Wireless Pers Commun (2017) 93:269–283
DOI 10.1007/s11277-016-3866-8

Secure Short URL Generation Method that Recognizes Risk of Target URL

Hyung-Jin Mun[1] · Yongzhen Li[2]

Published online: 24 December 2016

Abstract All the information and data on the Internet are connected based on URL. Although many people use URL to share and convey the information, it is difficult to transmit the information when URL is long and special characters are mixed. Short URL service is a service that transforms long URL with information into short form of URL and conveys the information, which makes it possible to access the page with necessary information. Recently, attackers who want to distribute the malicious code abuse the short URL through SMS or SNS to distribute malicious codes. With the short URL information, as it is difficult to predict the original URL, it has the vulnerability to Phishing attacks. In this study, a method is proposed, which writes the destination information when generating a short URL so that a user is able to check whether the destination is a web document or a file. The service provider of short URL monitors the risk of target URL page of the generated short URL and decides whether to provide service. By monitoring the modification of web-document, it measures and evaluates the risk of the webpage and decides whether to block the short URL according to the threshold, which prevents attacks such as "drive by download" through the short URL.

Keywords Short URL · Short link · URL shortening · Security · Malware · Phishing · Drive by download · Smishing · Shortener service

1 Introduction

Because of SNS activation from the advancement of ICT, in the process of transmission of messages, the necessity to convey various information and data on the Internet happens to arise. Diverse techniques about methods to transmit the information and data on the Internet

✉ Yongzhen Li
lyz2008@ybu.edu.cn

[1] Division of Information and Communication, Baekseok University, Cheonan 31065, Korea

[2] Department of Computer Science and Technology, Yanbian University, Yanji 133002, China

have been researched and developed. For example, as a way to send or share data or information, barcode, QR code, email, and files can be used. Since the location of the information or the data is presented in URL form, it is difficult to write the full address of the URL when one tries to convey or share it with others through SNS because of the restriction like the length of letters. Particularly, the part written by other languages unlike English is not able to convey the URL that is transformed into special characters. In order to solve this problem, the short URL transforms a long URL into a short URL to transmit the short URL to another so that it is able to transmit information and data easily. However, in case of the transformed short URL, as it doesn't reveal the features of linked information or data, unlike the original URL, it is difficult to check which file or web-document the short URL connects him to. Using this vulnerability, attackers use the short URL with Phishing or Smishing attacks [1–4]. Moreover, attackers are able to build a system to provide URL shortening service to distribute malicious codes. An attacker creates a short URL that either connects a user to a risky webpage that has malicious codes or makes him download the codes when it is clicked. The attacker either posts a short URL with a rickrolling article or sends an email or SMS with the short URL in order for the users to click the link of the short URL.

Generally, SNS-related businesses and trusted companies related to Internet generate the short URL and store the related information, and provide services based on them using the short URL. There are service providers who put desirable keywords into the short URL in order to effectively distribute URL with important information and data [5].

Recently, by giving a meaning to the short URL, there have been studies to infer to which page it moves [6]. Furthermore, trusted companies do not generate a short URL and prevent connection in case of an address linked to risky files including execution files. Recently, because it is hard to figure out to which page it links the user only with the short URL, more evolved attacks than the existing ones using email or malicious code via social engineering hacking could come out as a form of image or video linking a short URL, which induces user to click [7]. Especially, the cases where, via SMS and SNS, attackers send a short URL and lure users to click so that the malicious codes are installed in a smartphone have been increasing [8, 9]. Since smartphones are vulnerable to malicious codes and it is difficult to execute real time monitoring programs for many reasons and they store diverse and sensitive information, attacks on them have discreetly been tried.

The ratio of webpages hiding malicious codes that is able to attack a normal webpage accounts for .1–.6% [10, 11]. Klien et al. [12] after analyzing emails containing 5957 short URLs, found out that there are 4780 junk emails and 1177 non-junk emails. In other words, 80% of short URLs are linked to junk.

In this paper, we propose four requirements to solve the vulnerability of the short URL as follows. The four requirements is the requisite to defend against attacks exploiting the vulnerability.

R1 The user should be able to figure out if the target URL of a short URL is a file or webpage.

R2 In case of suspected filename extension, the risk should be noticed and the short URL not be generated.

R3 Service providers that provide shortening of URL should monitor the information of the target URL to secure credibility of the platform [13].

R4 According to the evaluation whether a malicious code is downloaded through the target URL in a short URL, or the URL is redirected to a suspected risky page, the device should restrict services.

In this paper, the solution to the vulnerability of the short URL is to be found in terms of the user and the system. Regarding the user, the user of a device is able to check which data that the target URL of the short URL links him to; As for system, apps in a device or a web browser should restrict any suspected short URL service according to the verification of an organization's credibility. A third party, a trusted authority, which judges the risk of a short URL in a device is necessary. The service with which a platform or a web browser automatically judges a verified short URL service from trusted organizations is necessary. Especially, the method that short URL service providers manage the information of the short URL and investigate whether it contains the malicious code and check there is the modification in the linked file and inspect whether it is risky page or not is proposed.

