A Proposal of Malicious URLs Detection based on Features Generated by Exploit Kits

Yuma Sato[†], Yoshitaka Nakamura[‡], Hiroshi Inamura[‡] and Osamu Takahashi[‡]

[†]Graduate School of Systems Information Science, Future University Hakodate, Japan
[‡]School of Systems Information Science, Future University Hakodate, Japan
{g2115016, y-nakamr, inamura, osamu}@fun.ac.jp

Abstract - With the spread of Web access, cyber attacks are increasing. Drive-by Download attack is a kind of cyber attacks which may happen when visiting a website. Drive-by Download attacks redirect Web users to malicious Web pages. Drive-by Download attacks force Web users to download malware by exploiting the vulnerabilities of Web browsers or plug-ins when these users visit malicious Web pages. Attackers use heavily Exploit Kits to build Web sites for Drive-by Download attacks. Some characteristics such as string length and the number of special symbol, depending on the types of Exploit Kit are seen in the URLs of malicious Web pages used for these attacks. In addition, domain name of malicious Web sites tends to be short-lived to avoid blacklisting. Therefore, it is difficult to detect these attacks by using blacklists of URLs. However, the characteristic of the path and query of URLs does not change if an attacker does not change Exploit Kit to use. Therefore we can detect an attack from these characteristic of the path and query of URLs even if the attacker changed the domain name of the Web sites to use for attacks. These characteristics are extracted by decision tree learning. In this paper, we propose a novel malicious URLs detection method of Drive-by Download attacks by using features of Path and Query components of URLs used in Exploit Kits.

Keywords: Drive-by Download Attacks, Web Security, Malware, URLs, Exploit Kits

1 INTRODUCTION

The threats of cyber attacks are increasing with the spread of using the Web. The Drive-by Download attack is one of the cyber attacks through the Web. And it is increasingly sophisticated and becomes the large threat in late years. Drive-by Download attack forces Web users to download malware unconsciously. Figure 1 shows number of detected cases reported by IBM TOKYO SOC Report[1]. It shows that 2,740 of Drive-by Download attacks are detected in the first half year period of 2015. Furthermore, 800 attacks are detected in all half year periods from 2013.

Drive-by Download attackers compromise legitimate Web sites and embed malicious contents[2]. Attackers guide Web users from legitimate Web sites to the malicious Web sites and transmit the malware to PC of users. Generally these attacks exploit software of Web users and transmit the malware. It is difficult to detect this type of attack because malware is downloaded without any user noticing. Some kind of script code is used in Drive-by Download attacks. Script codes are almost always obfuscated. In addition, the detection by IPS(Intrusion Prevention System) becomes difficult because the obfuscated code does not appear in the attack patterns memorized in IPS. In this way, Drive-by Download attacks are being sophisticated and complex during recent years.

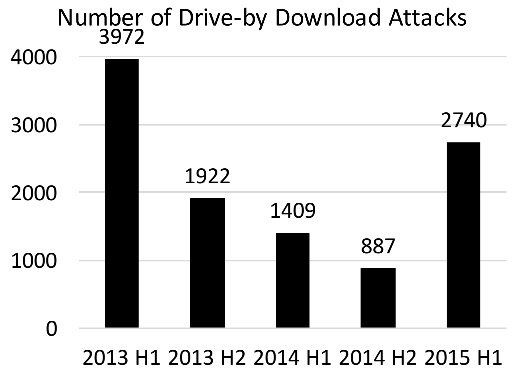

Figure 1: Number of Drive-by Download Attacks

Figure 2 shows typical flow of a Drive-by Download attacks. Firstly, attackers tamper with legitimate Web pages in order to redirect Web users to intermediate sites which guides Web access to the malicious sites. Secondly, Web users visit the compromised Web pages. Web users are redirected to multiple intermediate sites by compromised redirection. Usually, there are multiple redirections to make attack detection difficult. After multiple redirections, Web users are redirected to the exploit sites and the malware download sites. The exploit site attacks the vulnerabilities of operating system, Web browser, and plug-ins of Web browser. Finally, Web accesses of users are redirected to the malware download sites and distribution sites transmit malware to PC of Web users.

