IOP Conference Series: Earth and
Environmental Science

Research and Detection of Fast-flux Botnet

Research and Detection of Fast-flux Botnet

Lu Yang and Gang Gan[*]

School of Cyberspace Security, Chengdu University of Information Technology,
Chengdu 610225, China
Corresponding Author Email: test_me@cuit.edu.cn

Abstract. With the development of Internet technology, there are more and more network attack modes. APT attacks, as one of them, will have a very serious impact due to its concealment, long-term nature, and purpose. Fast-flux botnets are widely used in APT attacks due to their spreading power and destructive power. This article first introduces the basic structure and related knowledge of FFSNs, and then summarizes and compares the previous papers on Fast-flux botnet research. In order to effectively detect the Fast-flux botnet, this experiment removed features that require a lot of resources and uncertain factors, and adopted a more convenient feature AA, which became a more lightweight detection model. At the same time, because the automatic update module of the detection system will promptly feedback the current traffic situation in the network, it will greatly improve the accuracy. The experimental results of the CTU-13 and ISOT public data sets show that the detection method proposed in this article Compared with other methods, the accuracy rate is increased to 98%.

1. Introduction

In recent years, APT (Advanced Persistent Threats) attacks have become increasingly severe security challenges due to their concealment and persistence. APT attacks are roughly divided into the information collection phase, the invasion phase, the incubation phase, and the exit phase. The Fast-flux botnet has non-negligible applications in each stage, causing a large number of users to lose.

In a normal network, if a user waits too long when opening a web page, then he will choose to close the web page. Therefore, for convenience and speed, many large web pages now use CDNs (Content Delivery Networks) technology to support their operations. This technology is not simply putting content on different machines in the same place, but on different machines in different places. Above, CDNs technology can effectively achieve load balancing. However, the Fast-flux botnet also uses a Fast-flux technology similar to CDNs technology to establish their illegal network circle, which causes a certain amount of trouble for the inspectors. Attackers can also launch DDOS attacks through controlled hosts in FFSNs.

Therefore, the research on the Fast-flux botnet is meaningful. On the basis of summarizing other people's papers and experiments, we chose the DNS response package as the basis for implementation. And put forward the following new entry points.

- Propose a new feature AA: The domain name accessed by a user in a period of time will not change frequently. This means that when querying a domain name, the local DNS server has stored a cache of the domain name. Therefore, in the normal situation, the proportion of the value of the AA of 1 is very small. However, when visiting a domain name of a Fast-flux botnet, it is usually the first visit, so compared to the normal situation, the ratio of AA to 1 will be relatively larger.
- An automatically updated model is used to detect Fast-flux botnets: In past experiments[1,2,4,5], fixed models and fixed parameters are usually used to determine future traffic packets. This will cause the model to be unable to adapt to changes in network traffic. In this experiment, during the process of

creating the model, the newly received DNS response packet will be used as data to update the model in real time. Greatly enhance the portability and adaptability of the model.

- Lightweight detection model: In the traditional model, the ASN code in the DNS response packet is used as a feature, and this feature occupies a very large weight. To detect the ASN code, an interface or a local database needs to be called, which will use too much resources, and there are uncertain factors such as network delays. In this paper, we did not use this feature in the subject, and the detection accuracy rate has reached 98%.

2. System Implementation

The system is mainly divided into 4 parts (as shown in Figure 1), the first part is the filtering module, the second part is the information collection part, the third part is the detection model, and the fourth part is the automatic update part.

