J Inf Process Syst, Vol.15, No.1, pp.100~115, February 2019
https://doi.org/10.3745/JIPS.03.0107

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Automated Link Tracing for Classification of Malicious Websites in Malware Distribution Networks

Sang-Yong Choi*, Chang Gyoon Lim**, and Yong-Min Kim***

Abstract

Malicious code distribution on the Internet is one of the most critical Internet-based threats and distribution technology has evolved to bypass detection systems. As a new defense against the detection bypass technology of malicious attackers, this study proposes the automated tracing of malicious websites in a malware distribution network (MDN). The proposed technology extracts automated links and classifies websites into malicious and normal websites based on link structure. Even if attackers use a new distribution technology, website classification is possible as long as the connections are established through automated links. The use of a real web-browser and proxy server enables an adequate response to attackers' perception of analysis environments and evasion technology and prevents analysis environments from being infected by malicious code. The validity and accuracy of the proposed method for classification are verified using 20,000 links, 10,000 each from normal and malicious websites.

Keywords

Auto Link Tracer, Drive-by Download, Malicious Website, MDN, Real Browser and Forward Proxy

1. Introduction

The recent growth in online services has not only offered convenience to everyday life but also increased threats to Internet users. Services such as online banking, shopping, and social networking, which depend on personal or financial information, are particularly susceptible to threats. One of the most common threats is the drive-by download attack. Drive-by download attacks entice users to sites that distribute malicious code that is designed to infect the user PCs. Vulnerable PCs become infected with malicious code simply by accessing such sites, which is a reason this type of attack is considered one of the most critical online threats [1–3].

Research on methods of countering drive-by downloads can be classified into three general analytical approaches: webpage static [4–10], execution-based dynamic [11–13], and binary [14–17] analysis. However, such studies possess limitations because they rely on signatures such as anti-virus engines, similarity comparison with past (previously detected) data, and behavior analysis. Static analysis is constrained by issues such as distribution script obfuscation and high false positives; dynamic analysis

Manuscript received July 26, 2016; first revision November 29, 2016; accepted December 6, 2016.
Corresponding Author: Yong-Min Kim (ymkim@chonnam.ac.kr)
* Cyber Security Research Center, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea (csyong95@gmail.com)
** Major in Computer Engineering, Chonnam National University, Yeosu, Korea (cglim@jnu.ac.kr)
*** Dept. of Electronic Commerce, Chonnam National University, Yeosu, Korea (ymkim@chonnam.ac.kr)

uses analysis environments that can be easily identified by attackers, which increase their chances of bypassing the behavioral monitoring process. Binary analysis has the same limitations.

To overcome these limitations, this study proposes a method of analysis that does not require website content analysis, extraction of website links, or behavior analysis emulation. This study conducts a comprehensive analysis of the total cost involved in visiting websites through automated links and classifies websites as malicious and normal. The reliability and accuracy of the proposed method in identifying malicious websites are verified through normal and malicious website links collected from the Internet.

2. Related Work

2.1 Malware Distribution Network

To infect PCs with malicious code, attackers create a network that connects landing sites with malicious-code distribution sites. This network is known as a malware distribution network (MDN) [4]. JavaScript and iframe tags are used to enable automatic connections without any action on the part of the users. The inserted link information is obfuscated [18] to interfere with analysis. Vulnerable PCs become infected with malicious code simply by accessing such sites. To attract users to the distribution sites, normal websites having a high number of user connections are injected with code that automatically connects users to the distribution sites [19].

2.2 MDN Analysis Methods

With malicious code distribution emerging as a critical online threat, an extensive analysis of the distribution sites has been conducted. The major research areas are static [4–10] and dynamic analysis [11–13,20]. In some cases, the latter includes binary behavior analysis.

Common methods of static analysis are the signature-based method, which analyzes abnormal content in websites, and meta-information comparison, which involves a statistical analysis of meta-information for comparison with websites commonly used for malicious code distribution. Static analysis decodes content that has been obfuscated in web-sites and the rate of analysis is faster than dynamic analysis because the content is directly analyzed. However, it is limited in its ability to counter the diverse methods of obfuscation. Static analysis is thus less effective in combating new attacks using evolved obfuscation methods.

Dynamic analysis uses virtual machines or emulators to examine changes in PCs after direct visits to websites. Using analysis environments similar to those in actual computing, this method does not have to consider obfuscation. It can serve as an effective solution as long as a well-defined malicious profile exists. This is because it analyzes changes in the actual computer, such as to the registry, file system, network, and process. However, a well-defined profile is difficult to acquire and evasive codes are not easy to combat [16]. An analysis environment also faces the risk of being infected by malicious code. Binary behavior analysis, as an expansion of static analysis, analyzes the binaries downloaded during website visits.

The four existing detection technologies capable of bypassing analysis environments are hardware,

execution environment, external application, and action detection [16]. Hardware detection is a method of detecting virtual machine devices and can be used to detect network interface defined under VMware such as pcnet32. Execution environment detection is used to determine whether a binary execution environment is capable of monitoring the debugger status and other processes. External application detection detects whether known analytical tools such as the process monitor are running. Action detection monitors user actions such as mouse clicks and keyboard input to distinguish between malicious code environments and user environments and delays the time involved in process execution. Static analysis may not be effective in responding to malicious code built on intelligent bypass technology.

