PAPER Special Section on Log Data Usage Technology and Office Information Systems

A Malicious Web Site Identification Technique Using Web Structure Clustering

Tatsuya NAGAI[†a)], **Masaki KAMIZONO**[††], ***Nonmembers***, **Yoshiaki SHIRAISHI**[†,†††b)], ***Senior Member***, **Kelin XIA**[††††], ***Nonmember***, **Masami MOHRI**[†††††], ***Senior Member***, **Yasuhiro TAKANO**[†], ***Member***, ***and*** **Masakatu MORII**[†], ***Fellow***

SUMMARY Epidemic cyber incidents are caused by malicious websites using exploit kits. The exploit kit facilitate attackers to perform the drive-by download (DBD) attack. However, it is reported that malicious websites using an exploit kit have similarity in their website structure (WS)-trees. Hence, malicious website identification techniques leveraging WS-trees have been studied, where the WS-trees can be estimated from HTTP traffic data. Nevertheless, the defensive component of the exploit kit prevents us from capturing the WS-tree perfectly. This paper shows, hence, a new WS-tree construction procedure by using the fact that a DBD attack happens in a certain duration. This paper proposes, moreover, a new malicious website identification technique by clustering the WS-tree of the exploit kits. Experiment results assuming the D3M dataset verify that the proposed technique identifies exploit kits with a reasonable accuracy even when HTTP traffic from the malicious sites are partially lost.

key words: *website structure, malicious website, exploit kit, clustering*

1. Introduction

Malicious websites are severe cyber threats because they spread malwares. The malicious websites exploit vulnerabilities in web browsers and their plugins and perform drive-by download (DBD) attacks. DBD attacks start in users host as a background process. Figure 1 illustrates a DBD attack. The DBD attack is performed by landing, intermediate, exploit host and malware distribution host. The landing host is a compromised website and redirects a victim to intermediate host. The intermediate host identifies victim's system environment. The exploit host exploits vulnerabilities in the victim's system, which makes the victim automatically download and execute a malware via the malware distribution host.

For the DBD attacks, we have utilized *blacklists* to avoid or stop accessing to the path of DBD attacks. However, the blacklists must be updated frequently since URLs of the malicious websites emerge and disappear in a short duration. Akiyama et al. [1] proposed a method to improve blacklists by searching neighborhood of the URLs. Mamun et al. [2] performed lexical analysis of the URLs by machine learning algorithms. These methods can detect malicious websites by finding similarity is the URL strings. However, the URL can be completely different depending on the visited host if each phase in the DBD attack is composed of multiple hosts as mentioned in [3]. Thus, detection methods using one URL do not always succeed.

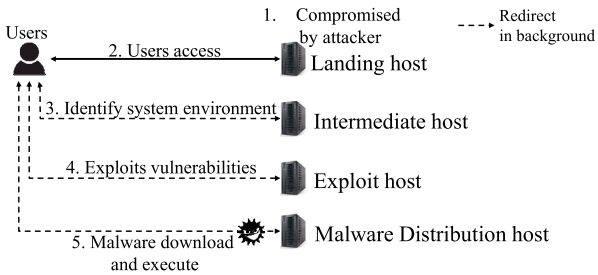


Fig. 1 Drive-by Download attack

It is reported that massive malicious websites are generated from specific exploit kits (e.g., [4]). An exploit kit has JavaScripts and management tools so that attackers can easily arm and update their weapons flexibly. The malicious websites can, moreover, use obfuscation codes and/or can change their behavior according to victims' browsers.

Malicious website detection techniques are, hence, necessary to cope with such the clever attacks. For example, detection methods focusing on the attack-codes have been studied. Kolbitsch et al. [5] proposed a greedy inspection technique for malicious websites by executing suspicious JavaScripts on all possible platforms. Takata et al. [6] proposed a URL extracting technique by constructing abstract syntax trees and analyzing them. Kim et al. [7] proposed a detection method by applying the strength analysis for obfuscated JavaScripts. Nevertheless, the website analysis methods only using HTTP contents do not perfectly detect all malicious sites, since attackers can also camouflage attack-codes as to avoid the defense strategies.

We can improve detection techniques by leveraging structure in traffic data. This is because some invariants are assumed to be left behind in the communicated protocols transmitted from a specific exploit kit. Under the assumption, Mekky et al. [8] analyzed HTTP redirection

Manuscript received November 12, 2018.
Manuscript revised April 7, 2019.
Manuscript publicized June 21, 2019.