Recently, attackers often use Exploit Kits at the time of Drive-by Download attack[3]. An Exploit Kit is the packaged tool kit consisting of some exploit codes which can exploit various type of vulnerabilities. Exploit codes attacking newly discovered vulnerabilities are added to Exploit Kit continuously. Exploit Kits can be managed by GUI, and the user without the technical knowledge can make effective attacks easily.

In the Drive-by Download attack, a fingerprinting technology distinguishing the environmental information of Web users is used. Because many exploit codes are included in Exploit Kit, the attacker can use code fitted to the environment of each Web users. Therefore if there is even one vulnerability in the environment of the Web users, the attack aimed at the vulnerability succeeds, and malware is downloaded to the user's terminal.

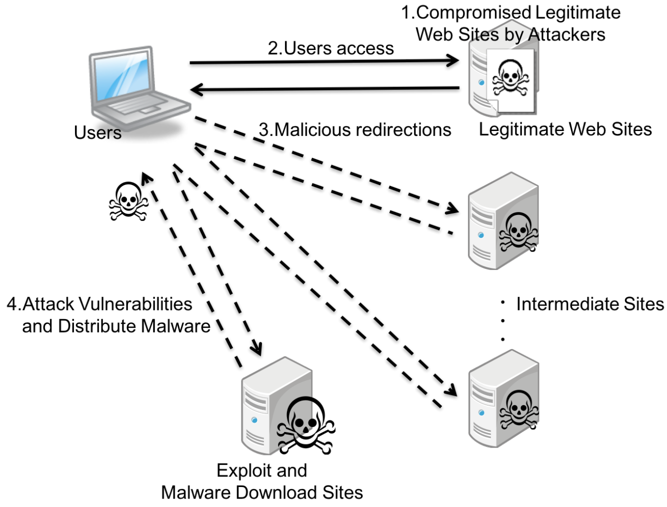

Figure 2: Typical Drive-by Download Attacks Flow

There are some types of Exploit Kits, and each Kits have features in generated URL. Web user can detect the pattern of the Drive-by Download attack by using these features and can prevent downloading of malware.

In this paper, we propose a detection method of malicious URLs of the Drive-by Download attacks based on features of URLs generated by Exploit Kits. As a contribution, even if the domain name of the website changed in a short term, our approach can detect malicious Web sites. Our approach does not need the blacklist management and can detect malicious Web sites at lower cost. And the proposed method shows possibility of the detection using URL's path and query information.

2 RELATED WORK

There are some approaches to detect Drive-by Download attacks.

Some approaches are blacklist type method. Google Safe Browsing is a typical example of such approaches. Generally, the blacklist is generated by 3 steps. First step is to trace links included in the Web page using Web crawler. Second step is to extract URLs considered to be malicious based on the features of HTML codes and URLs. As the last step, analyzers actually access Web pages with URLs considered to be malicious and evaluate the codes of HTML and JavaScript of Web pages and convert into scores. However, construction and maintenance of the blacklist requires large cost because domain name of malicious Web pages change frequently. To solve this problem, Invernizzi proposed low-cost construction method of blacklist by efficient Web crawling [4].

However, domain name of malicious Web sites tend to be short-lived to avoid blacklisting. Reference[5] shows that 40% of malicious URLs are changed within one month. And Ref.[6] describes that 80% of domain names of the malicious site are not used in six months. If attackers change domain name of malicious site, blacklisting can't detect attacks such as Drive-by Download attacks because changed domains is not listed. It is difficult to perfectly detect by blacklists. As a result, the blacklist-type detection becomes more difficult because the update of the blacklist does not catch up with the changes of domain names if attackers frequently change the domain names of the malicious sites.

