Domain Information Based Blacklisting Method For The Detection Of Malicious Webpages

Ralph Edem Agbefu [1], Yoshiaki Hori [1 2] and Kouichi Sakurai [1 2]
Department of Informatics, Kyushu University [1 2],
Room 712, Ito Campus, West Bldg. No.2,
744 Motooka, Nishi-ku, Fukuoka, 819-0395, Japan
Institute of Systems and Information Technologies and Nanotechnologies (ISIT) [2]
Fukuoka SRP Center Building 7F,
2-1-22, Momochihama, Sawara-ku, Fukuoka, 814-0001, Japan
edem@itslab.inf.kyushu-u.ac.jp,hori@inf.kyushu-u.ac.jp
sakurai@inf.kyushu-u.ac.jp

ABSTRACT

Malicious web pages that host drive by download exploits have become a popular means by which an attacker delivers malicious contents to computers across the internet. The popularity of the attack has led to researchers developing systems to detect and stop such attacks. These methods include dynamic solutions, static solutions and the use of blacklisting and whitelisting methods. Blacklisting and in particular URL blacklisting is one of such detection methods. URL blacklisting analyzes the structure of a web page URL. URL blacklisting are however prone to evasion attacks when the lexical structure of the URL changes. In this paper, we propose the usage of domain related information for the detection of drive by download web pages. These domain features are used to model a scoring mechanism classification system. We show the effectiveness of detecting malicious web pages using domain based by obtaining a high detection rate and a relatively low false negative.

KEYWORDS

Drive by download, Blacklist, domain information, registrar, domain duration, domain freshness.

1 INTRODUCTION

Technological advances have led to the internet playing a major factor in our daily activities. We depend greatly on the internet for a variety of activities including industrial activities, medical activities, banking activities and many more. In order to keep with these technological advances, a wide variety of functionalities have been added to the modern day web browsers and these advancements, have come with a number of vulnerabilities. The presence of these vulnerabilities require the high standard of security practices to be implemented in a browser. In addition to this, a user's computer usually contains a number of applications that are rarely updated or in some case never updated. According to a report [1], as much as 80% of web users are using outdated versions of Adobe Flash and Acrobat Reader and popular web browser plugins.

These outdated application introduce a number of vulnerabilities and these together with web browser vulnerabilities are very much exploited by adversaries. An effective way of performing this exploitation of computer vulnerabilities is through the use of the so called drive by download attack. In a drive by download, a user can be infected by just visiting a compromised website without necessarily having any other sort of interaction. The successful exploitation of vulnerabilities will lead to the downloading and installation of malicious software. The installed malicious software connect to a command and control infrastructure to form a botnet.

1.1 Background

Drive-by downloads attacks have been around for a number of years. A drive by download attack occurs when on visiting a web page, a user is redirected to a malicious web page that leads to the downloading of malware to the user's computer without his consent. These attacks exploit multiple unpatched vulnerabilities in the user's browser, browser plugin, application or operating system. Hackers can either lure users to malicious sites they have initially injected with malicious code or hack legitimate sites to host the malware. Because legitimate sites are generally trusted and may be popular high-traffic venues, they can be very successful for distributing malware to unsuspecting visitors through the browser. Also, due to the fact that certain landing pages are originally benign web sites, a number of users could access these sites thus attackers can effectively infect the users' machine with malware.

Moreover, the injected malicious codes are usually obfuscated Java codes [19]. The obfuscation of Java Codes makes analysis difficult and increases the success rate of drive by download attacks. A typical attack is illustrated in the Fig.1 and it usually involves the following steps:

(1) Initially an attacker injects malicious code into web server, which compromises the site.

(2) User visits compromised website.

(3) Injected malicious script causes redirection from one web server to another.

(4) After a number of redirects, the user is directed to the exploit server, which sends the exploit codes to the user's computer.

(5) On execution of such exploit codes, the attacker gains control of the victim's browser**.** Malwares are downloaded and executed on the user's computer.