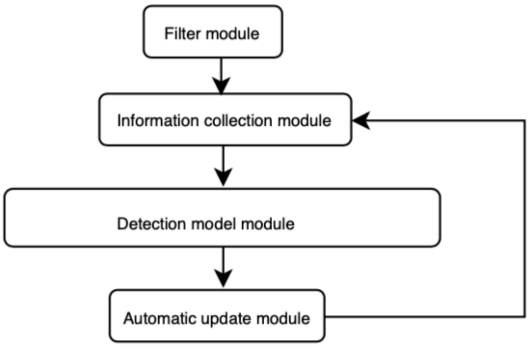


Figure 1. System model

2.1. Preparation Stage

In the preparation phase of this test, we first extracted all the fields in DNS to compare normal networks and Fast-flux botnets. Extract the AA, TTL, ANCOUNT, NSCOUNT, ARCOUNT, ANSWER fields in the DNS response packet. Experiments show that there is not much difference between the RA and RD values in normal networks and botnets.

The specific results are shown in Figure 2, Figure 3, Figure 4, and Figure 5.

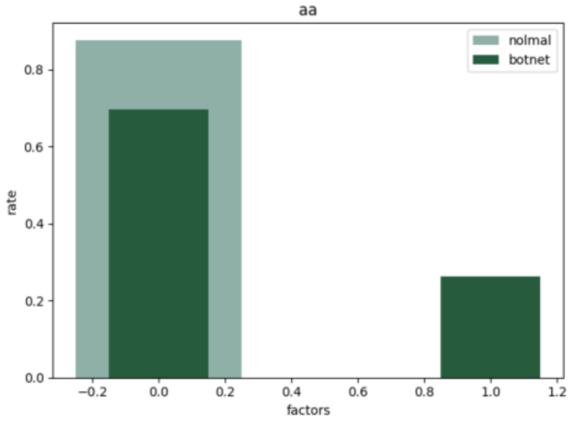


Figure 2. comparison of AA

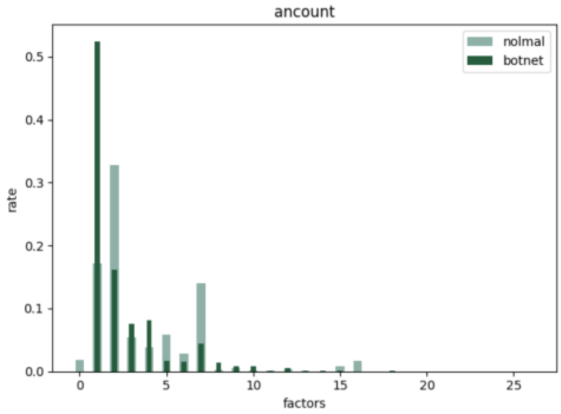


Figure 3. comparison of ANCOUNT

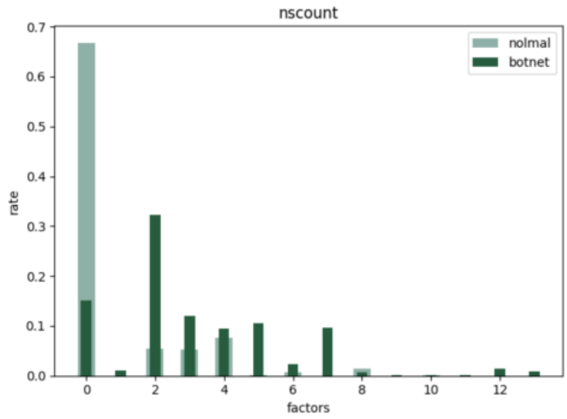


Figure 4. comparison of NSCOUNT

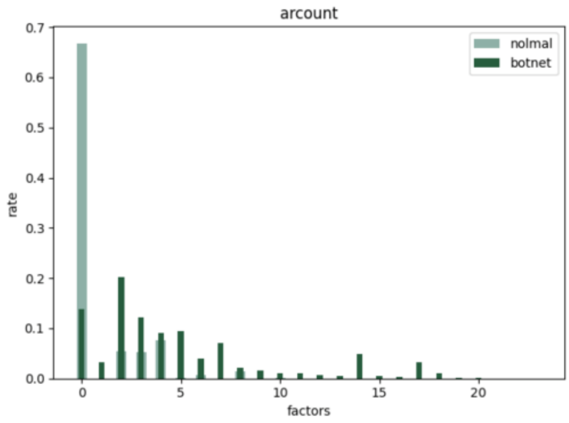
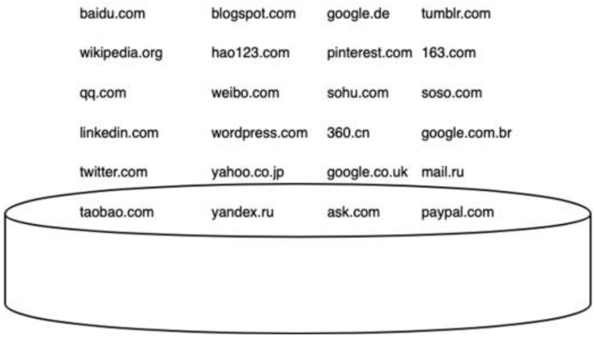


Figure 5. comparison of ARCOUNT

Compared with normal traffic, in the Fast-flux botnet, the proportion of the AA value of 1 reaches 30%, while the AA value in the normal network is all 0. Compared with normal traffic, the ANCOUNT number distribution of Fast-flux botnets is generally smaller. At the same time, we can also see from the figure that the normal network and the Fast-flux botnet also have a big difference in the NSCOUNT and ARCOUNT fields.

2.2. Filter Module

The filtering module mainly uses domain name whitelist technology. During the experiment, we found that some large websites often use CDN technology in order to maintain their connectivity. In order to prevent the model from determining this part of the domain name as a Fast-flux botnet, we have added a domain name whitelist as a filter, and some white domain list is shown in Figure 6.



baidu.com	blogspot.com	google.de	tumblr.com
wikipedia.org	hao123.com	pinterest.com	163.com
qq.com	weibo.com	sohu.com	soso.com
linkedin.com	wordpress.com	360.cn	google.com.br
twitter.com	yahoo.co.jp	google.co.uk	mail.ru
taobao.com	yandex.ru	ask.com	paypal.com

Figure 6. Domain whitelist

2.3. Information Collection Module

The function of this part is mainly to extract the required data in the DNS response packet, and perform data processing and processing on it to obtain numerical data. Need to cooperate with Wireshark.

