Fast-flux Service Network Detection Based on Spatial Snapshot Mechanism for Delay-free Detection

Si-Yu Huang
National Taiwan University of Science and Technology
d9815014@mail.ntust.edu.tw

Ching-Hao Mao
National Taiwan University of Science and Technology
d9415004@mail.ntust.edu.tw

Hahn-Ming Lee
National Taiwan University of Science and Technology
Academia Sinica
hmlee@mail.ntust.edu.tw

ABSTRACT

Capturing Fast-Flux Service Networks (FFSNs) by temporal variances is an intuitive way for seeking to identify rapid changes of DNS records. Unfortunately, the features regard to temporal variances would lead to the delay detection (more than one hour) of FFSN which could cause more damages, such as Botnet propagation and malware delivery. In this study, we proposed a delay-free detection system, Spatial Snapshot Fast-flux Detection system (SSFD), for identifying FFSN in real time and alleviating these potential damages. SSFD is capable to capture the geographical pattern of hosts as well as mapping IP addresses in a DNS response into geographic coordinate system for revealing FFSNs at the moment. The SSFD benefits from two novel spatial measures proposed in this study– spatial distribution estimation and spatial service relationship evaluation. These two measures consider the degree of uniform geographic distribution of infected hosts among FFSN composed of Bots, Content Distribution Network and general benign services. After that, Bayesian network classifier is applied to identify the FFSNs with the joint probability consideration against evading our proposed detection technique easily for attackers. Our experiment results indicate that the proposed SSFD system is more effective and efficient (within less than 0.5 second) with lower False Positive rate than flux-score based detection through one public dataset and two collected datasets.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Network**]: Security and Protection

General Terms

Security, Algorithms

Keywords

Fast-flux Service Network (FFSN), Botnet, Content Distribution Network (CDN), Bayesian Network Classifier

1. INTRODUCTION

Fast-flux service network exploits a DNS technique mostly used by Botnets to hide malicious behavior behind a complex ever-changing network of proxy servers (or called Fast-Flux Service Network, FFSN) [13]. The famous P2P Botnet used this technique is the storm worm for Botnet infrastructure (e.g. Bots communications), sending spam e-mails, and online scams [11, 9, 12, 16]. The current research on P2P Botnet detection is the most difficult task because of similar-looking traffic and behavior between P2P Botnet and normal P2P. Hence, many researchers focus on FFSN detection for identify the malicious behavior and even for finding out P2P Botnets.

In the current detection approaches, many researchers utilize the temporal-based characteristics (i.e. the same as the name “fast-flux”) with fast changes of IP addresses to identify FFSN. The detection delay would always happen in temporal-based detection mechanism because extracting the temporal-features regarded to the variances of time interval requires a period of time, e.g.: Time To Live (TTL) time. This kind of detection mechanism would result in the long delay for benign domains (e.g. 14,192 seconds for ‘yahoo.com’) so that users are troublesome. Furthermore for fast-flux domains, the detection delay for at least one TTL time (e.g. 3,600 seconds for ‘ndkgje.fallbroad.com’) could cause attack proliferation such as Botnet propagation and malware delivery [11, 9, 12, 16]. Hence, in our investigation, we concentrate on providing delay-free detection (i.e. real-time detection) for reducing the damages and waiting time.

Holz et al. [5] utilized the temporal-based characteristic and a spatial feature to identify FFSN. The temporal-based characteristic could cause that extracting the features takes the detector TTL+1 seconds (i.e. detection delay time). In addition, they utilized a spatial feature, Autonomous System Number (ASN) with a network group connecting one or more networks (clearly defined routing policy in [4]), to imply the simple concept of geographical distribution degree for hosts. The reason why to make use of ASN is that FFSNs tend to be located in different ASNs since the infected hosts are distributed across different ISPs. Furthermore, FluXOR system [12] reduces the detection delay time from TTL time to 1-3 hours because which TTL time is larger than three hours is regarded as the benign. However, all of the above related work has the same problem with the detection delay. Hence, our goal is that detecting FFSN

ASIACCS'10 April 13–16, 2010, Beijing, China.

and benign is as soon as possible (less than one second per domain) with effectiveness.

In our investigation, we propose Spatial Snapshot Fast-flux Detection (SSFD) system, replacing temporal-based characteristics with spatial snapshot mechanism, for providing delay-free detection (i.e. real-time detection). The used spatial snapshot mechanism means that all of IP addresses in each DNS response packet are immediately corresponded to the geographic coordinate system (i.e. latitude and longitude) by an open project, hostip.info [6]. For example, Figure 1 drawn by the Google maps illustrates the IP-geographical mapping.

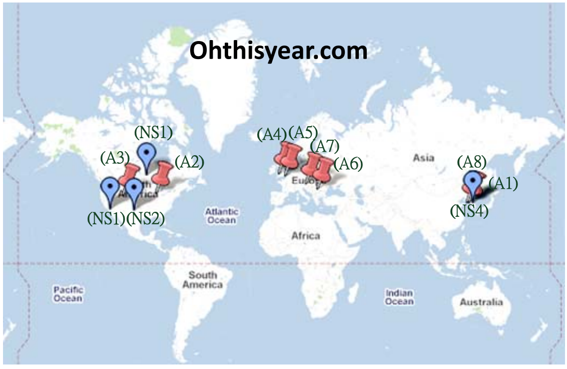
ANSWER SECTION			
(A1) 69.146.38.156	(A2) 75.45.165.219	(A3) 75.54.92.139	(A4) 76.176.179.150
(A5) 125.141.88.141	(A6) 210.96.195.64	(A7) 211.173.169.253	(A8) 62.245.114.120
ADDITIONAL SECTION			
(NS1) 66.165.197.187	(NS2) 210.123.24.9	(NS3) 59.149.105.240	(NS4) 222.239.58.242

(a) IP addresses in the public dataset for Ohthisyear.com

Format: (Latitude, Longitude)

ANSWER SECTION			
(A1) (37.756,128.896)	**(A2)** (42.366,-83.102)	**(A3)** (41.426,-105.515)	**(A4)** (51.800,-0.200)
(A5) (50.650,4.267)	**(A6)** (44.433,26.100)	**(A7)**(47.200,18.133)	**(A8)** (37.566,127.000)
ADDITIONAL SECTION			
(NS1) (51.033,-93.833)	**(NS2)** (33.548,-101.922)	**(NS3)** (34.042,-118.299)	**(NS4)** (37.566,127.000)

(b) Mapped in geographic coordinate system



(c) Representation by Google Maps

Figure 1: The illustration of IP-geographical mapping in spatial snapshot mechanism for a fast-flux domain from the public dataset. All of IP addresses are mapped into geographic coordinate system with a display by Google Maps.

