Exploring Global IP-Usage Patterns in Fast-Flux Service Networks

Ci-Bin Jiang, Jung-Shian Li*
Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan City 701, Taiwan.

* Corresponding author. Email: jsli@mail.ncku.edu.tw
Manuscript submitted July 23, 2015; accepted April 20, 2016.
doi: 10.17706/jcp.12.4.371-380

Abstract: In recent years, hackers have increasingly used fast-flux techniques to extend the lifetime of malware networks in order to conduct various Advanced Persistent Threat (APT) activities. Such activities typically target nations and or organizations for business or political motives and have the potential to cause immense disruption. Thus, it is essential to study the fast-flux service network and find possible attack behaviors. The literature contains various proposals for FFSN detection. However, these methods are either out of date in terms of the features they use for detection purposes or are unworkable under a new FFSN architecture identified in this study (denoted as N-flux networks), in which the IP addresses are swapped in and out at a speed normally associated with benign domains. Accordingly, the present study proposes a two-stage FFSN detection scheme in which a data mining algorithm is employed initially to detect possible FFSNs and a shared-domain detection algorithm is then applied to identify the nature of the FFSN through an analysis of its malware connections. The feasibility of the proposed scheme is demonstrated by analyzing five real-world datasets. It is shown that the proposed scheme achieves both a higher detection accuracy and a lower detection delay than existing schemes such as GRADE, Flux-Score, FFBD and SSFD.

Key words: Advanced persistent threat (APT), fast-flux service network (FFSN), N-flux, data mining.

1. Introduction

The bot controller, so-called bot master employs in such attacks, the bot master uses a fast flux technique to hide the malware delivery and phishing sites behind an ever changing network of compromised hosts, thereby rendering the network more resistant to discovery and counter-measures and prolonging its lifetime accordingly. Botnets have several key advantages from a hacker's perspective, for example, ease of operation, support for main nodes and limited audit trails.

Fig. 1 illustrates a typical attack scenario, in which a hacker sends a phishing email with a malicious attachment to multiple users. If the users interact with the email, their computers are compromised and then infect other computers in turn. Having established a network of infected machines, the hacker uses a RAT (Remote Access Tool) to control the machines in an attempt to exfiltrate critical data or sensitive information. In practice, the motives behind targeted attacks vary. Advanced Persistent Threat (APT) activities such as botnets have the potential to cause immense disruption. As a result, recognizing potential attacks at the earliest stage possible is essential in ensuring network security. However, APT attacks generally use a Fast-Flux Service Network (FFSN) or DGA (Domain Generation Algorithm) to periodically generate a large number of domain names to serve as rendezvous points with their controllers. Accordingly, the present study

proposes a two-stage FFSN detection scheme in which a data mining algorithm is employed to detect the existence of a FFSN and a shared-domain detection algorithm based on a clustering approach and a domain blacklist is then applied to identify the nature of the FFSN through an analysis of its connections and communications. The feasibility of the proposed scheme is demonstrated by analyzing five real-world datasets containing benign and FFSN domains.

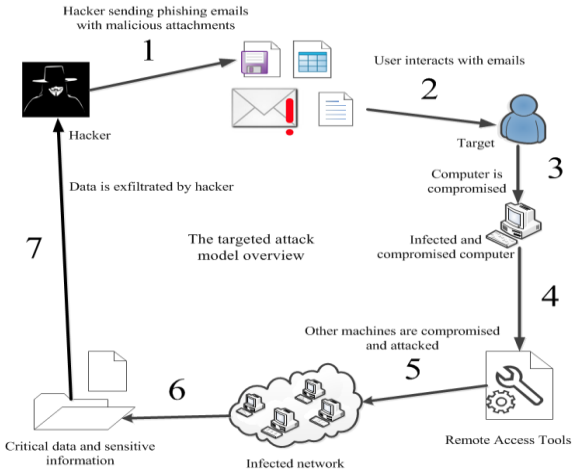

Fig. 1. Typical malicious attack model.

The experimental results show that the proposed two-stage scheme achieves both a higher detection accuracy than existing methods such as GRADE, Flux-Score, FFBD and SSFD and a lower detection delay. Notably, the results also identify a new class of FFSN architecture known as N-flux, in which the change rate of the IP addresses and name server addresses are slower than in a typical FFSN (in some cases even slower than those of some benign domains), but the IP addresses and name server addresses are shared among different families of fast-flux domains.

