Abnormal Traffic Detection of IoT Terminals Based on Bloom Filter

Fengjie DENG, Yubo SONG, Aiqun HU, Min FAN, Yu JIANG

School of Cyber Science and Engineering Southeast University

Jiangsu, Nanjing, China

fjdeng17@gmail.com, {songyubo, aqhu, jiangyu, seu_fm }@seu.edu.cn

ABSTRACT

As the size and speed of the network increase, the discovery of abnormal traffic becomes more difficult. It is not only necessary to accurately detect real-time traffic but also to determine the type of abnormality. Therefore, in view of the requirement for network anomaly discovery, this paper proposes a Bloom Filter (BF) based abnormal traffic detection framework. This framework could retrieve information from real-time data accurately under low time complexity. This article mainly analyzes two kinds of abnormal traffic (port scanning traffic and TCP flooding traffic). For port scanning traffic, with BF structure the framework could retrieve what ports this stream has accessed. If there is too much traffic on different ports, an abnormality could be determined. For the TCP flooding traffic, the Count Bloom Filter (CBF) is used to count the number of packets with similar length in each type of stream for a period of time. If a higher proportion of packets with similar length has been detected, an abnormality has a strong probability. Finally, the paper analyzes the proposed abnormal traffic detection framework in the real environment. The experiment finds that there is less false positive for normal traffic and it can correctly identify the above two abnormal traffic.

CCS CONCEPTS

• Insert CCS text here • Insert CCS text here • Insert CCS text here

KEYWORDS

Bloom Filter, Abnormal Detection, IoT

ACM Reference format:

Fengjie Deng, Yubo Song, Aiqun Hu, Min Fan and Yu Jiang. 2018. Abnormal Traffic Detection of IoT terminals Based on Bloom Filter. In TURC-AIS 2019: ACM TURC Conference on Artificial Intelligence and Security, May 17–19, 2019, Chengdu, China. ACM, New York, NY ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3321408.3326654

ACM TURC 2019, May 17–19, 2019, Chengdu, China

ACM ISBN 978-1-4503-7158-2/19/05…$15.00.
https://doi.org/10.1145/3321408.3326654

1 Introduction

The Internet of Things (IoT) is one of the most popular researches in the world, and is known as the third information technology revolution after computers and the Internet[1]. The main idea of the IoT is to embed communication protocols into everyday objects so that any item can be connected to the Internet. Therefore, IoT will not only change the way people and things communicate, but also change the way things interact with things[27,28]. Therefore, this technology has a broad application space in the future.

With the gradual development of the IoT, it has gradually been applied in the fields of industry and medical industry[2,24]. For example, in the industrial world, the sensor can monitor the production process in real time, and test and evaluate the product uploading the data to the management for reference. Therefore, the development of IoT not only enhances the economic benefits of the industry, but more importantly, changes the way the entire industry operates, making it more modern and integrated.[23]

However, there have been many security risks in the development of the Internet of Things[3,25]. For example, on October 21, 2016, the United States suffered the largest Internet attack in history, and a large number of Internet users were disconnected. The culprit is that hackers exploited the security vulnerabilities of various IoT devices, used a software named Mirai to invade a large number of cameras and video recorders, and used these bots to conduct DDoS attacks on domain name service manager[4]. In mid-2017, a similar DDoS attack occurred in China, mainly because a virus named http81 controlled about 50,000 IoT devices.

Therefore, from the above analysis it can be seen that the rapid development of the IoT requires attention to potential security risks, especially abnormal traffic in IoT devices[21]. The network abnormal traffic refers to the excessive deviation between the current traffic and the normal network traffic, which affects the network performance and causes network congestion[5]. In severe cases, the network is interrupted. At present, there are many reasons for abnormal network traffic in the Internet, including Dos attack, port scanning, and worm[6,22]. In the current IoT environment, network traffic data arrives at high speed exhibiting massive data characteristics, and cannot be stored locally. However, abnormal

traffic needs to be detected and discovered, so that abnormal traffic processing can be performed.[26]

2 Related work

This chapter will firstly introduce several common abnormal traffic detection methods, and analyze the main influencing factors of each method, and then compare these methods and analyze their advantages and disadvantages.