In order to achieve Spatial Snapshot Fast-flux Detection, we provide two spatial measures, spatial uniform distribution estimation and spatial service relationship evaluation, for detecting FFSNs. The former is used to estimate the uniform geographic distribution of infected hosts by information theory (e.g. entropy). The reason why to utilize the geographic distribution is that infected hosts are distributed over all the countries. Because geographic distribution is variant for each spatial snapshot, namely variant geographic distribution problem, the latter is used to be the invariance for solving this problem. Also, the spatial service relationship is used to improve the misclassification between FFSN and Content Distribution Network because of the close spatial service relationship for CDN. In our experiments, the results display that our proposed SSFD is more effective and efficient (within less than 0.5 second) with lower False Positive (FP) rate than FSD method through one public dataset and two collected datasets.

The major contributions of the paper can be summarized as follows:

- Contribute a new idea about spatial snapshot mechanism for delay-free detection (i.e. real-time detection)
- Provide a novel spatial measurement, spatial distribution estimation and spatial service relationship evaluation, to identify FFSN
- Publish our collected datasets for evaluating the effectiveness

The rest of the paper is organized as follows: The implemented system of the proposed SSFD is illustrated in Section 2. And then, Section 3 provides the experimental evaluation to show the effectiveness and efficiency. The limitations and evading techniques of our proposed SSFD are discussed in Section 4. Finally, we conclude in Section 5.

2. SPATIAL SNAPSHOT FAST-FLUX DETECTION SYSTEM (SSFD)

In order to provide delay-free detection mechanism for FFSN, we propose Spatial Snapshot Fast-flux Detection system (SSFD) with the critical techniques/component of spatial snapshot mechanism for reducing the detection delay. In this section, we illustrate our approach in full details. As discussed before, since FFSN comes with more uniform geographic distribution among infected hosts and more spread service relationship which is comparing with benign service, we intelligently leverages the intrinsic characteristic of FFSN to provide detection without delay. The SSFD, the system architecture is as shown in Figure 2, consists of three components: DNS Packet Monitor, Spatial Snapshot Feature Extractor and Fast-flux Attack Detection Engine. In addition, two data sources are used for supporting to SSFD, namely Spatial Coordinate Database and Standard Time Zones Database. These modules and data sources will be briefly described as follows.

2.1 DNS Packet Monitor

DNS Packet Monitor is implemented by Wireshark [14] for collecting the DNS response packets. The main work is for collecting real-time DNS response information with IP addresses in Answer Section and Additional Section, TTL time in each record, and other detailed information.

2.2 Spatial Snapshot Feature Extractor

Spatial Snapshot Feature Extractor can derive the spatial snapshot information for identifying FFSNs. The used snapshot implies that all of IP addresses in a DNS response packet are mapped into the geographic coordinate system. After that, we provide two spatial measures, spatial distribution estimation and spatial service relationship evaluation, for identifying FFSNs. Hence, this module has four components: IP Extractor, Spatial Address Finding Agent, Spatial Uniform Distribution Estimator and Spatial Service Relationship Estimator.

2.2.1 IP Extractor

This component is used to obtain IP addresses from real-time raw DNS packet. While snapshot in our system, our

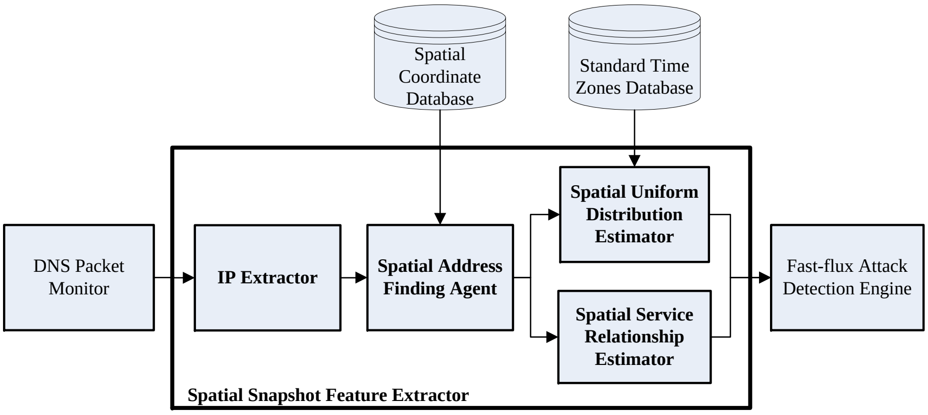

Spatial Coordinate Database
Standard Time Zones Database
DNS Packet Monitor
IP Extractor
Spatial Address Finding Agent
Spatial Uniform Distribution Estimator
Spatial Service Relationship Estimator
Fast-flux Attack Detection Engine
Spatial Snapshot Feature Extractor

Figure 2: System architecture of Spatial Snapshot Fast-flux Detection.

system only extracts all of IP addresses in Answer Section and Additional Section from a packet of DNS response, and these IP addresses will be mapped into the spatial coordinates.

2.2.2 *Spatial Address Finding Agent*

This component provides a spatial transformation mechanism from virtual IP addresses to the physical spatial coordinates from Spatial Coordinate Database maintained by an open project, hostip.info [6]. Our system will utilize the spatial coordinate to estimate the spatial information in next two subsections.

2.2.3 *Spatial Uniform Distribution Estimator*

Spatial Uniform Distribution Estimator is used to evaluate the degree of uniform geographic distribution for hosts because infected hosts are distributed across all of the countries [5]. In detail, we utilize "Time Zones" to discretize the geographic system space and make use of information theory (e.g. entropy) for measuring the degree of uniform distribution.

The time zones is required because infected hosts in the FFSN could remain working in time zones under working hours (e.g. daytime). Furthermore, the attackers could make these infected hosts in a FFSN scatter across different time zones or distribute over the same time zone under working hours for doing continuous malicious activity. Greenwich Mean Time (GMT) [3, 15] is applied to be as a transformation basis of time zones from spatial coordinates. Furthermore, compared with FFSN, most benign domains (about 95%) are in the same time zone because of the setup cost.

The insight of entropy might use an uniform measure for the spatial distribution as shown in Figure 3. Given IP addresses set Q in FFSN and an IP-spatial mapping function $C(Q) =< C_1(Q), C_2(Q) >$ with Latitude $C_1(Q)$ and Longitude $C_2(Q)$, the entropy-based function is used to estimate the uniform distribution, named Time Zone Entropy (TZE) function as shown in Equation 1:

$$TZE(C(Q)) = -\sum_{t \in GMT} \frac{N_t(C(Q))}{|Q|} \log \frac{N_t(C(Q))}{|Q|} \quad (1)$$

where $N_t(C(Q))$ is the number of times located in the tth time zone for hosts. TZE is applied to estimate the uniform degree of IP addresses (i.e. hosts) in Answer Section or in Additional Section. The reason is that an attacker can change IP addresses in Answer Section or in Additional Section according to three different attack types: Single flux, Name Server flux and Double flux [8].

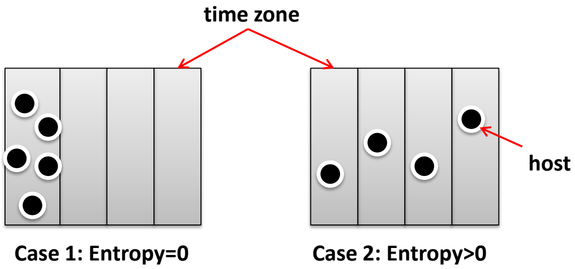


Figure 3: The concept of spatial uniform distribution using time zone entropy. Entropy is used to measure the degree of uniform distribution for hosts.