This paper is organized as follows. In the next section, a short URL service and transform function are introduced. In Sect. 3, the method and algorithm to generate a short URL with a destination revealed are suggested. In Sect. 4, risk inspection and proposal method are evaluated and there are discussion and conclusion in Sect. 5.

2 Related Work

2.1 Short URL

In order to provide accurate information, data, and the location, a long URL is necessary. Short URL is a short URL linked to a long URL, one to one. In other words, shortening service redirects a long URL to a short URL. Figure 1 shows a procedure in which with a user clicking on the short URL, TS with the function, "HTTP 301 Moved Permanently", redirects it to the targeted URL [6].

Figure 2 shows the process in which the user is connected to Target URL step by step. User goes through DNS (step 1) and he is provided with IP of a short URL transform server (step 2) and he has an access to TS (step 3). In order for TS to check the target URL linked

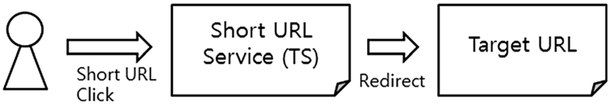

Fig. 1 Short URL service process

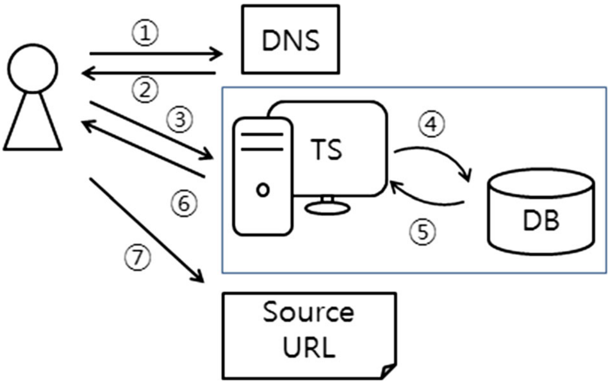

Fig. 2 The process to search for target URL through the short URL

to the short URL, it inquires the information of the database in TS (step 4, 5) and redirects to the target URL (step 6, 7).

Because it passes through TS of a company to access the target URL, if the shortening service provider were the trusted organization, it would be utilized as the line of defense that blocks TS's malicious codes or suspected risky pages.

In order to provide the short URL service, two are required. First, the generated short URL information should be one and only. In other words, the chance to collide should be excluded. Second, the target URL of the generated short URL should be stored and managed in Database. Besides, the generation and inquiry of the short URL should be processed in a short time.

Like Fig. 3, in order to exclude the chance to collide, it allocates URL information in order (a), or marks it as two information using hash function like CRC16 (b), or uses keyword extraction information and hash value and adds collision prevention information (c) [6].

As for Google, it provides the shortening URL service via http://goo.gl [14]. By utilizing the original URL information, when generating a short URL, it provides a different short URL according to login state and account. It seems to be an action to verify the subject that generates the short URL. Moreover, it creates a different URL when requested.

Table 1 generally shows the information table of URL shortening service providers. In case of a nonmember of Google, a short URL with five characters such as http://goo.gl/g7y*X is generated to a certain webpage. However, in case of a member of Google, the short URL information with six letters which are alphabetic or numerical or combined with both like http://goo.gl/akF**x can be generated, which means the number of cases reaches up to 56.8 billion.

Although the special characters like "_" that cannot be used in generating domain are used, it uniformly changes it into a short URL without inspection as seen in Table 2. The short URL providers like *goo.gl* and *bit.ly* block suspected execution files but other providers have been providing services with no restriction even if there are suspected files. In other words, since different organizations providing the short URL service do not examine

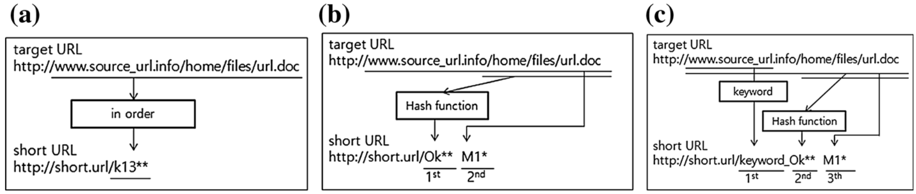
(a)
target URL
http://www.source_url.info/home/files/url.doc
in order
short URL
http://short.url/k13**
(b)
target URL
http://www.source_url.info/home/files/url.doc
Hash function
short URL
http://short.url/Ok** M1*
1st
2nd
(c)
target URL
http://www.source_url.info/home/files/url.doc
keyword
Hash function
short URL
http://short.url/keyword_Ok** M1*
1st
2nd
3th

Fig. 3 The hash value generation of a normal short URL. **a** In order, **b** using hash function, **c** using keyword