2.1 Statistical analysis

Analytical methods based on statistical analysis are a developed detection method and the most widely used detection method[7]. The basic principle of statistical analysis is that the normal flow rate is regular and stable in the long run, but the abnormal flow varies greatly in certain characteristics and is so different from normal flow. Therefore, according to the statistical characteristics, an abnormal flow detection mechanism can be developed. If the difference between the observed features and the statistical laws is large, the abnormality can be inferred. Denning[8] proposed an anomaly detection model based on variance, Markkov process, multivariate model and time series model. Cheng Guang[9] proposed an anomaly detection model for high-speed networks based on the rule of message length.

The implementation of such methods is relatively simple and the anomaly identification is accurate. Additionally, the judgment threshold can be continuously adjusted with time. However, it has a major shortcoming. In the current network environment, there are many types of traffic, so it is necessary to select appropriate statistical characteristics. Otherwise there will be a great probability of false positives and false negatives.

2.2 Signal processing

The signal processing-based analysis method mainly analyzes the flow by performing corresponding synthesis or decomposition on the received signal. The most mature method is the time series analysis method[10]. Time series analysis method is mainly to uniformly sample specific statistical objects to form time series, and analyze them through classical time series models such as AR, ARMA, ARMIA and EWMA, or other time series analysis methods[11]. At the same time, the wavelet analysis method has also been widely quoted[12]. Its main idea is to perform time-frequency domain decomposition on the network traffic signal by wavelet analysis, so as to highlight the local characteristics of the received signal, so frequency changes on abnormal traffic can be detected.

Compared with statistical analysis methods, signal processing methods do not need to find suitable statistics for monitoring, so such methods are more versatile and combined with time domain and frequency domain features greatly improve the accuracy of abnormal traffic detection[13]. However, current network traffic changes and complexity are high, it is difficult to accurately predict through a certain time series and the computational complexity of time series is high and it is difficult to guarantee real-time performance for high-speed networks. For signal analysis, the most notable problem is that anomalies can be found, but it is difficult to analyze the cause of the anomaly.

2.3 Machine learning

Machine learning methods are great for finding hidden information in big data and finding relevant association information[14]. So a lot of machine learning methods are also used in abnormal traffic detection[15]. The most commonly used one is the clustering algorithm, which firstly extracts the similar parts from the traffic and clusters the traffic according to their characteristics. The part of the massive traffic aggregation can be regarded as normal traffic. If traffic occurs away from the normal traffic aggregation area at some moments, these traffic can be judged as abnormal traffic. Support Vector Machine (SVM) is an another common machine learning algorithm[16]. It is mainly used to divide large amounts of data according to categories. Therefore, by selecting appropriate features and using SVM normal traffic and abnormal traffic can be separated from each other[17].

Most machine-based learning methods have low time complexity and space complexity[18]. However, the applicability of this algorithm is not strong. A specific algorithm needs to be designed for a specific scenario. Because the machine learning algorithm has higher requirements for the training set. Therefore, this defect has prevented large-scale application of machine learning algorithms.

3 BF based abnormal traffic detection

From the analysis in the previous chapter, it can be seen that the abnormal traffic detection needs to meet two requirements. Firstly, the real-time detection is need to be guaranteed. Secondly, the method could distinguish different abnormal traffic. For real-time requirement this article uses BF to achieve big data storage and query. Because BF uses hashing techniques to alleviate the contradiction between storage rate and memory capacity and the computational complexity is relatively low. Therefore, the storage speed and recognition accuracy are improved[19]. However, this structure does not have statistical functions, so this paper also introduces CBF. Therefore, this paper jointly uses BF and CBF to quickly identify abnormal traffic. In order to distinguish abnormal traffic this paper analyzes the behavior characteristics of port scanning and TCP flooding traffic, designs the detection model and proposes a joint detection framework.

