. The model use a two-layer Bi-directional LSTM structure, and first turn the sequence of bytes in each packet into a vector, and the results are input in the second LSTM layer which outputs the temporal feature. Before training, each flow should be trimmed to 8 packets, and each packet should be trimmed to 100 bytes. The structure of LSTM contains 5 layers:

One-Hot Encoding layer O1: O1 utilizes One-Hot encoding convert each byte in the packet to a 256-dimension vector, and each packet become a 100*256 sparse matrix.

LSTM layer L1: L1 has 100 LSTM units, the outputs are 100 256-dimension vectors.

Dense layer D1: D1 has 256 neurons.

LSTM layer L2: L2 has 8 LSTM units, the outputs are 8 256-dimension vectors.

Dense layer D2: D2 has 10 neurons, the output of D2 is a 10-dimension temporal feature vector.

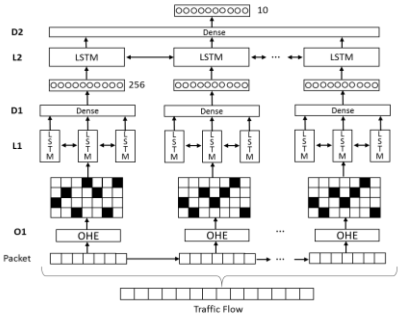

Figure 3. The structure of LSTM

III. Preliminary Results

We use softmax classifier to determine whether the input traffic is normal or botnet. Figure 4 shows the model performance which training by 5000 samples. In it, model A only uses CNN and model B only uses LSTM. And we can see, DBDS has good accuracy and low false positive rate.

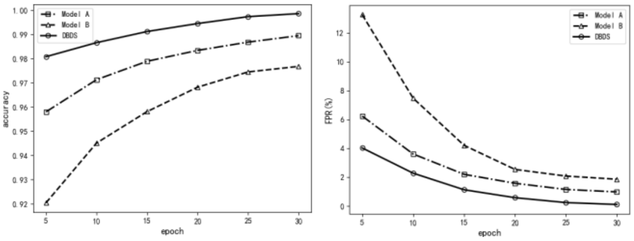

Figure 4. The performance of DBDS

IV. Conclusion

In this paper, we propose a botnet detection system based on deep learning, called DBDS. DBDS automatically studies features from spatial and temporal dimension, and characterizes the traffic globally. The preliminary results show that DBDS has good performance in botnet detection.

Acknowledgment

This work is supported by DongGuan Innovative Research Team Program(201636000100038), the National Key R&D Program of China (No. 2016QY08D1602), the Key Laboratory of Network Assessment Technology at Chinese Academy of Sciences and Beijing Key Laboratory of Network security and Protection Technology.

References

[1] Kondo S, Sato N. Botnet traffic detection techniques by C&C session classification using SVM[C]//International Workshop on Security. Springer, Berlin, Heidelberg, 2007: 91-104.

[2] Gu G, Perdisci R, Zhang J, et al. BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection[C]//USENIX security symposium. 2008, 5(2): 139-154.

[3] Xiang C, Binxing F, Lihua Y, et al. Andbot: towards advanced mobile botnets[C]//Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats. USENIX Association, 2011: 11-11.