Data collection for the analysis can be categorized into two methods. The first method mirrors all web traffic of the target environment [21]. However, it is less applicable to encrypted traffic. The second method collects user traffic using a forward proxy server [13]. Although the second method is effective in decoding encrypted traffic, it is relatively slow because all user traffics must be processed by the proxy server.

2.3 Characteristics of MDNs

As previously mentioned, an MDN can contain obfuscated pages or scripts that detect analysis environments. These characteristics are insufficient as a standard for classifying sites into malicious and normal webpages. This is because normal code is also obfuscated for protection and the obfuscation method may be similar to that of malicious code. That is, website classification cannot be based simply on the properties of singular webpages that constitute links. A more reliable method of classification is required to distinguish normal from malicious webpages.

This study analyzes the automated link structure of normal websites and MDNs to classify normal and malicious websites. Our analysis revealed the differences between connected links in five major areas. Clearly, analyzing an MDN configuration and normal link structure is difficult when the five properties are individually examined. By focusing on the necessity of certain properties and whether they can be easily modified by attackers, this study applied relative weights and performed a correlation analysis between the links.

1) **Host Properties between Root URI and Referrer:** The URIs of normal links are typically connected to resources within a host and make the host of the root URI the same as that of the referrer. Conversely, an MDN contains links connecting normal websites to distribution sites created by attackers. To attract as many users as possible to the distribution sites, attackers operate distribution sites that are unlike normal websites. Thus, the MDN is likely to have different host values for the root URI and referrer [4,12,19].
2) **Domain Properties between Root URI and Referrer:** Similar to host properties, the root URI and referrer have different domain values. In general web hosting, a single domain is used to accommodate several hosts. Table 1 displays several URIs and their referrers for http://www.msn.com. As indicated in Table 1, the URIs have the same domain despite having different hosts. Similar to host properties, an MDN can have different domains for the root URI and referrer. Table 2 illustrates a typical MDN where four websites are connected and users are led to download portable executable (PE) files. Different domains exist for each node within the automated links, including crows.co.kr, filehon.com, ytmall.co.kr, and coocera.com [12,21].

3) **Form of URI is IP Address:** Normal websites use systematic domains instead of IP addresses and assign URIs for various services. As indicated in Table 3, an MDN typically consists of IP addresses. This can be traced to the attackers' intention of enhancing the mobility of the distribution sites [10,22]. IP addresses are used when inserting links that connect hacked websites to distribution sites to avoid the cost of retaining domains or the unavailability of domains previously identified by detection systems. The use of IP addresses is an effective option for attackers running distribution sites, which are usually changed after a short period.

Table 1. Domain and host configuration of normal websites

URI	Host	Domain
http://c.msn.com/c.gif?udc=...< omitted >…	c.msn.com	msn.com
http://otf.msn.com/c.gif?evt=...< omitted >…	otf.msn.com	msn.com
http://rad.msn.com/ADSAdClient31.dll?GetSAd=...< omitted >…	rad.msn.com	msn.com
http://otf.msn.com/c.gif	otf.msn.com	msn.com
http://rad.msn.com/ADSAdClient31.dll?GetSAd=...< omitted >…	rad.msn.com	msn.com
http://c.msn.com/c.gif?udc=true&rid=...< omitted >…	c.msn.com	msn.com
http://otf.msn.com/c.gif	otf.msn.com	msn.com
http://otf.msn.com/c.gif	otf.msn.com	msn.com
http://otf.msn.com/c.gif	otf.msn.com	msn.com
http://g.msn.com/view/30000000000223966?EVT=...< omitted >…	g.msn.com	msn.com
http://g.msn.com/view/43000000000221750?EVT=...< omitted >…	g.msn.com	msn.com
http://otf.msn.com/c.gif	otf.msn.com	msn.com
http://otf.msn.com/c.gif	otf.msn.com	msn.com

Table 2. Domain and host configuration of an MDN

URI	Host	Domain
http://www.crows.co.kr	www.crows.co.kr	crows.co.kr
http://filehon.com/?p_id=dream1434&category_use=1&layout=01&category=&pop=y	filehon.com	filehon.com
http://ytmall.co.kr/vars/11/a.html	ytmall.co.kr	ytmall.co.kr
http://coocera.com/new/bbs_sun/files/s.exe	coocera.com	coocera.com

Table 3. An MDN composed of IP addresses

http://www.19x.co.kr/
→ http://223.255.222.85:8080/index.html
→ http://223.255.222.85:8080/ww.html
→ http://223.255.222.85:8080/ww.swf

Table 4. Country codes in an MDN

http://jmdc.onmam.com/[**KR**]
→ http://hompy.onmam.com/portal/bgmPlay.aspx?hpno=69504[**KR**]
→ http://cus.flower25.com/img/pop/rc.html[**KR**]
→ http://eco-health.org/upload/ad/index.html[**KR**]
→ http://count34.51yes.com/sa.htm?id=344119155&refe=&location= http%3A//ecohealth.org/upload/ad/index.html&color=32x&resolution=1024x768&returning=0&language=undefined& ua=Mozilla/5.0%20%28compatible%3B%20MSIE%2010.0%3B%20Windows%20NT%206.1%29[**CN**]

4) **Country Properties of URI:** Normal websites have the same country code as that of the domain of their related websites. Global services such as Google Analytics, Facebook, and YouTube may have different country information within automated links; however, the domains of the majority of the web services have the same country code. Moreover, attackers include malicious code distribution sites in website links that constitute an MDN. To avoid tracking distribution sites, the attackers may insert country information that is different from that of the hacked sites. As indicated in Table 4, the root URI and intermediate site in an MDN can have different country codes [10].
5) **File Properties of URI:** The ultimate purpose of drive-by download attacks is to infect user PCs with malicious code. In the majority of cases, the malicious code is downloaded in the form of an executable file in the MDN. If the user PC is not vulnerable, the executable file may not be downloaded, even when the user connects to the distribution site. The presence of an MDN cannot be determined solely by the properties of the downloaded file. This is even truer given the frequent changes to malicious code in the distribution sites. As illustrated in Table 2, if connecting to a certain website triggers the downloading of a PE or executable file, it is highly likely to be a constituent of an MDN [21].