The remainder of this paper is organized as follows. Section 2 reviews various fast flux detection methods presented in the literature. Section 3 describes the two-stage FFSN detection scheme proposed in the present study. Section 4 presents and discusses the experimental results. Finally, Section 5 provides some brief concluding remarks.

2. Related Work

FFSNs are commonly used by hackers as a means of hiding their DNS behaviors. A FFSN typically translates the host names into IP addresses and returns an A or NS record for each query. In the event of a suspected FFSN attack, network troubleshooting is generally performed using Domain Information Groper (DIG), a command-line tool for querying DNS name servers. The Honeynet Project [1] has categorized two different types of FFSN, namely single-flux and double-flux. Single-flux service networks change DNS records as often as every 3~10 minutes and hence the TTL is very short. Furthermore, even if one flux agent is shut down, other flux-agent nodes are standing by and quickly take its place.

Many methods have been proposed for the detection of FFSNs. For example, Wu *et al.* [2] analyzed fast-flux domains using a data mining method based on the Naïve Bayes classifier and a K-Nearest Neighbors (KNN) algorithm. Tyagi and Aghila [3] presented an Analysis Based Detection Technique (ABDT) for FFSN-based social bots using a geographically-dispersed set of proxy hosts. Wang *et al.* [4] presented a method for identifying FFSNs in real time using a localized spatial geolocation detection (LSGD) system. Many methods have been proposed for identifying FFSNs by means of temporal or spatial features [4]-[9], [10]. For example, Huang *et al.* [11] proposed a SSFD (Spatial Snapshot Fast-flux Detection) system for the real-time detection of FFSNs based on two novel spatial measures, namely spatial distribution estimation and spatial service

relationship evaluation. Kadir *et al.* [12] utilized a visualization approach to monitor and classify over 500 domains, and found a new type of fluxing behavior which they termed NSName-Flux (NF). Hu *et al.* [13] proposed a multi-level support vector machine (SVM) classifier for discriminating between malicious and benign domains based on their DNS behavioral characteristics. Lin *et al.* [14] proposed a novel detection scheme designated as Genetic-based ReAl-time DEtection (GRADE) for identifying FFSNs in real time based on two new characteristics, namely the entropy of domains of preceding nodes for all A records and the standard deviation of the round trip time to all A records.

The present study extends the detection scheme proposed in [14] to construct a new two-stage framework for analyzing and identifying FFSNs. In the proposed scheme, a data mining process is first performed to detect potential FFSNs and a shared-domain detection algorithm is then applied to identify the true nature of the FFSN through an analysis of its malware connections. According to the results and observations presented in [4]-[6], [8]-[14], FFSNs have two main features: The TTL value for each A record is very short (typically 1800 seconds), the number of distinct ASNs and A records is more than two distinct ASNs.

As will be discussed later, this study identifies a further class of FFSN designated as N-flux service networks, in which the IP addresses and name server addresses are shared among different families of fast-flux domain.

3. Proposed Two-Stage FFSN Detection Scheme

This section introduces the two-stage FFSN detection method proposed in this study. Fig. 2 illustrates the proposed approach.

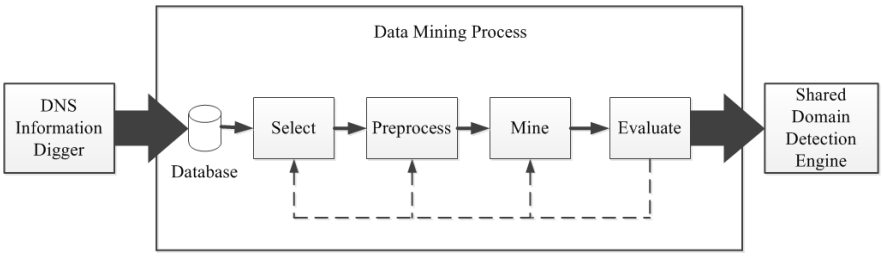

Fig. 2. System flowchart.