3.1 Bloom filter

BF is a special data structure whose main goal is to reduce the consumption of storage resources in high-speed data search. The main idea of this structure is to establish a mapping relationship between the set of elements S through the Hash function group K and a vector group V. There are n elements in the set S, and then the elements are mapped into the vector group V of length m by using k independent Hash functions, which means the results of the k hash functions are respectively set to 1 in the corresponding positions in V[20]. The principle is shown in Figure 1.

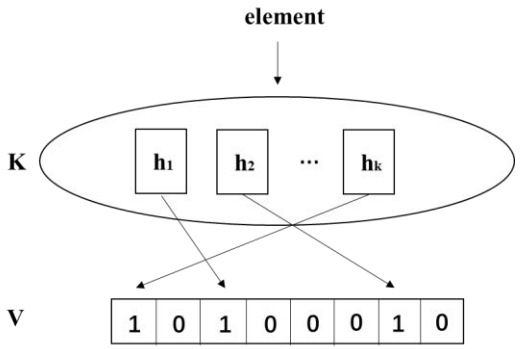

Fig.1 Schematic diagram of BF

The BF structure can search for the presence of elements in the set S in V at low time complexity and there will be no false negatives. However, BF will produce false positives. The elements that do not belong to the set S will be misidentified as the elements in S under certain probability. The false positive probability is as follows:

$$f = (1 - e^{-kn/m})^k \quad (1)$$

n represents the number of elements in the set S, m represents the length of the vector group V, k represents the number of hash functions.

when $\frac{df(k)}{dk} = 0$

$$k = \frac{m}{n} * \ln 2 \quad (2)$$

$$f_{min} = 0.5^{\frac{m}{n}*ln2} \quad (3)$$

According to Equation 2, in order to further reduce the probability of false positives, the length of the vector group V can be appropriately increased, but at the same time the number of hash functions is increased increasing the overall algorithm time complexity. In order to get a balance, appropriate parameters should be chosen.

3.1 Joint detection framework

The joint detection framework based on BF is divided into three parts, a recording module, a statistics module and a detection module. The main goal of recording module is to record the data packets. If unrecorded messages arrive, they need to be recorded firstly and then transmitted to the statistics module. The statistics module is the core of the joint detection framework, using BF and CBF to count the statistical characteristics. The detection module analyzes the recorded statistical characteristics. If the difference between the statistical characteristics in detected traffic and that in normal traffic is large, it can be judged that abnormal traffic occurs.

For the two types of abnormal traffic, port scanning and TCP flooding traffic, the statistics module and the detection module need to be separately designed due to different statistics. The specific design scheme will be elaborated in chapters 4 and 5. The overall framework flow chart is shown in the following figure.

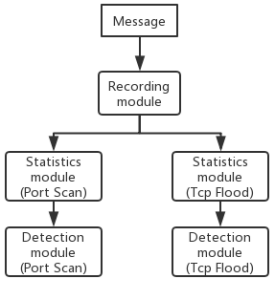

Fig.2 Schematic diagram of joint detection framework

4 Port scanning traffic analysis and modeling

Port scanning means that sender sends a message to each port of the target computer. According to the type of response received the sender could detect the condition of receiver. There are many ways to perform port scanning, either by manual scanning or by port scanning software. The commonly used port scanning methods are divided into TCP connection scanning, TCP (SYN) scanning and ICMP scanning.

4.1 Traffic characteristics analysis

Step 1: In order to analyze the characteristics of port scanning traffic, this paper uses the software named w*ireshark* to capture the data of normal traffic for a period of time at first. The source address and the destination address are used to distinguish various traffic. The number of traffic usage protocols, the number of access ports and the total number of packets are recorded to detect the characteristics of port scanning traffic.

The characteristics of normal traffic are shown in table 1.

Tab.1 Normal traffic

	Packets	Destination Ports	Protocols
1	293264	2	2
2	7362	1	1
3	254	1	1
4	207	1	1
5	56	1	1
6	53	1	1
7	51	6	1
8	16	1	1

Step 2: The port scanning traffic is actively injected into the normal traffic and the data of the abnormal traffic is captured by the software named *wireshark*. The analysis method is as described in the Step 1.