Table 1 DB information of Google's short URL service

Short URL	Target URL	Count	Etc.
xQD**Q	google.co.kr		
NAq**A	gmail.com		

Table 2 Generation results of short URLs of *goo.gl* and *bit.ly*

Target URL	Short URL of Goo.gl	Result	Short URL of Bit.ly	Result
source_URL.info/test.pdf	http://goo.gl/iv**ji	Ok	http://bit.ly/1Og**cV	Ok
source_URL.info/test.doc	http://goo.gl/HP**qe	Ok	http://bit.ly/1WM**my	Ok
source_URL.info/test.zip	http://goo.gl/pb**sp	Ok	http://bit.ly/1Zk**S5	Ok
source_URL.info/test.php	http://goo.gl/OJ**XJ	500	http://bit.ly/1rv**X2	500
source_URL.info/test.exe	This URL cannot be shortened. Please try another one	Unable	http://bit.ly/1Yb**i0	Stop
source_URL.info/test.dll	http://goo.gl/qm**gv	Wait	http://bit.ly/24w**Qv	Ok
source_URL.info/test.scr	http://goo.gl/uR**U8	Wait	http://bit.ly/1rv**5f	Stop
source_URL.info/test.dat	http://goo.gl/vr**rk	Ok	http://bit.ly/1rv**4R	Ok
source_URL.info/test.jsp	http://goo.gl/2g**Ry	Ok	http://bit.ly/1O1**uC	Ok
source_URL.info/test.abc	http://goo.gl/YB**Im	Ok	http://bit.ly/1Oj**p9	Ok
source_URL.info/test.exe.zip	http://goo.gl/Pn**8F	Ok	http://bit.ly/1VP**Zq	Ok

500: HTTP ERROR 500

STOP: STOP—there might be a problem with the requested link

wait: While the short URL is generated, when clicking, it shows the warning message, “this *goo.gl* shortlink has been disabled”

the information and content of the original URL, there is possibility of 2nd and 3rd damage.

2.2 Transform Function

In order to generate a short URL, the function that transforms the original address information into fixed length of a short URL is necessary. Each of the short URL service providers uses a variety of transform functions. The condition that transform function abides by is that the chance to collide should be low.

Given x, it is computationally infeasible to find y which is not equal to x such that $TF(x) = TF(y)$, where x, y are long URLs and $TF(x)$, $TF(y)$ are short URLs.

In addition, the function value should have a short and fixed length of output. The hash function is available to be used as the transform function. The hash function, one-way function, is transformed into the function with a fixed length of character string (Fig. 4). Besides, because there is no inverse function, the original message cannot be identified with the value of the hash function. With this characteristic, the password input is replaced with a hash value in the DB not to expose the password in the ID and password verification system.

The hash function in the field of the cryptology field is usually utilized as Message Digest to verify the integrity of the message. Especially, in case of an open source or a program distributed on the web, because an attacker can distribute them to set malicious codes, as an inspector to check any modification of a program, the hash functions such as MD5 and SHA-1 are utilized.

Although it may vary depending on the short URL service providers, because the length of the transform function for the short URL service is not long, there are no possible

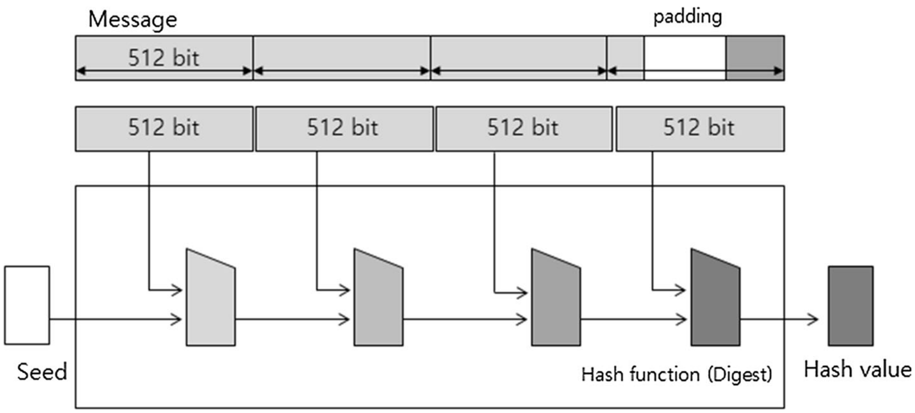

Fig. 4 Hash value generation

clashes, and a transformed hash function that is created in a short form is used. However, since there is no need to be one-way regarding the characteristic of the short URL, according to the order of requests, a short URL is generated and a service combined with various information is provided [15].

2.3 Drive by Download Attack

An attacker either runs a hidden site with malicious codes to spread the codes and to infect devices or lures users into a maliciously hidden website using the link to the site. In case of one user's PC lacking security patches, the attacker spreads the malicious code with the method of drive-by-download once a user accesses to a website. Even if the connected webpage didn't distribute the malicious code, the PC would be connected to the malicious-code-distributing site, and it would be infected with the malicious code by Drive-by-download Attack [16–20]. Drive-by-download Attack is an attack with which an attacker invades normal websites or infects by inserting a script related to the malicious code [18]. Either hiding the malicious code using iframe tag or Javascript on a webpage or refreshing with Meta tag connects users to a hidden site. In addition, using tags such as <object>, <embed> in a message board, it links to a site hiding the malicious code.

Figure 5 shows the Drive-by-download Attack step by step [16]. Using the vulnerability of the short URL, not only Drive-by-download Attack but other attacks including Phishing, Smishing, Pharming are possible to be made [1, 3, 16, 17].

