I. J. Computer Network and Information Security, 2016, 10, 37-44
Published Online October 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.10.05

TempR: Application of Stricture Dependent Intelligent Classifier for Fast Flux Domain Detection

Prabhjot Singh Chahal
M.Tech Student, Central University of Punjab, India
E-mail: prabhchahal.19@gmail.com

Surinder Singh Khurana Assistant Professor
E-mail: surinder.seeker@gmail.com

***Abstract*—**Fast-flux service networks (FFSN) helps the cyber-criminals to hide the servers used for malicious activities behind a wall of proxies (bots). It provides the reliability and detection evasion to a malicious server. FFSN use a large pool of IP addresses for proxies. Detection of FFSN is difficult as few benign technologies like Content distribution networks and round robin DNS have similar working characteristics. Many approaches have been proposed to detect FFSN and fast flux domains. However, due to dynamic behavior of FFSN, these techniques suffer from a significant number of false positives. In this paper, we present a Temporal and Real time detections based approach (TempR) to detect fast flux domains. The features of fast flux domains and benign domains have been collected and classified using intelligent classifiers. Our technique illustrates 96.99% detection accuracy with the recent behavior of fast flux domains.

***Index Terms*—**Content Distribution Network, Domain Name System, Fast-flux Networks, Machine learning, Botnet, Malware.

I. Introduction

Malware is not a new term in the computer world. Malware cause damage to software, data, and other computer resources. Malwares are not just limited to mentioned damage; some malwares have sophisticated capabilities like providing remote access to the infected system. A large pool of such infected systems executes the commands given by attacker and such pool is known as a botnet. These botnets perform a number of cyber-crime activities including distributed denial-of-service (DDoS) attacks, spam, phishing, and identity theft [1]. Any person who has the control over botnet is called botnet master. Some of the cyber-crimes rely on the botnet infrastructure. Botnet masters use these botnets for rent and pay per services. These botnets can be used as Fast flux service networks on pay per use basis.

Fast flux service network (FFSN) is the refined application of botnets. Fast flux service networks use a DNS technique named Fast flux to hide phishing and malware delivery servers behind an attacker controlled network of bots acting as proxies [2]. In fast flux technique, DNS records of the domain (both DNS A record and NS record) which associates the domain to bots are changed rapidly. These rapid changes make it difficult to track and block such criminal operations. The domain name of a website hosted behind fast-flux service network is called Fast-flux domain. FFSN exploits a network of bots to conduct illegal activities such as spam, phishing, illegal content hosting and other malicious activities using DNS record manipulation techniques.

The remainder of this paper is organized as follows: Section II to V explains Fast Flux domains, their significance & types and similar technologies related to Fast Flux domains. Section VI reviews the related work. Section VII presents the proposed TEMPR approach. Section VIII & IX explains features used for classification and collected dataset. Section X analyses the performance. Section XI compares the performance of TEMPR approach with existing approaches. The facts we observed during the work are mentioned in section XII. Finally Section XIII concludes the paper.

II. Why Fast-flux Domains are Used?

Fast-Flux Service Network (FFSN) architecture has been used to increase the productivity, availability and to extend the lifetime of domain names linked to the fast-flux service networks [3]. The reasons behind the use of FFSN are:

A. Frequent and Dynamic Resolution of Domain Names to a large Pool of Ip Addresses

A domain name resolute to a large pool of IP addresses rather than a single IP address. So a domain name is always resolved to a controlled and live flux agent, which ensures the availability of mothership server all the time. If a flux agent is detected, the attacker can abandon the detected flux agent without disrupting the malicious services hosted by mothership server.

B. Indirect Connection between Mothership Server and Victim

In FFSN, there is no direct connection between the victim and mothership server. This architecture provides few benefits such as in any case authorities manage to locate the bot, the mothership server will remain intact and hidden. The attacker can use another bot as a proxy to forward the traffic and mothership server can stay online. Flux agent produces fewer anomalies than traditional bots so these are less susceptible to detection.

