**ISBN** 978-981-11-7861-0
**Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)**
*Bangkok, 28-30 June, 2018, pp.417-422*
doi: 10.18178/wcse.2018.06.073

# HRB-EM Fast-Flux Scoring Model

Hamid Reza Bolhasani [1], Ebrahim Mahdipour [1+]

[1] Department of Computer Engineering, Science and Research Branch Islamic Azad University, Tehran, Iran

**Abstract.** In the recent years, DNS misuse is increasing significantly and domain name system due to its special properties, has become one of the most attractive areas for the hackers and botmasters. Current botnets are using two main strategies for intrusion and getting C&C of the victim's machines: DGA (Domain Name Generation Algorithm) and Fast-Fluxing. In this paper, we propose a new model for detecting fast-flux service networks which we called it HRB-EM. The accuracy of this model is 99.71% with 0.86% false positive and 0.09% false negative rate.

**Keywords:** Network security, botnet, DNS, fast-flux

## 1. Introduction

With tremendous growth of internet and rapid development of network interconnections, connected devices and broadband services, information and network security has become one of the most serious challenges of organizations and an important research field with much more concern [1]. Thus we should have a deeper understanding of network threats for detecting and defending against them. Viruses, worms, Trojan horses, malwares and bots are among the threats that network security administrators worry about [2].

A bot is a type of malware, which can get command, and control of the infected computer. A Botnet is a large collection of these compromised machines (bot) for doing some malicious activities such as distributed denial of service (DDoS) attack, steal data, spam, key-logging, click-fraud, etc.

Over the recent years, many botnets exploit DNS for their destructive purposes. First case of these attacks found in 2007, which its name was Srizbi. Since 2015, more than 30 other cases have been discovered [3]. These DNS misuses can be categorized in two main classes: DGA and Fast-Fluxing. In DGA method, each bot based on its algorithm generates a large collection of domain names and query them until one of them could be resolved. After successful resolving of the name to IP, botmaster starts connecting to the infected computer and get its control. One of the advantages of this method for hackers is that they can't be stopped by detection of malicious IP addresses on client side because botmasters consequently can change their IP addresses. Conficker is one of these first botnets that found in 2008. Its algorithm generates domain names with a relatively high rate. Another bot is Torping, which uses twitter topic trends for generating random names as domain. Some other similar bots utilize English pool of suffixes, verbs, and nouns to make these domain names [4].

Fast-Fluxing is another well-known method of DNS exploitation that is our main concern in this research, so we discuss deeper with more details about it. In this method, one single domain is used in the malware and its related IP addresses are changed frequently by the botmaster. One of the major limitations of fast-flux service networks is that if the domain got detected by victim's security systems, communication channel of botmaster will also blocked consequently and thus the malware would not be functional more. In Figure 1, you can find a brief overview of this pattern and related events.

[+] Corresponding author. Tel.: + 989126110260; fax: +982144865154
*E-mail address*: mahdipour@srbiau.ac.ir

As shown in Figure 1, in first step (1) victim's computer will be infected by the malware through an email spam, infected hard disk or some other ways. This malware contains some data such as the domain name that is used for DNS misuse. In the next step (2), malware starts doing DNS query for its containing domain name. (3) This request will go to Flux agent and consequently (4) to botmaster. Then (5) botmaster will return its IP address and (6), (7) will delivered to the victim via DNS protocol. In (8) botmaster eventually will get the C&C of the infected computer. If IP address of botmaster got detected by the attacked computer, botmaster can change it and continue doing its malicious goals. Since IP address of botmaster may be detected by victim's security software and thus botmaster decide to change it, usually a low value of TTL will assigned to them for preventing DNS query caching.