2.2.4 *Spatial Service Relationship Estimator*

Spatial Service Relationship Estimator is used to measure the closeness degree of spatial service relationship. Because Spatial Uniform Distribution Estimator cannot distinguish between FFSN and Content Distribution Network, FFSN-like geographic distribution, the service relationship is proposed to improve this problem. Hence, we utilize the characteristic, CDN is more close spatial service relationship than FFSN, to classify them.

In addition, we consider another problem: if the same time zone distribution for infected hosts in FFSN, Spatial Uniform Distribution Estimator is ineffective to identify FFSNs because of because of the value of time zone entropy is equal to zero. Hence, the service relationship is also used to solve this problem because of more spread service relationship for FFSN in the same time zone. On the other hand, the spatial service relationship is an invariant relationship against the variant geographic distribution problem.

The concept of spatial service relationship is shown in Figure 4. Spatial service relationship describes the relationship between a provider and a consumer by a service

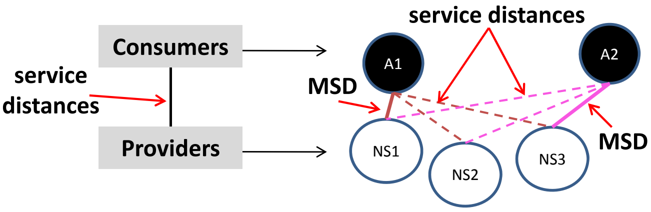


Figure 4: The concept of spatial service relationship. The service distance is regarded as the cornerstone in spatial service relationship. And MSD of A1 is the minimal service distance in all of service distances for A1.

distance. The consumers mean IP addresses in Answer Section such as 'A1' in the figure, and the providers imply IP addresses in Additional Section such as 'NS1' in the figure. The service distance is used to represent a measurement of the consumer-provider relationship. Also, Minimalizing Service Distances (MSD) for each consumer is required because MSD for benign is almost nearly equal to zero. In this paper, the spatial service relationship is combination of the average and standard deviation of minimal service distances (MSDs).

We define $d_{mm'}$ as the spatial service distance and a 2-norm distance in Euclidean space from mth IP address q_m in Answer Section (i.e. $q_m \in Q_{AS}$) to m'th IP address $q_{m'}$ in Additional Section (i.e. $q_{m'} \in Q_{NS}$) as shown in Equation 2.

$$d_{mm'} = \sqrt{(C_1(q_m) - C_1(q_{m'}))^2 + (C_2(q_m) - C_2(q_{m'}))^2} \tag{2}$$

Also, for each q_m, d_m is defined as MSD shown in Equation 3.

$$d_m = \min_{m' \in Q_{NS}} \{d_{mm'}\} \tag{3}$$

Furthermore, the average $\bar{d}$ and standard deviation S_d of MSDs as shown in the Equation 4 are used to estimate the closeness degree of service relationship. In brief, standard deviation S_d is mainly used to the spread relationship measurement. However, if the value of standard deviation S_d is equal to zero, the average $\bar{d}$ is used for the measurement.

$$\bar{d} = \frac{\sum_{m=1}^{|Q_{AS}|} d_m}{|Q_{AS}|} \; and \; S_d = \sqrt{\frac{\sum_{m=1}^{|Q_{AS}|} (d_m - \bar{d})^2}{|Q_{AS}| - 1}} \tag{4}$$

2.3 Fast-flux Attack Detection Engine

We utilize Bayesian network classifier to detect FFSNs. The reason why this classifier is considered is that the joint distribution of the spatial features could make the attackers difficult to evade our proposed spatial snapshot. In addition, K2 algorithm [1] is used to learn the dependence relationship from the labeled data. On the other hand, the input of this detection engine is shown in Table 1.

3. EXPERIMENT RESULTS

This section presents experiment results of our proposed system SSFD, and we divide this section into five subsections. Datasets are first introduced in Subsection 3.1, and data analysis based on spatial measurement is then shown in Subsection 3.2. Performance evaluation is discussed in Subsection 3.3. All of the experiments are designed to answer the following questions:

Table 1: The input features description of detection engine

#	Description
x_1	Number of unique IP addresses in Answer Section
x_2	Number of unique IP addresses in Additional Section
x_3	Time zone entropy in Answer Section
x_4	Time zone entropy in Additional Section
x_5	Average of minimal service distances
x_6	Standard deviation of minimal service distances

- *Effectiveness*
  How quality is FFSN classification based on our proposed SSFD? We compared the proposed SSFD with the flux-score based method [5] in Subsection 3.4.
- *Efficiency*
  How fast is the proposed SSFD? For the detection time with feature extraction and instance classification, the proposed SSFD was compared with the current detection methods in Subsection 3.5.

Finally, we provide the real cases study to discuss our proposed approach based on our spatial snapshot mechanism in Subsection 3.6.

3.1 Datasets Description

In this paper, we utilized one public dataset and two collected datasets for evaluating the effectiveness and efficiency. Public dataset was derived from authors' website [1] published at Network and Distributed System Security Symposium (NDSS) conference in 2008, and collected datasets were synthesized from different sources on the Internet. These data would be used in our experiments for evaluating the effectiveness and efficiency.

In addition, the variant geographic distribution problem (i.e. concept drifts) is considered because of infectiveness result for our proposed spatial features. The reason why this problem produces is that captured geographical distribution for infected hosts would be changed in the different periods. Hence, we collected DNS data during different periods in 2009, and the detailed time will be described in Subsection 3.1.1.

This section will introduce the collected data sources in Section 3.1.1 and the synthesis of the collected data in Subsection 3.1.2.

3.1.1 Data Sources

Labeling whether an instance is an attack or not is an open problem in the experiment verification. In our paper, we labeled the data based on the data source. The labeling work of FFSN is based on the famous Fast-Flux blocklist Website (FFWeb) such as ATLAS [2], DNSBL [3] and FluXOR

[1] The dataset of fast-flux, http://pi1.informatik.uni-mannheim.de

[2] ATLAS, http://atlas.arbor.net/summary/fastflux

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

[4] from the information security experts or the state-of-the-art detection systems. Also, the benign data was collected from the different sources, famous top websites as Alexa [5] and top blogs as Blogs On Top (BOT) [6], because these benign data sources have a verification mechanism to filter the malicious websites and blogs. In addition, which Alexa and BOT are with different properties would result in different geographic information.

Table 2 is the information of different data sources for benign and FFSN. One of these different data sources is the public dataset published in 2008 that contains contains 12,878 instances for benign and 75 instances for FFSN. In addition, the benign data is 92,178 and 953 instances for Alexa and BOT, and the FFSN data is 26,930 and 615 instances for FluXOR and FFWeb. On the other hand, these raw DNS data sets were appeared during different periods with late 2007 and the beginning of 2009.