1) DNS Information Digger: The collected information includes A records, NS records, TTL values, and so on. The information collected by the digger module is stored in a database and is then processed by a data mining algorithm.
2) Data Mining Process: The data mining process commences by selecting certain attributes to mine, as shown in Table 1. The information retrieved from the database in accordance with the selected features is then preprocessed (normalized) and converted to ASCII values for subsequent data mining. In performing the mining process, three different supervised learning approaches are considered, namely SVM (Support Vector Machine), Naïve Bayes and K-Nearest Neighbors (K-NN).

Table 1. Experimental Feature Set

date	domain_name	A_rec	NA_rec	TTL	ASN_number
2014-04-10	www.igps.es	81.88.48.79	dns2.nominalia.com	300	39729
…	…	…	…	…	…

The details of the three classifiers are provided in the following.

a. SVM (Support Vector Machine) classifier [15], [16]: In this study, the performance of four different SVM kernels was evaluated, namely linear, polynomial, radial basis function (RBF) and sigmoid. The four kernels are defined respectively as

$$\text{Linear: } K(X_i, X_j) = X_i^T X_j \quad (1)$$

$$\text{Polynomial: } K(X_i, X_y)=(\Upsilon X_i^T X_j + r)^d, \ \Upsilon>0, \quad (2)$$

$$\text{RBF: } K(X_i, X_y)=\exp(-\Upsilon||X_i-X_y||^2), \Upsilon>0, \quad (3)$$

$$\text{Sigmoid: } K(X_i, X_y)=\tanh(\Upsilon X_i^T X_j + r), \quad (4)$$

where Υ, r and d are kernel parameters.

b. Naïve Bayes classifier [17]: The Naïve Bayes classifier is based on Bayes' theorem, a simple probabilistic model. Passerini *et al.* [18] used the Naïve Bayes classifier to discriminate between benign domains and fast-flux domains based on nine selected features. In the present study, five features are used in the first stage of the detection process, as shown in Table 2.

Table 2. Classification Feature Quantification

Classification Feature	DNS Record	Domain Type Classification Groups
F1. Total number of unique IPs	A	1. [CDN, non-CDN, FFx1_Narec, MAL]
		2. [FFx2, FFx1_Arec]
	NA	1. [FFx2, CDN]
		2. [FFx1_Narec]
		3. [MAL, non-CDN, FFx1_Arec]
F2. A & NA record overlap	A & NA	1. [non-CDN, CDN]
		2. [FFx1_Narec, FFx1_Arec, MAL, FFx2]
F3. Number of distinct ASNs	A	1. [CDN]
	NA	2. [MAL, non-CDN, FFx2, FFx1_Arec, FFx1_Narec]
F4. Time-to-live of DNS records	A	1. [CDN]
		2. [MAL]
		3. [FFx1_Narec, non-CDN]
		4. [FFx2, FFx1_Arec]
	NA	1. [CDN]
		2. [MAL]
		3. [FFx1_Arec, non-CDN]
		4. [FFx2, FFx1_Narec]
F5. IP from wrong continent	A	1. [CDN]
		2. [MAL, non-CDN, FFx2, FFx1_Arec, FFx1_Narec]

In the present study, the five features shown in Table 2 were applied to five real-world datasets in order to determine the corresponding domain types. In Table 2, F1 classifies the domains in accordance with their distinct IP address, while F2 groups the domains based on duplicated DNS records. F3 categorizes the domains in accordance with the number of distinct ASNs. F4 classifies the domains in accordance with their TTL values. Finally, F5 categorizes the domains in accordance with the geographical proximity of the IP to the user. The second column in the table shows the record type(s) processed by each feature. Finally, the third column shows the potential domains identified by each feature.

c. K-Nearest Neighbors classifier [19]: The K-Nearest Neighbors (K-NN) classifier is based on similarly measurements and is a form of lazy learning method. In the present study, the KNN classifier was implemented using the Euclidean distance metric.

3) Shared-Domain Detection Engine: This module performs a cluster-based detection process based on *malcom* [20] to determine the sharing of IP addresses and name servers among inter-domains, inter-families and intra-families of the FFSNs identified in the data mining process. Table 3 shows the

share-factor of a typical FFSN. Basically, the share-factor indicates the degree of sharing of the ANS and NS records among the identified FFSN domains. For example, in Table 3, 52 ANS records are shared among 367 domains, giving a share factor of 7.06.