Some methods are proposed to prevent Drive-by Download attack without blacklists. One method of Ref.[7] is detecting Drive-by Download attack by analyzing attack script codes and extracting features of frequently appearing characters used in codes. Because algorithm used in attack code varies according to the kinds of Exploit Kit, this feature can be used for the detection. It can detect attacks by these differences. Reference[8] introduce the method to detect malicious Web sites used for attacks by monitoring Web communication log, and the method to detect attacks by analyzing HTTP header information.

3 PROPOSED METHOD

3.1 Basic Concept

There are some features in the Web pages generated using Exploit Kit in Drive-by Download attacks. At first there is feature that URLs generated by Exploit Kits are longer than those of legitimate Web page. And different features appear in these URLs depending on Exploit Kits used for genetation. It is known that these URLs can be detected by regular expressions.

In this paper, we propose a method to detect URLs of malicious Web pages based on features of URLs generated by such Exploit Kits. Our approach focuses on URLs path and query information as features. If the Exploit Kit using for generation is the same, the features of the path and query of URLs do not have any change even if attackers change domain name of malicious sites frequently. These features can be used for the detection of malicious sites. Therefore we use features of URL paths and queries for the detection of malicious sites in the Drive-by Download attack. Our method vectorizes the features of paths and queries of URLs used in malicious URLs, and constructs the decision tree using these vectors. In this way, we can detect malicious URLs of Drive-by Download attacks.

3.2 URL's path and query

In this paper, “URL's path and query” is defined as the string which combined path of URL with query of URL using character “?”.

For example, in the URL of “http://www.example.com/dir/file.html?key=value”, part of “dir/file.html” becomes the path of this URL, and part of “key=value” becomes the query of this URL. In other words this URL's path and query becomes “dir/file.html?key=value”.

3.3 Binary decision tree

Binary decision tree is classifier that classify input data into premade classes. It has leaf nodes, root nodes, and internal nodes. Leaf nodes represent the premade multiple classes. Internal nodes and root nodes except leaf nodes represent the test for input data. Binary decision tree can classify which class the input data applied to by repeating tests for the input data from the root node to a leaf node.

Weka (Waikato Environment for Knowledge Analysis) is open source software for machine learning, which was made by machine learning group at the University of Waikato[9]. Weka has many functions such as data preprocessing and data visualization in machine learning.

Our method uses J48 classifier which is an implementation of C45 algorithm developed by Quinlan[10] for the generation of binary decision tree.

3.4 Detection method of malicious URLs

3.4.1 Detection process

The proposed method uses URL's path and query to detect malicious URL. As preparations, the method needs to collect the information of legitimate URLs and malicious URLs beforehand. Firstly the method extracts URLs that occurred by communication from the communication data which accessed to the legitimate Web pages, and from attack communication data of Exploit Kit. Secondly, URL's path and query is extracted from these legitimate and malicious URLs. Thirdly, these URL's path and query are vectorized using multiple features of malicious Web sites like the next subsection. Binary decision tree for classifications to detect malicious URLs is made by these vectors and C4.5 algorithm. This decision tree classifies whether the URL was generated by Exploit Kits. By using this result the proposed method can determine whether the URL is malicious or the legitimate.

3.4.2 Vectorization of URL's path and query

The extracted URL's path and query is converted into a vector using item (1) to item (9) of Table 1.

In Ref.[11], L. Xu et al. describe that the average length of the legitimate URL is 18.23, and the average length of the malicious URL is 25.11. They also describe that long and random character string tends to be used in malicious URL. From this result, we use the length of the URL's path and query as a component of the vectors (item (1)).
Reference[11] also describe that the average number of special symbols of the legitimate URL is 3.36 and the average number of special symbols of the malicious URL is 2.93. From this result, we use the number of special symbols as a component of the vectors (item (2)).