[†]The authors are with the Department of Electrical and Electronic Engineering, Kobe University, Kobe-shi, 657–8501 Japan.
[††]The author is with PwC Cyber Services, Tokyo, 100–0004 Japan.
[†††]The author is with Center for Mathematical and Data Sciences, Kobe University, Kobe-shi, 657–8501 Japan.
[††††]The author is with Nanyang Technological University, 637371 Singapore.
[†††††]The author is with the Department of Electrical, Electronic and Computer Engineering, Gifu University, Gifu-shi, 501–1193 Japan.
a) E-mail: t.nagai@stu.kobe-u.ac.jp
b) E-mail: zenmei@port.kobe-u.ac.jp
DOI: 10.1587/transinf.2018OFP0010

graphs of the malicious sites by using machine learning algorithms. Stringhihni et al. [9] also investigated malicious HTTP redirection graphs from extensively collected web surfing datasets. HTTP redirections are, however, marked by HTTP header and/or HTTP contents. Thus, these methods are also difficult to cope with the camouflage problem.

As an advanced detection technique, Zhang et al. [3] took an approach by constructing trees of attack hosts from their URL's hierarchical structure. The method extracts common features between malicious websites by generating signature from a malicious central host. Since DBD attacks can have multiple hosts, the analysis assuming a single host is also difficult to capture the behavior of the malicious websites.

The above researches aim to classify an HTTP flow into benign or not. When an incident caused by malicious web sites occurred in organizations, system administrator and security personnel shall investigate in which environment the incidents occurred, what client was redirected, and what vulnerability was exploited in order to understand the cause and damage. After the HTTP flow classification, understanding behavior of website from malicious HTTP flow is necessary in order to early recover or to minimize the damage.

Under the aim of understanding website behavior, we propose a new exploit kit identification method by using all URLs which belong to DBD attack hosts. It should be emphasized that our method is feasible even if a part of HTTP redirection data has lost, since it only utilizes URL lists accessed from a user. Our method can identify exploit kits by clustering the malicious website structures. Effectiveness of the proposed method is evaluated by using D3M dataset [10].

This paper is organized as follows. Section 2 details the background of this study. Section 3 proposes the new malicious website identification method. Section 4 presents experiment results to show effectiveness of the proposed method. We conclude the paper in Sect. 5.

2. Related Work about Exploit Kit

2.1 Exploit Kit's Components

Kotov et al. [11] showed typical components of exploit kits based on the source codes analysis of 30 exploit kits. Maio et al. [12] also analyzed the source codes of exploit kits and reported behavior of exploit kit at the server side. According to [11] and [12], exploit kits have the following four components:

Offensive Component that attacks vulnerable machines according to user's environments identified from the user-agent and/or the accept-language in the HTTP protocol.

Defensive Component that performs the IP blocking for multiple-visit users in order to avoid detection by malware scanners.

Management Component which facilitates attackers to install the exploit kit and provides control and monitor functionality of the attack activities.

Protection Mechanisms which prevent from reverse-engineering by obfuscating themselves, since the exploit kits are sharewares.

2.2 Related Work

We note that ordinary attackers are difficult to change the protected codes. Thus, the websites using the same exploit kit are assumed to have similarity although they are customized by the management component. Under the assumption, detection techniques have been studied by investigating features in a) the attack codes, b) the URLs, or c) the web session trees of the malicious websites:

a) Stoke et al. [13] generates a signature to identify exploit kits from keyword, identifier, string and delimiter by parsing tokens in the JavaScripts. Luo et al. [14] proposed a clustering method by abstracting JavaScript codes. The method identifies the type of exploit kits and the obfuscator version. Although these methods can be performed without actually executing of JavaScript codes, the identification performance is not always guaranteed for polymorphic attack codes [13].

b) Machine learning approaches [15], [16] can be utilized to identify whether a URL was generated by an exploit kit. However, the approaches simply inspecting a single URL can fail to detect the malicious websites, since the defensive component can abort the redirection for machines which they recognize no-targeted user environment. It should be noticed that, moreover, a complete URL sequence accessed during a DBD attack is required to capture all the URL redirections correctly.

c) Taylor et al. [17] proposed a detection method by using *web session tree* generated from web requests. The method identifies malicious websites by comparing the web session trees of suspicious sites with that of known exploit kits. However, in practice, we can observe the subtrees of web session trees only. This is because the redirection information cannot always be obtained perfectly due to the defensive component.