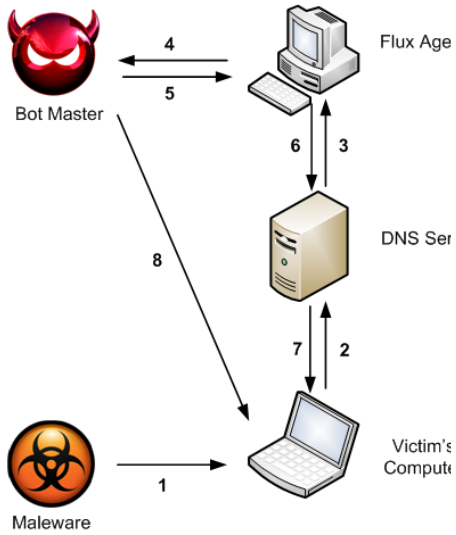


Fig. 1: Fast-flux service network architecture.

There are two types of Fast-Flux service network: Single-Flux and Double-Flux. Single-Flux networks change the DNS record for their front-end node IP addresses often as every 3 to 10 minutes' time intervals, which leads to frequently changing A records in each DNS query. Double-Flux service networks are more complicated in comparison of single-flux ones because of one additional redundancy layer. Actually, in double-flux service networks, both the DNS A record sets and authoritative NS records for a malicious domain continually changed in a round-robin manner. For better understanding, DNS query results of one single-flux and double-flux domain in two attempts with 30 minutes' intervals are compared and presented in Figure 2 and Figure 3 [5].

```
                                                    ;; WHEN: (~30 minutes/1800 seconds later)
divewithsharks.hk. 1800 IN A 24.85.102.xxx          divewithsharks.hk. 1800 IN A 24.85.102.xxx
divewithsharks.hk. 1800 IN A 69.47.177.xxx          divewithsharks.hk. 1800 IN A 69.47.177.xxx
divewithsharks.hk. 1800 IN A 70.68.187.xxx          divewithsharks.hk. 1800 IN A 70.68.187.xxx
divewithsharks.hk. 1800 IN A 90.144.43.xxx          divewithsharks.hk. 1800 IN A 90.144.43.xxx
divewithsharks.hk. 1800 IN A 142.165.41.xxx         divewithsharks.hk. 1800 IN A 142.165.41.xxx

divewithsharks.hk. 1800 IN NS ns1.world-wr.com. divewithsharks.hk. 1800 IN NS ns1.world-wr.com.
divewithsharks.hk. 1800 IN NS ns2.world-wr.com. divewithsharks.hk. 1800 IN NS ns2.world-wr.com.

ns1.world-wr.com.  85248 IN A 66.232.119.xxx    ns1.world-wr.com.  85248 IN A 66.232.119.xxx
ns2.world-wr.com.  82991 IN A 209.88.199.xxx    ns2.world-wr.com.  82991 IN A 209.88.199.xxx
```

Fig. 2: Two consecutive DNS query for a single-flux domain with 30 minutes' interval.

As shown in Figure 2, two new IP addresses has been resolved in second query which has been highlighted but no new name server is seen. Based on what we discussed earlier, these are characteristics of single-flux. While in double-flux case which is presented in Figure 3, some IP addresses of both A (web serving hosts or hosts for the target) and NS (authoritative name server for the domain) records are new in the second query. This provide a more dynamic environment for the hackers or criminals that want to be

resilient against security detection systems. For these type of flux networks, domain registrars should allow webmaster the ability to change its name servers frequently, which is not something normal and usual [5].


Fig. 3: Two consecutive DNS query for a double-flux domain with 30 minutes' interval

## 2. Related Works

Honeynet project [5] was the first comprehensive research for detecting behaviour and threats of FFSNs[1].