The characteristics of abnormal traffic are shown in table 2.

Tab.2 Abnormal traffic

	Packets	Destination Ports	Protocols
1	293264	2	2
2	7362	1	1
3	254	1	1
4	207	1	1
5	56	1	1
6	53	1	1
7 (port scanning)	435	202	2
8 (port scanning)	426	202	2

As can be seen from Table 1, in normal traffic, whether it is large traffic or small traffic, they access fewer target ports and the maximum number is no more than six. In Table 2, the 7th and 8th

streams are port scanning traffic. Although they only have about 430 messages, the number of target ports accessed in a short period of time exceeds 200, and the number of packets accessing different target ports accounts for about 50% of the total number of packets.

Therefore, according to Table 1 and Table 2, the characteristics of port scanning traffic are summarized as follows.

- The source address does not change and destination is the victim host.
- The source port and the destination port keep changing.

According to the traffic characteristic analysis, the port scanning traffic does not necessarily generate a large amount of traffic. However, the packet will apply for access to many common ports, such as 80 and 443, which has a large difference from the normal traffic. This paper extracts the source address and destination address of each stream as features and uses the number of ports covered as the detection standard. When the number of ports to be accessed is too large, it can be determined that a port scanning attack occurs.

4.2 Detection module

The port scanning detection model is divided into three parts. The first part is used to extract stream from the network. The second part is used to count the number of different ports that each stream requests to access. The third part is used to detect whether it appears abnormal.

4.2.1 Recording module.

The Recording module uses BF to identify the stream.

When the packet arrives, the source address and the destination address of the packet are extracted as the identifier of the stream, and then BF is used to determine whether the packet belongs to a new stream. The method of judging is as follows.

Hash functions $H_i(1 \le i \le k)$ are used to map the message to the k positions of BF. If the values of k positions of BF are all 1, the flow information to which the message belongs has been recorded. If the value of any position is not 1, it indicates that the stream has not been recorded yet. If the flow to which the message belongs has been drawn, the message will be transmitted directly to the statistics module, otherwise the stream will be recorded firstly.

The pseudo code of the recording module is as follows.

```
Input: m(captured message), V(BF vector)
    function MessageRecord(m,V)
      for i=0;i<k;i++ do
        srcip = Getsrcip(m)
        dstip = Getdstip(m)
        index = Hi(srcip+dstip)
        if V[index] = 0 then
          m has not been recorded, record m
          Break
        end if
      end for
end function
```

4.2.2 Statistics module.

The statistics module uses BF to record the number of access port numbers for the same stream.

When the message arrives, its source address, destination address and destination port are used as the identifier, and then Hash functions $H_i(1 \le i \le k)$ are used to map the message to the k positions of BF. If all values of these positions are equal 1, it indicates this feature group (source address + destination address + destination port) has been recorded.

4.2.3 Detection module.

The detection module maintains an array of common port numbers and Num_i represents the number of ports that the i-th stream requests to access.

Firstly, the recorded stream P_i (source address + destination address) is obtained from the recording module, and then P_i is stitched with different common port (source address + destination address + port). $P_i + port$ is mapped to k positions of BF using Hash functions $H_i(1 \le i \le k)$. If all values of these positions are equal 1, it indicates that this port appears in the stream. So Num_i could be added. Assuming that Num_j is too large, the j-th stream can be considered as port scanning traffic.

5 TCP flooding traffic analysis and modeling

When user makes a standard TCP connection, there is a 3-way handshake. The SYN flooding refers to that after the service sends a SYN-ACK message to the requesting party. Because requesting party uses the source address spoofing, the service keeps waiting to receive the requesting party ACK message at a certain time. If a malicious attacker sends such a connection request in rapid succession, the TCP connection queue available to the server will be blocked quickly. Therefore, the available resources of the system will be drastically reduced, the available bandwidth of the network will be rapidly reduced, and the server will not be able to provide normal legal services to the user.