Table 2: Information of different data sources

Sources	Instances	Collection time	Label
NDSS'08	12,953	Nov. 23, 2007	X
Alexa	92,178	Mar. 8, 2009	Benign
FluXOR	26,930	Mar. 12, 2009	FFSN
FFWeb	615	Mar. 9 - May 1, 2009	FFSN
BOT	953	May 4, 2009	Benign

3.1.2 Data Synthesis

Data synthesis is based on the approximate collection time, such as Alexa and FluXOR on March, for evaluating the effectiveness from the tolerance of concept drifts. Also, because the real network environment is considered, the number of instances for benign is more than for FFSN. Hence, we synthesized four data sources into two datasets. One of the two datasets is the data synthesis with Alexa and FluXOR. After that, we split this dataset into two parts, 66% of this dataset was as train-1, and the other was as test-2. The other is the data synthesis with FFWeb and BOT, namely test-3. Table 3 is the summary of the above mentions.

Table 3: Summary of the data synthesis

Datasets	Description
train-1	66% of data synthesis with Alexa and FluXOR
test-1	Public dataset at NDSS conference in 2008
test-2	34% of data synthesis with Alexa and FluXOR
test-3	Data synthesis with FFWeb and BOT

In this experiment, train-1 was considered as the training data in order to construct the model profile for Bayesian Network classifier, and test-1, test-2 and test-3 were regarded as testing data for evaluating the effectiveness and efficiency. Figure 5 displays the data distribution in the training and testing datasets. In the training dataset, there are 60,833 instances for benign and 17,778 instances for FFSN in train-1 dataset . In the test datasets, test-2 dataset contains 31,345 benign instances and 9,152 FFSN instances, and other information for testing datasets is mentioned in Subsection 3.1.1.

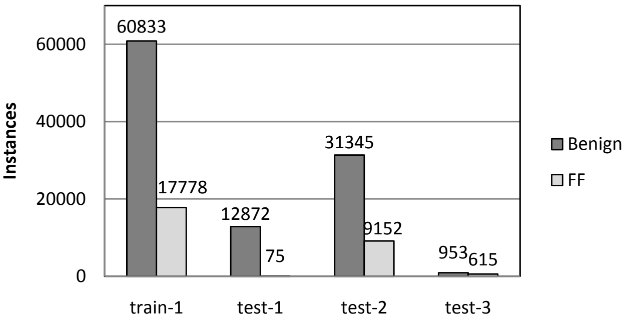


Figure 5: Data distribution for benign and FFSN in the training and testing datasets.

3.2 Data Analysis Based on Spatial Measurement

This subsection shows data analysis based on spatial measurement from the public dataset [10]. The spatial measurement is spatial distribution estimation and spatial service relationship evaluation. The former is analyzed in Subsection 3.2.1, and the latter is discussed in Subsection 3.2.2.

3.2.1 Spatial Distribution

Figure 6 shows the comparison of Spatial Distribution with the entropy concept with (a) benign and (b) FFSN. The geographic distribution of FFSN is more uniform (higher entropy value) than benign because infected hosts in FFSN are distributed over all the countries.

3.2.2 Spatial Service Relationship

Figure 7 displays the comparison of Spatial Service Relationship with the standard deviation concept with (a) benign and (b) FFSN. The spatial relationship of benign is more close (lower standard deviation value) than FFSN because of the close spatial service relationship for benign.

3.3 Performance Evaluation

In order to evaluate the effectiveness, a confusion matrix [7] is applied to measure the accuracy, recall rate and False Positive (FP) rate in our experiments. Furthermore, we emphasize FP rate because a higher FP rate shows that the most benign domains could always classified into FFSNs to make users troublesome. Also, we consider the cost-benefit factor, Area Under the ROC (AUC) [2], to measure the trade-off between True positive (TP) rate and FP rate. The higher AUC value means the higher TP rate (Benefit) with lower FP rate (Cost).

On the other hand, we utilized wall clock time to measure the efficiency. The machine, an Intel P4 2.4GHz with 3.5Gb ram, is used to test the performance in these two experiments.

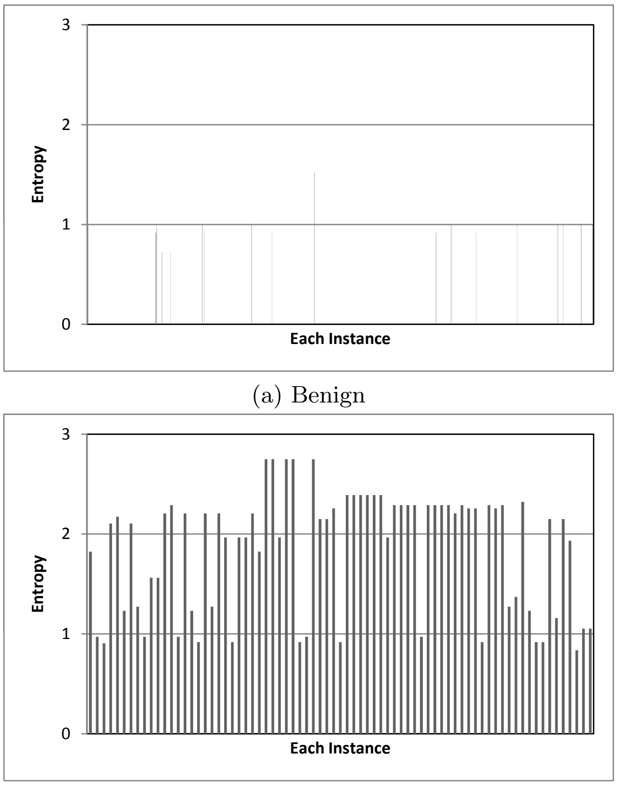
3.4 Experiment 1: Effectiveness Comparison

In Experiment 1, the proposed SSFD was compared with Flux-Score based Detection (FSD) method proposed by Holz

[4] FluXOR, http://pi1.informatik.uni-mannheim.de

[5] Alexa, http://www.alexa.com

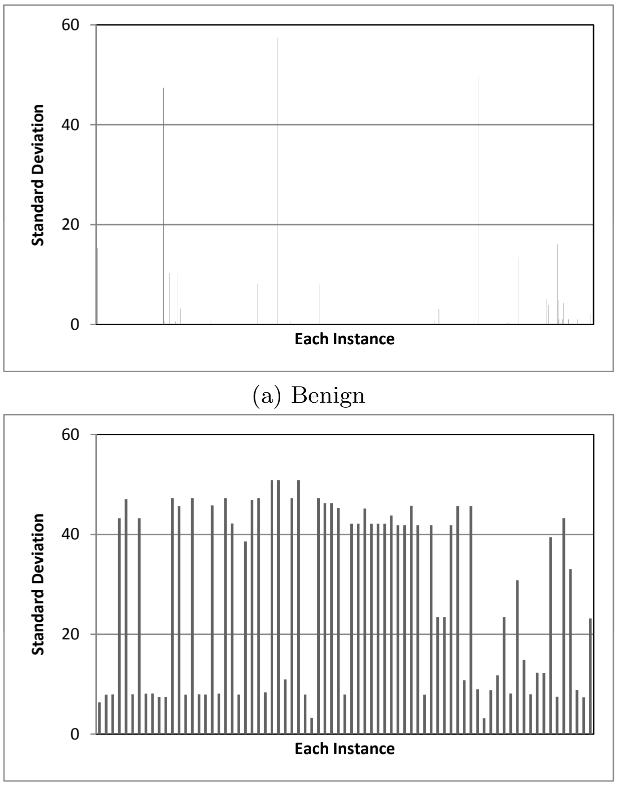
[6] Blogs On Top (BOT), http://www.blogsontop.com



(a) Benign

(b) FFSN

Figure 6: Comparison of Spatial Distribution with the entropy concept with (a) benign and (b) FFSN. The geographic distribution of FFSN is more uniform (higher entropy value) than benign because infected hosts in a FFSN are distributed over all the countries.