Holz et al. [6] presented the first practical work for measuring and detecting fast-flux service networks. In this research, they have done deep analysis on fast-flux networks behaviours and discovered related features. Using SVM[2] chunking algorithm, for each feature a weight is stated, thus the result is a formula for measuring fluxiness of a domain as you can find it below as in (1).

$$f(x) = 1.32\, n_A + 18.54\, n_{ASN} + 0\, n_{NS} \tag{1}$$

where $n_A$, $n_{ASN}$ are respectively number of unique A records and related unique autonomous system numbers (ASNs) that are resolved in DNS query.

$$\begin{gathered} f(x) - b > 0 \quad if\ x\ is\ Fast - Flux \\ f(x) - b \leq 0 \quad if\ x\ is\ Benign \\ b = 142.38 \end{gathered} \tag{2}$$

According to (2) we can find out whether a domain is benign or fast-flux. Based on the paper's statement, detection have been done with accuracy of 99.98% but the limitation is that the coefficients might need to be modified if some new behaviours of FFSNs appear in the future.

Bilge et al. [7] introduced a system called EXPOSURE, which can detect malicious domains based on their patterns. Four feature sets including 15 features has been used in this research that 9 of them were novel and had not been proposed before. These feature sets are:

F1: Time-Based Features

F2: DNS Answer-Based Features

F3: TTL Value-Based Features

F4: Domain Name-Based Features

[1] Fast-Flux Service Networks
[2] Support Vector Machine

Among mentioned features, F4 shows the highest error rate in detection and $F_{all}$ which utilizes all the features for recognition had the lowest error rate.

Hsu et al. [8] presented a novel approach for discovering fast-flux networks based on response time differences. This research has opened a new area regarding this topic. The most brilliant part of this work is that it has focused on time response differences, which is introduced for the first time. In this research, it is supposed that fast-flux networks do not use load-balancing technic for using dynamic DNS, and usually they compromise hosts that are connected to internet with relatively medium to low speed such as ADSL. Since these compromised computers are not dedicated servers and most of the times they are some standard personal desktop or laptop, and because of this, they are running many other applications simultaneously, so response time for each requests that are sent to them with a high probability may change. Therefore, response time differences would be a good metric for scoring fluxiness. Fast-Flux Score for a domain ($dom_e$) between time interval $t_s$ and $t_e$ is calculated based on this metric and some other metrics, which have been extracted from previous researches as in (3).

$$FF - Score_{t_s}^{t_e}(dom_e) = \emptyset_{t_s}^{t_e} \times (\alpha_{t_s}^{t_e} + \beta_{t_s}^{t_e}) \tag{3}$$

where $\emptyset_{t_s}^{t_e}$ is equal to:

$$\begin{array}{l} 0 \quad if \ |\gamma_1 \cup \gamma_2 \cup \ldots \cup \gamma_n| \leq 3 \\ else \ \ |\gamma_1 \cup \gamma_2 \cup \ldots \cup \gamma_n|/|\gamma_1| \end{array} \tag{4}$$

And $\gamma_i$ is set of A records that have been resolved in $i^{th}$ query.

$$\alpha_{t_s}^{t_e} = srtd_{t_e}^{t_e}(dom)_e/[(r-1) \times i] \tag{5}$$

$$srtd_{t_e}^{t_e}(dom)_e = \sum_{i=1}^{k} srtd_{t_s}^{t_e}(IP_i) \tag{6}$$

$$srtd_{t_e}^{t_e}(IP)_k = \sum_{i=2}^{r} \left|\rho_{ki} - \rho_{k(i-1)}\right| \tag{7}$$

Which srtd means sum of response time differences for a specific domain ($dom_e$) between time interval $t_s$ and $t_e$. And $\rho_{ki}$ is the time response of the $i^{th}$ request related to the $IP_k$.

Schales et al. [3] cooperated for another innovative research in this area that is published recently in 2016. This scientific work mainly focuses on DNS analysis of Big Data and uses two main algorithms for detecting DGA and Fast-Flux Networks: MapReduce and Feature Collection and Correlation Engine (FCCE). In this research, a comprehensive survey on DGA and FF methods is presented. Precision of the mentioned algorithms is about 82% that is noticeable in case of Big Data.