5.1 Traffic characteristics analysis

- The source address is falsified and irregular, but the destination address is the victim host IP address.
- The length of packets in the traffic is relatively close and the difference between two packets is small.
- There are a large number of SYN, SYNACK and FIN packets in the SYN flooding traffic, so the length of TCP packet is short.

According to the traffic characteristic analysis, the most obvious feature in TCP flooding is that the randomness of the packet length is poor, which means the lengths of a large number of packets are relatively close. In the case of SYN flooding traffic, a large number of SYN and FIN packets appear in the same stream collection. These characteristics are quite different from regular traffic because there are a large number of ACK packets in the normal TCP stream and the message length is relatively random.

Therefore, the destination address, destination port, and packet length are extracted as the characteristics of the stream and the SYN and FIN flags in the packet will be detected at the same time.

5.2 Detection module

The TCP flooding detection model is divided into three parts. The first part is used to extract and merge the same stream from the network. The second part detects the characteristics of the stream and counts the total number of each stream. The third part is used to detect whether it appears abnormal.

5.2.1 Recording module.

The recording module is same as the recording module described in 4.2.1.

5.2.2 Statistics module.

The statistics module uses two CBF structures to record the total number of each type of stream. The first CBF is used as a flow identifier based on the destination address and the destination port to count the total number of streams. The second CBF structure uses the destination address, the destination port, and the packet length as the flow identifier to collect the number of packets of a certain length in this type of flow.

After the packet enters the statistics module, the packet destination address and the destination port feature are extracted firstly and the Hash function $H_i(1 \le i \le k)$ is used to map the message to the k positions of the first CBF whose value is increased by 1. Secondly, the SYN and FIN flag bits are judged. If one of the flag bits is set to 1, the message length is zero. Otherwise, the message length will be up to an integer multiple of 10. The packet length is set to an integer multiple of 10, which is used to classify packets of similar length into one type thereby improving the accuracy of detection. Then the message will be mapped to the k positions of the second CBF with Hash functions $H_i(1 \le i \le k)$ using these three features as the stream identifier. Each value of these k positions will by increased by 1.

The pseudo code of the statistics module is as follows.

```
Input: m(captured message), V1(first CBF), V2(second CBF)
    function TCPFlooding_StatisticModule(m,V1,V2)
        for i=0;i<k;i++ do
            dstip = Getdstip(m), dstport = Getdstport(m)
            len = Getlen(m)
            if m is SYN or m is FIN then
                len = 0
            else
                len = (len/10+1)*10
            end if
            index = Hi(dstip+dstport)
            V1[index] += 1
            index = Hi(dstip+dstport+len)
            V2[index] += 1
            end for
end function
```

5.2.3 Detection module.

Firstly, detection module calculates the total number of the message in the first CBF and records the number as Num_a. Secondly the module calculates the total number Num_b of the current message in the second CBF. In normal traffic. Because of the randomness of the message length, the ratio of Num_b/Num_a is small. However, in TCP flooding traffic the number of packets of the same length is large, so the ratio of Num_b/Num_a of such packets is large.

Therefore, only Num_b/Num_a needs to be calculated. If the value is too large and exceeds the set threshold, it can be determined that a TCP flooding attack occurs in this stream. At the same time, according to the length of the current packet, if the length is 0, it indicates that the traffic belongs to the SYN flooding traffic, otherwise it is determined to be ordinary TCP flooding traffic.

6 Experiment

The algorithm running environment of this paper is as follows: CPU, Intel Core i5-6200U, 2.4GHz; memory, 4GB; operating system, Windows10; development tools and development language, Eclipse IDE, Java. TCP flooding is generated by the software named *scapy* and the source address is forged. The port scanning traffic is implemented by the software named *namp* and the traffic packet is captured by the software named *wireshark*.