2.3 Website Structure Tree

This subsection reviews the conventional technique to describe a web session tree, where this paper refers the tree as *website structure (WS)-tree*. The structure of a DBD attack-site can be obtained from HTTP traffic data by finding redirection events. Specifically, we can obtain URLs used in the DBD attack by searching Referer and Location properties.

Figure 2 shows an example URLs used for a DBD attack, where dashed arrows indicate redirection events. As depicted in Fig. 2, the URL sequence has a tree structure, where the top-most "/" is the root node. A URL can be delimited by the slash character ("/"), where the first strings denote the host part linked to the root node. The rests are

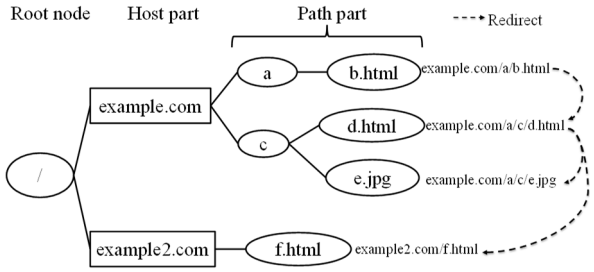


Fig. 2 Example of WS-tree

the path part. The WS-tree can be utilized to identify websites. As above-mentioned, however, what we can observe is, practically, a subtree of that shown in Fig. 2, since the defensive component can obscure the Referer and Location properties.

3. Our Approach

Therefore, we propose a new malicious website identification technique which can properly handle incomplete redirection information observed in traffic data from attack-sites.

Figure 3 shows overview of the proposed method. The method is composed of clustering and identification phases. In the clustering phase, first of all, we describe structure of malicious websites captured in training dataset, where the labels of exploit kits are determined by leveraging known CVE reports, etc. We perform, then, a hierarchical clustering algorithm for the WS-trees. In the identification phase, we identify labels of the exploit kits for estimated WS-tree of unknown malicious websites in test data.

3.1 Constructing WS-Tree

As discussed in Sect. 2, we can detect events of suspected DBD attacks by utilizing the WS-tree. However, the defensive component may prevent from capturing the redirection events correctly. The dynamic analysis approaches [8], [17] can compensate for the missing observations. Unlike the conventional approaches, we notice that a series of URL accesses incurred by the DBD attack happens in a certain duration. This subsection, hence, proposes a new WS-tree construction procedure, after defining *observed- and effective-periods* for a WS-tree representing a DBD attack activity.

Definition: *Observed-period* of a WS-tree is defined by the minimum and maximum time-stamps accessed to the URLs corresponding to leaf nodes of a WS-tree.

Definition: *Effective-period* of a WS-tree is duration that added a certain margin to the observed-period.

WS-tree construction procedure is shown in Fig. 4. Note that by Step 2 and 3, we can estimate the WS-tree for websites that are potentially used for a DBD attack. Even if the redirection information is lost partially, our technique

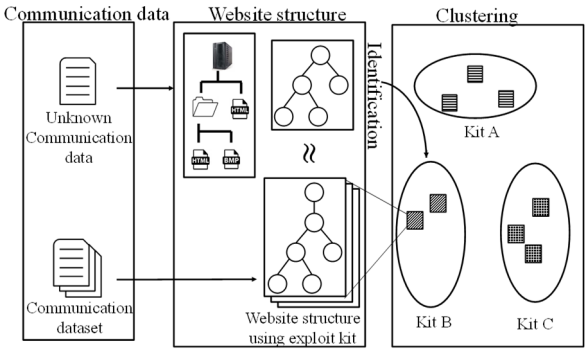


Fig. 3 Overview of proposed method

Step 0: Compose initial WS-trees according to the Referer and Location properties in HTTP traffic data, as described in Section 2.3.
Step 1: In the effective-period of the initial WS-trees, search HTTP requests without Referer and/nor Location properties and obtain a URL list.
Step 2: If the URL has the same host-part as that of the initial WS-trees, append the URL as a leaf node of the matched WS-tree.
Step 3: Otherwise, create new WS-trees for the unmatched URLs.

Fig. 4 WS-Tree construction procedure

Table 1 Number of sites constructed from D3M dataset by exploit kit

Eleonore	53
Seo	23
Cry217	8
Fragus BLK	6
CrimePack	8
Phoenix	95
BlackHole	18
Bleeding Life	12
Unknown1	16
Unknown2	17

can identify the malicious websites by performing the following clustering algorithm.