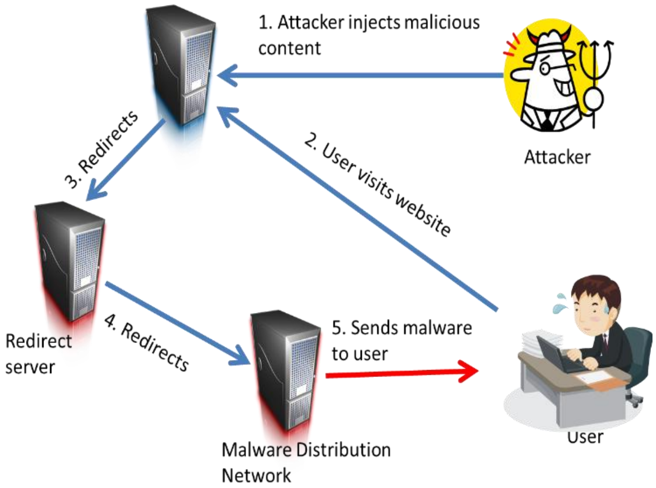

Fig 1: Drive by download attack

1.2 Injecting malicious contents

The starting point for an attacker is to obtain control of a legitimate web server and inserts malicious code into such web servers. These malicious codes are very much obfuscated contents. In order for this to be successful, an attacker usually takes advantage of certain conditions. A non-secure web server, user contributed content, advertisements and third party widgets are examples of such situations.

Keeping the web server security and applications regularly up-dated with the most recent security patches ensures the web server is secure. Malicious content can be injected as an entry in databases using SQL injection techniques.

A number of websites allows users to contribute to their page in the form of blogs, comments, reviews etc. However, most of the pages lack proper input validations and allows users to input virtually anything. Attackers can insert "iframes" that will expose other users to the dangers of a drive by download attack.

Ad syndication is one of the ways through which an attacker can insert malicious content into a web page. Ad syndication allows an advertiser to sell or lease out the advertising space to another company which in turn can sell it out to another company. This implies that the advertising script is not directly under the control of the web page's administrator and if the controller happens to be an adversary, he can easily inserts malicious code into the web page.

Third party widgets are tools provided by third parties used commonly to provide extra functionality to a web site. If the third party in question happens to be an attacker he can easily inserts malicious script code into the provided widget.

1.3 Exploiting strategies

A popular way of exploiting the user's computer in the past was by exploitation of network services. However, due to the abundance of technologies such as Network Address Translation and Firewalls, these exploitation techniques have become less successful. As a result, attackers are forced to find other means of exploiting vulnerabilities in a user's computer.

Attackers look to lure users to connect to malicious servers on the internet through the use of a combination of exploit software and social engineering techniques. Scripting support allows a web page to collect information about the browser's computing environment. While these capabilities can be employed for legitimate purposes, it also offers for adversaries the opportunity to learn of the vulnerabilities on a user's computer.

Tricking of users usually happen in cases where the attacker was not able to find a vulnerability on the user's computer. Users are asked to download a certain program in order to play certain media files. These programs may in fact look like the normal "codec" or "plug-ins", however they are malicious software. Once installed, they compromise the user's computer and finally lead to the downloading and installation of malware.

Exploiting software may either look to exploit vulnerabilities in the browser or automatically launch external programs and extensions. Less number of users pay much attention to upgrading of applications and software on their computers, as evidenced by Secunia PSI Report [17].

1.4 Motivation

Web pages are very common and web browsers that are used to retrieve web pages have existed for a long time. In the creation of the web browser, the possibility of an attack being mounted through the web browser wasn't envisaged. As such, web browser security became an afterthought.

The danger of the drive by download attack stems from the fact that very little or no user interaction is required for a user's computer to be compromised. A simple click on a URL that points to a malicious web pages is sufficient to trigger the attack. If the exploitation of vulnerabilities is successful, malicious contents are downloaded unto the user's computer and all this is done without the user's consent.