(a) Benign

(b) FFSN

Figure 7: Comparison of Spatial Service Relationship with the standard deviation concept with (a) benign and (b) FFSN. The spatial relationship of benign is more close (lower standard deviation value) than FFSN because of the close spatial service relationship for benign.

et al. [5]. Because FSD method needs two DNS information instead of our collected snapshot information in these datasets, the adjustment mechanism for the best case of detection accuracy for FSD method was mentioned in first subsection. Furthermore, the effectiveness comparison was described in final subsection.

3.4.1 Adjustment Mechanism for FSD Method (FSD and FSD#)

In order to satisfy temporal characteristic for FSD method (i.e. FSD method needs two DNS information), we first describe the simulation of the best case for the three temporal features. On the other hand, because FSD cannot detect the instances with the same time zone distribution for infected hosts, the improvement of FSD Method is finally proposed.

- *Adjustment of our Collected DNS Information for FSD method*
  In order to satisfy the need with two DNS information for FSD method, we adjusted these collected datasets to obtain the best case for detection accuracy. The best case is highest detection accuracy in all of which the cases with the simulated second DNS information is considered. In detail, compared with the first DNS response information (e.g. IP addresses in Answer Section and Additional Section), the estimated information of the second DNS response is invariable for benign and is extremely different for FFSN. For example, if the unique number of IP addresses in the first DNS response are four, the unique number of IP addresses in the two DNS responses are four for benign (i.e. 4+0=4) and is eight for FFSN (i.e. 4+4=8).

- *Improvement of FSD Method (FSD#)*
  Because FSD method is ineffective for the instances with the same time zone distribution for infected hosts (i.e. the Case 1 in Figure 3), we improved FSD method (namely FSD#) to learn from the new instances in Case 1. The reason why FSD is ineffective is that the model profile is from test-1 dataset with no FFSN instances in Case 1. Hence, improved FSD method is through retraining the model profile using train-1 dataset based on Bayesian Network classifier. In detail, the differences between FSD and FSD# are training data and classifier.

3.4.2 Effectiveness Comparison with FSD and Our Proposed SSFD

As shown in Figure 8, compared with FSD [5] with detection delay, our proposed SSFD with delay-free is more effective with higher accuracy (SSFD: 98.16%, FSD: 90.80%). And compared with FSD# with the different features, SSFD performs a higher accuracy (SSFD: 98.16%, FSD#: 90.18%) with lower FP rate (SSFD: 0.00398, FSD#: 0.09704) and higher confidence (AUC value, SSFD: 0.9841, FSD#:0.910).

The reason why SSFD is better result is that the features of FSD method (The features of FSD# is the same as FSD method) can not identify FFSN with less of IP addresses in the same time zone (The detailed discussion will be in Section 4.1). In addition, false positive for our proposed SSFD results from ambiguous spatial distribution. Most FP instances with CDN service are the same spatial characteristic as FFSN.

On the other hand, we focus on the accuracy of FSD (90.80%) and FSD# (90.18%). Because of the best case for FSD, all of the benign instances are correctly classified so that the accuracy rate for FSD is higher than FSD#. Furthermore, the recall rate is used to the measure improvement of FSD method, and the average recall rates of FSD and FSD# in test-2 and test-3 are 77.52% and 56.52% respectively.

3.5 Experiment 2: Efficiency Comparison

In this experiment, we demonstrated how efficient our proposed SSFD is (i.e. benefit from spatial snapshot mechanism). The detection time was estimated in Section 3.5.1 according to the description of detection method proposed in FSD [5] and FluXOR [12], and we then compared our proposed SSFD with previous detection methods in Section 3.5.3.

3.5.1 Estimation of Detection Time for Previous Work

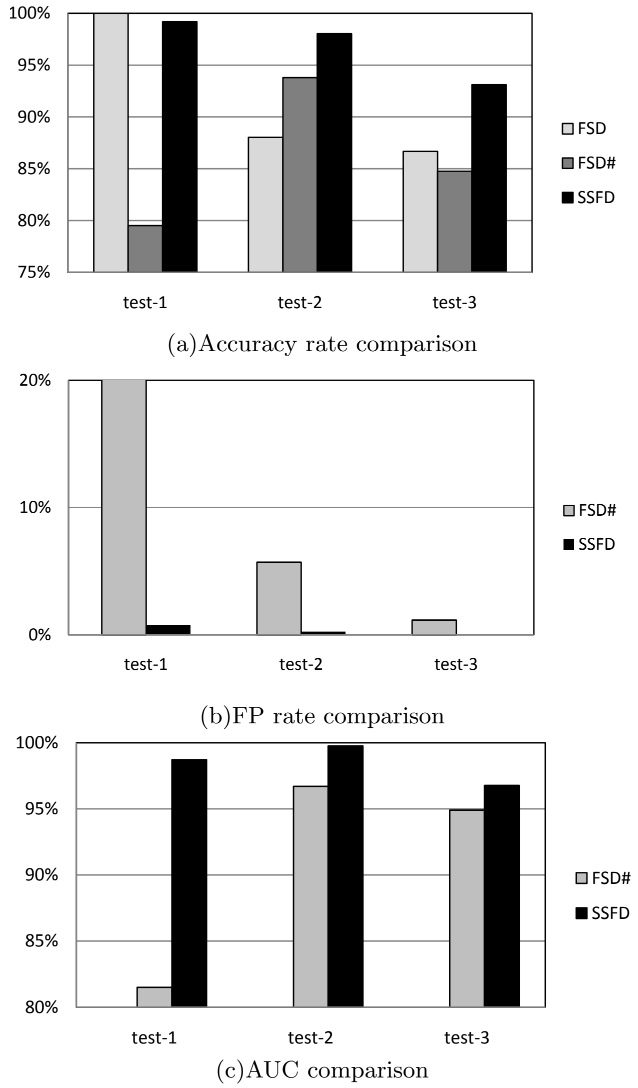
In this subsection, we illustrate how we estimated the detection time for current famous detection methods, FSD and FluXOR based on the description of detection method. We must determine the detection time of each domain from the large number of the same TTL or different TTLs. In detail, the minimal TTL time for all of 'A' records is regarded as the detection time (i.e. the best case for efficiency). Hence, the estimated detection time of FSD is the minimal TTL time for each domain. On the other hand, based on the description in [12], the detection time of FluXOR is one, two, or three hours because of three model profiles in FluXOR system. For the estimated detection time for FluXOR system, if the minimal TTL time is less than one hour, the estimated detection time is one hour, similarly, if more than three hours, it is regard as three hours, and for the other, the detection time is two hours. The summary of the above estimation is as shown in Table 4.