The aim of share domain detection engine is to make malware analysis, detect central command and control (C&C) servers and observe DNS fast-flux infrastructures. Its approach is to find neighboring nodes and cross-reference network communication with known malware sources. By *malcom*, we found the connections via the shared IP among the cluster-based fast-flux domains.

Table 3. Share-Factor of Typical FFSN

Type	Number	Share-factor (average)
Domain	367	*n / a*
Authoritative Name Server	52	367 / 52 = 7.06
Name server	174	367 / 174 = 2.11

4. Experimental Results

This section presents and discusses the experimental results obtained in this study for the proposed two-stage FFSN detection system. Section 4.1 introduces the datasets used for training and testing purposes. Finally, Section 4.2 presents and evaluates the experimental results.

4.1. Experimental Datasets

The feasibility of the proposed detection system was evaluated using a database consisting of benign and FFSN domains. Five datasets were drawn from public sources, namely Zeus and Spyeye Tracker [21], ATLAS [22], Alexa [23], DNSBL [24] and DNSBH [25]. The details of each dataset are shown in Table 4.

Table 4. Experimental Datasets

Dataset	Instances	Collection Time	Category
Alexa	10,000	May, 2014	Benign
***Zeus* and *Spyeye* Tracker**	526	September 2013 - April 2014	FFSN
ATLAS	1,200	September 2013 - April 2014	FFSN
DNSBL	13,010	September 2013 - April 2014	FFSN
DNSBH	2,137	September 2013 - April 2014	FFSN

Table 5. Training and Testing Datasets

Dataset	Description
TR-1	70% of data training with *Zeus* and *Spyeye* Tracker, Alexa, DNSBL, DNSBH, and ATLAS
TR-2	80% of data training with *Zeus* and *Spyeye* Tracker, Alexa, DNSBL, DNSBH, and ATLAS
TE-1	30% of data training with *Zeus* and *Spyeye* Tracker, Alexa, DNSBL, DNSBH, and ATLAS
TE-2	20% of data training with *Zeus* and *Spyeye* Tracker, Alexa, DNSBL, DNSBH, and ATLAS
TE-3	Top 500 benign domains and all FFSN domains

As shown in Table 4, the Alexa dataset contains 10000 benign domains, while Zeus and Spyeye Tracker, ATLAS, DNSBL and DNSBH contain 526, 1200, 13010 and 2137 FFSN domains, respectively. Moreover as shown in Table 5, two datasets (TR-1 and TR-2) comprising 70% and 80% of the data in the five experimental datasets, respectively, were used for training purposes, while three datasets (TE-1, TE-2 and TE-3) were used for testing purposes. The dataset of TE-3 were collected from the top 500 Alexa domains [26], and the FFSN domains are the same as the dataset TE-2.

4.2. Experimental Results

Table 6 summarizes the experimental results obtained for the testing datasets using the three different

classification methods (SVM, Naïve Bayes and K-NN).

Table 6. Experimental Results

Dataset	Method	Accuracy Ratio (%)	Precision (%)	Recall (%)	F-Measure (%)	ROC Area (%)
TE - 1	SVM	92.8	93.8	94.6	94.2	97.5
	Naïve Bayes	78.6	83.6	85.1	84.3	91.4
	K-NN	88.1	89.4	92.2	90.8	96.1
TE - 2	SVM	96.7	96.5	98.1	97.3	98.9
	Naïve Bayes	85.3	88.1	89.5	88.8	92.2
	K-NN	92	93.1	94.3	93.7	97.3
TE - 3	SVM	86.3	91.8	93.1	92.5	94.3
	Naïve Bayes	77.7	87.1	87.5	87.3	90.7
	K-NN	84.4	91.1	91.6	91.4	95.2

As shown in Table 6, the performance of the proposed scheme was evaluated using five standard performance measures [27], namely the accuracy, the precision, the recall, the F-Measure, and the ROC curve area. Note that the results presented in the table relate to the first-stage (i.e., the data mining stage) of the two-stage detection process. In other words, shared-domain detection is not performed. The various performance measures are defined as follows:

Accuracy ratio:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

F-Measure:

$$\text{F - Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Finally, for the ROC curve metric, a larger area under the curve indicates an improved performance. In the equations above, TP indicates correctly classified fast-flux domains, TN denotes correctly classified benign domains, FN indicates the misclassification of fast-flux domains as benign domains, and FP denotes the misclassification of benign domains as fast-flux domains.