3.2 Comparing WS-Tree

We defined distance between two WS-trees T_a and T_b as:

$$D_s(T_a, T_b) = \frac{TED(T_a, T_b)}{|T_a| + |T_b|},$$

where $|T|$ is the number of nodes in tree T. The function $TED(T_a, T_b)$ is the tree edit distance [17] between trees T_a and T_b. The tree edit distance is the minimum of the sum of costs for the insertion, deletion, and substitution operations required to transform T_a from T_b. We may set all the operation costs at 1. By definition, $D_s(T_a, T_b)$ takes value

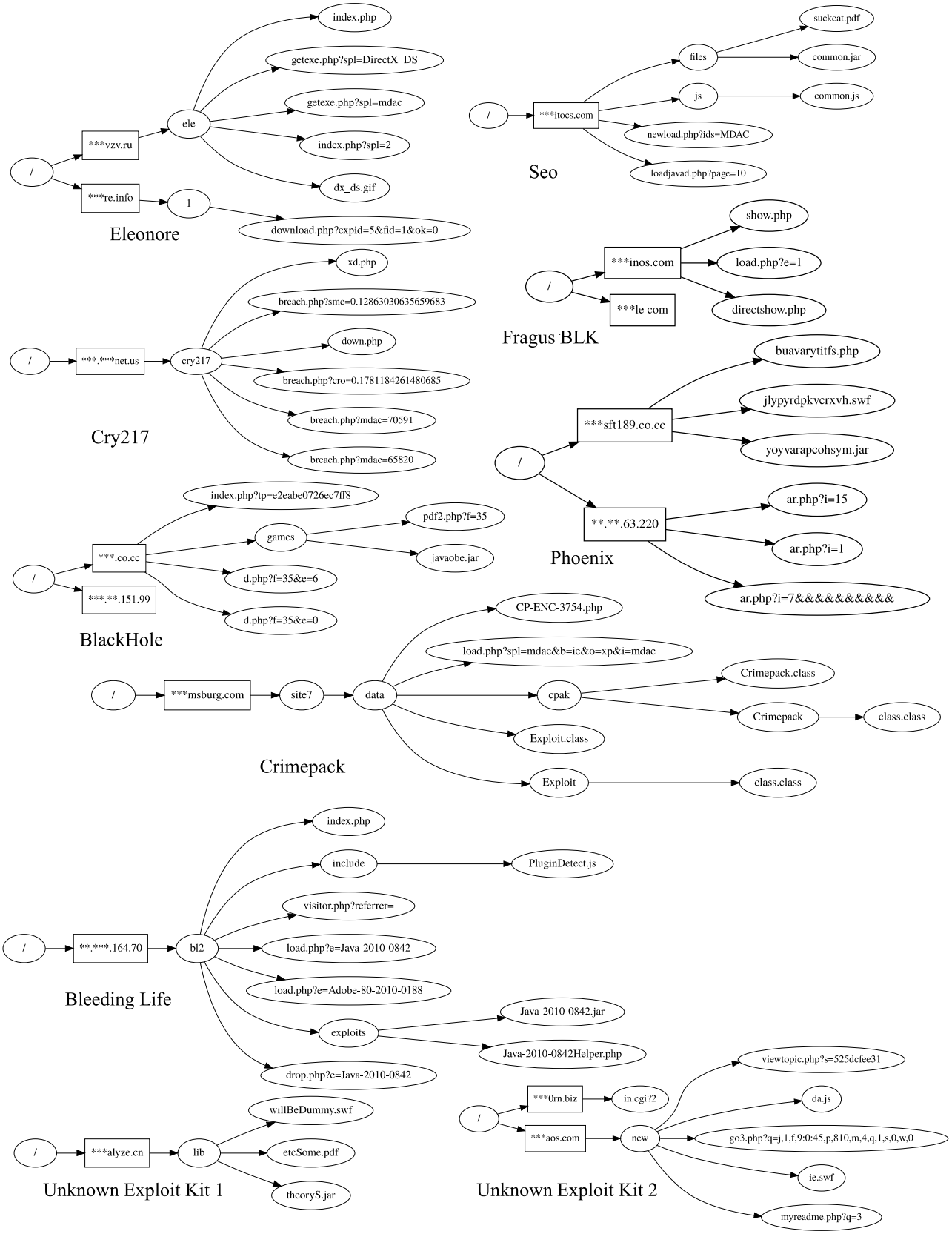


Fig. 5 Examples of WS-tree using exploit kit

Table 2 Confusion matrices of 10 times experiments

Predict labels \ Real Labels	Eleonore	Seo	FragusBLK	Cry217	BlackHole	Phoenix	Bleeding	CrimePack	Unknown1	Unknown2
Eleonore	.92	.03	.05	0	0	.01	0	0	.07	.28
Seo	.01	.92	0	0	.06	0	0	0	.03	0
FragusBLK	.01	0	.76	0	0	.01	0	0	0	.06
Cry217	.01	0	0	.75	0	0	0	0	0	0
BlackHole	0	0	0	0	.94	.03	0	0	0	0
Phoenix	.01	.03	0	.25	0	.91	0	0	.31	.03
Bleeding	.01	0	.05	0	0	.02	.97	0	0	0
CrimePack	0	0	0	0	0	0	0	1	0	0
Unknown1	.01	0	.14	0	0	.02	0	0	.48	.03
Unknown2	.03	.03	0	0	0	.02	.03	0	.10	.61

between 0 and 1. Note that $D_s(T_a, T_b) \approx 0$ indicates that the similarity between T_a and T_b is significantly high.

3.3 Clustering

The proposed technique utilizes the hierarchical agglomerative clustering with average linkage method. We initialize clusters as to have one WS-tree. We iteratively merge clusters into the closest cluster according to the cost function:

$$D_c(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{T_k \in C_i} \sum_{T_l \in C_j} D_s(T_k, T_l),$$

until the stopping criterion $D_c(C_i, C_j) > D_{th}$, $\forall(i, j)$ holds for a threshold D_{th}. The notation $|C|$ is the number of WS-trees in cluster C. We labeled the each cluster with exploit kit which the website in the cluster used.

3.4 Classifier

The classifier outputs a label of the identified exploit kit for an unknown WS-tree T_u. Specifically, the label is determined by $\mathrm{argmin}_i D_c(T_u, C_i)$ for $\forall i$.

4. Evaluation

We have implemented our method and evaluated it using a communication dataset. The following sections describe the experimental setup and evaluation results.

4.1 Experiment Setup