C. Better Management of Mothership Server

Another benefit is better management of mothership server. An attacker can host the malicious content on a dedicated server. It is also easy and reliable to manage the server instead of multiple bots hosting malicious services. Attackers have physical access to mothership server so they can maintain the server in a better way.

III. Benign Technologies Similar to FFSN

Some aspects of FFSN exhibits the characteristics like legitimate technologies like RRDNS and CDNs. These characteristics include returning multiple IP addresses in response to a DNS query and short TTL values of DNS records [4].

A. Content Distribution Network

A content delivery network (CDN) is a network of distributed servers (network) used to improve accessibility, maximize bandwidth, and maintain correctness. It provides reliable and fast services by distributing content to cache or edge servers located close to users. Content Delivery Networks provide services with enhanced utilization and balanced load[5]. CDN is a combination of content-delivery, request-routing, distribution, content management services and accounting infrastructure. These networks are effective in high-speed delivery of content of websites with high traffic and large user pool. To return the IP addresses of the best available servers to a client request, a CDN utilizes sophisticated techniques to compute information such as network topology and link characteristics. When responding to a DNS lookup, it returns multiple DNS A records. A low TTL value is employed by CDN to enable them to react quickly to changes in link characteristics.

B. Round Robin DNS

Round Robin domain name system is a technique typically used for load balancing, and fault tolerance. It works by responding to a client's DNS query with a list of A records instead of a single A record to provide multiple, redundant IP service hosts. The list of A records of RRDNS domain is cycled in a round-robin manner for consecutive queries [6]. Therefore, a series of queries to RRDNS domain are directed to different geographically distributed servers [7] and thus effectively balance the load.

IV. Why Benign Domains are Mistaken as Fast Flux Domains?

Both CDN and RRDNS exhibit the same characteristics such as low TTL, multiple IP addresses in DNS A record and geographic dispersion of these IP addresses. Since most of FFSN detection techniques rely on these characteristics and these benign domains may show behavior similar to FFSN domains. So, an effective fast flux domain detection technique must consider some other features to differentiate benign domains from fast flux domains.

V. Types of Fast-flux

Fast flux service networks are growing at a rapid rate, and changes to known fast flux mechanisms provides more lifetimes to these fast flux domains. Based on the different combinations of change of name server records and change of associated IP addresses classify the fast-flux mechanism into following three types:

A. Single Flux

Most basic type of flux is single flux. In single flux a domain is resolute to a different IP address. Only IP addresses related to domain, change frequently [8]. A bot or server serves as name server for the domain. The name server remains a weak point. If zone file from name server is changed, this will create the problems for domain resolution. Single-flux service networks change the DNS records for their front end node IP address as often, so even if one flux-agent node is shut down, many other infected bots are standing by and available to quickly take its place.

B. Double flux

In double flux, IP addresses associated with the domain as well as name servers are frequently changed [9]. The name server zone file is loaded on several bots. These bots serve as a name server for fast flux domains. Double-flux networks are complicated and provide an extra level of redundancy.

C. N-Level flux

N-level flux is the latest observed trend in fast-flux service networks. Instead of frequently changing name server, N-level flux uses n long chain of name servers. The name server domains are like ns*.ns*.ns*... [9].

VI. Related Work

There are many techniques to detect whether a domain is a fast flux domain or not. These techniques use different characteristics possessed by a FFSN to identify them. There are a number of ways a Fast Flux domain can evade these detection techniques. These techniques use a set of features extracted from DNS responses to identify the fast flux domains.

Most of the techniques use a set of features and classification of obtained data by machine learning

algorithms to reach a conclusion whether the domain is fast flux or not. The methods for collection of data may be active, passive or a combination of both. On the basis of features used by detection techniques, these can be classified into two types: temporal based techniques and Real time techniques.