Through the experiments of Holz[1], Zou[2], etc. and the characteristics of the Fast-flux botnet, we know that the Fast-flux botnet has a large IP pool, so this module returns the number of IPs for each response packet, and the corresponding domain name The number of IPs ($total_{ip}$) is counted.

But unlike the Fast-flux botnet, although normal networks have multiple IP addresses, their IPs do not change frequently. That is, in the next visit, the IP address in the response packet has not changed. The total amount has not increased. The Fast-flux botnet will not only return multiple IPs in one response, but also return data different from the previous one in the next response. In order to distinguish between the normal network and the Fast-flux botnet, we introduced the circulation φ as one of our characteristics. Through the processing of this module, we will get the value of the circulation φ.The value of the circulation φ is: the number of IPs known so far/the number of IPs known last time.

At the same time, according to the characteristics of the Fast-flux botnet, in order to detect the Fast-flux botnet using Double-flux technology, in this module, we also set the TTL value in the ANSWER field and the Authoritative Nameserves (hereinafter referred to as NS) field The TTL value of and the TTL value in the Additional Records (hereinafter referred to as AR) field are processed according to the data returned in the automatic update module.

The reason why we did not deal with other information in the NS field and AR field is because other papers have found that these two fields have little weight on the experimental results. In addition, the information collection module counts the NSCOUNT and ARCOUNT that behave differently in the normal network and the Fast-flux botnet during the preparation phase.

2.4. Detection Model Module

In this article, we have selected the following features.

AA, the DNS response packet indicates that the responding server is an authorized resolution server that queries the domain name.

n_{ip}, the number of different IP addresses returned by each DNS response packet (in the rdata in the ANSWER field).