3.5.2 Detection Time for Our Proposed SSFD

The detection time for SSFD is inclusive of processing time in Spatial Snapshot Feature Extractor and classification time in Fast-flux Attack Detection Engine. Spatial Address Finding Agent costs more time than other processing time and classification time because of searching for the spatial coordinate in the large number of data. Hence, Spatial Coordinate Database is optimized for efficiency search.

3.5.3 Efficiency Comparison

Figure 9 presents the comparison of average detection time with FSD, FluXOR and the proposed SSFD. The X-axis is the number of instances with random sampling from testing datasets, and Y-axis is detection time (Unit is seconds) with function ($\log y + 1$). The result displays that SSFD is more efficient (around 0.5 second) than FSD (around 21,083 seconds) and FluXOR (around 6,186 seconds) because of replacing temporal characteristic with spatial snapshot mechanism.



(a)Accuracy rate comparison

(b)FP rate comparison

(c)AUC comparison

Figure 8: The results comparison of (a) accuracy, (b) FP rate and (c) AUC with FSD, FSD# and SSFD. The result displays that SSFD is more effective with lower FP rate and higher confidence (AUC).

3.6 Real Cases Study

Serval interesting real world cases are give for demonstrating the detection capability against to FFSNs based on our spatial snapshot mechanism. Furthermore, two cases are considered: (1) hosts are scattered over the same time zone (i.e. entropy value is equal to zero, Case 1) and (2) across different time zones (i.e. entropy value is larger than zero, Case 2). In addition, Google Map is used to show the visualization of spatial snapshot for the benign and FFSN. In this subsection, we focus on Time Zone Entropy in Answer Section, $TZE(Q_{AS})$, and spatial service relationship, MSD,

Table 4: The summary of estimated detection time for FSD and FluXOR.

Methods	Description
FSD [5]	The minimal TTL time (i.e. best case) in Answer Section for a domain is regarded as the detection time.
FluXOR [12]	The detection time is one, two or three hours. If the minimal TTL is less than one hour, the estimated time is considered as one hour. Similarly, if more than three hours, it is as three hours. And for the other minimal TTL, the detection time is two hours.

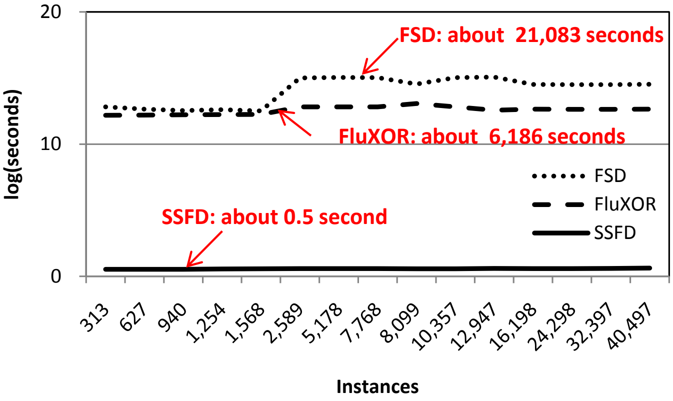
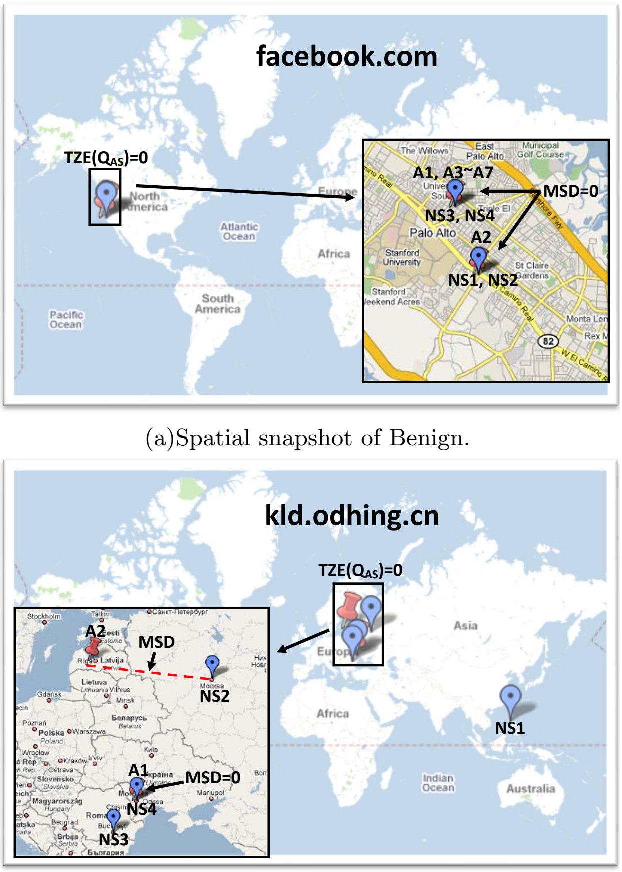


Figure 9: The comparison of average detection time with the previous work, FSD [5] and FluXOR [12]. X-axis is the number of instances with random sampling from testing datasets, and Y-axis is the detection time with log function. The result displays that a instance is detected by our SSFD for less time (around 0.5 second) than by the previous work in average.

for illustrating the differences between benign and FFSN.

Figure 10 shows the comparison of spatial measurement with benign and FFSN in Case 1. The benign domain, facebook.com, from test-1 dataset and the fast-flux domain, kld.odhing.cn, from train-1 dataset are shown in Figure 10 (a) and (b) respectively. In Case 1, FSD method [5] can not detect FFSN because there are less IP addresses, but our proposed SSFD can identify it mainly relying on the far service distance. Compared with Figure 10 (a) and (b), the service relationship is close (i.e. MSD value is nearly equal to zero) for most benign domains (about 96%) but is more spread for FFSN (The distance of MSD for A2 is far).

The comparison of spatial measurement in Case 2 with benign domain, microsoft.com, from test-1 dataset and the fast-flux domain, augustbody.com, from test-3 dataset is shown in Figure 11 (a) and (b) respectively. In Figure 11 (a), although the Microsoft makes use of Content Distribution Network(CDN) service to distribute the contents across different time zones ($TZE(Q_{AS})$=1.000), CDN service is the close service relationship. On the contrary, the fast-flux domain is more uniform geographic distribution ($TZE(Q_{AS})$=3.122) and more spread service relationship (S_d=66.010). Hence, our proposed SSFD can distinguish between FFSN and benign with CDN service by which FFSN is more spread of MSDs and more uniform geographic distribution for infected hosts than benign.