As shown in Table 6, SVM yields the highest recall value (98.1) in all the performance metric of the three classifiers for all three testing datasets. Furthermore, the maximum value of the ROC area is obtained when applying SVM to TE-2. Comparing the experimental results for datasets TE-1 and TE-2, all three classifiers achieve an improved performance for the latter dataset due to the greater number of instances (80%) used for training purposes. In general, the results presented in Table 6 show that the K-NN classier outperforms the Naïve Bayes classifier for all three datasets. Again, this finding is reasonable since the Naïve Bayes classifier is based on the so-called Bayesian theorem with a balanced class distribution, whereas in the present experimental datasets, the classes (i.e., benign or FFSN) are imbalanced. All three classifiers achieve a relatively poor performance when applied to the TE-3 dataset since most of the domains in this dataset are fast-flux service networks.

In the first stage of the proposed detection framework, misclassification errors readily occur since some of the interactions appear to have a normal behavior with no fast-flux characteristics. Accordingly, in the second stage of the proposed framework, a Shared-Domain Detection Engine is employed to more reliably determine the true nature of the FFSNs identified in the data mining process.

Fig. 3 compares the accuracy of the proposed two-step framework with that of GRADE [14], FFBD [10], SSFD [16] and Flux-Score [28]. It is seen that the proposed system achieves the highest accuracy of the five schemes. It is noted that GRADE also achieves a good accuracy since this method was also proposed by entropy of domains and the standard deviation of the round trip time. Notably, Table 6 shows that the proposed scheme achieves a maximum accuracy of 96.7%, whereas Fig. 3 gives a slightly higher value of 97.6%. The higher value in the latter case is reasonable since the results in Table 6 are obtained using data mining only, whereas the result in Fig. 3 is obtained using both data mining and *malcom* clustering. In other words, the results confirm the effectiveness of the *malcom* clustering process in minimizing the number of FN outcomes.

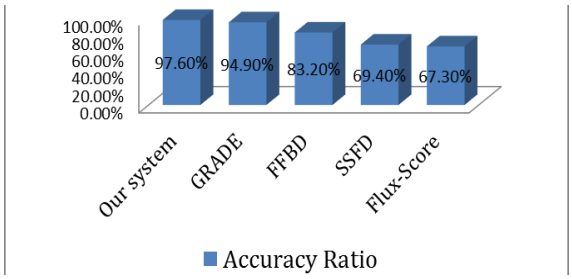

Fig. 3. Accuracy ratio of compared schemes.

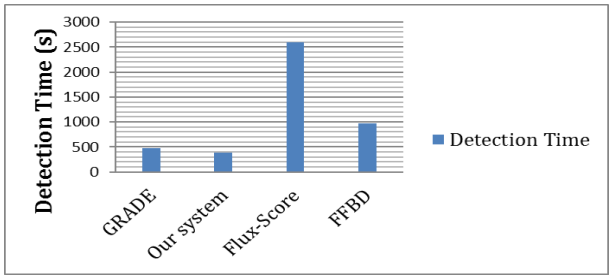

Fig. 4. Detection time of compared systems.

SSFD and Flux-Score both achieve a relatively low accuracy since they suffer from the missing value problem [5]. Moreover, they cannot identify shared inter-domains, inter-families and intra-families since they focus only on malware infection. Fig. 4 compares the detection time of the proposed two-stage framework with that of GRADE, Flux-Score and FFBD. It is seen that the proposed framework has the lowest detection time of the four schemes. Moreover, the detection time of GRADE is significantly lower than that of Flux-Score or FFBD. However, GRADE does not consider the share-factor that we discussed before, see Table 3. The reliability of the experimental results was evaluated using the k-fold cross-validation method, in which the data were segmented into k equal-sized partitions.