The D3M2010~2015 dataset [10] is utilized to verify the effectiveness of the proposed technique. The D3M dataset contains DBD attacks and the malware traffic data collected by Marionette which is a high-interaction client honeypot. The traffic of DBD attacks is extracted by using a Ruby's HTTP parser library called *packetfu*. The corrupted packets for which *packetfu* does not accept are excluded from the input. The WS-trees are estimated by performing the procedure shown in Sect. 3.1, where the margin time of the effective-period of WS-tree is set at 60 seconds[†].

In this experiment, the WS-trees are labeled as any of 10 exploit kits: Eleonore, Seo, Cry217, Fragus BLK, CrimePack, Phoenix, BlackHole, Bleeding Life, and two unknown exploit kits. According to our clustering method, the D3M dataset includes 256 malicious websites. Table 1 shows the details of the sites constructed by the exploit kits. Moreover, Fig. 5 depicts examples of the WS-trees.

We perform 2-fold cross-validation (CV) by randomly choosing 206 training and 50 test data. In the training phase, we perform clustering for the 206 sites after obtaining the WS-trees. In the test phase, as shown in Sect. 3.4, the classifier identifies unknown 50 test data.

4.2 Results

Table 2 shows a confusion matrix obtained from the 10 CVs. The threshold D_{th} is set at 0.5 in order to maximize the purity which is a performance measure of clustering.

The Y-axis represents the real labels of the exploit kits, whereas the X-axis shows the identified kits by the classifier. The average accuracy is 86.2%. It should be emphasized that, as observed from Table 2, the proposed technique achieves the accuracy more than 90% for the exploit kits such as {Eleonore, Seo, BlackHole, Phoenix, Bleeding Life, CrimePack} that follow our assumption.

Figure 6 shows the WS-tree of Eleonore that the proposed technique identifies successfully. As shown in Fig. 6, the WS-tree of Eleonore is composed of two clusters. In the first cluster, the attack-codes exploit ActiveX vulnerability, where the websites have 432.js and load.php and perform 5734.js finally. In the second cluster, a vulnerability of Java is exploited, where the websites have dx_ds.gif exploiting CVE 2008-0015 and the downloader getexe.php.

[†]We experimentally find that the DBD attacks observed in the D3M dataset are executed in 60 seconds at most.