In certain instances, drive by download attack's starting point of malware infection is originally benign web site. This has the tendency to draw a lot of users who will inadvertently be infected with the malware. Also, different from other forms of attacks, users will generally not be able to notice the malware infection because they believe they have accessed benign web sites. This contributes to make detection of drive by download attacks difficult and one of the most effective form of malware infection.

Blacklisting methods which focus on the use of URL information are one of the popular ways of defense against the drive by download attack. However URL based blacklisting methods have been found to ineffective due to their reliance on the syntax of a given URL. Hence alternatives to URL blacklisting need to be considered. Our work is motivated by the observation that attackers tend to employ similar domain characteristics in deploying malicious web pages. Attackers also tend to abuse certain domain features during this process. By basing a detection system on such features we can effectively detect and blacklist the malicious web pages.

1.5 Contribution

In this work, we propose a domain based blacklisting method for the detection of drive by download pages. In order to achieve this, we did the following. First, we collected a set of data of known malicious web pages and known benign web pages and from these set of data obtained a set of domain related information. The collected domain information are analyzed for the frequently used information by attackers in the deployment of malicious web pages. For the set of identified features which are most abused by attackers, we construct a scoring based mechanism that gives the degree of abuse by the attacker. The score values of the domain features are used to model a detection algorithm where a score value total lower than a pre-determined threshold value is said to be malicious, otherwise the page is identified as benign web page. A score value is a numerical representation of the degree of usage of an identified domain information.

We evaluate the effectiveness of our approach through the use of and evaluation data set. We show that such an approach has capability of detecting malicious web pages that participate drive by download attacks.

We summarize our contributions in the following points:

1. We determine a set of domain features that are abused by attackers.
2. We determine the extent to which these domain features are abused.
3. We make use of a scoring detection mechanism.
4. We show the effectiveness of a domain based blacklisting of drive by download pages.

2 RELATED WORKS

2.1 Survey reports

The effectiveness of the drive by download attacks have resulted in a number of entities focusing on the detection and protection against the attack. These entities actively study malicious web sites and on regular basis publish white papers of their findings [2], [3], [4], [5], [6], [7][18]. Majority of the white papers report a significant increase of web-based client-side attacks recently, leading to the web becoming the primary attack vector to infect user with malware.

Websites found in the top 100 most popular web sites seem to have also been found to not be immune to attack. In their work, Provos et al. observed that as much as 10% of suspicious URLs received on the Google search engine were actually found to be malicious [8], [9]. They identified four key areas of content control: authored content, user-contributed content, advertising, third-party widgets. In order to avoid detection, attackers use techniques such as obfuscation of the exploit code, distributing binaries across different domains and continuously re-packing the binaries. Furthermore, they found botnet-like structures based on binary updates which can be interpreted as command and control. They report that from the top one million URLs appearing in the Google search engine results, about 6,000 belong to sites known to cause drive by download attacks [9].

These results highlight the significance of drive by download. A browser that loads a malicious web page can be redirected via multiple pages on numerous hosts until the actual exploit is delivered by a central exploit server. They also observed that 82% of malicious web pages identified make use of a central exploit server. The central exploit servers serve the numerous malicious web page with the appropriate exploit which is transmitted to the user's computer.

2.2 Detection Systems

We group the detection systems based on their approach of detecting drive by download attacks: whitelisting and blacklisting, signature based and dynamic analysis.

Whitelisting and Blacklisting

Whitelisting and blacklisting are a popular method to the detection of drive by download attacks. In a whitelisting type of detection, a database containing the set of programs and applications that are allowed to run on a computer is maintained. Any other kinds of programs or applications that are not included in such a database are denied access to execute on the machine.