## 3. HRB-EM Fast-Flux Scoring Model

HRB-EM is a new scoring model for fast-flux networks, which is made from authors of this work's names. We design this model based on our analysis and results of related works, which have been introduced in the previous section. For designing this model, first we considered the features that has direct relation with fluxiness of a domain:

- Number of unique A records in DNS query ($n_A$)
- Number of unique Time Zones ($n_{TZ}$)
- Number of unique ASNs ($n_{ASN}$)
- TTL Value ($v_{TTL}$)

Feature selection is the process of choosing a subset of the original feature spaces according to discrimination capability in an affect to improve the quality while reducing the dimensionality of data [9].We have done our analysis based on three datasets. For considering benign domains, we used Alexa [10] 100,000 top domains and for fast-flux networks, 25,000 domains from ATLAS and DNSBL databases are considered [11], [12].

Since in Holz Model [6] Number of unique ASNs ($n_{ASN}$) is used, in this research, to check the problem from another point of view and in order of opening new challenge, we utilize Number of unique Time Zones ($n_{TZ}$) as a new metric. Comparing these two parameters shows that value of them for fast-flux networks are almost similar as shown in Figure 4.

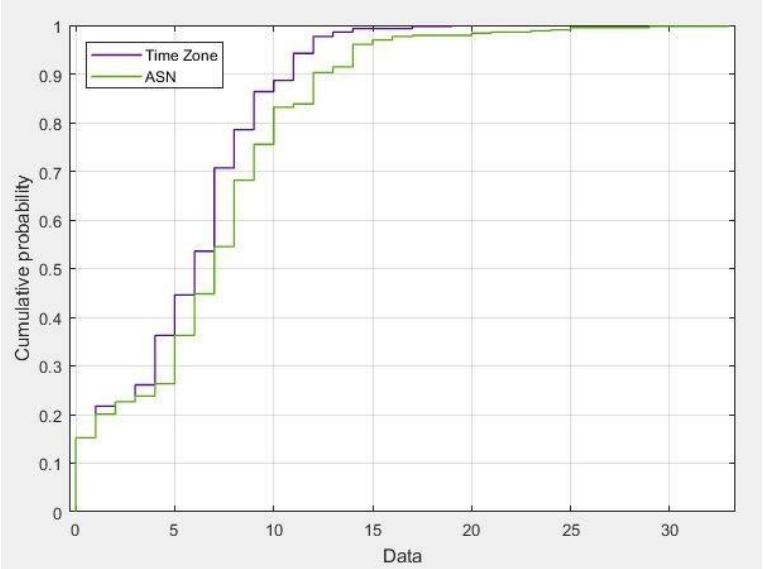

Fig. 4: Comparing number of time zones and ASNs in fast-flux networks.

As shown in Figure 4, comparing Number of unique Time Zones and ASNs clarify that they are approximately similar but for this work, we utilize the first one in our model ($n_{TZ}$).

Another mentioned metric is value of TTL. For this metric, we did analysis on TTL value of about 25,000 Fast-Flux domains and compared them with benign domains. Setting low TTL is mostly for two different conditions: CDNs, which uses Round-Robin DNS to make their availability of services as much as possible and on the other hand, malicious packets that are made for attacks or intrusion [7]. Hackers usually set the TTL of their packet as low as possible because the probability of detecting and put them in DNS black lists become very low.

We collected TTL of all Fast-Flux Domains from our dataset. After preprocessing and removing outliers from it, eventually average value equal to 186.18 has been achieved. With considering average TTL of 190, now we can finalize all of our findings and present our scoring model in (8).

$$S_{ff}(d_i) = \emptyset_i(w_1.n_A + w_2.n_{TZ} + \lfloor 190/v_{TTL} \rfloor) \tag{8}$$

where $S_{ff}$ ($d_i$) is Fast-Flux Score for a specific domain i. And $\emptyset_i$ *is* equal to:

$$0 \quad if \ |\gamma_1 \cup \gamma_2 \cup \dots \cup \gamma_n| \leq 3 \tag{9}$$
$$else \ \ |\gamma_1 \cup \gamma_2 \cup \dots \cup \gamma_n|/|\gamma_1|$$

