Fast-Flux Detection Method Based on DNS Attribute

Jiajia Wang\textsuperscript{1*}, Yu Chen\textsuperscript{1}

\textsuperscript{1}School of Information Technology, Taizhou Polytechnic College, Taizhou, Jiangsu, 225300, China

Corresponding author’s e-mail: wangjiajia_99@163.com

Abstract. Fast-flux service network provides redirection/proxy services for a large number of malicious and illegal content based on botnet, and provides a high level of protection for botmaster. In this paper, a method is proposed to distinguish fast-flux service network from legal website. Firstly, the differences between the attributes of the above two are analyzed in detail, then the detection features are constructed, and decision tree algorithm is used to detect the existence of fast-flux service network.

1. Introduction

FFSN (Fast-flux service network) continuously updates malicious domain name entries during regular intervals, such as quickly update IP addresses. Repeatedly and rapidly updating IP addresses can hide the true location of malicious servers and avoid blacklist settings. This process increases the communication level of FFSN and improves the robustness and anonymity of malicious behavior to a great extent. The combination of FFSN and botnet brings great harm to the stable operation of Internet. CDN (Content Delivery Network) and RRDNS (Round Robin Domain Name System) adopt similar features. In order to load balance and increase the reliability and availability of regional services, CDN and RRDNS are associated with multiple DNS (Domain Name System) records, too. Therefore, how to effectively distinguish FFSN from the other two technologies has become an urgent problem to be solved.

The structure of this paper is as follows: section 1 describes the related work in the current field and analyzes the characteristics of the existing FFSN detection methods. Section 2 explains the working principle and process of FFSN. Section 3 abstracts and optimizes the related features of data packets according to the characteristics of FFSN, then sums up the detection flow of FFSN. Section 4 describes the detection algorithm used in this paper, and good algorithm can further improve the detection efficiency. Section 5 is the experiment and result, and finally the conclusion and prospect.

2. Related work

Based on the comprehensive analysis of existing literature, it can be concluded that the FFSN detection methods can be divided into the following categories.

2.1. Based on characteristics of DNS reply packets

At present, most FFSN detection works are utilizing the characteristics of DNS response packets. Lombardo\textsuperscript{1} performed passive analysis on DNS communication, identified the main characteristics of FFSN in almost real-time, and detected the data mining method, the depth analysis of FFSN domain name can prove the reliability of the proposed method. However, the method needed to track a large number of DNS streams, whether legal or illegal. Knysz\textsuperscript{2} detected the size of a DNS packet, but the
size of packet is not strictly related to the attack. Miller\cite{3} had adopted more complex detection means, requiring detection of packet size, sequence information, etc., which can easily produce more delay.

2.2. Based on characteristics of domain names
Changling\cite{4} proposed a passive detection method, which used random forest algorithm to detect the existence of FFSN by recording the domain name query history of the real campus network and combining with the detection features. However, this method has a high false alarm rate. Martinezbea\cite{5} proposed a method by collecting domain name resolution records, the more comprehensive the collection, the higher the detection accuracy. But there are many domain names in the network, the comprehensive collection requires a lot of storage space. Queries can also cause delays.

2.3. Based on characteristics of space
Wang\cite{6} proposed a FFSN detection method based on spatial geographic location, which is enhanced by autonomous system numbering. Tyagi\cite{7} focused on detecting geographically dispersed FFSN. However, in some countries, IP positioning needs special permission, if only according to the spatial characteristics of detection, the maneuverability is not strong.

2.4. Based on characteristics of time
Martinezbea\cite{8} proposed a real-time detection method based on domain name resolution records, but this method needed to actively query every domain name, and the result was only to detect the existence of FFSN, easily to produce a high false alarm rate. Almomani\cite{9} proposed a fast-flux detection method with both real-time classification and long-term monitoring, which had high detection rate and low false alarm rate, but could not accurately locate the data. It would affect the effectiveness of the method.

Based on the advantages and disadvantages of the above methods, the detection features used in this paper include a variety of detection types, not limited to one of them. Celik\cite{10} showed that the combination of various types of features can achieve higher accuracy. From the overall point of view, some features that are not feasible and unnecessary are excluded, and the detection features are constantly optimized, which are described in detail in the fourth section.