2.4 Considerations in Analysis

In addition to analyzing previous research, this study proposes three considerations for effective analysis of an MDN. First, the proposed system must be capable of effectively combating new distribution methods including obfuscation. Rather than analyzing meta-information, signatures, and other content that can be easily evaded by attackers, analysis should focus on elements essential to the MDN, such as website link structure and webpage type. Second, the system must be capable of responding to bypass technology. The proposed system relies on real browsers and does not require that any detection programs be installed in the analysis environments. Finally, the system must be able to process encrypted traffic. To achieve this goal, forward proxy servers are employed in our study. Another advantage of using proxy servers is that the filtering of inbound traffic prevents analysis environments from being infected with malicious code. If an executable file is present in the response data, it is logged and deleted by the proxy server.

3. AutoLink-Tracer for Classification of Malicious Website

3.1 Definition to Automated Webpage Links (AutoLink)

Links constituting web services can be classified into <a href> tags that enable access through mouse

clicks and other user actions, as well as <iframe> or <JavaScript> that enable automatic connections without clicks. In general, the latter is used in an MDN. Other than iframe and JavaScript, links can use meta-tags such as location. A group of automatically linked websites can be expressed in the form of nodes and relations as illustrated in Fig. 1.

In this case, the nodes represent a webpage linked to the src properties of iframe or JavaScript. The relations indicate that the node is connected to other nodes.

3.2 Automated Link Analyzer

The website first visited by a user is the *Root_Node*, and the node that is automatically linked is the *Hopping_Node*. The final node that possesses no further automatic linking is labeled as the *Last_Node*. All automated links can be expressed in graphical form. The definition of the node is given by Eq. (1).

$$\begin{aligned} Node_{(RN)} &= Root\_Node\ \ of\ \ Automated\ \ Link\ Graph \\ Node_{(HN)} &= Hopping\_Node\ \ of\ \ Automated\ \ Link\ Graph \\ Node_{(LN)} &= Last\_Node\ \ of\ \ Automated\ \ Link\ Graph \end{aligned} \tag{1}$$

In Fig. 1, the strength of the connection between the nodes is the relative strength of the logical connections between $Node_{(RN)}$ and the remaining nodes. For instance, when a specific node is linked to the same site as $Node_{(RN)}$, the connection strength is relatively greater than that of non-$Node_{(RN)}$ connections. The greater the connection strength, the lower the cost involved in connecting $Node_{(RN)}$ to the corresponding node, and vice versa. Connection strength is thus inversely proportional to the cost of visit. The automated link analyzer calculates the cost of visit by considering connection weights and the cost of visit for each node. It then performs the classification of MDNs and non-MDNs.

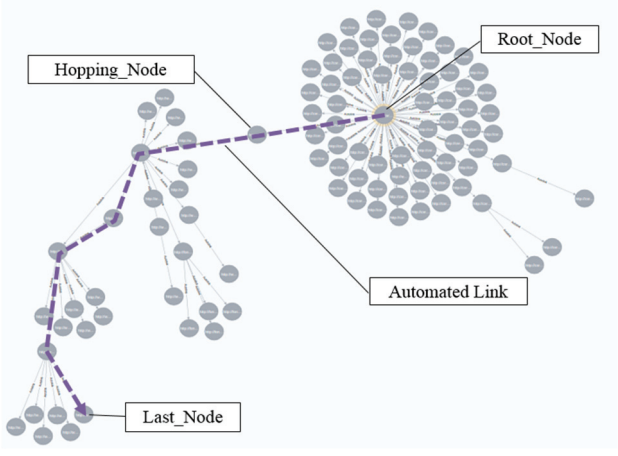


Fig. 1. Configuration of automated linked pages.

3.3 Connection Strength of Automated Link

To calculate the cost of visit to the node of an automated link, the simple connection strength (SCS) of each node must be measured. As indicated in Table 5, SCS is represented by zero for a strong connection

and one for a weak connection. That is, a strong connection results in a lower cost of visit. The five characteristics analyzed in Section 2.3 serve as the criteria for measuring SCS.

An MDN can contain $Node_{(HN)}$ or $Node_{(LN)}$ with different hosts, domains, or country codes from $Node_{(RN)}$. Further, in an MDN, the downloaded file is usually in the form of a PE or other executable. If the URI of the node is an IP address, it is likely to be the $Node_{(HN)}$ or $Node_{(LN)}$ of the MDN. Table 6 indicates the connection strength of each node. We allocate a small weight to characteristics that can be easily evaded by attackers and a high weight to all others. A high weight is assigned to characteristics essential for the MDN configuration; otherwise a low weight is assigned. For example, an IP address can be easily modified by attackers to a non-IP address. An IP address is also not a requirement for an MDN. However, the downloading of executable files is essential for the MDN configuration, even though the file type can be modified. Because normal sites are hacked to become malicious code distribution sites, host and domain names should be different. These are difficult to bypass and a URI is necessary for the MDN. Connection weights (CW) are assigned to the five characteristics, as presented in Table 6.