A. Temporal Based Techniques

Temporal based methods passively observe DNS query responses for a specific time period. During this period, characteristics such as the heterogeneity of IP addresses or autonomous system numbers from several DNS queries are recorded. The suspected domains are monitored by querying DNS A records for a time longer than TTL or more than this. All the IP addresses obtained from A records are stored. If the observed parameters reach their threshold value, the suspected domain is declared fast flux domain. Though these temporal-based characteristics match the basic behavior of FFSN and provide good detection accuracy, they also introduce considerable detection delay because observation requires at least one Time to Live (TTL) period. These techniques suffer from long detection time periods probably few days. In such long period, attacker may change their domain. FluXOR [3] and Flux-Score [7] are temporal based detection techniques. Flux-Score based technique uses temporal-based characteristics and spatial features to measure the extent of this threat. To provide a deterministic decision, a general metric named Flux-Score was proposed to count the number of

- Unique A records in overall DNS lookups,
- NS records in a single DNS lookup, and
- Unique Autonomous System Numbers (ASNs) for overall A records.

If the Flux-Score value is positive, the domain is classified as a FFSN domain; otherwise, the domain is considered benign. Although Flux-Score was considered highly accurate, yet the temporal-based characteristics require at least detection time equal to of TTL period. It especially takes a long time when detecting benign domains with characteristically long TTLs. FluXOR aimed to reduce the latency in detection of fast flux domains. It uses nine different features as detection parameters. The FluXOR has three principles: domain of the suspected hostname, degree of availability and heterogeneity of the hosts of the target network. It worked better than Flux-score. But in FluXOR time limit for filtering benign domains was 3 hour that introduce a significant delay. If any domain had larger time threshold, it would be classified as benign domain. New trends in fast flux service networks like high TTL and few IP addresses in DNS A record makes temporal based techniques less effective.

B. Real time detection techniques

Since temporal based techniques take long detection time, network security system can't afford such long time. There may be enough damage during the detection period. So to overcome these issues real-time detection techniques were suggested. These techniques take only a few seconds to detect fast flux domains. These are more effective than the temporal based techniques, but there is concerning false positive rates. Because some CDN have similar behavior like FFSN, so sometimes these can be marked as Fast flux domain. In [10], Huang et al. presents a real-time detection system, named Spatial Snapshot Fast-Flux Detection (SSFD) system, for identifying fast flux domains. The design principle of SSFD was based on spatial features that capture the dispersed nature of FFSN IP addresses. SSFD utilises two spatial distinguishing approaches comprised of spatial distribution estimation and spatial service relationship evaluation. Although the results showed a high accuracy rate and low detection time, SSFD often fails because the required geographic information was not always available. This approach is fully dependent on the geographic information that can subject to legal issues in few provinces, so its implementation is not feasible. SSFD is also not able to detect fast flux domains in case of dynamic DNS [11]. This issue significantly limits the effectiveness of this scheme in detecting FFSNs.

In [12], Hsu et al. focused on the proxy-based architecture of Fast flux service networks and proposed real-time detection system based on the network delay features collected by checking the responses of the fast flux domains. Since bots have limited resources, it puts significant delays in responding to users. Authors proposed a Fast-Flux Bot Detection (FFBD) approach based on features such as delays in fetching documents such as web page, variable network delays, and long processing delays. This system has low detection delays and average accuracy. This system is not effective as the network congestion can affect these features. Any network failures, congestion or denial of service can be classified as fast flux domains.

In [7], Lin et al. proposed a real-time detection system based on genetic algorithms. Authors used two new characteristics called the Entropy of Domains of Preceding Nodes (E-DPNs) and the Standard Deviation of Round Trip Time (SD-RTT). Later is a spatial based feature. It also uses number of ASN's and number of IP address in a single lookup. This system has better performance than temporal based techniques and other real time based systems. Authors acknowledged two different scenarios where fast flux domains can evade this detection mechanism. In the first case, it is not effective to detect domains returning single IP with TTL=0. Another is related to the geographic dispersion of IP addresses. Domains, with close geographical locations of IP addresses, can be misclassified. Another factor that reduces the effectiveness of the technique is related to Genetic algorithm. Genetic algorithms provide good accuracy, but are very complex and take significant time to build data models.