Blacklisting works contrary to whitelisting, blacklist methods maintain a database of programs and applications that are not permitted to run on a user's machine. Any other any program or applications are allowed to run. Whitelisting and blacklisting methods suffer from outdated database. This will imply programs or applications that shouldn't be allowed to run can be run on a user's computer. Blacklisting methods in particular are not proactive in their detection methods. Fukushima et al proposed the usage certain domain features in detecting and blacklisting malicious web pages [10]. The features identified were used in a reputation evaluation system, where web sites with low reputation values are classified as malicious whereas high reputation value web sites are classified as benign. The domain based features used in this work are the IP address block and the registrar information.

Felegyhazi et al explored the possibility of a proactive domain blacklisting [11]. They obtain malicious domains for seeding and extracting name server features from the malicious domains as properties for determining and predicting a drive by download page. They use the freshness of domains and self-resolution as features for determining malicious domains.

Static Detection

ARROW developed by Zhang et al obtains a log of HTTP redirection traces from compromised web sites using honey pots [12]. ARROW groups domains with the same IP addresses into a Hostname-IP Clusters. From the logs, it identifies the most commonly used property referred to as the central server. Regular expression signatures are generated based on these central servers. Due to the conditional redirections, where queries from honey pot are redirected to benign pages or to search engines, ARROW cannot in certain instances obtain HTTP traces. ARROW considers the URL structure in their approach, however, our work doesn't analyze the URL structure.

BrowserGuard by Fu-Hau et al records the download scenario of every file through a web browser and based on the download scenario, blocks any file that is loaded without the consent of a web user [18]. They do not analyze the source code or script file and hence do not need to maintain an exploit code sample.

Dynamic detection

Honeypots are an alternative to intrusion detection systems and are described as security devices designed to detect intrusions and attacks on a computer. A honeypot is a vulnerable device that looks to lure attackers to it. Attackers that scan for vulnerable web servers will eventually find this web server and attack it. In the honeypot environment, the attacker's actions can be observed and traced. To detect drive by download attacks, a new honeypot system was introduced: the client honeypot.

Client honeypot focus on malicious web servers, which they interact with by driving a web browser on the dedicated honeypot system. By monitoring of the changes to a list of system files, configurations and directories, they can detect the occurrence of a drive by download attack. Client honeypot can be classified into low interaction and high interaction. The group of high interaction honeypot clients run web browsers in a well-controlled environment. It looks out for any change that will indicate a successful drive by download

3 OUR APPROACH

Adversaries have been known to abuse certain domain properties in the setting up of malicious web pages. As a result, a number of research has been done on the analysis of web page's domain properties especially in phishing and scam attacks. It is imperative that such studies be extended to relatively new attacks such as the drive by download attack.

In this paper, we analyze the characteristics of drive by download hosting web pages with respect to their domain information. The aim is to identify methodologies and traits that attackers use in deploying them. We considered the following domain information: domain registration date, domain expiration date, domain duration, how recent the domain was created, the registrar and the geographical location of the domain. Our analysis provides an insight into how much of these features are common among attackers and the level of popularity.

By assigning score values to these features based on their popularities with either benign or malicious web pages, we design a detection scheme for malicious web pages. The detection system is based on a score value. The score values are a numerical representation of the relative strength or otherwise of a given feature. The lower the score value for a feature, the higher the possibility of that feature been abused by attackers.

In order to create the scoring mechanism we must decide on the amount of points required before a web page is considered malicious. For this research work, we choose a negative total score value as an indication of maliciousness of a page.

3.1 Dataset

The data set used in this research consisted of known malicious and known benign web pages. For malicious dataset, we obtained the data from Malware Domain List (MDL), a web service which provides daily updated domain lists of malicious web pages [13]. Since the MDL database consists of more than just drive by download webpages, we had to extract only drive by download pages. This was done using descriptions of the web pages as provided in the original database. The "description" tab of the database gives the cause of maliciousness. For drive by download web pages, we focus on keywords such as "exploit", "redirect" or "drive by". The malicious data was obtained for the period of July 2012 to December 2012.

Benign dataset was obtained from Alexa, which is a web service that rates web sites based on traffic generated [14]. In Alexa, web sites such Google and Facebook will be ranked higher in their list of websites compared those that are less frequented.