(a)Spatial snapshot of Benign.

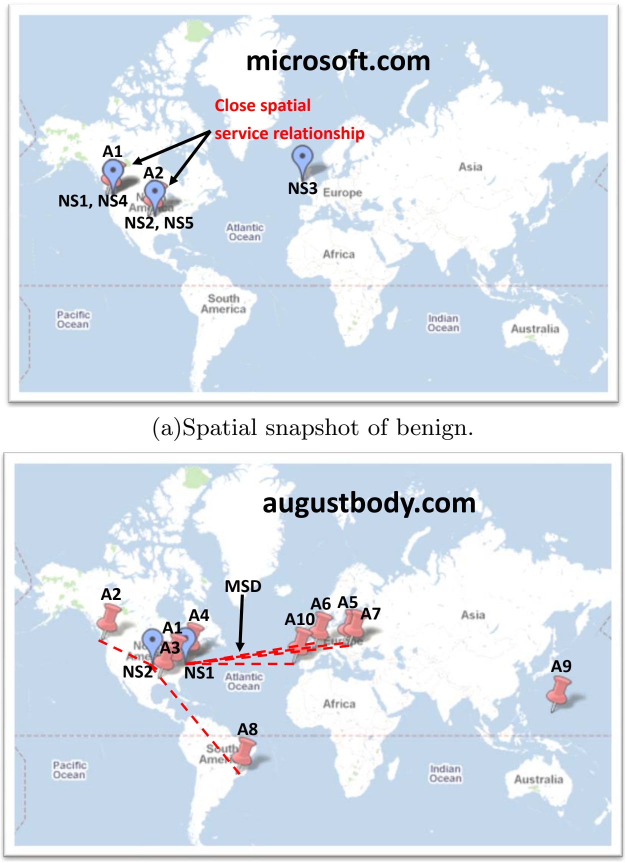
(b)Spatial snapshot of FFSN.

Figure 10: The comparison of spatial measurement with (a) the benign domain from test-1 and (b) fast-flux domain from train-1 in the same time zone distribution for hosts (Case 1). The result displays that MSDs are nearly equal to zero for the benign (i.e. close service relationship), but MSD of A2 is far for the FFSN (i.e. far service relationship).

4. DISCUSSION

In this section, all of the discussions are designed to answer the following questions:

- *Advantages of Effectiveness*
  Why does our proposed SSFD perform the effective result? We will discuss the reason why the SSFD is more effective than FSD method [5] in Section 4.1.
- *Limitations*
  What are the limitations of our proposed SSFD system? The limitations will be illustrated in Section 4.2.
- *Evading Techniques*
  How could the attackers evade the detection of our



(a)Spatial snapshot of benign.

(b)Spatial snapshot of FFSN.

Figure 11: The comparison of spatial measurement with (a) the benign domain from test-1 and (b) fast-flux domain from test-3 in the different time zones distribution for hosts (Case 2). The result shows that the spatial distribution of the infected hosts is more uniform than the normal servers. In addition, MSDs are nearly equal to zero (close spatial service relationship) for benign, but MSDs are spread and far (i.e. far service relationship).

proposed SSFD system? The evading techniques will be made a discussion in Section 4.3.

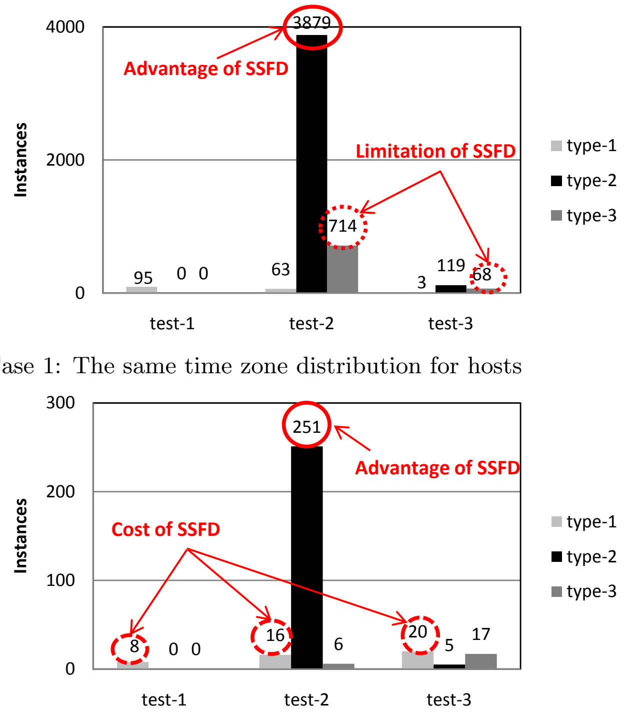
For convenience to analysis and discussion, we make use of the type errors concept in statistics to evaluate the cost, advantage and limitation for our proposed SSFD. The cost, namely type-1, means that the instances can be detected by FSD but can not be detected by SSFD. On the contrary, which the instances can be classified by SSFD but cat not classified by FSD is named the advantage or type-2. On the other hand, the limitation is that both can not identify instances. Table 5 lists the summary of the above descriptions.

4.1 Advantages of Effectiveness

Section 3.4 illustrates that our proposed SSFD performs the better result than Flux Score based Detection (FSD). And in this section, we discuss the reason why our proposed SSFD is more effective. Figure 12 shows that the instances in type-2 is more than in other types for Case 1 and Case 2. In other words, the advantage of SSFD is that there are more correctly classified instances in Case 1 and Case 2 than FSD method. The result implies that SSFD is able to identify FFSNs with the same time zone distribution for infected hosts (Case 1), but FSD is hard to detect them. The reason why FSD is ineffective is that the attacker evades the detection of FSD method (i.e. less IP addresses in Case 1).

Table 5: The summary of error types compared with the classified results of FSD and SSFD

Error	FSD	SSFD	Description
type-1	V	X	Cost or limitation for SSFD
type-2	X	V	Advantage for SSFD
type-3	X	X	Limitation for FSD and SSFD



Case 1: The same time zone distribution for hosts

Case 2: The different time zones distribution for hosts

Figure 12: The advantage comparison in Case 1 and Case 2 with FSD and our proposed SSFD. This result demonstrates that SSFD is able to classify FFSNs in Case 1, but FSD is hard to detect them.

4.2 Limitations

We list and discuss the limitations as follows:

- *IP-coordinate Database Maintenance*
  In our proposed SSFD system, we make an additional effort to maintain the IP-coordinate database because of the spatial snapshot mechanism. On the other hand, this database must update or renew the information at night by off-line update. In other words, our proposed SSFD system does not query on the internet frequently in the detection. However, if there are no geographic information for some of IP addresses (i.e. missing value problem), our proposed SSFD does

not work (the detailed discussion as follows).