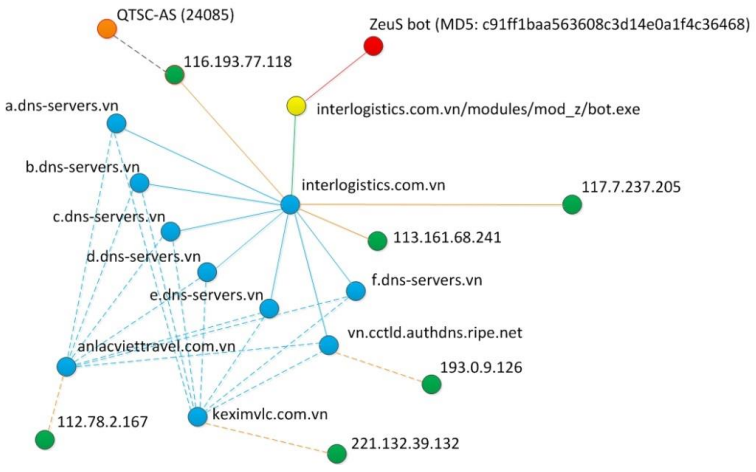

Fig. 5. Misclassification of fast-flux domain.

In Fig. 5, the solid lines represent Zeus bot infections. In the first stage of the proposed two-stage classification process, these interactions appear to have a normal behavior with no fast-flux characteristics,

and are thus easily misclassified as benign. The dotted lines relate to fast-flux techniques employed by the bot to prolong its survival. The notation QTSC-AS (24085) indicates that the ASN number is 24085. The node is thus another Zeus bot. In the first stage of the proposed detection framework, misclassification errors readily occur since some of the interactions appear to have a normal behavior with no fast-flux characteristics. Accordingly, in the second stage of the proposed framework, a Shared-Domain Detection Engine is employed to more reliably determine the true nature of the FFSNs identified in the data mining process.

5. Discussion and Conclusion

This study has proposed a two-stage framework for the detection of FFSNs comprising a data mining operation followed by a shared-domain detection process. In the proposed approach, the data mining process is used to distinguish between benign domains and FFSN domains and the shared-domain detection process is then used to confirm whether or not the identified FFSN domains are truly malicious. Notably, through the shared-domain detection engine, the proposed scheme has the ability to detect a new class of FFSNs, denoted in this study as N-flux, in which the domains exhibit a normal DNS behavior, but have a high degree of inter-domain, inter-family and intra-family sharing. Traditional FFSN detection schemes use temporal and spatial features to identify the existence of FFSNs. However, such schemes frequently have a long detection delay, thereby prolonging the life of the network and permitting continued malware distribution. Our proposed system in the first stage uses data mining approach to find FFSN from datasets. In the second stage, we proposed the share-factor to classify whether it is FFSN or not. The share-factor is used to compute malware connections to identify a FFSN. Since, today's FFSN conceals with lots of normal domains, as a result, the two-stage detection system can efficiently and effectively classify the possible FFSN domains in the shortest amount time. By contrast, the experimental results obtained in this study have shown that the proposed scheme has the ability to detect FFSNs with a minimal detection time. As a result, it provides network administrators with a more effective countermeasure in thwarting malicious attacks.

Acknowledgment

This study was supported by the Ministry of Science and Technology (MOST) of Taiwan under Contract Nos. MOST 100-2218-E-006-029-MY3, MOST 103-2221-E-006 -146 -MY3 and MOST 104-2221-E-492 -014 -MY2.

References

[1] Know Your Enemy: Fast-Flux Service Networks. Retrieved July 21, 2015, from http://www.honeynet.org/papers/ff

[2] Wu, J. Y., Zhang, L. W., L., Jian, & Qu, S. (2010). A comparative study for fast-flux service networks detection. *Proceedings of 2010 Sixth International Conference on Networked Computing and Advanced Information Management*.

[3] Tyagi, A. K., & Aghila, G. (2012). Detection of fast flux network based social bot using analysis based techniques. *Data Science & Engineering*.

[4] Wang, H.-T., Mao, C.-H., Wu, K.-P., & Lee, H.-M. (2012). Real-time fast-flux identification via localized spatial geolocation detection. *Proceedings of IEEE 36th International Conference on Computer Software and Applications.*

[5] Thorsten, H., Christian, G., Konrad, R., & Felix, C. F. (2008). Detection and mitigation of fast-flux service networks. *Proceedings of the 15th Annual Network & Distributed System Security Symposium.*

[6] Hsu, C.-H., Huang, C.-Y., & Chen, K.-T. (2010). Fast-flux bot detection in real time. *Proceedings of the*

13th International Conference on Recent Advances in Intrusion Detection.