$total_{ip}$, the number of different IP addresses corresponding to each domain name.

φ, the circulation φ is equal to the number of IPs known so far/the number of IPs known last time.

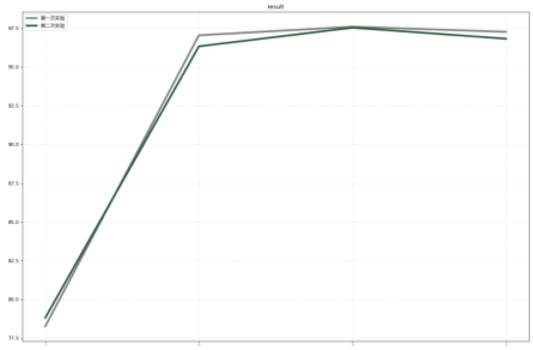
TTL_{AN}, TTL_{NS}, TTL_{AR}, the information collection module processes the TTL returned in the AN ,NS,ARfield into [0,5] levels.

$NSCOUNT$, a field value representing how many domain name server information there are in the DNS response packet.

$ARCOUNT$, a field value representing how many additional information is in the DNS response packet.

N_{ASN}, represents how many different asn codes are in $total_{ip}$.

By comparing the four machine learning methods of decision tree($x = 1$), KNN($x = 2$), random forest($x = 3$), and ExtraTree classifier($x = 4$), (as shown in Figure 7), we found that the effect of the ExtraTree classifier set and the random forest are comparable, the best, and in other experiments The better-performing decision tree in the medium performs poorly.



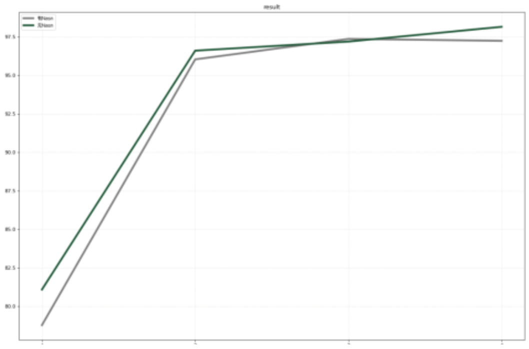


Figure 7. Comparison of Methods

Figure 8. Comparison of Model

Therefore, we chose random forest as our judgment model.

In order to compare other experiments, we introduced the N_{ASN} value in the established model, but found that the detection accuracy did not change much, which means that our feature selection is more feasible and has a high accuracy rate, achieving a lightweight model. We show the changes in the detection results after the introduction of N_{ASN} in Figure 8.

Through comparison, we know that the method in this article has high accuracy while being adaptable and portable.

2.5. Automatic Update Module

The module will return the result of the classification to the data collection module.

First, the module extracts the TTL values of these three fields, counts and analysis the data (as shown in Figure 9, 10, 11). Divide the interval, starting from 0, using 1000 as the increasing interval, and ending at 10000. Through experiments, it is found that the TTL value distribution of normal network and Fast-flux botnet is shown in the figure.