3 Proposal Method

When generating a short URL with the given original URL, the user should be able to discern whether the short URL points to a file or a web document after looking at the short URL.

SHRT tried to solve the vulnerability of the short URL, by adding keywords from the target URL to the information of the short URL when generating the short URL to solve the problem of the short URL that it cannot find the information of the target URL [6]. A short URL is generated after selecting keywords, extracting consonants of domain of

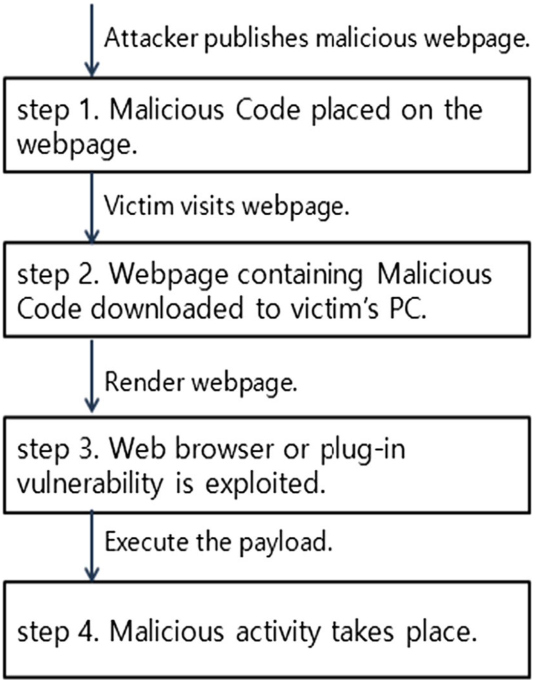
Attacker publishes malicious webpage.
step 1. Malicious Code placed on the webpage.
Victim visits webpage.
step 2. Webpage containing Malicious Code downloaded to victim's PC.
Render webpage.
step 3. Web browser or plug-in vulnerability is exploited.
Execute the payload.
step 4. Malicious activity takes place.

Fig. 5 Step-by-step scenario of Drive-by-download Attack

Target URL. For example, in case of "http://www.sourceURL.info/attack.aspx", it generates "http://short.URL/srcrl_B*7".

However, when the address of the target URL is "http://www.source_URL.info/***/malware.exe" and "malware.exe" is the filename and the extension of the file is execution file, although it is likely to be a malicious code, the SHRT method cannot check it, which is the very weakness of the method.

Attackers upload malicious codes into a hacked webpage for distributing or malicious code distributing website. Attackers generate the short URL for the malicious code to be automatically downloaded. If a file is downloaded when generating a short URL, TS should recognize that and block the generation of the short URL according to the inspection result on harmfulness.

Portable executable (PE) file format has a data structure into which Windows OS loader capsulized the information of execution code. Malicious codes have PE structure [21]. Recently, even though the short URL service providers have been blocking the generation of execution files, in case of *dll* or *scr* which can be abused as malicious codes, the structure should be examined and thereby processed. Regarding uploaded files compressed with malicious codes, since they are generated without verification, downloading the malicious code is possible.

Table 2 shows when the extension of the filename is *scr*, it can be used as the malicious code, nevertheless, generation of the short URL is possible. As remarked on the blog of AlYac, a targeted attack named Spear Phishing has been tried via résumé [22]. Spear Phising is a social engineering hacking method to lure receivers to execute the file in the email attached by an attacker [23]. With the method disguising the file attribute to be seen as a document file, attacks have been tried. As a more evolved method, chances are that making a new extension can be used to attack in the future.

3.1 Destination-Recognizable Short URL Generation Method

A new generation procedure by which the user can check the target URL with the information of the short URL is shown in Fig. 6.

In Fig. 6, after checking whether the analysis result of the given URL is a file or a web document that is interpreted by the server, (1) if it is a webpage, it will confirm the degree of risk and generates a short URL, (2) or if it is a file, it will check if it is a malicious code by a vaccine engine. Like "http://www.virustotal.com", utilizing an inspection engine for malicious code and file, it checks the harmfulness. If the inspection result for harmfulness is (3) higher that the threshold, (4) it bocks the generation of the short URL and access; If the result is lower than the threshold, (5) it examines the type of the file, and it generates the short URL according to the file.

Figure 7 is a pseudo code that presents the algorithm for the proposal method (Fig. 6). For example, as seen in Fig. 8, when the short URL of "http://www.short_URL.info/***/URL.doc"" is transformed into a form like "http://short.URL/doc_sKu***", the user can easily figure out it is the document file of MS.