3. Working Principle and Process of FFSN
FFSN is a technique used by attackers to improve the robustness of malware communication. FFSN was first proposed in 2007, which referred to a network composed of controlled hosts, on which the DNS resolution records were constantly changing every few minutes\cite{11}. There are many types of DNS resolution records, as shown in Table 1. This article mainly uses type A resolution records.

| Type   | Number | Description        | Type   | Number | Description        |
|--------|--------|--------------------|--------|--------|--------------------|
| A      | 1      | IPv4 address       | PTR    | 12     | Pointer Record     |
| NS     | 2      | Name Server        | HINFO  | 13     | Host Information   |
| CNAME  | 5      | Canonical NAME     | MX     | 15     | Mail Exchange Record |

The process by which one client sends http requests to the mothership to apply for fast-flux service is as follows:

- One client sends a DNS request that queries the domain name and IP address of the mothership;
- The local DNS server responds when it receives the request, returning a series of resolved IP addresses to the client;
- The client uses one of the IP addresses to request http service;
- The proxy host corresponding to the IP address redirects the request to the mothership;
- The mothership which containing malicious content answers the client.

4. Related Characteristics and Detection Flow of FFSN
A large number of historical data shows that there are differences in TTL, the number of IP addresses, the change rate of IP address, the volatility of domain name response time and the number of whois
information updates between legitimate sites and FFSN. If the uniform distribution rate index is used, e.g., the analysis of the number of ASN or the number of countries, is even worse for the crowded network, the network would be overwhelmed. If the availability index of the network is used, e.g., to check whether the IP address could provide available service or not, it will take a lot of query and wait for the response. In this paper, more convenient indicators are used.

4.1. Attributive Analysis

We pay attention to fast-flux detection. First, we give the differential analysis of related attributes.

• TTL

In general, the TTL value of single-flux does not exceed 300s, the TTL value of double flux does not exceed 600s and the quantity is much smaller than single-flux. Most of legitimate sites set TTL value exceed 600s\(^{[13]}\), which usually defaults to 3600s. Figure 1 shows the average TTL values for legitimate sites and FFSN.

• The number of IP address \(IP_{CN}\)

\(IP_{CN}\) is the number of IP addresses occupied by sites during detection time. The IP addresses occupied by legitimate domain names are servers, which are highly available and relatively stable, the increase rate of new IP is relatively slow. In order to maintain availability, FFSN usually takes up more IP addresses. These IP are often terminal hosts which lack of protection, online time is unstable and may be offline at any time. FFSN has to continue to add new IP rapidly during its survival, as shown in figure 2.

• The change rate of IP address \(IP_{CS}\)

If a domain name wants to provide with normal network service, legitimate IP address must be used for resolution, the IP address whether considered to be changed or not in accordance with the service condition. They are stable and reliable servers provide services for legitimate domain names, almost no frequent replacement, and the scope of IP address replacement is also within a limited number of address pools to facilitate the improvement of the stability and reliability. Most of the services provided for the FFSN are terminal hosts with weak security protection capabilities. Because of the instability of terminal host itself and in order to avoid to be detected, the attacker have to change the IP of the domain name map frequently, and the new IP address should be used for almost every IP address change.

• The volatility of domain name response time \(RT_{V}\)

Response time refers to the waiting time between the current packet sending and receiving the response packet. If the IP corresponding to the domain name is relatively stable, the waiting time is also relatively stable. If the host where the corresponding IP located is often turned off or offline, there will be a greater uncertainty about the response time.

The legal network is usually based on stable and reliable servers, the IP addresses are stable and reliable, and the DNS domain name resolution and connection stages are almost the same, so the
response time is relatively stable and $R_{TV}$ is stable. In view of the terminal host’s own instability, such as shutdown, poisoning, inappropriate software used and so on, FFSN often needs to be re-resolved the domain name and even establish a connection to a new host, which is uncontrollable and $R_{TV}$ is large.

- Number of whois information updates $W_{UC}$

All legitimate domain names have whois information. In order to improve the quality and popularity of service, the owners of legitimate domain names update the whois information at intervals, and the rate of update per unit time is higher. The owners of FFSN only use these domain names to carry out malicious activities. The updating times of whois information are long and the updating rate per unit time is lower.

4.2. Detection Features

- The change rate of IP address $I_{PCS}$

In the unit detection time $T(T>TTL)$, the number of IP addresses divided by $T$ is the IP change rate $I_{PCS} = I_{PCN}/T$, the $I_{PCS}$ of legitimate domain name tends to 0, and the $I_{PCS}$ of FFSN increases rapidly.

- The volatility of domain name response time $R_{TV}$

In the unit detection time $T(T>TTL)$, the current response time of the domain name $RT$, response time volatility is that $R_{TV} = RT/T$, the legal domain name response time is short and stable, $R_{TV}$ tends to be stable, and the FFSN response time is unstable. $R_{TV}$ shows an uncertain state of change.

CDN and RRDNS also use small $TTL$, but the response time $RT$ is relatively stable, so it is still in a stable state, which is obviously different from FFSN.