6.1 Experiment on port scanning traffic

Port scanning experiment data comes from traffic fetched from the real environment. To ensure the lowest false positive rate and false negative rate any port less than 500 is considered to be a normal port. Therefore, if the destination port of the traffic access is less than 500, it is judged that the traffic scans a port. The value of *a* is set to 50. That is, the total number of different ports that port scanning traffic accessing exceeds 50. A type of traffic that satisfies these two elements is determined as port scanning traffic and the destination address will be detected as well.

In order to test the performance of the BF-based port scanning traffic detection method, all port scanning traffic is marked in the captured real traffic.

Tab.3 Port scanning traffic

Destination address	Different destination ports
119.75.217.26	150
58.205.221.214	150
202.119.24.205	200
58.205.217.1	200
58.205.212.207	250
58.205.214.142	250
180.163.155.12	300
222.192.186.23	350
60.210.19.136	400
58.205.221.224	450

From Table 3, it shows that there are 10 port scanning traffic in the experiment and the number of hosts scanned and the number of ports scanned are different.

The false alarm rate will be tested firstly. The target address of the detected port scanning traffic and that of the real abnormal traffic will be compared. If the addresses are equal, the detection is correct, otherwise a false alarm is generated. Secondly, the false negative rate will be tested. If the actual abnormal traffic is not detected, then a false negative will occur.

Figure 3 shows the overall traffic situation and table 4 lists the results.

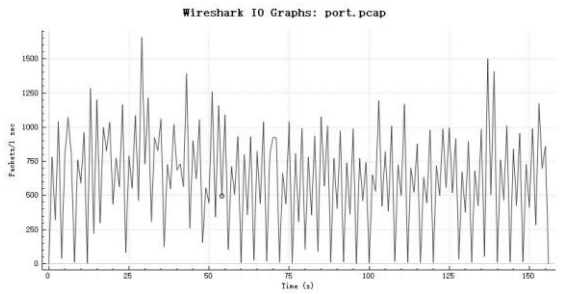

Fig.3 Traffic situation

Tab.4 Results

Detected address	Detected different ports
119.75.217.26	158
58.205.221.214	158
202.119.24.205	209
58.205.217.1	208
58.205.212.207	255
58.205.214.142	254
180.163.155.12	304
222.192.186.23	352
60.210.19.136	404
58.205.221.224	451

The experiment results show that the detection framework proposed in this paper is inaccurate for the detection of the number of scanning ports. For example, in the traffic of the target host of 119.75.217.26, 150 ports are scanned in the real traffic, but 158 ports are detected to be scanned; in the traffic of the target host 202.119.24.205, 200 ports are scanned in the real traffic, but a total of 209 ports were scanned. However, during the entire detection process, less false positives or false negatives were generated and the information of the detected port scanning attacks were correct.

6.2 Experiment on TCP flooding traffic

TCP flooding experimental data comes from traffic that is fetched from the real environment. In order to ensure the lowest false positive rate and false negative rate, set *a* to 3000packets/s, which means TCP flooding average traffic must reach 3000 packets per second, otherwise it is not judged to be TCP flooding traffic; set *b* to 50% That is, the number of packets of similar length in a type of traffic must reach 50% of the total number of packets. A type of traffic that satisfies these two elements is determined as TCP flooding traffic and for the detected suspected TCP flooding traffic whose target address and target port will be detected.

In the experiment, this paper utilizes the software named *scapy* to monitor TCP flooding attack. The core code for SYN flooding and other TCP flooding attacks is described in picture 3. The source address and source port are chosen randomly.

```
class sendSYN(threading.Thread):
        global target, port
        def __init__(self):
                threading.Thread.__init__(self)

        def run(self):
                IPlayer = IP()
                IPlayer.src = "%i.%i.%i.%i" % (random.randint(1,254),random.randint
(1,254),random.randint(1,254),random.randint(1,254))
                global target, port
                IPlayer.dst = target

                TCPlayer = TCP()
                TCPlayer.sport = random.randint(1,65535)
                TCPlayer.dport = port
                TCPlayer.flags = 'S'

                pkt = IPlayer / TCPlayer
                send(pkt)
```

Fig.3 Core code in TCP flooding

In order to observe the performance of the BF-based TCP flooding traffic detection method, all the TCP flooding traffic is marked in the captured real traffic.