C. Hybrid detection techniques

Temporal based techniques and real-time detection techniques have their own pros and cons. To overcome

the limitations of both hybrid techniques have been developed. These techniques use both temporal based mechanism and real-time mechanisms for fast flux detection. Firstly, the suspicious domains are identified by using real-time mechanisms. Then these suspicious domains are monitored to collect DNS information for a long period such as two days using temporal based techniques. If any suspicious domain shows fast flux behavior in 2nd stage, it will be classified as a fast-flux domain; otherwise it will be whitelisted [4].

VII. TempR Approach to Identify Fast-flux Domains

To overcome the problem of false positive with existing detection technique, this work presents an efficient approach called TempR to Identify Fast-flux Domains. As represented in Fig 1. TempR has divided into two stages:

1) The first stage uses real-time detection approach. If any DNS A record has TTL=0, the domain will be declared suspicious. If TTL is not equal to 0, number of IP addresses in A record (na), ASN diversity of IP addresses (no. of ASN/na), standard deviation in round trip times of returned IP addresses (SD-RTT), name server diversity etc. are observed. By using these parameters and the pre-learned pattern, classification algorithm declares the subjected domain as suspicious or benign.
2) The second stage is the temporal based approach. The suspicious domains will be monitored for a long period, and this stage collects a total number of unique IP addresses returned, unique ASN numbers, and the number of IP addresses of the name server, etc. Based on the collected features, either domain is classified as fast-flux domain or it will be whitelisted as benign domain.

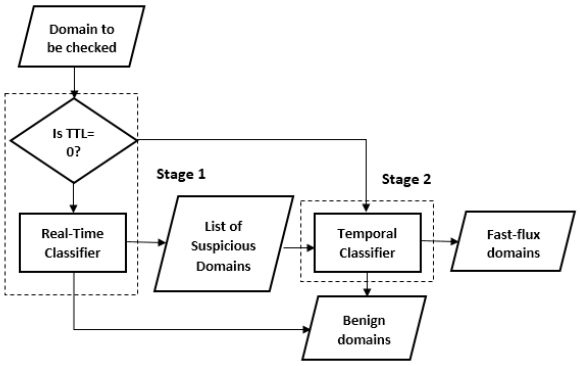

Fig.1. Stages in TempR Approach

VIII. Details of Features Used

Some other hybrid approaches also exists. However, a different set of input parameters are used in case of TempR. TempR use a different set of features for both the stages. These feature sets are described below.

A. Real-time Features

Real-time features provide a fast and efficient detection of suspected domains. These features describe the behavior of a domain at a particular time instant.

In this experimental setup, only authoritative DNS responses were used to obtain the information. The feature collection module collects the following real-time features.

Number of IP Addresses in A Record
ASN Diversity of IP Addresses in A Record
Time to Live of A Record
Standard Deviation of Round Trip Times of A Record IP Addresses
Number of Name Server IP Addresses in NS Record
Time to Live of NS Record
ASN Diversity of IP Addresses in NS Record
Temporal Based Features

The second type of features of the domain that data collection program collects are called temporal features. The domains are queried, and these features are collected periodically and are summed up for a specified monitoring period. Temporal features are incremental in nature and describe the behavior of a domain within specified monitoring period. In this feature collection module, the period between two queries is Time to live of DNS A record.

Like real-time features, temporal features were collected from the authoritative DNS responses. The data collection module collects the following temporal features.

Total no. of IP addresses in all A records fetched during the monitoring time
Total no. of name server IP addresses in all NS records fetched in given monitoring time
ASN diversity of all A record IP addresses
ASN diversity of all NS record IP addresses
Network prefixes of IP addresses in A records
Network prefixes of IP addresses in NS records
Fluxiness

IX. Dataset Description

For the purpose of training the classifier and testing the performance of TempR, a dataset has been prepared. To prepare the dataset many domains were monitored. The alive benign domains were selected from OpenDNS's public domain list hosted on Github ("Public-domain-lists/opendns-top-domains", 2015) and Alexa top 500 domains[14]. The criteria for selection of Fast-flux domains and details of the prepared dataset has been mentioned in following sections.