In Ref.[12], J. Ma et al. describe that the longest and average path lengths of URLs are available as malicious detection factor. Therefore, path length is used as a component of the vectors (item (3)). And they also describe that the number of digits included in URLs is available as malicious detection factor. Therefore, number of digits is used as a component of the vectors (item (3)).

Because the number of alphabets and keys in query are thought to increase in malicious URLs by features of item (1), these parameters are used as component of the vectors (item(6), (7)).

Because many redirections are used in the Drive-by Download attack, redirection URL may be included in query. Therefore we use the information whether the query includes character string "http" or whether the query includes IP address as components of the vectors (item(8), (9)).

Table 1: Vector Components

Item	Vector Components	Values
(1)	Length of URL's path and query	Integer
(2)	Number of special symbols	Integer
(3)	Path length	Integer
(4)	Number of digits	Integer
(5)	Query length	Integer
(6)	Number of alphabets	Integer
(7)	Number of keys in query	Integer
(8)	Inclusion of string "http"	0 or 1
(9)	Inclusion of IP addresses	0 or 1

4 EXPERIMENTAL EVALUATION

4.1 Overview of experimentation

This experiment intends for only a URL query path with disregard to the operation of the Web browser by the user. We extracted only URL's path and query in a text file and use the text file as input data.

In this experiment, we make the binary decision tree which classifies legitimate URL and malicious URL according to the kind of Exploit Kit. The construction of the decision tree uses C4.5 algorithm. Specifically, we use J48 implemented in Weka. This experiment evaluate the proposed method with 10-fold cross validation by using constructed decision tree for input data.

4.2 Experimental data

URLs of malicious and legitimate communication data are used as experimental data. About the malicious data, we use the PCAP type format data including in Malware - Traffic - Analysis.net from 2013 through 2015[13]. These data include malicious and legitimate communication data, and downloading communication data of malware. The number of the URL's path and query included in the experimental data is 7,212. We also extract the set of Exploit Kits from the same PCAP data. These Exploit Kits are classified using file name. PCAP data is named each Exploit Kits used. The types and number of Exploit Kits included in the experimental data are as follows.

We use DMOZ as legitimate communication data[14] because data have many URLs included many path and query. DMOZ is the largest Web directory that constructed and maintained by a global community of volunteer editors. In this experiment, we extract 300 of URLs of Web pages indexed in DMOZ at random. We randomly extract 2,368 of URL's paths and queries as legitimate data from the URL's path and query of the request URL occurred at the time of access to these URLs.

Table 2: Types and Number of Exploit Kits

Types	Number
Angler Exploit Kit	1,947
Blackhole Exploit Kit	39
Cool Exploit Kit	18
Dotkachef Exploit Kit	54
Fiesta Exploit Kit	1,071
Flashpack Exploit Kit	225
Goon Exploit Kit	225
Hello Exploit Kit	9
KaiXin Exploit Kit	18
Magnitude Exploit Kit	1,128
Neutrino Exploit Kit	408
Nuclear Exploit Kit	1,314
Rig Exploit Kit	387
Styx Exploit Kit	135
Sweet Orange Exploit Kit	234

4.3 Evaluation process

In this experiment, we evaluate the detection of the proposed method using five evaluation items such as true positive rate ($TPrate$), false negative rate ($FNrate$), true negative rate ($TNrate$), a false positive rate ($FPrate$), and accuracy (ACC). We evaluate every URL's paths and queries included in a request to any Web page. In the following formulas, "URL's P & Q" denotes to "URL's path and query".