Some of them redirect executable files. As shown in above of Fig. 6, no redirection between different hosts is observed. It should be noted that, hence, our method can identify malicious websites even if URLs and/or redirection information are partially lost due to the protection component of the exploit kits.

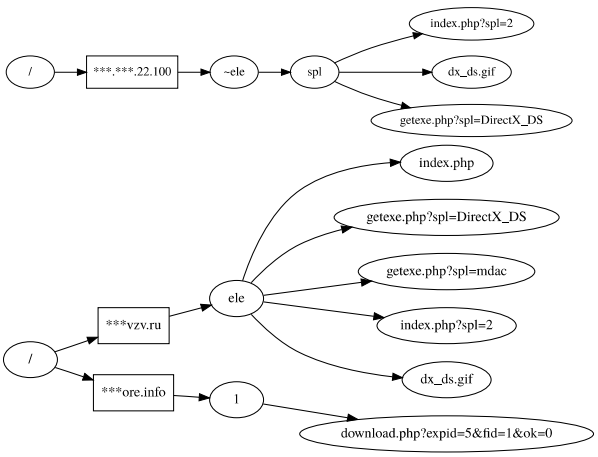


Fig. 6 Example of WS-tree succeeded in identification using Eleonore exploit kit

Figure 7 shows the WS-trees that the proposed method failed in identifying of Bleeding Life. Bleeding Life has some behavior like Eleonore's case. The failed case is caused by the fact that the WS-trees in training dataset do not have some nodes related to jQuery library even though the WS-trees in the test dataset have many nodes generated by loading jQuery library. As shown above, if training dataset do not have a WS-tree which is similar structure to unknown WS-tree, the proposed method cannot identify unknown WS-tree as known exploit kit. If a new exploit kit arises, it is expected to identify it by re-training the dataset so as to include it. This is a limitation of the proposed method.

Figure 8 (a) shows a WS-tree constructed by Taylor *et al.*'s technnique in the case of BlackHole exploit kit. Figures 8 (b) and 8 (c) show the partial WS-trees which assumed lost redirect information because of obfuscated .jar file. The tree of Fig. 8 (b) is almost same as the tree of Fig. 5, but the tree of Fig. 8 (a) is divided into the three parts shown in Fig. 8 (c). This observation verifies that the proposed method captures structural feature more accurately than the conventional technique.

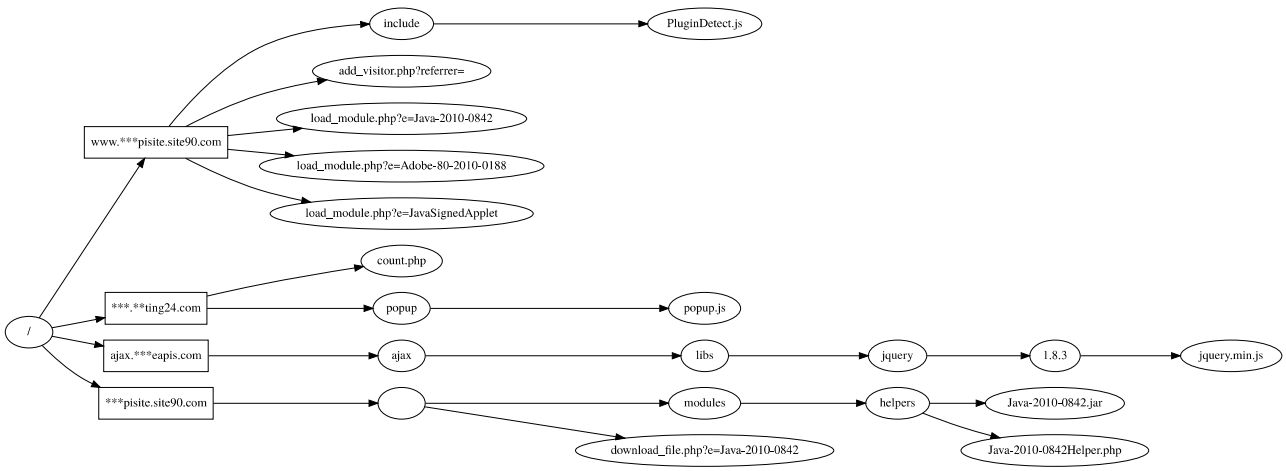
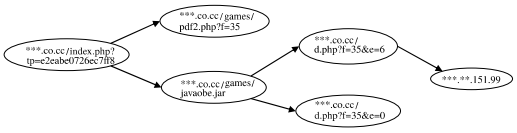
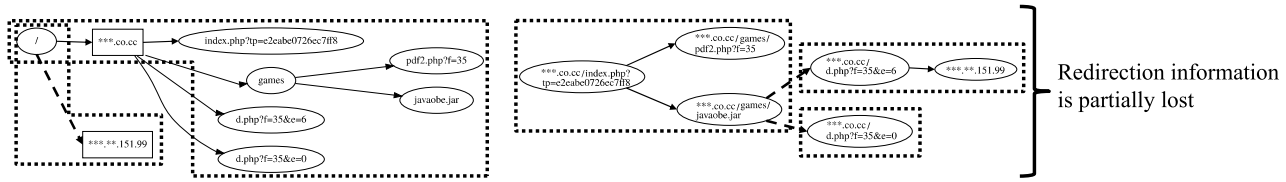