Selection of Fast-flux Domains from Blacklists

Candidate Fast-flux domains used in the dataset were selected from different malware trackers and malicious blacklists, the sources of these list are mentioned in table 1.

Table 1. Blacklists and Their Sources

Sr. No.	Blacklist name
1.	DNS Blackhole domain list[14]
2.	URL blacklist [15]
3.	DNS Blackhole Zeus gameover domains list[16]
4.	Malware domains Zeus gameover domains list[17]
5.	DNS Blackhole conficker domains list
6.	DNS Blackhole DGA domains list
7.	Malware Domain list (domain.txt and zeus domains)
8.	ZEUS tracker[18]

The two phases of the selection procedure are as:

Phase-1: Pre-filtering of domains from blacklists

All the domains in a blacklist were pre-filtered using a python script. The script checked whether the domain is alive or not. Then based on the following pre-filtering criteria domains were selected.

1. Domains with TTL of **A** record less or equal 5.

OR

2. Domains with following specific feature values associated with corresponding **A** record.

i. TTL<=3600 and
ii. Number of IP addresses >2 and
iii. Number distinct ASNs >2

Phase-2: Final selection of candidate fast-flux domains

In this phase, temporal features of the selected domains in phase-1, were collected for 12 hours. At the end of 12-hour period, the domains with fluxiness value greater than 1 were selected as fast-flux domains and included in the dataset.

To prepare the dataset numerous domains were considered, and the related statistics are summarized in the table 2.

Table 2. Number of Domains Considered to Prepare Dataset

Total Malicious domains picked from blacklists	2511007
No. of domains after Pre-filtering	30932
No. of Domains with TTL less than or equal to 5	245
Candidate fast-flux domains	48
Fetched authoritative NS and DNS A records	116144

The dataset used for experimental evaluation consists of instances from collected data related to benign domains and fast-flux domains. There were 384 benign labeled instances and 48 fast-flux labeled instances.

The details of collected feature are mentioned in Table 3. These features confirm the significant double flux behavior in tracked fast-flux domains. Bots were being used as both content hosts and name servers. Double flux behavior makes botnets more resilient to takedown.

Table 3. Details of Collected Features

Total number of tracked fast-flux domains	48
Total number of IP addresses in A records (AIPs)	4007
Total number of ASN numbers of AIPs	1328
/16 Network prefixes of AIPs	1736
Total number of Name server IP addresses (NSIPs)	2828
Total number of ASN number of NS IPs	1131
/16 Network prefixes of NSIPs	1593

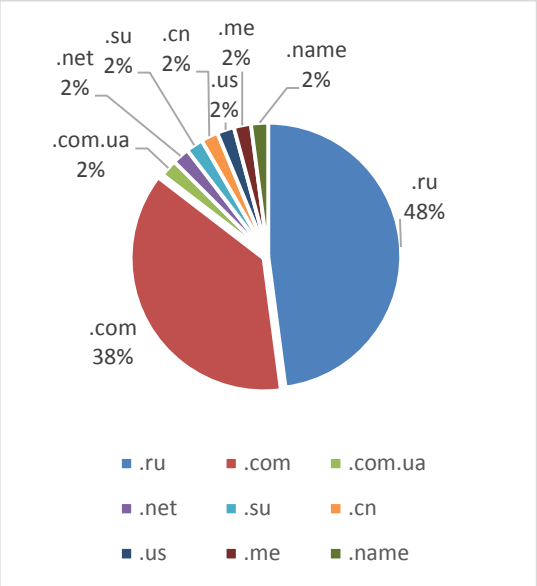

Fig.2. Distribution of Tracked Fast Flux Domains over Different TLD

As represented by in fig. 2., .ru and .com are most abused top level domains among the tracked domains, both contributing the 86% of total tracked domains. However, we can't make any such general statement, this may be the coincidence, and these values depend on the collected domain list.