Tab.5 TCP flooding traffic

Destination address	Destination port	Type
112.124.47.27	80	TCP flooding
119.75.217.26	21	TCP flooding
42.120.21.30	443	SYN flooding
119.75.217.26	20	SYN flooding
58.205.212.207	80	TCP flooding
58.205.214.142	21	SYN flooding
60.210.19.136	443	TCP flooding
58.205.221.224	80	TCP flooding

As can be seen from Table 5, there are 8 kinds of TCP flooding traffic in the real traffic. The third, fourth and sixth abnormal traffic belong to SNY flooding traffic and the other 5 are other TCP flooding traffic. The destination addresses in TCP flooding traffic are different from each other.

This chapter tests the false positive rate firstly. The model will detect the three parameters: target address, target port and exception type. Comparing these parameters with the real abnormal flow characteristics, if the detection parameters and the original parameters are equal, it means the model detects TCP flooding traffic successfully, otherwise a false alarm will occur. Secondly, the false negative rate will be tested and if any of abnormal traffic is not detected successfully, a false negative will be generated. Table 6 lists the results.

Tab.6 Results

Address	Port	Type
112.124.47.27	80	TCP
119.75.217.26	21	TCP
42.120.21.30	443	SYN
182.248.50.109	80	SYN
119.75.217.26	20	SYN
58.205.212.207	80	TCP
58.205.214.142	21	SYN
60.210.19.136	443	TCP
58.205.221.224	80	TCP

From table 6, the detection framework detects three important features: target address, target port and the type of TCP flooding.

It is clear that one false positive occurs. The fourth stream was a normal traffic but was judged as SYN flooding incorrectly. The main reason for the mistake is that poor network environment resulted in a large number of SYN retransmission packets in the network. When it comes to the other traffic, the detected results are same as the real abnormal traffic, which means they are detected correctly.

However, compared with port scanning framework the parameters, such as the average traffic, are essential to the TCP flooding traffic. In order to obtain the optimal performance, the measurement of real traffic is considerably important before abnormal traffic detection.

7 Conclusion

A joint detection model based on BF structure is proposed for two kinds of abnormal traffic (port scanning and TCP flooding traffic). The method mainly utilizes the excellent retrieval performance of BF and CBF in large-scale traffic. Secondly, with the behavior characteristics analysis of the above two kinds of

abnormal traffic the model extracts some parameters that have large difference between abnormal flow and normal flow. Finally, the detection of abnormal traffic is converted into the detection of these parameters, which greatly simplifies the complexity of the model. Therefore, the detection model can accurately identify abnormal traffic in a large amount of data. The simulation results verify the effectiveness of the proposed method.

ACKNOWLEDGMENTS

The work in this paper was supported by the science and technology project of State Grid Corporation of China: "Research on Key Technologies of Marketing Site Terminal Security Access" (Grand No. SGGR0000XTJS1800079).

REFERENCES

[1] Perera, C., Chi, H. L., & Jayawardena, S. (2017). The emerging internet of things marketplace from an industrial perspective: a survey. IEEE Transactions on Emerging Topics in Computing, 3(4), 585-598.

[2] Perera, C., Chi, H. L., Jayawardena, S., & Min, C. (2017). A survey on internet of things from industrial market perspective. IEEE Access, 2, 1660-1679.

[3] Islam, N., & Islam, N. (2017). Botnets and internet of things security. Computer, 50(2), 76-79.

[4] Xin, Y., Mo, X., Wang, C., & Xin, Y. (2017). Research on Real-Time Flow Abnormal Traffic Detection System Based on DDoS Attack.

[5] Marnerides, A. K., Schaeffer-Filho, A., & Mauthe, A. (2014). Traffic anomaly diagnosis in internet backbone networks: a survey. Computer Networks, 73(C), 224-243.

[6] Chen, M. , Chen, S. , & Cai, Z. . (2017). Counter tree: a scalable counter architecture for per-flow traffic measurement. IEEE/ACM Transactions on Networking, PP(99), 1-14.