When the proposed method correctly classifies the URL's path and query of the malicious URL as malicious, we define it as true positive. It is true positive if malicious URLs are classified as any Exploit Kits. The true positive rate is a ratio of true positive number in all malicious URL's paths and queries. This rate is expressed as formula (1).

$$TP\ rate\ = \frac{\#\ of\ malicious\ URL's\ P\&Q\ classified\ as\ malicious}{\#\ of\ malicious\ URL's\ P\&Q} \tag{1}$$

False negative is that malicious URL's paths and queries incorrectly classified as legitimate. The false negative rate is a ratio of false negative number in all malicious URL's paths and queries. This rate is expressed as formula (2).

$$FN\ rate\ = \frac{\#\ of\ malicious\ URL's\ P\&Q\ classified\ as\ legitimate}{\#\ of\ malicious\ URL's\ P\&Q} \tag{2}$$

True negative is legitimate is classified as legitimate. True negative is that legitimate URL's paths and queries correctly classified as legitimate. The true negative rate is a ratio of true negative number in all legitimate URL's paths and queries. This rate is expressed as formula (3).

$$TN\ rate\ = \frac{\#\ of\ legitimate\ URL's\ P\&Q\ classified\ as\ legitimate}{\#\ of\ legitimate\ URL's\ P\&Q} \tag{3}$$

False positive is that legitimate URL's paths and queries incorrectly classified as malicious. The false positive rate is a ratio of false positive number in all legitimate URL's paths and queries. This rate is expressed as formula (4).

$$FP\ rate\ = \frac{\#\ of\ legitimate\ URL's\ P\&Q\ classified\ as\ malicious}{\#\ of\ legitimate\ URL's\ P\&Q} \tag{4}$$

The accuracy is defined as a ratio of true positive and true negative number in all URL's paths and queries. This rate is expressed as formula (5).

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

5 RESULT AND DISCUSSION

5.1 Result

Table 3 shows the result of classification using the proposed method by each evaluation item. Table 4 shows the classification result by Exploit Kits. From this result, the classification of the Exploit Kits with much used number achieves high detection precision. On the other hand, the classification of the Exploit Kits with a little used number tends to have a low detection precision.

Table 3: Result of Classifying

Evaluation Items	Value
True Positive (TP)	89.02%
False Negative (FN)	10.98%
True Negative (TN)	81.67%
False Positive (FP)	18.33%
Accuracy	87.20%

5.2 Discussion

Our proposed method has a strong point to be able to detect the URL of malicious Web page, even if the domain name of the malicious Web site is changed, because our method focuses on URL's path and query occured in Exploit Kits.

From the result of experiment, the more number of URLs are generated by the Exploit Kits, the higher detection precision is achieved. In order to increase the detection precision of malicious URLs, it is necessary to collect much more access data made by Exploit Kits.

Our approach only vectorize from path and query in URLs. For instance, when URLs path and query is "/", vector of legitimate URLs may be the same as vector of malicious. If it comes to that, legitimate URLs are classified as malicious and vice versa. For this problem, in order to block exploit attacks and downloading malware precisely, we need to make improvement adding some other components such as the number of redirections. For the future, we are going to enhance the capability to detect by classify in detail URLs generate by Exploit Kits.

Table 4: Result of Classifying by Exlopoit Kits

			Number of Classifying															
			A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Exploit Kits	A	Angler	1572	0	0	1	9	3	5	0	2	77	9	100	0	3	0	166
	B	Blackhole	0	25	0	0	0	0	3	0	0	0	0	0	0	0	0	11
	C	Cool	0	0	5	0	0	2	0	0	0	0	0	0	0	0	0	11
	D	DotKachef	3	0	0	36	0	0	3	0	0	0	0	0	0	0	0	12
	E	Fiesta	10	0	0	0	889	2	12	0	0	34	2	37	6	0	0	79
	F	Flashpack	1	0	2	0	3	139	3	0	0	1	15	5	0	0	2	54
	G	Goon	12	0	0	1	9	1	105	0	1	12	2	26	11	0	3	42
	H	Hello	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	3
	I	KaiXin	3	0	0	0	0	0	0	0	3	0	0	1	0	0	0	11
	J	Magnitude	86	0	0	0	27	1	15	0	0	859	6	54	6	2	0	72
	K	Neutrino	25	0	0	0	4	11	3	0	0	11	237	57	0	0	2	58
	L	Nuclear	73	0	0	0	28	2	8	0	0	55	30	1013	5	0	2	98
	M	Rig	5	0	0	1	12	0	9	0	0	7	1	17	304	3	0	28
	N	Styx	5	0	0	0	2	0	0	0	0	6	0	1	2	70	0	49
	O	Sweet Orange	5	0	0	0	3	0	3	0	0	1	0	4	2	0	118	98
Legitimate	P	Legitimate	97	3	4	5	53	26	19	1	0	61	37	82	13	13	20	1934