- *IP-coordinate Transformation*
  There are limitations in IP-coordinate transformation problem as follows:

  (1) **Sleep domain problem**
  In the collected datasets, we find out the sleep domains with the virtual IP addresses (e.g. 10.0.0.1) or local host IP address (i.e. 127.0.0.1). The reason why to setup such information of IP addresses is that attackers usually prepare for doing a evildoing or for stopping their current malicious activities in order to evade blocking by DNS registration. Hence, the above mentioned IP addresses are not transferred into the spatial coordinates so that the proposed SSFD can not detect them.

  (2) **Missing value problem in database**
  The missing value problem means that some of IP addresses have no corresponding geographical coordinates. However, in our collected datasets, the missing value rate is nearly equal to zero listed in Table 13. If unfortunately, we meet the missing values problem, IP addresses are ignored in our proposed SSFD system.

- *Attack Misclassification*
  Several cases which would lead the proposed method into attack misclassification are shown as following:

  (1) **Single IP address problem**
  The problem is originally used to evade temporal-based detection systems (e.g. FSD method). Single IP address shows that the same IP address (or maybe multiple repeated IP addresses) is in the Answer Section and Additional Section. Furthermore, the single IP address is regarded as a point in geographic coordinate system. Our system is hard to detect because the value is equal to zero for all of spatial features. But this problem is considered as an ineffectiveness case because the single IP address problem can result in the overload for the infected host so that an end-user is easily aware that the computer is inefficiency.

  (2) **One-to-one or one-to-many relationships**
  The instances with one-to-one and one-to-many relationships (i.e. consumer-provider relationship) is hard to classify because of zero value for all of spatial features. Furthermore, these relationships perhaps implies dynamic DNS service. However, one consumer (i.e. infected host) could be inefficient to provide the continuous malicious activity if the machine (i.e. consumer) is ineffective.

4.3 Evading Techniques

In this subsection, we discuss that the attackers could evade the detection mechanism for our proposed spatial features. The evading techniques could be separated into three parts for illustration: evading for spatial uniform distribution, spatial service relationships, or both.

For spatial uniform distribution, geographic distribution for infected host would be less uniform or even a point for attackers. But the disadvantage is that ineffective infected hosts could result in the inefficiency malicious behavior. More attackers could set up the same time zone distribution existing in test-2 and test-3 dataset but not in the published test-1 dataset. However, we could make use of the spatial service relationship with invariance approach to detect this existing evading.

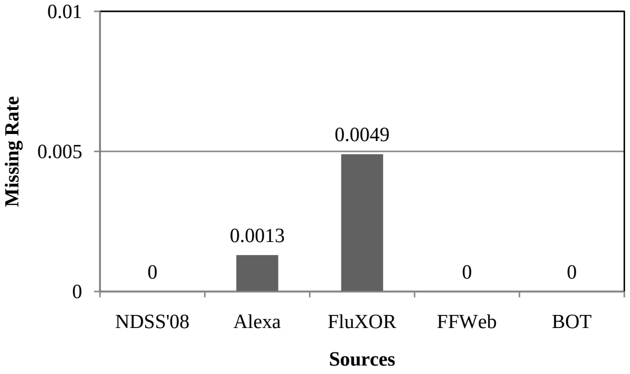


Figure 13: Missing value problem in collected datasets

For spatial service relationships, attackers could utilize the dynamic DNS service to evade this relationships. For example, a dynamic domain with name servers in Taipei is used to register a infected computer also in Taipei for doing the malicious activity. This relationship is hard to be classified but less infected computers could result in the inefficient attacks. On the other hand, attackers would make an additional effort to select similar IP addresses for being consumers and providers. This evading result could be difficult to detect by our proposed SSFD, but the additional effort could make motherships inefficient. Also, in reality, the similar IP addresses could be less.

For evading both characteristics, the attackers could make infected hosts be located in the near area, but this is a difficult task. If this were happened, our proposed SSFD could not detect.

On the other hand, we do not adopt the other characteristics such as ASN and TTL time. The reason is that ASN could make additional effort to query WHOIS messages, and TTL time is easier to set up by the attacker to evade the detection.

5. CONCLUSIONS AND FURTHER WORK

In this study, we contribute a new idea about spatial snapshot mechanism, replacing temporal characteristic with spatial snapshot mechanism, to provide the delay-free detection. Also, the novel spatial measurement, spatial distribution estimation and spatial service relationship evaluation, is proposed to evaluate the infected hosts distribution and relationships. Furthermore, our experiment results show that the proposed SSFD system is more effective and efficient with lower FP rate and higher confidence (higher AUC value) than FSD method.

Our further research focuses on analyzing and detecting the attack using the dynamic DNS service. In addition, we will extend our research to analysis of the sleep domain problem and the other zero day FFSN problem (existing malware or spam).

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Council of Taiwan under grants NSC96-2221-E-011-064- MY3, moreover by the Taiwan Information Security Center (TWISC), the National Science Council under grants NSC 98-2219-E-011-001. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

7. REFERENCES

[1] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.

[2] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[3] Greenwich2000. Greenwich mean time (GMT). [Online]. Available: http://wwp.greenwichmeantime.com/

[4] J. Hawkinson and T. Brisco, *RFC 1930: Guidelines for creation, selection, and registration of an Autonomous System.* RFC Editor, 1996.

[5] T. Holz, C. Gorecki, K. Rieck, and F. Freiling, "Measuring and detecting fast-flux service networks," in *Proceedings of the 15th Network & Distributed System Security Symposium (NDSS)*, 2008.

[6] HOSTIP Project. My IP Address Lookup and GeoTargeting Community Geotarget IP. [Online]. Available: http://www.hostip.info

[7] R. Kohavi and F. Provost, "Glossary of terms," *Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, vol. 30, pp. 271–274, 1998.

[8] M. Konings, *Initial Report of the GNSO Fast Flux Hosting Working Group.* Generic Names Supporting Organization (GNSO), 2009.

[9] M. Konte, N. Feamster, and J. Jung, "Dynamics of online scam hosting infrastructure," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement (PAM)*, 2009.

[10] Laboratory for Dependable Distributed Systems University of Mannheim. The dataset of fast-flux. [Online]. Available: http://pi1.informatik.uni-mannheim.de

[11] J. Nazario and T. Holz, "As the net churns: Fast-flux botnet observations," in *Proceedings of the 3th International Malicious and Unwanted Software (Malware)*, 2008.

[12] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi, "FluXOR: detecting and monitoring fast-flux service networks," in *Proceedings of the 5th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 2008.

[13] Wikipedia. The concept of fast-flux. [Online]. Available: http://en.wikipedia.org/wiki/Fast_flux

[14] Wireshark Foundation. Network protocol analyzer: Wireshark. [Online]. Available: http://www.wireshark.org/

[15] WorldTimeZone. Standard time zones. [Online]. Available: http://www.worldtimezone.com/standard.html

[16] C. V. Zhou, C. Leckie, and S. Karunasekera, "Collaborative detection of fast flux phishing domains," *Journal of Networks*, vol. 4, pp. 75–84, February 2009.