Table 3 checks the linkage relation between the short URL and the original URL to find out the original one is linked to file or website, and when it is linked to a file, it checks what extension it has, whether it is a malicious code or a suspected file, and it rates the degree of risk; checksum is the information with which in case that it is not a web document, the modification of a file by an attacker can be checked. At the point of generation of a short URL, the hash value is stored and whether for it to be modified is recorded. It is likely that when an attacker generates a short URL, he links it with a normal web document and he would modify the web for distribution of malicious codes later. Because of this, regular

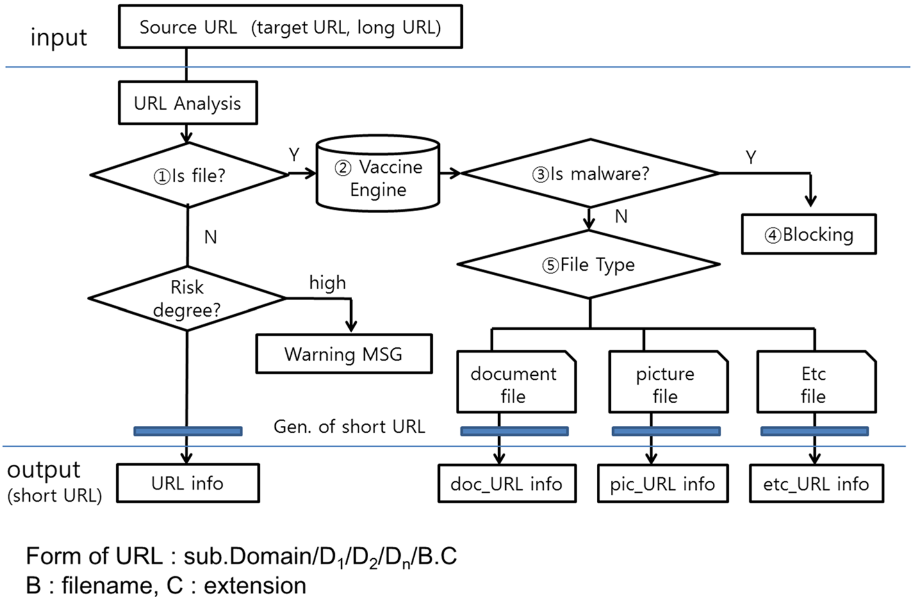

Fig. 6 Generation process of short URL

```
if (B.C is webpage) then
    if (webpage is secure) then
        h←hash(source URL)    // hash function for SHORT_URL
        result←T(h) //generation of short URL
    else break
else
    VirtualMachine(file) //file : B.C (B:filename, C:extension)
    if (file is malware) then
        break
    else
        h←hash(source URL)    // hash function for SHORT_URL
        result ←C_T(h) // C : extension, T(h) : letter transform function
    end    // result : generation of short URL
end
```

Fig. 7 Short URL generation algorithm

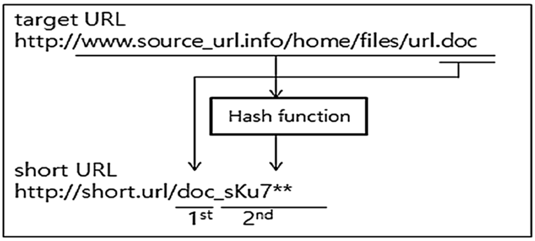
target URL
http://www.source_url.info/home/files/url.doc
Hash function
short URL
http://short.url/doc_sKu7**
1st
2nd

Fig. 8 Adding of meaning to each section of the information of short URL

Table 3 Additional DB information of the short URL service of the proposal method

Shortened URL	Targeted URL	File type	Risk degree	Hash	Checksum	Etc.
xBn**7	source_URL.info	–	.2			
doc_B7B**d	source_URL.info/hello.doc	doc	.17	h(hello.doc)	–	
YB5**d	source_URL.info/hello.php	–	.37			
scr_DeK**Z	source_URL.info/attack.scr	scr	.9	h(attack.scr)	Modified	
pdf_Ca7**n	source_URL.info/hello.pdf	pdf	.21	h(hello.pdf)	–	

monitoring on the web document is necessary, and based on that, it measures the degree of risk [13].

3.2 Transformation and Processing Process of Short URL

When clicking on the link of a short URL generated with the new short URL generation method like Fig. 6, the processing process is as seen in Fig. 9.

Figure 9 shows a process that as a user clicks on the given short URL, web browser or Internet processing module installed in the device platform judges whether it is a file or a web document and processes it. When target URL is a webpage, in order to checked if it is a malicious webpage, the threshold is repeatedly monitored in the transforming service in a

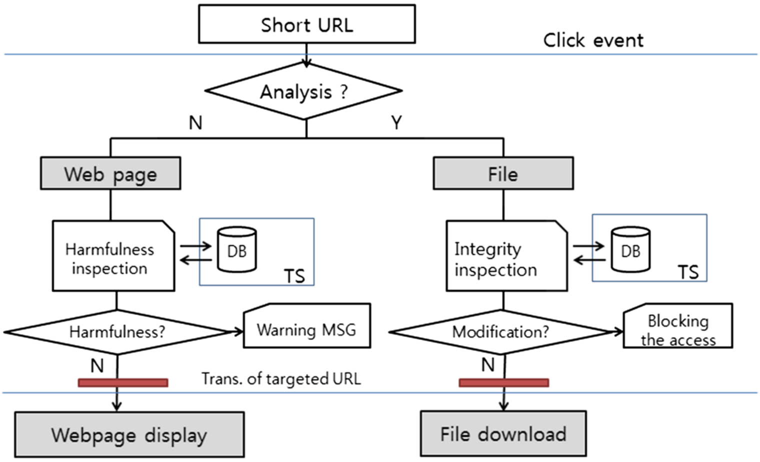

Fig. 9 Process when clicking short URL

certain period and related information is stored as a form of the threshold (Table 3) [13]. If the target URL is a file, after the process to check whether the file is modified by the attacker, the file is to be downloaded, being accessed.