Higher traffic generated web sites are less likely to be malicious because such sites are well maintained due to their popularity among internet users.

The collected data was grouped into a training and evaluation data set. Training set is defined as a set of data used to discover potentially predictive relationships. The training set is used at the initial stage of the proposal to determine patterns or similarities between the different set of data obtained. The evaluation set is used to verify the set of patterns or similarities that was discovered during the training stage. Our training dataset consisted of 1202 malicious web pages and 500 benign web pages. The evaluation data set consisting of all malicious web pages was used to determine the effectiveness of our proposed approach by way of detection rate and false negatives.

3.2 Domain feature analysis

For each of web pages we collect the following domain information:

1. Registrar
2. IP Address
3. Domain Registration date
4. Domain Expiration date.

These information are obtained by performing a WHOIS query on the web page.

Geographical location (Country):

The country or geographical location of the domain is obtained from the use of GeoIP [16]. GeoIP is a web service that determines the geographical location of a domain based on its IP address. The result represents the physical location of the web page.

The results of the geographical location for our data set are shown in Table 1 and Table 2. The geographical location results for malicious domains show the leading contributor of malicious domains to be the USA, contributing at least one-third of malicious web pages. This results suggests U.S.A to be most likely to host malicious web pages. However in Table 2 for benign domains table, USA again, is the leading contributor with the percentage contribution towards benign pages being more than that for malicious distribution web pages. Due to the fact that the source of our database for malicious data set is user contributed, the location of the source of our data can greatly affect the results obtained for this feature. For example, if such web service is to be in USA, it is much likely the users contributing to the overall database are mostly in the USA. This can be a contributing factor for the geographical locations distributions.

Table 1: Geographical distribution of malicious domains

Country	**Domains**	**Percentage (%)**
U.S.A	343	36
Russia	94	10
France	52	5
Germany	48	5
Turkey	44	5
Netherlands	38	4
Australia	36	4
Canada	30	3
U.K	24	3
Italy	22	2

Table 2: Geographical distribution of benign domains

Country	**Domains**	**Percentage (%)**
U.S.A	169	47
China	59	16
Germany	14	4
U.K	11	3
Japan	10	3
Russia	10	3
Canada	9	3
France	8	2

India	8	2
Hong Kong	5	1

Domain duration

The domain duration for a web page is obtained by calculating the difference between the domain expiration date and domain registration date. Figures 2 and 3 show the result obtained for malicious domains and benign domains respectively. The domain duration graph for malicious shows as much as 75% of the malicious web pages have a duration of less than five years. A look at the benign web pages graph shows an average domain duration of 15 years compared to 4 years of malicious web pages. From these results there is a strong indication of domain duration less than or equal to one year been popular with attackers.