[7] Geng, T., Wang, Z., Xia, Y., Chen, J., Shi, X., & Chao, Z., et al. (2017). CEFF: An efficient approach for traffic anomaly detection and classification. Computers & Communications.

[8] Denning, D. E. (2006). An intrusion-detection model. IEEE Transactions on Software Engineering, SE-13(2), 222-232.

[9] Cheng G, Gong J, Ding W. (2003). A real-time detection model based on sampling measurement in a high-speed network. Journal of Software, 14(3), 594-599.

[10] Wei, Y. , & Jun, Z. . (2017). Network traffic anomaly detection based on time series analysis. Journal of Jilin University.

[11] Barbhuiya, F. A. , Roopa, S. , Ratti, R. , Biswas, S. , & Nandi, S. . (2012). An active detection mechanism for detecting icmp based attacks.

[12] Barford, P., Kline, J., Plonka, D., & Ron, A. (2002). A signal analysis of network traffic anomalies. Proc Acm Sigcomm Internet Measurement Workshop.

[13] Nurohman, H., Purwanto, Y., & Hafidudin. (2015). Traffic anomaly based detection: Anomaly detection by self-similar analysis. International Conference on Control.

[14] Zhang, Z., He, Q., Jing, G., & Ming, N. (2018). A deep learning approach for detecting traffic accidents from social media data. Transportation Research Part C Emerging Technologies, 86, 580-596.

[15] Zou, M., Wang, C., Li, F., & Song, W. Z. (2018). Network phenotyping for network traffic classification and anomaly detection.

[16] Kong, L., Huang, G., & Wu, K. (2017). Identification of Abnormal Network Traffic Using Support Vector Machine. International Conference on Parallel & Distributed Computing.

[17] Yan, G. (2017). Network Anomaly Traffic Detection Method Based on Support Vector Machine. International Conference on Smart City & Systems Engineering.

[18] Ciptaningtyas, H. T. . (2017). Network Traffic Anomaly Prediction Using Artificial Neural Network. International Conference on Education. 5th International Conference on Education, Concept, and Application of Green Technology.

[19] Peng, X., Li, Z., Qi, H., Qu, W., & Yu, H. (2017). An Efficient DDoS Detection with Bloom Filter in SDN. Trustcom/bigdatase/ispa.

[20] Kumar, A., Xu, J., Li, L., & Jia, W. (2003). Space-code bloom filter for efficient traffic flow measurement. Acm Sigcomm Conference on Internet Measurement.

[21] Shang, G. , Zhe, P. , Bin, X. , & Yubo, S. . (2016). Secure and energy efficient prefetching design for smartphones. IEEE International Conference on Communications. IEEE.

[22] Shang, G. , Zhe, P. , Bin, X. , Aiqun, H. , & Kui, R. . (2017). FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks. IEEE INFOCOM 2017 - IEEE Conference on Computer Communications. IEEE.

[23] Xue, N. M. , Wang, N. J. , & Hux, N. A. . (2016). An enhanced classification-based golden chips-free hardware Trojan detection technique. 2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST). IEEE Computer Society.

[24] Jiang, Y. , Hu, A. , & Huang, J. . (2018). A lightweight physical-layer based security strategy for internet of things. Cluster Computing.

[25] Benson, T., & Chandrasekaran, B. (2017). Sounding the bell for improving internet (of things) security.

[26] Andoh-Baidoo, F. K., & Osei-Bryson, K. M. (2007). Exploring the characteristics of internet security breaches that impact the market value of breached firms. Expert Systems with Applications, 32(3), 703-725.

[27] Saez, M., Maturana, F. P., Barton, K., & Tilbury, D. M. (2018). Real-time manufacturing machine and system performance monitoring using internet of things. IEEE Transactions on Automation Science & Engineering, PP(99), 1-14.

[28] Abhishta, Joosten, R., & Nieuwenhuis, L. J. M. (2018). Comparing alternatives to measure the impact of ddos attack announcements on target stock prices.