Figure 10 exteriorized the section related to the integrity inspection of Fig. 9 when it is a file. Even though a short URL generated by a trusted provider is generated after checking whether it is a malicious code in the generation process, it can be modified into a malicious code after that. Since chances are that it can be modified as a malicious code later, in order to verify the integrity. The hash value of the firstly generated file is to be stored; if there is any difference between two, it will mark checksum item and block download of the file; if they are identical, the file will be downloaded (Table 3).

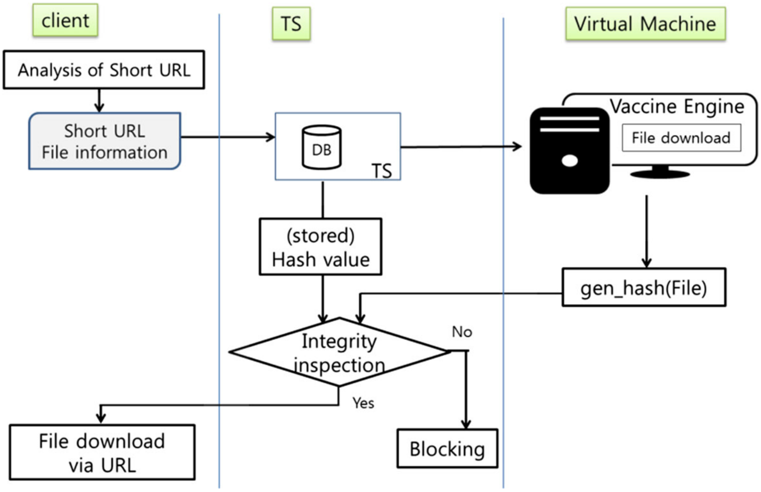

Fig. 10 Checking modification of file linked to short URL

4 Analysis and Assessment

4.1 Inspection for the Harmfulness of the File Linked to Short URL

Figure 11 is a detailed figure of the short URL generation process about the file in Fig. 6. Before generating the short URL, judging the harmfulness of the linked file, if any problem is not detected, it generates the short URL and stores the hash value of the file. In order to inspect the harmfulness of the file pointed by a short URL, it is executed by a virtual machine and judged whether it is the malicious code. If necessary, code analysis is implemented through a vaccine engine like in *Virustotal.com* [24]. Even after the generation of a short URL by generating a hash value, it is utilized in integrity inspection to check further modification of the file.

4.2 Analysis and Assessment of Proposal Method

As seen in the result from running a detection program for site hiding malicious codes from 2008 to 2009, the number of malicious code routing site is three times more than the number of malicious code distributing sites [25]. By collecting the information of the target URL and using search information of malicious code hiding site of Google and detection programs like MCFinder, it analyze the webpage and rates the harmfulness. Digitizing the harmfulness, judging whether to provide the short URL service is required [25].

There are two countermeasures that are related to the inspection of harmfulness of a webpage. As a method to analyze the web document that arrived at the last destination, first, the static analysis, after analyzing the web document, it figures out if there is a shell code or functions related to the distribution of malicious codes. Analyzing the meta information of the webpage, it refers to the data to judge the harmfulness of the page.

Second, the dynamic analysis, by implementing DOM structure analysis or the script included in the web document, analyzes the action and the result. Particularly, when a file is downloaded, a virtual machine executes the file and analyzes what difference it gives, and judges whether it is the malicious code (Fig. 10).

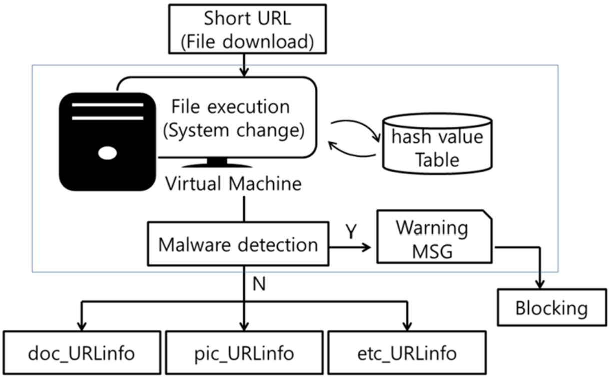

Fig. 11 Checking process of malicious code

Through the static and dynamic analysis, the harmfulness is accumulated in the information of Table 3, and whether to access the target URL from the short URL is judged. If it is judged that the degree of risk of the page of the target URL to be high and there is risk, it outputs a warning message to the user and runs the inspection for the webpage, and checks if there is a malicious code or it is the page with Drive-by-download Attack that automatically installs the malicious code.

This countermeasure is a protection action limited to the trusted short URL service. In case of the short URL service providers who lack protection methods or the short URL provided by an attacker, users cannot but be vulnerable. That is, proposal method cannot be a fundamental solution. As a fundamental solution, the step in which a system checks short URL and judges if it is trustworthy and whether to provide the service should be added. Thus, a trusted authority (TA) is necessary in order to judge the credibility of a short URL providing organization through web browsers or Internet connection module.