Table 5. Measurement of SCS in automated links

Node attribute	Measured value
Relation between current node and $Node_{(RN)}$ (eq/non-eq)	
Hostname	0 / 1
Domain	0 / 1
Hosted Country	0 / 1
Current node	
Type of URI	
IP	1
Non-IP	0
File attribute	
Executable	1
Non-executable	0

Table 6. Connection weight of SCS characteristics

Connection weight		High need		Low need	
		Feature	Weight	Feature	Weight
Possibility of analysis avoidance	High	File attribute	2	Type of URI	1
	Low	Hostname/Domain	4	Hosted country	3

Eq. (2) is the connection strength (CS) measurement for a single node in an automated link considering SCS and CW.

$$CS_{(Node)} = \sum_{k=1}^{5} \left\{ SCS_{(k)} \times CW_{(K)} \right\}$$
$$Node \in \left\{ Node_{(HN)}, Node_{(LN)} \right\} \quad (2)$$

3.4 Cost of Automated Link Visit

CS is the strength of the connection between $Node_{(RN)}$ and the present node. The higher the value of CS, the lower the cost of visit. Thus, the cost of visit to $Node_{(LN)}$ of automated link (CAL) is the sum of CS of all the nodes constituting the automated link and the absolute distance (AD) between the nodes multiplied together (see Fig. 2).

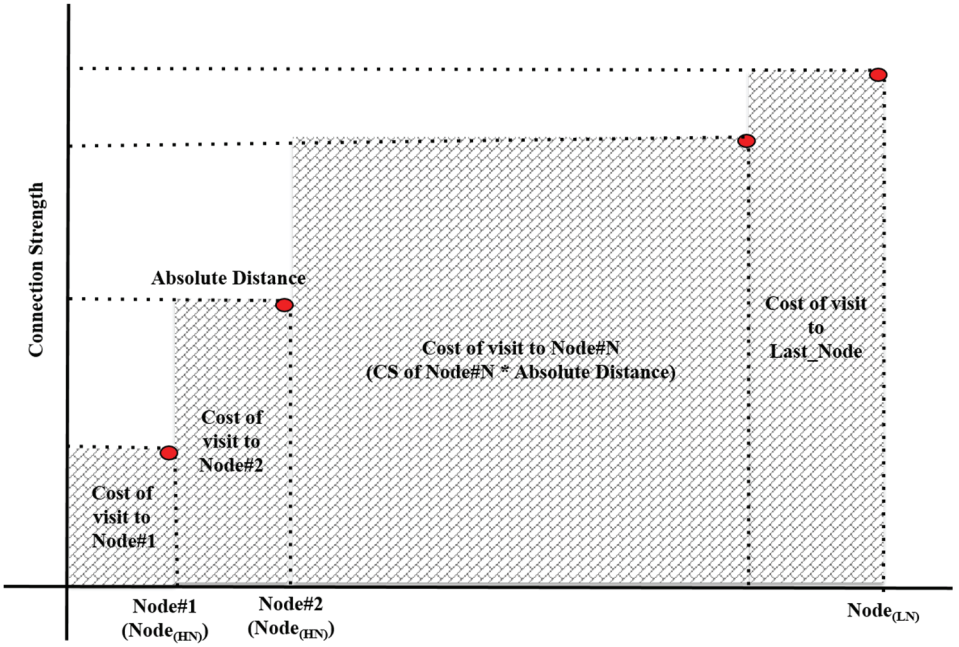


Fig. 2. Total cost to visit all nodes of automated link.

However, because automated links consist of a different number of nodes, summing the CS values may be insufficient for classifying MDNs and non-MDNs. For a relative comparison, normalizing the CS of the automated links using a different number of nodes is necessary. Assuming N is the maximum number of nodes, the relative distance (RD) can be obtained by dividing *N* by the number of nodes. CAL, which is the sum of all CS and RD values multiplied together, can be derived by Eq. (3).

$$
\begin{aligned}
CAL_{(Link(i))} &= \sum_{k=1}^{NC_{(Link(i))}} \left\{ CS_{(Node(k))} \times RD \right\} \\
NC_{(link(i))} &= Node\ \ Count\ \ of\ \ link_{(i)} \\
RD &= \frac{N}{NC_{(link(i))}}
\end{aligned}
\tag{3}
$$

Because an MDN consists of normal nodes that also contain nodes inserted by the attackers, a high number of nodes means that the link is likely to be an MDN. As indicated in Eq. (4), the final decision cost (DC) can be derived by adding the node count rate (NCR) to CAL.

$$DC_{(Link(i))} = CAL_{(link(i))} + \left(CAL_{link(i))} \times NCR\right)$$
$$NCR = \begin{cases} NC_{(link(i))} \le 10 : \dfrac{NC_{(link(i))}}{10} \\ NC_{(link(i))} > 10 : 1 \end{cases} \quad (4)$$

Table 7 presents the operating processes of the system.