- The ratio of the number of IP addresses $I_{PCN}$ to the number of whois information updates $W_{UC}$

From the previous analysis, it can be concluded that the number of IP addresses of legitimate domain names $I_{PCN}$ tends to be stable with the increase of time, and the number of whois information updates $W_{UC}$ would increase steadily. For the number of IP addresses of FFSN $I_{PCN}$ would increase rapidly. The number of whois information updates $W_{UC}$ is almost the same. Therefore, with regard to the ratio of the number of IP addresses to the number of whois information updates $W_{UC}$, the legal domain name will gradually tend to 0, while the FFSN will grow rapidly.

4.3. Detection Process

The FFSN detection process designed in this paper is as follows:

- Preprocessing part

(1) Fast-flux technology is a special DNS technology, so we only detects port 53 and UDP protocol packets, which are the characteristics of DNS protocol. This step can filter most of packets in the network.

(2) In view of the fact that fast-flux packets account for a small proportion of DNS packets and there have been considerable research results, black and white lists are used in this paper. The first $10^6$ records listed in the www.Alexa.com site are listed as whitelist, if the source of the current packet matches the record in the whitelist, directly forward. The FFSN blacklist, which has been detected in previous research, can also be used to share, not only fast-flux blacklist, but also other types of blacklist can also be used. Such as the blacklist in the www.malwaredomains.com website.

Packets preprocessed composed a set of samples $S$.

- Specific detection part

(3) For each packet $X(X \in S)$ that passes through preprocessing, extract feature vecto $F_X = \{F_{X1}, F_{X1} = I_{PCS}\}
\{F_{X2}, F_{X2} = R_{TV}\}
\{F_{X3}, F_{X3} = I_{PCN}/W_{UC}\}$.

(4) The related algorithm is used to detect the features, the classification results are obtained by comparing the training data, that is, the detection site is determined to be FFSN whether or not. The specific process is shown in figure 3.
5. FFSN Detection Algorithm

It is the focus of current work to detect and distinguish FFSN and legitimate networks according to multiple characteristics, not to occupy too many network resources. The final conclusion is obtained by combining several factors, we take the decision tree algorithm as the classifier. Decision tree is a tree structure composed of nodes and directed edges. Each node in the tree has the characteristic of maximum information gain, and the data can be classified until the leaf node, that is, a path from the root node of the tree to a certain leaf node represents a classification rule.

Among all the decision tree algorithms, according to the existing conclusion, the C4.5 algorithm is one of the best. The specific algorithm is described below:

1) Preprocessing of training dataset. If the dataset has continuous attributes, it needs to be discretized first.
2) The data are classified according to the attributes, and the corresponding information gain rate is calculated for each classification result.
3) Select a feature from the current set of leaf nodes to calculate the information gain Gain(A_i) = Gain(A_i) - Entropy(A_i);
4) Calculate attribute split information metric SplitInfo(A_i) = - \sum_{t=1}^{w} \frac{|A_i|}{|D|} \log_2 \frac{|A_i|}{|D|};
5) Calculate information gain rate GainRatio(A_i) = \frac{Gain(A_i)}{SplitInfo(A_i)}.

(3) According to the attributes corresponding to the maximum information gain, the current data set is divided into different subsets, and the corresponding decision tree branches are established to form new child nodes.
(4) For the newly generated nodes, step 2, 3 is called iteratively until the record class number in all nodes are the same, then the decision tree is built.

The experimental results can be divided into two kinds: legitimate site or FFSN, C4.5 algorithm can be dealt with as binary problem. In this article, the trust level is set to 0.25, and the minimum class for each leaf node is set to 2 for pruning. Since the decision tree has been established during the training phase, it is known that the best features that distinguish legitimate site from FFSN, the attributes that generate the maximum information gain have more powerful authentication capabilities. The obtained results can be used as a filtering method to sort the features according to the calculated information gain values. Finally, according to the given threshold, the characteristics that could best distinguish the experimental data can be determined.

The detection features used in this paper do not have continuous attributes, the number of features is small, but it is effective, it is impossible that the model is too complex. When C4.5 algorithm is applied to this paper, the efficiency will not be reduced due to excessive fitting.

6. Experiments and results

6.1. Training dataset and experimental dataset
We select training data from www.Alexa.com site and www.malwaredomains.com site, select 5000 legitimate domain names from www.Alexa.com site, divided into 5 groups as training data, and select 2000 malicious domain names from www.malwaredomains.com website, divided into 5 groups as training data. Five groups of training data \( D_{T1}, D_{T2}, D_{T3}, D_{T4}, D_{T5} \) included 1000 legitimate domain names and 400 malicious domain names, respectively.

The training data do not need to go through the preprocessing part, directly enter the specific detection part to carry on the feature extraction calculation. The total training time of the five groups of data is about 4 hours.

The experiment data comes from the real data of campus network. In this paper, the domain name data collected within 30 days of campus network is used as the experimental data, involving about 400 hosts.