Which $\gamma_j$ is number of unique A records in each distinct set of DNS queries of a specific domain. For calculating $w_1$ and $w_2$ which are coefficients of $n_A$ and $n_{TZ}$, we solved the problem through Neural Networks by MATLAB and find these values for them: $w_1$= -4.98, $w_2$= 10.08. Thus eventually we can state (10) as our final formula for calculating Fast-Flux score.

$$S_{ff}(d_i) = \emptyset_i(-4.98\, n_A + 10.08\, n_{TZ} - 1.20 + [190/v_{TTL}]) \quad (10)$$

Calculating $S_{ff}(d_i)$ for any domain name is very easy and doesn't need some complex tools or conditions. If its value for a domain equals to zero or less, it can be considered as benign, otherwise is a fast-flux.

## 4. Model Evaluation and Results

We developed a Python program, utilizing our final formula for detecting whether a domain is benign or fast-flux, and benchmarked it with our dataset. Result shows that this method works and the accuracy was 99.71% with 0.86% false positive and 0.09% false negative rate, which in details are presented here:

TP=11,593 FP=38 TN=4,375 FN=4

False Negative Rate (%) =0.091%

False Positive Rate (%) =0.86%

$$Accuracy = \frac{TP + TN}{TP + FP + FN + FP} = 99.71\%$$

## 5. Conclusion

In this paper, two main approaches for DNS misuse have been introduced and then Fast-Flux method is deeply studied. We considered almost all related works and introduced our scoring model. We propose HRB-EM model based on significant results of related works and also our finding in this research. This model is different from similar works because of using number of unique time zones and value of TTL as main parameters in flux scoring of a domain. Fortunately, evaluation results show that this model works with accuracy of 99.71% and 0.86% false positive and 0.09% false negative.

## 6. References


[1] Yang Xia Luo, "Malicious detection based on relief and boosting multidimensional features," in Journal of Communications, VOL. 10, NO. 11, Nov 2015.

[2] D. Geer, "Malicious bots threaten network security," in *Computer*, vol. 38, no. 1, pp. 18-20, Jan. 2005.

[3] D. L. Schales et al., "Scalable analytics to detect dns misuse for establishing stealthy communication channels," in IBM J. RES. & DEV. VOL. 60 NO. 4 PAPER 3 JULY/AUGUST 2016.

[4] Sandeep Yandav, Ashwath K. K Reddy, and A.L. Narasimha Reddy, "Detecting algorithmically generated malicious domain names," in IMC'10 , November 1-3, 2010, Melbourne, Australia.

[5] William Salusky, Robert Danford, "Know your enemy: fast-flux service networks," in Naperville, IL, USA, July 2007. [Online] Available: http://www.honeynet.org/papers/ff/

[6] T. Holz, C. Gorecki, K. Rieck, and F.C. Freiling, "Measuring and detecting fast-flux service networks," in Proc. 16th Annu. Netw. Distrib. Syst. Security Symp., Feb. 2008, pp. 1-12.

[7] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: finding malicious domains using passive dns analysis," in Proc. 18th Annu. Netw. Distrib. Syst. Security Symp., Feb 2011, pp. 1-17.

[8] Fu-Hau Hsu, et al., "Detect fast-flux domains through response time differences," in IEEE JOURNAL ON SELECTED AREA IN COMMUNICATIONS, VOL. 32, NO. 10, OCTOBER 2014.

[9] Jingping Song, Zhiliang Zhu, and Chris Price, "Feature grouping for intrusion detection based on mutual information" in Journal of Communications, VOL. 9, NO. 12, Dec 2014.

[10] Alexa – The Web Information Company. [Online]. Available: http://www.alexa.com

[11] ATLAS. Global fast flux, Burlington, MA, USA. [Online] Available: http://www.atlas.arbor.net/summary/fastflux

[12] DNSBL. http:// dnsbl.abuse.ch/fastfluxtracker.php