Table 7. Operation of AutoLink-Tracer

input *URL: one of URLs in database*

output *MaliciousLink[]: Array of [The linked nodes by automated]*

declare *NURL: Now selected URL in scheduler*

SCS : Simple Connection Strength of current node

CS : Connection Strength of current link

CAL : Cost of visit to Automated link

DC : Decision Cost to visit current link

CW : Connection Weight of each Feature

NCR : Node Count Rate

algorithm *Auto-Tracer*

begin

STEP 1: *// initialization*

MaliciousLink[] = NULL

STEP 2: *// Visit Website*

loop

if exist not accessed URL in scheduler ***then***

URL = select not accessed URL in scheduler

Capture URI, Referrer, Hostname, DomainName, Country, Filetype by Proxy

Insert Capture data to database

STEP 3: *// Calculate Connection Strength*

loop

select URI, Referer, Hostname, DomainName, Country, Filetype from Database

SCS = Calculate Simple Connection Strength of each node

*CS = sum(SCS * CW) of all node in current link*

*CAL = sum(CS * 10/Nocd_count) for all node in current link*

*DC = CAL + (CAL * NCR)*

if DC include boundary of MDN

MaliciousLink = Now link

return Malicious_Link

end *Auto-Tracer*

4. Experimental Evaluation and Analysis

4.1 An Architecture of the Proposed Method

The overall structure of the prototype, called AutoLink-Tracer, is presented in Fig. 3. AutoLink-Tracer is a method that automatically traces links constituting web services and classifies MDNs based on the connection characteristics between the links. It consists of a link-tracing and link-analysis component. The link-tracing module is composed of real browsers and a forward proxy. A forward proxy utilized *Mitmproxy* [23], which is an open source software to analyze both http and https protocols. The automated link analyzer performs link analysis based on logs recorded by link tracing.

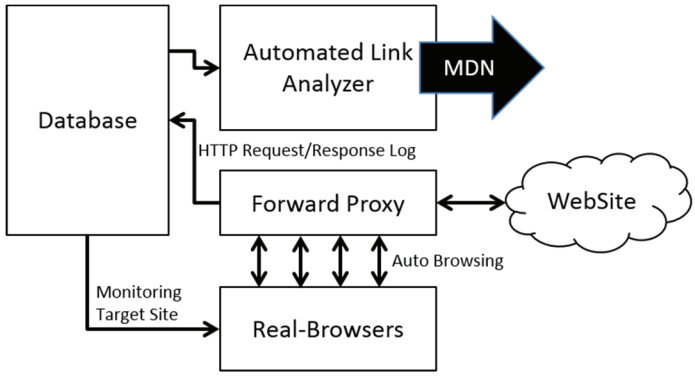


Fig. 3. AutoLink-Tracer: a prototype architecture for the proposed method.

4.2 Experimental Goals and Procedures

The first goal of our study was to determine whether the proposed method can be a standard for classifying malicious and normal links. The second goal was to apply the classification method to malicious and normal links and then assess the performance of the AutoLink-Tracer. Table 8 presents the experimental data and procedures. These links were used as training and experimental data at a ratio of 7:3.

Table 8. Experimental goals and procedures

Goal	Procedures	Dataset
Validity test	Analysis of the distribution situation for malicious/benign link using proposed method Classification equation induction ROC analysis	Malicious link: 7,000 Benign link: 7,000
Classification performance	Optimal cross error rate (CER) derivation False positive and false negative analysis for CER	Malicious link: 3,000 Benign link: 3,000

Ten-thousand MDNs were collected from KAIST Cyber Security Research Center [24] and ten-thousand normal links were collected from normal websites. The collection of normal links is browsing via real web browser. Thus all links are automated linking from webpage in the root website to webpage

in the last leap website same as drive-by download attacks. The configuration of real web browser to collecting normal links is shown in Table 9.

Table 9. Configuration of real browsing

Host OS	Web browser version and options	Plug-in and application
Windows 7 SP1 32 bit	Internet Explorer 10.0 Internet Option - Security Setting : Minimum	Adobe Flash Player 14 .NET 4.0 SDK 1.6 MS Office 2007

4.3 Validity of Experiments

To validate the proposed method, Eq. (4) was applied to extract the DC per link for 7,000 MDNs and 7,000 normal links. Further, the links were classified based on the number of nodes. Fig. 4 presents the automated link distribution with the *x* and *y* axes representing the number of nodes and DC, respectively. The size of the circle is indicative of the number of nodes at a corresponding point. As indicated in Fig. 4, malicious and normal links classified under the proposed method were distributed to different locations. This demonstrates that the proposed method is effective in classifying malicious and normal links.