6.2. Experiment
Windows 7 operating system is used to run the detection algorithm. The experimental data were divided into 5 groups with an interval of 10 min, and the whole experiment took about 8 hours.

The following parameters are mainly considered, as shown in Table 2. The experimental results are shown in Table 3.

| Parameters | Parameter Declaration | Test NO | FP | FN | FPR | RR |
|------------|-----------------------|---------|----|----|-----|----|
| TP         | The number of correctly identified FFSN | 1       | 46 | 1  | 4   | 97.9% | 92.0% |
| FN         | Number of FFSN identified as legitimate domain names | 2       | 46 | 1  | 3   | 97.9% | 93.9% |
| FP         | Number of legitimate domain names identified as FFSN | 3       | 47 | 0  | 3   | 100%  | 93.9% |
| R          | detection rate R = TP / (TP + FN) | 4       | 47 | 0  | 2   | 100%  | 95.9% |
| P          | correct rate P = TP / (TP + FP) | 5       | 47 | 0  | 2   | 100%  | 95.9% |

From the above analysis, it can be seen that the detection rate and correct rate of this method will be optimized with the increase of using frequency. Among the similar detection methods, the detection rate of Soltanaghaei[14] is only 94%. The detection rate of Lombardo[11] is the same with us, but the time and space complexity is higher, so we have obvious advantages compared with other similar literatures.

7. Conclusion and prospect
An FFSN detection method is proposed and verified, which uses several types of detection features and has small computational and space-time complexity. In this paper, the packet is preprocessed and only
the DNS packet is detected. Filtering known types of domain names with black and white lists can reduce the burden of detection, select three simple but effective detection features, and C4.5 decision tree algorithm is used for FFSN detection. The experimental results show that the proposed method can effectively identify FFSN, and distinguish it from other legal types of networks; the detection rate and accuracy are ideal. The next work is to study a more accurate detection system.

Acknowledgments
This work was financially supported by Taizhou Polytechnic College Boshuo Fund(TZYBS-17-6), and Taizhou Polytechnic College Students Innovation and Entrepreneurship Training Program (YJDC2018017).

References
[1] Lombardo P , Saeli S , Bisio F , et al. (2018). Fast Flux Service Network Detection via Data Mining on Passive DNS Traffic. https://arxiv.org/pdf/1804.06316.pdf.
[2] Knysz M, Hu X, Shin K G. (2011) Good guys vs. Bot Guise: Mimicry attacks against fast-flux detection systems In: 2011 Proceedings IEEE INFOCOM. pp:1844-1852.
[3] Miller D J, Kocak F, Kesidis G. (2012) Sequential anomaly detection in a batch with growing number of tests: Application to network intrusion detection. https://ieeexplore.ieee.org/document/6349793.
[4] Changling Z , Kai C , Xuxiao G , et al. (2016) Detection of Fast-Flux Domains Based on Passive DNS Analysis. http://www.en.cnki.com.cn/Article_en/CJFDTotal-BJDZ201603003.htm.
[5] Martinezbea S, Castilloperez S, Garciaalfaro J. (2013) Real-time malicious fast-flux detection using DNS and bot related features. https://ieeexplore.ieee.org/document/6596093.
[6] Wang H T, Mao C H, Wu K P, et al. (2012) Real-Time Fast-Flux Identification via Localized Spatial Geolocation Detection. https://ieeexplore.ieee.org/document/6340149.
[7] Tyagi A K, Aghila G. (2012) Detection of fast flux network based social bot using analysis based techniques. https://ieeexplore.ieee.org/document/6281898.
[8] Martinezbea S, Castilloperez S, Garciaalfaro J. (2013) Real-time malicious fast-flux detection using DNS and bot related features. https://ieeexplore.ieee.org/document/6596093.
[9] Almomani A , Gupta B B , Atawneh S , et al. (2013) A Survey of Phishing Email Filtering Techniques[J]. IEEE Communications Surveys & Tutorials, 15(4), pp:2070-2090.
[10] Celik Z B, Oktug S. (2013) Detection of Fast-Flux Networks using various DNS feature sets. https://ieeexplore.ieee.org/document/6755058.
[11] The Honeynet Project: Know Your Enemy: Fast-Flux Service Networks (2007), http://www.honeynet.org/book/export/html/130.
[12] Almomani, Ammar. (2016) Fast-flux hunter: a system for filtering online fast-flux botnet. Neural Computing and Applications, 29(7), pp:483-493.
[13] TRUONG DINH T U (2015). HTTP--Based Botnet Detection Using Network Traffic Traces. http://www.wanfangdata.com.cn/details/detail.do?_type=degree&id=Y3142003.
[14] Soltanaghaei E , Kharrazi M . (2015) Detection of Fast-Flux Botnets Through DNS Traffic Analysis, Scientia Iranica, 22(6),pp:2389-2400.