X. Performance Evaluation

To evaluate TempR, a training data set from known fast-flux & benign domains has been prepared, and an appropriate data model for both stages was generated from the dataset using various intelligent classifiers. To test the performance of various classifiers we use 10 fold cross validation approach. The performance of various classifier during training & validation is mentioned in table 4. The performance was evaluated on the basis of accuracy and false positives.

Table 4. Percentage of Correctly Classified Instances

Algorithm	Real-time data	Temporal data (12 hours)	Temporal data (24 hours)	Temporal data (36 hours)	Temporal data (48 hours)
J48	95.14	95.37	97.22	97.22	97.68
Random Forest	95.37	94.44	96.99	97.68	96.99
Random tree	91.90	94.21	95.60	96.30	94.68
NBtree	94.91	94.44	97.45	96.53	96.99
Genetic programming	92.82	92.59	93.28	93.98	94.44
LMT	94.21	94.44	95.83	96.75	97.45
ADtree	94.44	95.37	97.22	96.53	96.99

In TempR, the first stage (real-time detection) classifies the benign domains and domains suspicious to be fast-flux, and 2nd stage (temporal detection) monitor those suspicious domains. In this experiment, we used 3 data classification algorithms (J48, Random Forest, and NBTree) that have achieved a higher accuracy as compared to other algorithms during training and testing of the dataset. The purpose of evaluating data related to different time periods is to identify minimum effective temporal detection period for the proposed technique.

To assess the performance of TempR, we measured all the performance metrics i.e. True Positives (TP), False Positives (FP), True Negatives (TN), False Negatives (FN) using various data classification algorithms. These performance metrics for "Real-time and temporal detection" are calculated by following steps:

True positives and false negatives remain the same as those were in real-time detection.

False positives that are whitelisted in temporal detection (2nd stage) are then added to true negatives.

The number of whitelisted domains is subtracted from the number of false positives generated by real-time detection (1st stage).

Then we calculated the accuracy of proposed 2-stage approach for detection of fast-flux domains. The accuracy has been calculated from performance metrics with following formula.

Accuracy= (TN+TP)/N

Where N= Total number of instances in dataset

Total number of TP, FP, TN, FN observed during the evaluation of TempR have been mentioned in table V.

Table 5. TP, FP, TN, FN Observed During the Evaluation Using Various Classifiers

Classifier	True Positives	False Positives	True Negatives	False Negatives
J48	35	0	384	13
Random Forest	34	0	384	14
NBtree	33	0	384	15

Classifiers that have achieved a higher accuracy as compared to other algorithms along with corresponding detection accuracy rate have been mentioned in Table VI. In table VI, column entitled "real-time detection" describes the performance metrics on real-time features (1st stage). The column named as "Real-time and temporal detection" provides the detection accuracy when results of both real-time detection and temporal detection were combined.

Table 6. Detection Accuracy Rate of Tempr Using Various Classifiers

Classifier	Real-time Detection	Real-time and Temporal Detection (12-Hours data)	Real-time and Temporal Detection (24-Hours data)	Real-time and Temporal Detection (36-Hours data)	Real-time and Temporal Detection (48-Hours data)
J48	95.14	96.30	96.99	96.99	96.99
Random Forest	95.37	96.76	96.76	96.76	96.76
NBtree	94.91	96.53	96.30	96.53	96.53

When using J48 classification algorithm, TempR illustrates highest detection accuracy of 96.99% with feature set collected by 24 hours monitoring. However, with Random Forest classification algorithm it achieved 96.76% detection accuracy with feature set obtained by monitoring the domains just for 12 hours.