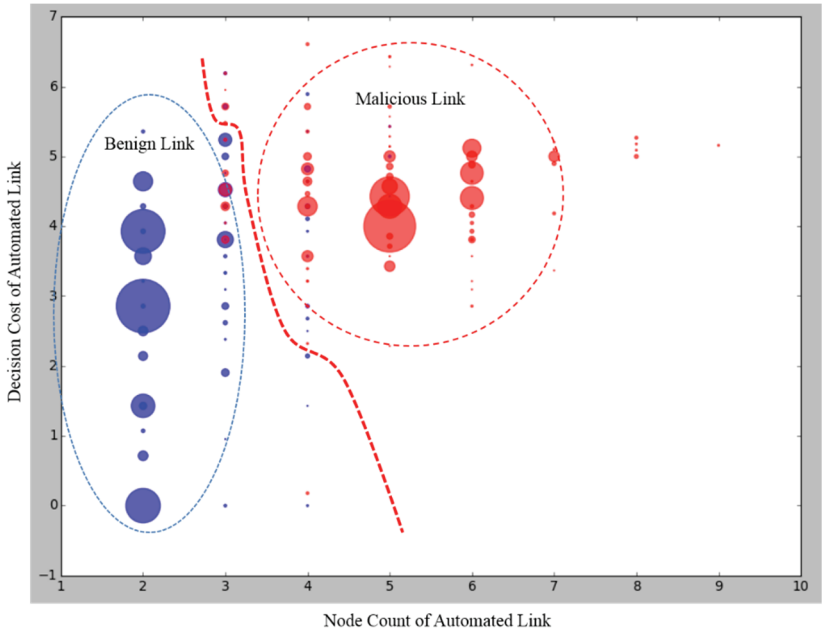


Fig. 4. Distribution of automated links.

To assess the performance of the proposed method for classifying malicious and normal links, ROC curves were compared based on linear and circle equations.

First, as Fig. 5(a) reveals, a linear equation with the negative of the boundary between the malicious and normal links was used as a gradient. Among the malicious links, the gradient was measured using

the (x, y) and (x', y') coordinates at those points having the lowest y-axis and lowest x-axis, respectively. A straight line passing through the two points was drawn. Although the x-axis value varies, the ROC curve of Fig. 5(c) is derived from measurements of true positive rates and false positive rates.

From the ROC curve and its relation to changes in X (from a'' to a' of Fig. 5(a)) presented in Fig. 5(c), the method of classifying malicious and normal links based on the linear equation can be considered highly reliable. Second, the equation of the circle was used as the boundary between malicious and normal links. As illustrated in Fig. 5(b), the center of the circle was (a, b), which had the highest distribution of malicious links. Similar to the linear equation, the ROC curve was derived as the radius increased.

The ROC curve in relation to the radius is indicated in Fig. 5(d). The circle equation demonstrates that the proposed method of classification is highly reliable.

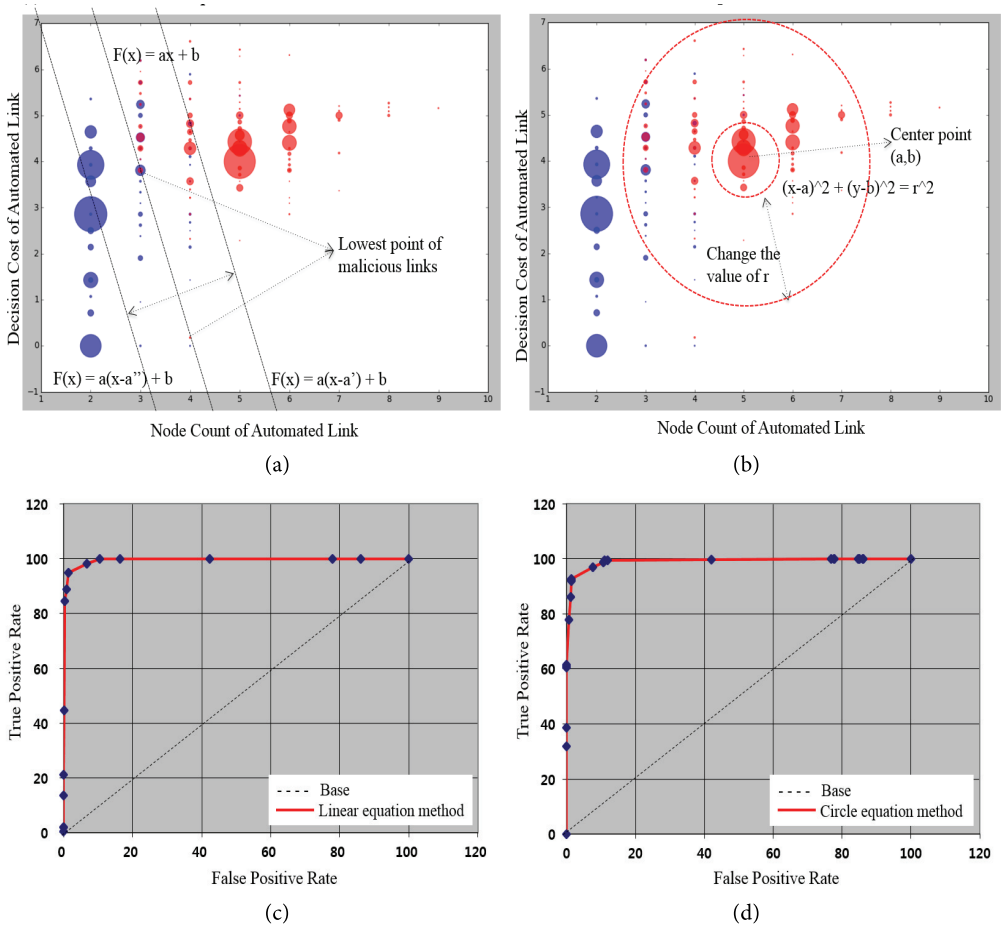


Fig. 5. Results of validity test. (a) Derivation of linear equation, (b) derivation of circle equation, (c) ROC of linear equation, and (d) ROC of circle equation.