Fig. 7 Example of WS-tree of Bleeding Life exploit kit in the failed identification case



(a) Taylor et al.'s method



(b)Proposed method (c)Taylor et al.'s method

Fig. 8 Examples of WS-trees of BlackHole exploit kit in the failed identification case

5. Conclusions

This paper has proposed a novel malicious website identification technique by clustering the WS-trees of exploit kits. Under the fact that a DBD attack happens in a certain duration, the proposed method can estimate the WS-tree even if the HTTP traffic data from a malicious website is partially lost. According to the experiment results shown in this paper, the proposed technique achieves a reasonable identification accuracy of 86.2% for the D3M dataset. Moreover, this paper showed WS-trees of actual 10 exploit kits. We plan to conduct evaluation for the exploit kit raised after 2015 to the proposed method. Future work of this study is, based on the WS-tree analysis, to improve identification performance further.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 16K00184.

References

[1] M. Akiyama, T. Yagi, and M. Itoh, "Searching structural neighborhood of malicious URLs to improve blacklisting," The 11th IEEE/IPSJ International Symposium on Applications and the Internet, pp.1–10, Munich, The Germany, July 2011.
[2] M.S.I. Mamun, M.A. Rathore, A.H. Lashkari, N. Stakhanova, and A.A. Ghorbani, "Detecting Malicious URLs Using Lexical Analysis," The 10th International Conference on Network and Security, pp.467–482, Taipei, Taiwan, Sept. 2016.
[3] J. Zhang, C. Seifert, J.W. Stokes, and W. Lee, "Arrow: Generating signatures to detect drive-by downloads," Proc. 20th International Conference on World Wide Web, pp.187–196, Hyderabad, India, March 2011. DOI: 10.1145/1963405.1963435
[4] C. Grier, A. Pitsillidis, N. Provos, M.Z. Rafique, M.A. Rajab, C. Rossow, K. Thomas, V. Paxson, S. Savage, G.M. Voelker, L. Ballard, J. Caballero, N. Chachra, C.J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, and A. Nappa, "Manufacturing compromise: the emergence of exploit-as-a-service," Proc. 2012 ACM Conference on Computer and Communications Security, pp.821–832, North Carolina, The USA, Oct. 2012.
[5] C. Kolbitsch, B. Livshits, B. Zorn, and C. Seifert, "Rozzle: De-cloaking internet malware," IEEE Symposium on Security and Privacy, pp.443–457, San Francisco, The USA, May 2012. DOI:10.1109/SP.2012.48
[6] Y. Takata, M. Akiyama, T. Yagi, T. Hariu, and S. Goto, "MineSpider: Extracting hidden URLs behind evasive drive-by download attacks," IEICE Trans. Inf. & Syst., vol.99, no.4, pp.860–872, Jan. 2016. DOI: 10.1587/transinf.2015ICP0013
[7] B. Kim, I. Chae-Tae, and H. Jung, "Suspicious malicious web site detection with strength analysis of a javascript obfuscation," International Journal of Advanced Science and Technology, vol.26, pp.19–32, Dec. 2011.
[8] H. Mekky, R. Torres, Z.-L. Zhang, S. Saha, and A. Nucci, "Detecting malicious HTTP redirections using trees of user browsing activity," The 33rd Annual IEEE International Conference on Computer Communications, pp.1159–1167, Toronto, Canada, April 2014. DOI: 10.1109/INFOCOM.2014.6848047
[9] G. Stringhini, C. Kruegel, and G. Vigna, "Shady paths: Leveraging surfing crowds to detect malicious web pages," Proc. 2013 ACM SIGSAC Conference on Computer and Communications Security, pp.133–144, Berlin, Germany, Nov. 2013. DOI: 10.1145/2508859.2516682
[10] M. Kamizono, M. Akiyama, T. Kasama, J. Murakami, M. Hatada, and M. Terada, "Datasets for Anti-Malware Research ~MWS Datasets 2015~," Computer Security Group, vol.2015-CSEC-70, no.6, pp.1–8, Nagoya, Japan, July 2015.
[11] V. Kotov and F. Massacci, "Anatomy of exploit kits," Proc. 5th International Symposium on Engineering Secure Software and Systems, vol.7781, pp.181–196, Paris, France, Feb. 2013.
[12] G.D. Maio, A. Kapravelos, Y. Shoshitaishvili, C. Krugel, and G. Vigna, "Pexy: The other side of exploit kits," International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp.132–151, London, United Kingdom, July 2014. DOI: 10.1007/978-3-319-08509-8_8
[13] B. Stock, L. Benjamin, and B. Zorn, "Kizzle: A signature compiler for detecting exploit kits," The 46th Anuual IEEE/IFIP International Conference on Dependable Systems and Networks, pp.455–466, Toulouse, France, June 2016. DOI: 10.1109/DSN.2016.48
[14] T. Luo and J. Xing, "Next-generation of exploit kit detection by building simulated obfuscators," Black Hat USA 2016, Mandalay Bay, The USA, July 2016.
[15] B. Eshete and V.N. Venkatakrishnan, "Webwinnow: Leveraging exploit kit workflows to detect malicious URLs," Proc. 4th ACM Conference on Data and Application Security and Privacy, pp.305–312, Texas, The USA, March 2014. DOI: 10.1145/2557547.2557575
[16] Y. Sato, Y. Nakamura, H. Inamura, and O. Takahashi, "A proposal of malicious URLs detection based on features generated by exploit kits," International Workshop on Infomatics, Riga, Latvia, Aug. 2016.
[17] T. Taylor, X. Hu, T. Wang, J. Jang, M.P. Stoecklin, F. Monrose, and R. Sailer, "Detecting malicious exploit kits using tree-based similarity searches," Proc. Sixth ACM Conference on Data and Application Security and Privacy, pp.255–266, Louisiana, The USA, March 2016. DOI: 10.1145/2857705.2857718
[18] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," The Society for Industrial and Applied Mathematics Journal on Computing, vol.18, no.6, pp.1245–1262, Feb. 1989. DOI: 10.1137/0218082