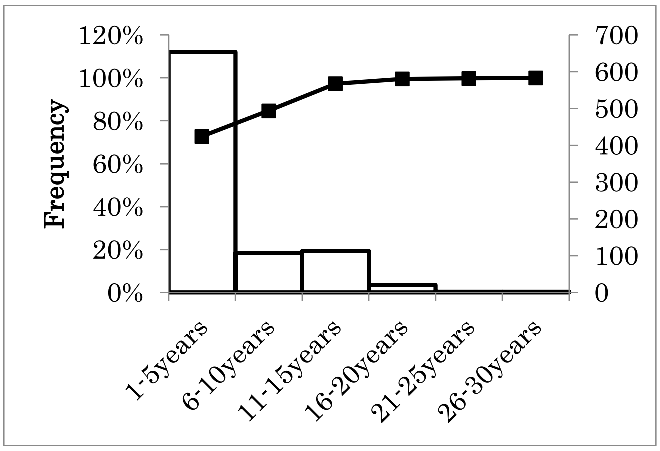

Fig.2: Domain duration distribution for malicious domains

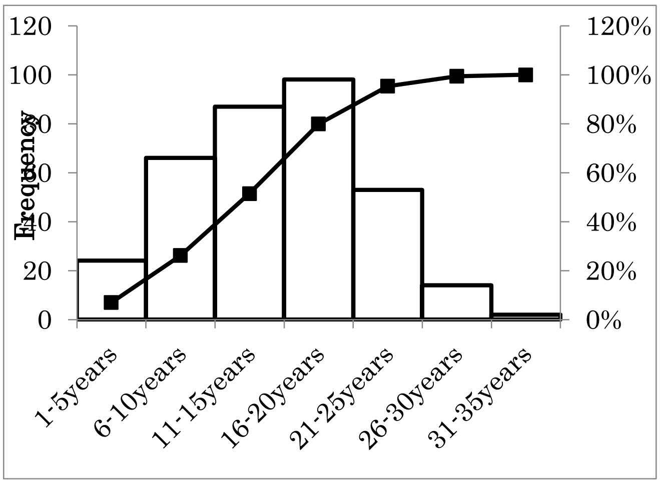

Fig.3: Domain duration distribution for benign domains

Registrar

Registrar refers to the organization which has been accredited to assign and manage the reservation of internet domain names.

For this feature, we compared our results of registrars for both types of data with the top ten domain registrars worldwide. The top ten registrars based on their market share according to [16], is shown in Table 5. We also show results for our top malicious and benign registrars in Tables 3 and 4 respectively. The result shows attackers do patronize the most popular of registrars as much benign web masters do.

Table 3: Registrar distribution for malicious domains

Registrar	Domains	%
GoDaddy	120	15
Network Solutions	71	9
Enom, Inc.	63	8
OVH	46	6
PDR Ltd	42	5
Tucows Inc	34	4
Naunet-Reg-Ripn	33	4
Internet.Bs.Corp	30	4
Click Registrar Inc	29	4
Regru-Reg-Ripn	19	2

Table 4: Registrar distribution for benign domains

Registrar	Domains	%
Mark Monitor	91	24
GoDaddy	44	11
Network Solutions	42	11
Enom. Inc	22	6
Hichina Zhicheng Ltd	16	4
CSC Corporate Inc	15	4
Melbourne IT Ltd	15	4
Tucows Inc	12	3
Ru-Center-Reg-Ripn	9	2

Register.com	7	2

Table 5: Top 10 Domain Registrars

Registrar	**Domains**	**%**
GoDaddy	28,718,464	32.47
Enom Inc	7,131,517	8.06
Tucows	6,011,360	6.80
Network Solutions	5,408,778	6.12
Schlund+Partner	4,267,855	4.83
Melbourne IT	2,830,735	3.20
Wild West Domains	2,485,786	2.81
Register.com	1,823,513	2.06
ResellerClub.com	1,606,005	1.82
Moniker	1,404,097	1.59

Domain freshness

The domain freshness properties of web page is indication of the “recentness” of the web page, that is, when the web page was created with reference to a particular date. This can be represented by the equation:

Domain freshness = Referenced date – Creation date

The referenced in our work was the date of performing the classification, the creation date can be obtained from a “WHOIS” query as the domain registration date.