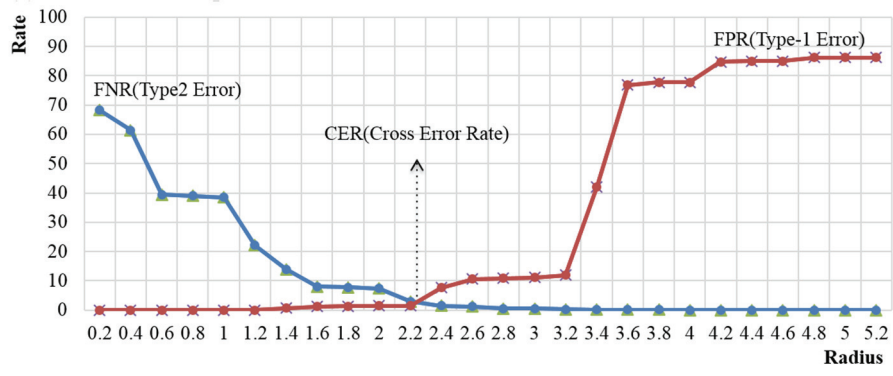
4.4 Evaluation of Classification Performance

Fig. 6 displays our analysis of Type 1 (false positive rate) and Type 2 (false negative rate) errors, which was conducted to derive the optimal values for the two classification methods. As indicated in Fig. 6, the

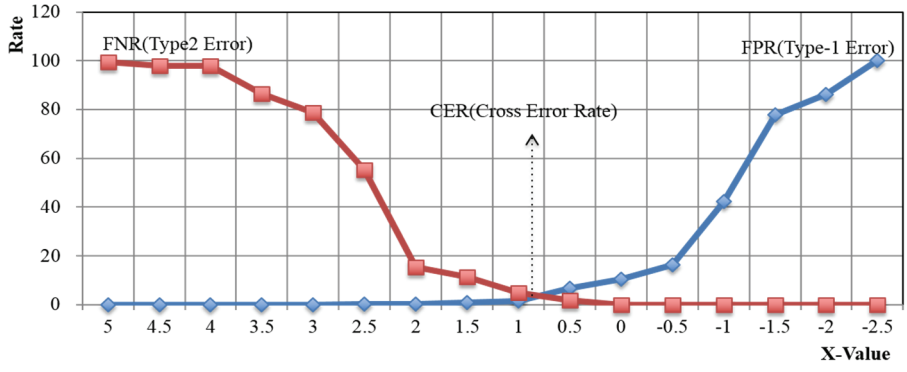
cross error rate (CER) is found at a radius between 2.2 to 2.4 for the circle equation, and at an *x*-value ranging from 0.5 to 1.0 for the linear equation.

The CER values derived for the two equations were used to classify the 3,000 malicious and normal links previously excluded in the validity test. Table 10 illustrates that when the radius is 2.3, the false positive (FP) and false negative (FN) are 7.7% and 2.83%, respectively, with an accuracy of 94.73%. Table 11 indicates that when the *x*-coordinate is *x*+0.8 of the linear equation, the FP is 1.6% and the FN is 2.16% with an accuracy of 98.08%. This indicates that the linear equation is capable of classifying malicious and normal links with a high accuracy of 98.08%.

We can conclude from the two results that the linear equation is superior in terms of accuracy and thus more effective at classifying MDNs and normal links than the circle equation. With a high accuracy of 98.08%, the linear equation fulfills the five measurement criteria defined for classification by AutoLink-Tracer.



(a)



(b)

Fig. 6. Cross error rate (CER) for each equation: (a) circle equation and (b) linear equation.

Table 10. Results of circle equation

Radius	TN (%)	TP (%)	FP (%)	FN (%)	Accuracy (%)
2.20	96.30	92.33	7.67	3.97	94.18
2.30	97.17	92.30	7.70	2.83	94.73
2.40	97.20	90.90	9.10	2.80	94.05

Table 11. Results of linear equation

X	TN (%)	TP (%)	FP (%)	FN (%)	Accuracy (%)
1.0	96.00	98.60	1.40	4.00	97.30
0.9	97.80	98.33	1.67	2.20	98.07
0.8	97.83	98.33	1.67	2.17	98.08
0.7	97.90	97.30	2.70	2.10	97.60
0.6	98.23	97.30	2.70	1.77	97.77
0.5	98.53	95.43	4.57	1.47	96.98

Analysis of the experimental data revealed that the majority of normal links classified as FP were websites that provide global services. These links included YouTube (9EA), advertisement (14EA), and CDN (27EA) links. Websites that provide global services may have domains, host information, and country codes that are different from those of the root node. These services may exhibit a similar link structure to an MDN. However, the greatest difference between MDNs and the global service links is that the latter contain a considerably greater number of child nodes. In developing an actual system, white list filters can be used to reduce the number of false positives.

5. Conclusion

MDNs, which distribute malicious code on the Internet, have surfaced as a critical online threat. Although static, dynamic, and binary analyses have been employed to protect user PCs from malicious code, the development of malicious technology has prevented previous studies from providing effective countermeasures.

This study proposed a new method known as AutoLink-Tracer to classify malicious and normal websites. Forward proxy and real browsers were used to collect website information automatically and the connections of the collected websites were classified according to five characteristics: domain, host, IP server, country code, and webpage type. With AutoLink-Tracer, signature or profile management is not necessary because neither content analysis nor classification is possible based on common an MDN characteristics, even if attacks become more sophisticated.