Tatsuya Nagai received the B.E. and M.E. degrees in electrical engineering from Kobe University in 2017 and 2019, respectively. His research interests include machine learning-based malware analysis and cyber threat intelligence.

Masaki Kamizono received the B.E. and M.E. degrees in Computer Engineering from the University of Tokushima in 2003 and 2005, respectively. He is currently a researcher at PwC Cyber Services LLC, Japan. His research interests include malware dynamic analysis, malware static analysis, and malicious web site detection and analysis technology. He received the Best Paper Award at the 2010, 2011 anti-Malware engineering WorkShop (MWS 2010, 2011). He has also conducted research presentations on security technology at international conferences such as AVAR.

Yoshiaki Shiraishi received the B.E. and M.E. degrees from Ehime University, Japan, and the Ph.D. degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kindai University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IPSJ.

Kelin Xia received his Ph.D. degree from Graduate University of Chinese Academy of Sciences, China in 2013. He is currently with Nanyang Technological University as an assistant professor. His research interests include topological data analysis, mathematical modeling of biomolecular systems, and scientific computing.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received Ph.D. degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa Junior College, Japan. From 1998 to 2002 she was a research associate of the Department of Information Science and Intelligent Systems, the University of Tokushima, Japan. From 2003 to 2007 she was a lecturer of the same department. From 2007 to 2017, she was an associate professor at the Information and Multimedia Center, Gifu University, Japan. Since 2017, she has been an associate professor at the Department of Electrical, Electronic and Computer Engineering in the same university. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE.

Yasuhiro Takano received the Ph.D. (Info. Sc.) and Dr.Sc. (Tech.) degrees, respectively, from Japan Advanced Institute of Science and Technology (JAIST) and the University of Oulu, Finland, in 2016. He is currently with Kobe University as an assistant professor. His research interests include signal processing for communications engineering.

Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering, Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronic Engineering, Faculty of Engineering, Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. He is a member of the IEEE.