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

[8] Caglayan, A., Toothaker, M., Drapaeau, D., Burke, D., & Eaton. G. (2009). Real-time detection of fast flux service networks. *Proceedings of Cybersecurity Applications & Technology Conference for Homeland Security.*

[9] Huang, S.-Y., Mao, C.-H., & Lee, H.-M. (2010). Fast-flux service network detection based on spatial snapshot mechanism for delay free detection. *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security.*

[10] He, W. S., Hu, G. M., & Zhou, Y. J. (2012). Large-scale IP network behavior anomaly detection and identification using substructure-based approach and multivariate time series mining. *Telecommunication Systems, 50(1),* 1-13.

[11] Huang, S.-Y., Mao, C.-H., & Lee, H.-M. (2010). Fast-flux service network detection based on spatial snapshot mechanism for delay free detection. *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security.*

[12] Andi, F. A. K., Raja, A. R. O., & Normaziah, A. A. (2012). Behavioral analysis and visualization of fast-flux DNS. *Proceedings of European Intelligence and Security Informatics Conference.*

[13] Xin, H., Matthew, K., & Kang, G. S. (2011). Measurement and analysis of global IP-usage patterns of fast-flux botnets. *IEEE INFOCOM*.

[14] Lin, H.-T., Lin, Y.-Y., & Chiang, J.-W. (2013). Genetic-based real-time fast-flux service networks detection. *Elsevier Computer Networks, Special Issue on Botnet Activity: Analysis, Detection and Shutdown, 57(2),* 501-513.

[15] Christopher, J. C. B. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.

[16] Iftikhar, A., Azween, A., Abdullah, A., & Muhammad, H. (2013). Optimized intrusion detection mechanism using soft computing techniques, *Telecommunication Systems, 52(4),* 2187-2195.

[17] Kamber, J. H. M. (2006). *Data Mining: Concepts and Techniques* (2nd ed.). 186-206.

[18] Passerini, E., Paleari, R., *et al*. (2008). Fluxor: Detecting and monitoring fast-flux service networks. *Detection of Intrusions and Malware, and Vulnerability Assessment Lecture Notes in Computer Science, 5137,* 186-206.

[19] Liao, Y., & Vemuri, V. R. (2002). Use of K-nearest neighbor classifier for intrusion detection. *Computers & Security, 21(5)*, 439-448.

[20] Malcom. Retrieved July 21, 2015, from https://github.com/tomchop/malcom

[21] Zeus and Spyeye Tracker. Retrieved July 21, 2015, from http://www.abuse.ch

[22] ATLAS: Global fast flux report. Retrieved July 21, 2015, from https://atlas.arbor.net/summary/fastflux

[23] FluXOR and Alexa of SSFD public dataset. Retrieved December 30, 2014, from https://sites.google.com/site/huangpublication/datasets/-1-fast-flux-attaackdatasets

[24] DNSBL. Retrieved July 21, 2015, from http://www.dnsbl.info

[25] DNSBH. Retrieved July 21, 2015, from http://www.malwaredomains.com/wordpress

[26] Alexa. Retrieved July 21, 2015, from http://www.alexa.com/topsites

[27] Tan, P.-N., Michael, S., & Vipin, K. (2006). *Introduction to Data Mining. PEARSON Addison Wesley*, New York.

[28] Gao, Y., Li, Z., & Chen, Y. (2006). A dos resilient flow-level intrusion detection approach for high-speed networks. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems* (p.

39).

Ci-Bin Jiang received his M.S. degree from the Department of Information Management of Southern Taiwan University, Taiwan, in 2009. His current research interests are in the areas of data mining and network security.

Jung-Shian Li is a full professor in the Department of Electrical Engineering, National Cheng Kung University, Taiwan. He graduated from the National Taiwan University, Taiwan, with BS in 1990 and MS degrees in 1992 in electrical engineering. He obtained his PhD in 1999 in computer science from the Technical University of Berlin, Germany. He teaches communication courses and his research interests include wired and wireless network protocol design, network security, and network management. He is currently involved in funded research projects dealing with optical network, VANET, cloud security and resource allocation, and IP QoS architectures. He is the deputy director of computer and network center, NCKU. He serves on the editorial boards of the International Journal of Communication Systems.