The effectiveness of the proposed method for classifying normal and malicious websites was verified using a test of normal website links and MDNs. Because traffic that enters an analysis environment can be controlled in the proxy server, the method offers the added advantage of protecting the analysis environment from malicious code. However, the use of a proxy server impairs the collection speed. Further research is required to prevent the reduced speed and to extract properties from a diverse collection of links.

References

[1] European Union Agency for Network and Information Security, *ENSIA Threats Landscape 2014*. Heraklion, Greece: European Union Agency for Network and Information Security, 2015.

[2] F. Y. Rashid, "Department of Labor website hacked to distribute malware," 2013; https://www.securityweek.

com/ department-labor-website-hacked-distribute-malware.

[3] J. Pepitone, "NBC hack infects visitors in 'drive by' cyberattack," 2013; http://money.cnn.com/2013/02/22/technology/security/nbc-com-hacked-malware/index.html.

[4] N. Provos, P. Mavrommatis, M. Abu Rajab, and F. Monrose, "All your iFRAMEs point to us," in *Proceedings of the USENIX Security Symposium*, San Jose, CA, 2008, pp. 1-16.

[5] K. Z. Chen, G. Gu, J. Zhuge, J. Nazario, and X. Han, "WebPatrol: automated collection and replay of web-based malware scenarios," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, Hong Kong, China, 2011, pp. 186-195.

[6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: an application of large-scale online learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Canada, 2009, pp. 681-688.

[7] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 1245-1254.

[8] Y. Shindo, A. Satoh, Y. Nakamura, and K. Iida, "Lightweight approach to detect drive-by download attacks based on file type transition," in *Proceedings of the 2014 CoNEXT on Student Workshop*, Sydney, Australia, 2014, pp. 28-30.

[9] G. Wang, J. W. Stokes, C. Herley, and D. Felstead, "Detecting malicious landing pages in malware distribution networks," in *Proceedings of 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Budapest, Hungary, 2013, pp. 1-11.

[10] G. Stringhini, C. Kruegel, and G. Vigna, "Shady paths: leveraging surfing crowds to detect malicious web pages," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, Berlin, Germany, 2013, pp. 133-144.

[11] A. Moshchuk, T. Bragin, D. Deville, S. D. Gribble, and H. M. Levy, "SpyProxy: execution-based detection of malicious web content," in *Proceedings of the USENIX Security Symposium*, Boston, MA, 2007, pp. 1-16.

[12] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," in *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, NC, 2010, pp. 281-290.

[13] M. Akiyama, M. Iwamura, Y. Kawakoya, K. Aoki, and M. Itoh, "Design and implementation of high interaction client honeypot for drive-by-download attacks," *IEICE Transactions on Communications*, vol. 93, no. 5, pp. 1131-1139, 2010.

[14] Cuckoo Sandbox [Online]. Available: https://cuckoosandbox.org/.

[15] The International Secure Systems Lab (iSecLab) [Online]. Available: https://iseclab.org/.

[16] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Computing Surveys*, vol. 44, no. 2, article no. 6, 2012.

[17] K. Mathur and S. Hiranwal, "A survey on techniques in detection and analyzing malware executables," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3(=, no. 4, pp. 422-428, 2013.

[18] B. I. Kim, C. T. Im, and H. C. 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, 2011.

[19] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser: analysis of web-based malware," in *Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots)*, Cambridge, MA, 2007.

[20] S. Y. Choi, D. Kim, and Y. M. Kim, "ELPA: emulation-based linked page map analysis for the detection of drive-by download attacks," *Journal of Information Processing Systems*, vol. 12, no. 3, pp. 422-435, 2016.

[21] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S. J. Lee, M. Mellia, C. Kruegel, and G. Vigna, "Nazca: detecting

malware distribution in large-scale networks," in *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, San Diego, CA, 2014, pp. 23-26.

[22] S. C. Jeeva and E. B. Rajsingh, "Intelligent phishing URL detection using association rule mining," *Human-centric Computing and Information Sciences*, vol. 6, article no. 10, 2016. https://doi.org/10.1186/s13673-016-0064-3.

[23] Mitmproxy [Online]. Avaialable: https://mitmproxy.org.

[24] KAIST Cyber Security Research Center, "SecureSurf," [Online]. Avaialable: http://csrc.kaist.ac.kr/bbs/board.php?tbl=report.

Sang-Yong Choi https://orcid.org/0000-0001-5152-3897

He received his B.S. degree in Mathematics and M.S. degree in Computer Science from Hannam University in 2000 and 2003, repectively, and Ph.D. degree in Interdisciplinary of Information Security from Chonnam National University, Korea in 2014. He is a research associate professor at Cyber Security Research Center of the Korea Advanced Institute of Science and Technology (KAIST). His research interests are in web security, network security and privacy.

Chang Gyoon Lim https://orcid.org/0000-0002-2295-568X

He received his Ph.D. in Dept. of Computer Engineering in Wayne State University, USA in 1997. Since September of 1997, he has been working for Major in Computer Engineering, Chonnam National University, Yeosu, Korea, as a professor. His current research interests include machine learning, soft computing, intelligent robot, IoT, cloud computing, and embedded software.

Yong-Min Kim https://orcid.org/0000-0002-5066-3908

He received his Ph.D. in Computer Science and Statics in Chonnam National University, Korea. He is a professor at Department of Electronic Commerce, Chonnam National University, Yeosu, Korea. His research interests are in security and privacy, system and network security, application security as electronic commerce.