6 CONCLUSION

In this paper, we proposed malicious URLs detection method based on features generated by Exploit Kits by using URLs path and query. This method builds binary decision tree from practical communications, and classifies malicious or legitimate URLs with paths and queries. From the experimental evaluation, the method achieved 89.02% of true positive rate and 81.67% of true negative rate.

As a future work, we will classify Exploit Kits in detail and need propose a detection method in extreme precision with string pattern or regular expressions.

REFERENCES

[1] IBM, The First Half of 2015 Tokyo SOC information analysis Report, Available: https://www-304.ibm.com/connections/blogs/tokyo-soc/resource/PDF/tokyo_soc_report2015_h1.pdf?lang=ja.

[2] T. Matsunaka, A. Kubota, and T. Kasama, An Approach to Detect Drive-By Download by Observing the Web Page Transition Behaviors, Proceedings of the 9th Asia Joint Conference on Information Security (ASIA JCIS2014), pp. 19–25 (2014).

[3] Trend Micro Incorporated., 3Q 2015 Security Roundup. Available: http://www.trendmicro.co.jp/cloud-content/jp/pdfs/security-intelligence/threat-report/pdf-sr2015q3-20151119.pdf?cm_sp=threat-_-sr2015q2-_-lp-btn.

[4] L. Invernizzi, and P. M. Comparetti, EvilSeed: A Guided Approach to Finding Malicious Web Pages, Proceedings of the 2012 IEEE Symposium on Security and Privacy, pp. 428–442 (2012).

[5] M. Akiyama, T. Yagi, and M. Itoh, Searching structural neighborhood of malicious URLs to improve blacklisting, Proceedings of the IEEE/IPSJ 11th International Symposium on Applications and the Internet(SAINT2011), pp. 1–10 (2011).

[6] M. Akiyama, T.Yagi, and T. Hariu, Measuring Lifetime of Malicious Website Based on Redirection from Compromised Websites, The Special Interest Group Technical Reports of IPSJ, Vol. 2014-SPT-8, No. 10, pp. 1–6 (2014).

[7] M. Cherukuri, S. Mukkamala, and D. Shin, Similarity Analysis of Shellcodes in Drive-by Download Attack Kits, Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing(CollaborateCom2012), pp. 687–694 (2012).

[8] T. Matsunaka, A. Kubota and T. Kasama, An Approach to Detect Drive-by Download by Observing the Web Page Transition Behaviors, Proceedings of the 9th Asia Joint Conference on Information Security (AsiaJCIS2014), pp. 19–25 (2014).

[9] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Available: http://www.cs.waikato.ac.nz/ml/weka/index.html.

[10] J. R. Quinlan. C4.5: Programs for Machine Leaning. Morgan Kaufmann, (1993).

[11] L. Xu, Z. Zhan, S. Xu, and K. Ye, Cross-layer detection of malicious websites Proceedings of the third ACM conference on Data and application security and privacy (CODASPY'13), pp. 141-152, (2013).

[12] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, Beyond blacklists: learning to detect malicious web sites from suspicious urls, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining(KDD'09), pp. 1245-1254, (2009).

[13] Malware-Traffic-Analysis.net, http://www.malware-traffic-analysis.net/.

[14] DMOZ - the Open Directory Project. https://www.dmoz.org/.