Like Fig. 12, TA-based short URL service is necessary. In Fig. 12, the web browser of a user (step 1, 2) with the verification of the credibility of the short URL service (step 3) requests the short URL service. After that, (step 4) he is provided with the service after connected to the target of the short URL.

In the first section, we mentioned four requirements to solve the vulnerability of the short URL. We met four requirements with the proposal method.

- The user can check whether it is web document or file by the prefix of the information of the generated short URL (Fig. 8).
- Before generating a short URL for a file, it implements the inspection for the harmfulness of the file through virtual machines (Fig. 11).
- Regarding the generated short URL, the service provider regularly monitors to inspect the harmfulness and measures the degree of risk (Table 3). Thus, the proposal method, running Table 3, is able to restrict the redirection service according to the degree of risk (Fig. 9).
- The device can restrict the redirection service based on the credibility of the short URL service organization (Fig. 12).

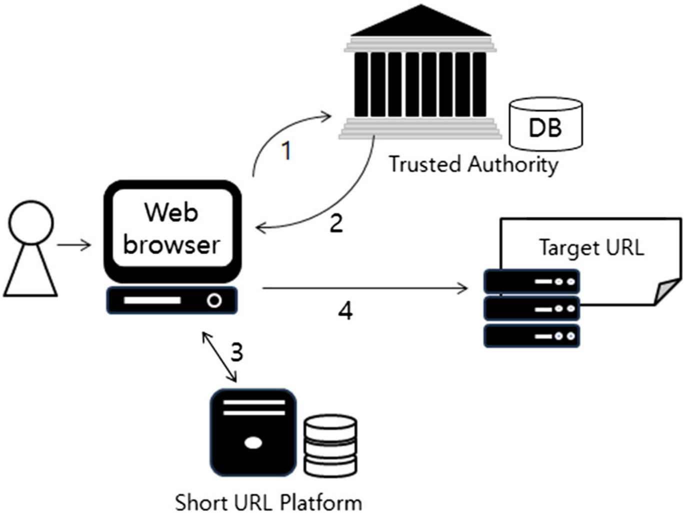

Fig. 12 TA-based short URL service providers and service restriction method

Table 4 Security analysis of SHRT and proposal method

	SHRT [6]	Proposal method
To check risk of destination	Vulnerable	Available
To check file connection	Unavailable	Available
Service blocking function	Unavailable	Available
To draw destination domain	Available	Unavailable

As a method that solve the problem that the destination of a short URL is unknown, SHRT is proposed [6]. Table 4 compared SHRT with the proposal method in terms of security.

While SHRT that extracts keywords of target URL to put them in a short URL enables users to draw the domain of a destination, it doesn't secure the stability about target URL. Though, in SHRT, target URL doesn't discern a web document or a file, the proposal method does that and included service blocking function.

5 Discussion and Conclusions

Many organizations have been transmitting information, utilizing short URL service. It is common that short URLs are transmitted via SNS and SMS. With domain, anyone can use a short URL service. In this circumstance, attackers can easily make the short URL service, and using it, they are able to do attacks such as Phishing, Smishing, and drive-by-download via the short URL of SNS and SMS.

The vulnerability of the short URL is that one is not able to figure out the target URL until he clicks. In other words, he doesn't know the target URL is linked a web document, or whether it would download the malicious code. To solve this, first, the user should be able to check if it is web document or file from the information of the short URL, and it is verified by TA to solve the service credibility problem of the URL in terms of system, and the verified short URL service should be supported.

Using a smartphone, when clicking on an accessible link to Internet through Twitter or SMS, and as Intent call comes out, like Fig. 12, if checking whether it is a credible short URL or not is made in advance, the vulnerability of the short URL can be resolved. Application of the verification-based technology to utilize the short URL to the existing system that generates the short link only with input information in SNS or SMS service using the short URL frequently is required.

In this study, like Fig. 13, the process to generate a short URL on its way was not considered. As a condition to secure stability, with a trusted organization, the verification system to generate short URLs should be applied (Fig. 12). If there is a trusted organization, the first step to generate a short URL in the condition where the second short URL is verified from TA can be considered as the way to add additional information of keywords of the second URL.

As further studies, after building a trusted organization and based on this, the technique that is able to judge whether the accessible link of SNS or SMS on the Internet is the short

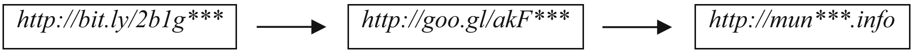

Fig. 13 Multi level short URL

URL or not is required. Via a trusted organization, application of web browser technology that connects users to credible webpages and checks whether it is the trusted service provider is needed.