XI. Comparison of Tempr Performance with other Detection Techniques

This section represents the comparison between the TempR performance in terms of detection accuracy and that of the previous fast-flux detection techniques such as Flux-score[6], GRADE [7], SSFD[10], and, FFBD [12]. We consider the accuracy of other techniques as calculated and mentioned in [7]. The graph in Fig 3. compares the accuracy of these techniques. TempR performed better than flux-score, FFBD, and SSFD. As compared to GRADE, the accuracy of the proposed approach is a little low. However, as compared to GRADE, TempR is also able to address the problem of single IP with TTL=0, whitelisted all the false positives after temporal detection stage.

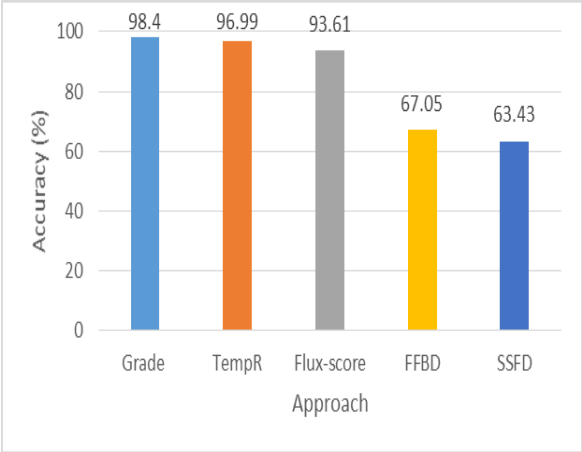

Fig.3. Comparison of TempR with Existing Detection Techniques (In Terms of Accuracy)

XII. Observations

The facts observed during this research work are:

1. .ru and.com are most abused top level domains among the tracked domains, both contributing the 86% of total tracked domains.
2. Collected data shows the significant double flux behavior. Bots are being used as both content hosts and Name servers.
3. There are malicious domains with TTL=0 and with TTL<5.
4. Some malicious domains use cloud hosting and number of associated IP addresses in such cases is more than benign domains (256 in one case, other is 512). However, these IP addresses belong to the same ASN.
5. There is a trade-off between detection accuracy and the temporal detection period. The detection accuracy increases as the monitoring period increases but becomes constant after 24 hours.
6. All false positives of real-time detection (stage-1), were whitelisted in the temporal detection stage.
7. The proposed 2-Stage detection has detection accuracy of 96.99% with 24 hours of temporal detection using J48 classifier, and it is 96.76% with 12 hours dataset when using Random Forest algorithm.

XIII. Conclusions

Fast-flux is a DNS manipulation technique that makes botnets more resilient to takedowns. Fast-flux service networks are often confused with benign technologies such as content distribution networks and round robin DNS. It's challenging to differentiate fast-flux domains from benign domains. A significant number of false positives occur during detection of fast-flux domains.

In this paper, we presented a Multi-stage FFSN detection approach entitled TempR. This approach is intended to increase accuracy and reduce the false positives. We collected the features (real-time features and temporal features) of benign and fast-flux domains. Then these instances in datasets were classified with classifiers as implemented in WEKA. The performance of TempR was evaluated in the terms of average accuracy and false positive, true positive, false negatives, and false positive rate.

The results represent that 96.99% detection accuracy can be achieved using TempR with 24-hour temporal monitoring and the temporal stage successfully whitelisted all the false positives. The results of temporal detection represent that there is a trade-off between accuracy in the temporal detection and temporal monitoring period. TempR illustrated improved detection accuracy over existing detection techniques like FFBD, Flux-score, and SSFD. However, the detection accuracy is 1.4% less than that of GRADE. But unlike GRADE, TempR can also detect fast-flux domains returning single IP address with TTL=0, successfully whitelisted all the false positives and cannot be evaded easily by the attackers.

However, TempR has eliminated the false positives but it still suffers from few false negatives generated in the real-time detection stage. In the future, work can be done to reduce these false negatives that can improve the performance in real-time detection stage and overall performance of TempR. Future work can be focused on the study of IP sharing among various fast-flux domains, and the role of IPv6 addresses in FFSN, etc.

References

[1] M. Feily, A. Shahrestani and S. Ramadass, "A Survey of Botnet and Botnet Detection," in Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Glyfada, 2009.