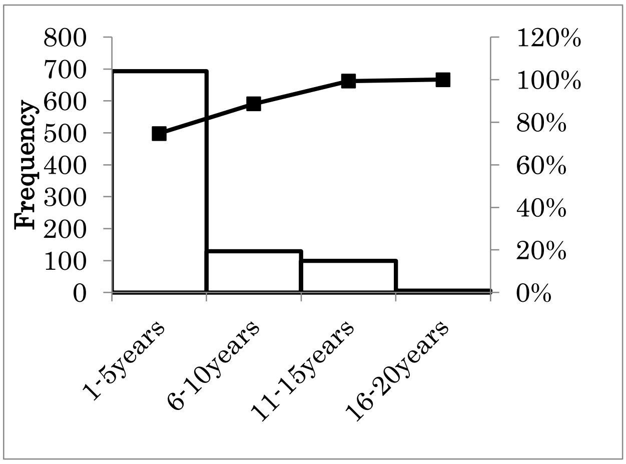

Fig. 4: Domain freshness distribution of malicious domain

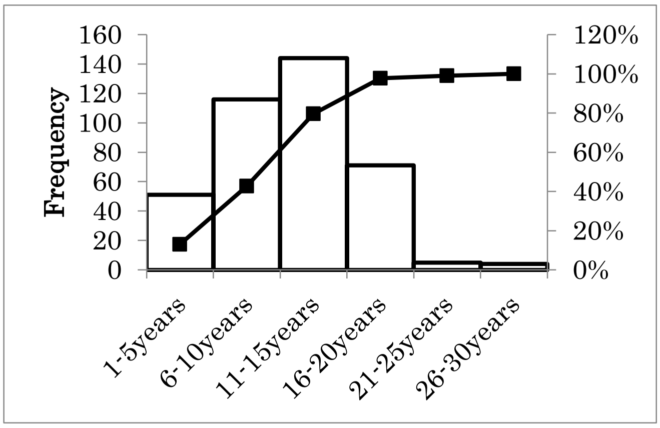

Fig. 5: Domain freshness distribution of benign domains

For our data set of malicious web pages, the result shows that as much as 72% of these were created in the past four years. Whereas for benign pages, about 40% of these have been around between 11 to 15 years. Averagely, the domain freshness for the benign data set was about 11 years, whereas that for malicious data set was 4 years.

4 CLASSIFICATION AND EVALUATION

4.1 Classification

In order to effectively detect drive by download pages using the domain information obtained, a score value system was introduced. A score value system makes use of numerical values that are assigned to a candidate web page domain features. These score values are determined based on how popular that particular feature is among attackers. A feature that shows a higher usage in the malicious data set compared with its usage in the benign data set, is an indication that the feature is most patronized by attackers. The calculation of the score values involve two main steps:

1. For any given feature in our database, we first obtain the percentage difference of usage in both set of data. This can be represented by the equation:

Percentage difference usage = benign_percentage_usage – malicious_percentage_usage

Depending on the usage of a given feature, the result obtained can be positive, negative or neutral (zero). A positive value will indicate that the given feature is less likely to be used by attackers in setting up malicious web pages whiles a negative result indicates otherwise.

2. The score values are then calculated from the obtained percentage difference as shown in Table 6 below:

Table 6: Score values

Score value	Percentage Difference
±(1)	±(1-5%)
±(2)	±(5-10%)
±(3)	±(>10%)

A percentage difference ranging from 1% to 5% is assigned a score value of 1, ranging from 5% to 10% assigned a score value of 2 and any other difference assigned a score value of 3. In this classification we consider a negative total score value result as a malicious web page. A positive result represents a benign web page.

From our dataset and given a feature of Registrar to be GoDaddy, the score values will be determined as follows:

Percentage difference = 11% - 15%, resulting in a percentage difference of *-4%*.

As the percentage difference falls within the ranges of ± *(1%-5%)*, this feature will be assigned a score value of *-1*. The determined individual score values of the features are then added up. A negative sum implies the web page is malicious else it is classified as being benign.

4.2 Evaluation

We evaluate the capability of our proposed scheme using the combinations of domain duration, domain freshness, registrar and country.

The critical factor for evaluating our proposal are the detection rate and false negatives of such an approach.

For the evaluation test, we obtained a data set of 334 known malicious web pages. These are known malicious web pages that currently carry drive by download exploits or at one time did. The web pages were obtained from MDL and dates of collection spans over a three month period, specifically they were collected between the period of November 2012 and January 2013.

For this work, we obtained a detection rate of 85% and a false negative rate of 15% using the proposed classification method. This indicates the ability to correctly detect 85 web pages as malicious given a dataset of 100 web pages and incorrectly classifying the rest as benign web pages.