References

1. Yearwood, J. L., Mammadov, M., & Webb, D. (2012). Profiling Phishing activity based on hyperlinks extracted from Phishing emails. *Social Network Analysis and Mining, 2*(1), 5–16.
2. Mun, H. J., & Oh, S. (2016). Injecting subject policy into access control for strengthening the protection of personal information. *Wireless Personal Communications, 89*(3), 715–728.
3. He, R., Qin, Z., Wang, F., Chang, C., & Qin, X. (2009). Security strategy for mobile police information system using SMS. *Wireless Personal Communications, 51*(2), 349–364.
4. Kang, A., Lee, J. D., Kang, W. M., Barolli, L., & Park, J. H. (2014). Security considerations for smart phone Smishing attacks. *Advances in Computer Science and its Applications, LNEE, 279*, 467–473.
5. Bitly, *Bitly URL shortener and link management platform*. https://bitly.com/
6. Yoon, S., Park, J., Choi, C., & Kim, S. (2013). SHRT: New method of URL shortening including relative word of target URL. *The Journal of the Korean Institute of Communication Sciences, 38*(6), 473–484.
7. Maan, P. S., & Sharma, M. (2012). Social engineering: A partial technical attack. *International Journal of Computer Science Issues, 9*(2–3), 557–559.
8. Kim, K. J., & Kim, J. (2015). A study on the Markov chain based malicious code threat estimation model. *Wireless Personal Communications*. doi:10.1007/s11277-015-3018-6.
9. Lu, H., Zhao, B., Su, J., & Xie, P. (2014). Generating lightweight behavioral signature for malware detection in people-centric sensing. *Wireless Personal Communications, 75*(3), 1591–1609.
10. Seifert, C., Steenson, R., Holz, T., Yuan, B., & Davis, M. A. (2007). Know your enemy: Malicious web servers. *The Honeynet Project*. http://www.honeynet.org/papers/mws/
11. Wang, Y.-M., Beck, D., Jiang, X., Roussev, R., Verbowski, C., Chen, S., & King, S. (2006). Automated web patrol with strider HoneyMonkeys: Finding web sites that exploit browser vulnerabilities. In *Proceedings of Network and Distributed Systems Security Symposium* (pp. 35–49).
12. Klien, F., & Strohmaier, M. (2012). Short links under attack: Geographical analysis of spam in a URL shortener network. In *Proceedings of the 23rd ACM conference on Hypertext and social media* (pp. 83–88). doi:10.1145/2309996.2310010
13. Mun, H.-J. (2015). Polling method based on weight table for efficient monitoring. *Journal of the Convergence Society for SMB, 5*(4), 5–10.
14. Google, *Google URL shortnener*. https://goo.gl
15. UNPLAY, *Free shortener service*. http://muz.so
16. Le, V. L., Welch, I., Gao X., & Komisarczuk, P. (2013). Anatomy of drive-by download attack. In *Proceedings of the Eleventh Australasian Information Security Conference (AISC 2013)* (Vol. 138, pp. 49–58).
17. JooHyung, O., Im, C., & Jeong, H. (2010). Technical trends and response methods of drive-by download. *Communications of the Korean Institute of Information Scientists and Engineers, 28*(11), 112–116.
18. Cova, M., Kruegel, C., & Vigna, G. (2010). Detection and analysis of Drive-by-download Attacks and malicious JavaScript code. In *Proceedings of the 19th International Conference on World Wide Web* (pp. 281–290).
19. Egele, M., Wurzinger, P., Kruegel, C., & Kirda, E. (2009). Defending browsers against drive-by downloads: Mitigating heap-spraying code injection attacks. In *Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment*, LNCS5587 (pp. 88–106).
20. Egele, M., Wurzinger, P., Kruegel, C., & Kirda, E. (2009). Defending browsers against drive-by downloads: Mitigating heap-spraying code injection attacks. In *Proceedings of the 6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment'*, DIMVA'09 (pp. 88–106). Berlin: Springer-Verlag.
21. Park, C., Chung, H., Seo, K., & Lee, S. (2012). Research on the classification model of similarity malware using fuzzy hash. *Journal of the Korea Institute of Information Security and Cryptology, 22*(6), 1325–1336.
22. Alyac blog, *Case study of malicious code with false résumé document file*. http://blog.alyac.co.kr/242

23. Sohn, Y.-s., Nam, K.-h., & Goh, S.-c. (2013). On the administrative security approaches against spear Phishing attacks. *The Korea Institute of Information and Communication Engineering, 17*(12), 253–2762.
24. VIRUSTOTAL, http://www.virustotal.com
25. Shin, H., & Moon, J.-S. (2011). A study on minimizing infection of web-based malware through distributed and dynamic detection method of malicious websites. *Journal of the Korea Institute of Information Security and Cryptology, 21*(3), 89–100.

Hyung-Jin Mun received his BS, and Master degree in Mathematics from Chungnam National University, Republic of Korea in 1996 and 2002. He received Ph.D. degrees in Computer Science from Chungbuk National University in 2008. He was an associate professor in Yanbian University Science and Technology in China. He is currently lecture in Baekseok University in Korea. His research interests include personal privacy, access control, and network security.

Yongzhen Li received his BS, and Master degree in Physics from Yanbian University, China in 1994, and 1997. He received the Ph.D. degree in computer science from Chungbuk National University, Korea, in 2007. He is currently a professor of Department of Computer Science and Technology at Yanbian University in China. His research interests are wireless network protocols, network security and cryptography algorithm.