[2] K. Govind, S. Selvakumar,"Auto-Pattern Programmable Kernel Filter (Auto-PPKF) for Suppression of Bot Generated Traffic", IJCNIS, vol.6, no.1, pp.48-54, 2014. DOI: 10.5815/ijcnis.2013.01.07

[3] E. Passerini, R. Paleari, L. Martignoni and D. Bruschi, "FluXOR: detecting and monitoring fast-flux service networks," in Detection of intrusions and malware, and vulnerability assessment, 2008.

[4] Z. Futai, Z. Siyu and R. Weixiong, "Hybrid Detection and Tracking of Fast-Flux Botnet on Domain Name System Traffic," Communications, China, vol. 10, no. 11, pp. 81-94, 2013.

[5] K. Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks," Grid Computing and Distributed Systems Laboratory, University of Melbourne, 2007.

[6] T. Holz, C. Gorecki, K. Rieck and F. C. Freiling, "Measuring and Detecting Fast-Flux Service Networks," in NDSS, 2007.

[7] H.-T. Lin, Y.-Y. Lin and J.-W. Chiang, "Genetic-based Real-time Fast-Flux Service Networks Detection," Computer Networks, vol. 57, no. 2, pp. 501-513, 2013.

[8] B. N. Al-Duwairi and A. T. Al-hammouri, "Fast flux watch: A mechanism for online detection of fast flux networks," Journal of advanced research, vol. 5, no. 4, pp. 473-479, 2014.

[9] W. Xu, X. Wang and H. Xie, "New Trends in FastFlux Networks," in Black Hat Conference, 2013.

[10] S.-Y. Huang, C.-H. Mao and H.-M. Lee, "Fast-flux Service Network Detection Based on Spatial Snapshot Mechanism for Delay-free Detection," in ASIACCS '10, 2010.
[11] A. Kamal, A. ALmomani, A. Manasrah, and M.M. Kadhum. "A survey of botnet detection based on DNS." Neural Computing and Applications (2015): 1-18.
[12] C. H. Hsu, C. Y. Huang and K. T. Chen, "Fast flux bot detection in real time," in 13th International Symposium, RAID 2010, Ottawa, Canada, 2010.
[13] "Alexa Top 500 Global Sites," Alexa Internet, Inc., [Online]. Available: http://www.alexa.com/topsites. [Accessed January 2015].
[14] "DNS Blackhole blocklist," DNS-BH project, [Online]. Available: http://www.malwaredomains.com/. [Accessed February 2015].
[15] "URLblacklist.com," [Online]. Available: http://urlblacklist.com/cgi-bin/commercialdownload.pl?type=download&file=bigblacklist. [Accessed February 2015].
[16] "DNS-BH – Malware Domain Blocklist," Jaunuary 2015. [Online]. Available: http://www.malwaredomains.com/.
[17] "MDL- Downlaodable Lists," [Online]. Available: http://www.malwaredomainlist.com/forums/index.php?topic=3270.0. [Accessed February 2015].
[18] "Zeus Tracker," zeustracker.abuse.ch, [Online]. Available: https://zeustracker.abuse.ch/blocklist.php. [Accessed January 2015].

Authors' Profiles

Prabhjot Singh completed Master of Technology in Cyber Security from Central University of Punjab, India and Bachelor Degree from Punjab Technical University, India. His research interests include information security, network security and network forensics.

Surinder S. Khurana is an Assistant Professor at Centre for Computer Science & Technology, Central University of Punjab, India, He received his Master's degree in computer science & engineering from PEC University of Technology, India in 2009. He has published many papers in refereed journals and conference proceedings. His research interests include networks security, cyber forensics and algorithm design.

How to cite this paper: Prabhjot Singh Chahal, Surinder Singh Khurana,"TempR: Application of Stricture Dependent Intelligent Classifier for Fast Flux Domain Detection", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.10, pp.37-44, 2016.DOI: 10.5815/ijcnis.2016.10.05