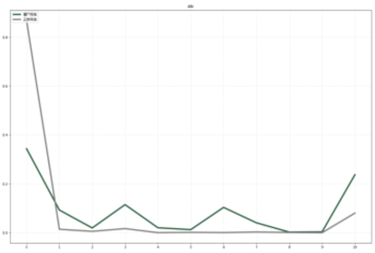


Figure 9. Comparison of TTL in AN

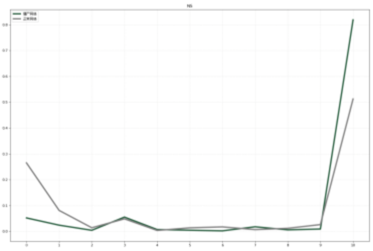


Figure 10. Comparison of TTL in NS

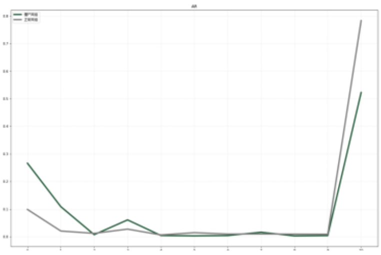


Figure 11. Comparison of TTL in AR

From the experiment, we can see that the distribution and proportion of normal networks and Fast-flux botnets are quite different. However, in the real network environment, the technology of Fast-flux botnets is constantly updated. Although they will essentially adopt Single-flux technology or Double-flux technology, they will also update their technology to prevent detection. Therefore, when TTL value data is processed, we do not directly fix a value to divide the basis, we will dynamically update our divided value for detection. Obtain a division value suitable for the current network environment, so that the detection model can achieve higher accuracy. In the initialization process, the module uses 5000 as the division value, and bring this division interval into the experiment, and the result passes the detection model, it turns out that the effect is not ideal. Therefore, in the automatic update module, it will modify the next distinguishing value to 1000, 2000. The module returns this result to the information collection module. Obtaining the new division result, we will find that the accuracy rate has been improved to a certain extent. In this data set, the model finally chose distinguishing value of 1000, 2000.

3. Experimental Environment

This experiment uses the ISOT data[6] set and part of the CTU-13 data set. The Waledoc botnet in the ISOT data set uses Fast-flux technology and normal DNS traffic in CTU-13.

Type	CTU-13	ISOT
Normal	3583	0
Botnet	0	9966

4. Experimental Results

The experimental data shows (as shown in Figure 12) that the data results are optimized. With the continuous development of Fast-flux botnet concealment technology and control methods, traditional detection methods require a lot of resources, and there are some uncertain factors. Therefore, this paper proposes a brand-new feature AA, which removes the feature N_{ASN}, which requires a lot of resources and has uncertain factors, and becomes a more stable, lightweight, and adaptable detection model. At the same time, due to the automatic update of the module, the current traffic situation in the network will be fed back in time, so the accuracy rate will be greatly improved, and the traffic changes in the network can be better adapted. The experimental data table maze, the stable, lightweight, automatically updated, and more adaptable model in this article has an accuracy rate of 98%, which has a higher accuracy and precision.

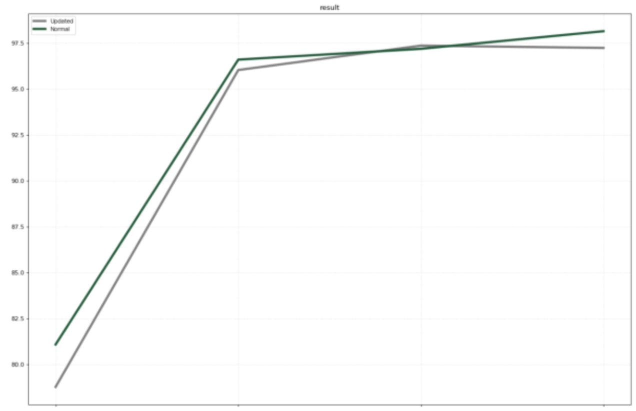


Figure 12. The result of Update

5. References

[1] Holz T, Gorecki C, Rieck K, et al. Measuring and Detecting Fast-Flux Service Networks[J]. Ndss, 2008, 1(5):487 - 492.

[2] Zou, Futai, Zhang, et al. Hybrid detection and tracking of fast-flux botnet on domain name system traffic[J]. Communications China, 2013.

[3] Fast-flux Botnet Detection Method Based on Spatiotemporal Feature of Network Traffic.

[4] Linan Huang, Quanyan Zhu. A dynamic games approach to proactive defense strategies against Advanced Persistent Threats in cyber-physicAl systems[J].Computers & Security,2020,(89)

[5] Assadhan B, Bashaiwth A , Al-Muhtadi J , et al. Analysis of P2P, IRC and HTTP traffic for botnets detection[J]. Peer-to-Peer Networking and Applications, 2018.