5 CONCLUSION AND FUTURE WORK

Malicious pages that launch drive by download attacks are very much on the increase. This work proposed the use of domain information properties of a web page to detect drive by download pages.

As future work, a number of areas of our proposal can be looked in order to improve on current results. One of such areas is the data set. It is imperative that the current data set be expanded in order to improve on the integrity and effectiveness of such an approach. Another area is the adjustment of score values. It is possible to adjust score values so as to improve on the detection rate and false negatives of this approach. Although this proposal considers the false negatives, future work can look at the false positives of using such an approach.

We do not view our proposal as a replacement of existing blacklisting methods but as a complement. Our experimental results show that a domain based blacklisting method has capability of detecting drive by download pages. Such a proposal can be implemented together with the slower dynamic solutions to improve on the detection latency.

Acknowledgement

This work was partially supported by Proactive Response Against Cyber-Attacks Through International Collaborative Exchange (PRACTICE), Ministry of Internal Affairs and Communications, Japan. The first author of this research is supported by Monbukagakusho (MEXT) scholarship.

REFERENCES:

1. Danchev, D.: 80% of Web users running unpatched versions of Flash Acrobat (Online) available from http://blogs.zdnet.com/security/?p=4097. (accessed February 2013).
2. WEBSENSE INC. Home page, 1994. Available from: http://www.websense.com; accessed on February 2013.
3. WEBROOT SOFTWARE INC. Home page, 1997. Available from: http://www.webroot.com/; accessed on February 2013.
4. SOPHOS. Home page, 1997. Available from http://www.sophos.com; accessed on February 2013.
5. MCAFEE, INC. Home page, 2005. Available from http://www.siteadvisor.com/; accessed on February 2013.
6. FINJAN. Home page, 1996. Available from http://www.finjan.com; accessed on February 2013.
7. SCANSAFE. Home page, 1999. Available from http://www.scansafe.com; accessed on February 2013.
8. Provos,N., Mavrommatis, P., Rajab, M.A. and Monrose F.: All your iFRAMEs point to us, In proceedings of USENIX Security Symposium, pp 1-15,(2008).
9. PROVOS, N., MCNAMEE, D., MAVROMMATIS, P., WANG, K., AND MODADUGU, N. The ghost in the browser: Analysis of web-based malware. In *HotBots'07* (Cambridge, 2007), Usenix.
10. Fukushima, Y., Hori, Y. and Sakurai, K.: Proactive Blacklisting for Malicious Web Sites by Reputation Evaluation Based on Domain and IP Address Registration, In Proceedings of International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, pp. 352-361 (2011)
11. Felegyhazi, M., Kreibich, C. and Paxson, V.: On the potential of proactive domain blacklisting, In Proceedings of the 3rd USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET'10),pp. 99-107, (2010).
12. Zhang, J., Seifert, C., Stokes, J. and Lee, W.: ARROW: GenerAting SignatuRes to Detect DRive-ByDOWnloads In proceedings of WWW'11 , pp. 187-196, (2011).
13. Malware Domain List(MDL) Available from: http://www.malwaredomainlist.com/mdl.php; accessed December 2012.
14. Alexa-The Web Information Company. Available from: http://www.alexa.com/; accessed December 2012.
15. WebHostingInfo: available from: http://www.webhosting.info; accessed February 2013.
16. Geo IP Tool, available: http://www.geoiptool.com/, accessed February, 2013.
17. Interesting statistics from the Secunia PSI, Available from: http://secunia.com/blog/18/ accessed February 2013.
18. Fu-Hau Hsu, Chang-Kuo Tso, Yi-Chun Yeh, Wei-Jen Wang, and Li-Han Chen, "BrowserGuard: A Behavior-based Solution to

Drive-by-Download Attacks," IEEE Journal on Selected Areas in Communications, Volume 29, Issue 7, pages 1461 - 1468, August 2011.

19. M. Johns. On JavaScript malware and related threats – Web page based attacks revisited. Journal in Computer Virology, 4(3):161–178, 2008.