SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2012; **5**:1181–1192
Published online 6 February 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.420

SPECIAL ISSUE PAPER

A quantitative approach to estimate a website security risk using whitelist

Young-Gab Kim[1], Minsoo Lee[2], Sanghyun Cho[3] and Sungdeok Cha[1]*

[1] Department of Computer Science and Engineering, College of Information and Communication, Korea University, 1, 5-ga, Anam-dong, Sungbuk-gu, 136-701 Seoul, Korea
[2] Division of Computer Science, Department of EECS, Korea Advanced Institute of Science and Technology (KAIST), 335 Gwahangno (373-1 Guseong-dong), Yuseong-gu, 305-701 Daejeon, Korea
[3] Portal Service Security Team, NHN Business Platform, Venture Town Bldg, Jeongja-dong, Bundang-gu, Seongnam-si 25-1 Gyeonggi-do, Korea

ABSTRACT

Despite much research on defense against phishing attacks, incidents continue to occur where sensitive (e.g., personal or financial) information is stolen using social engineering and technical spoofing techniques. Most approaches use the notions of blacklists versus whitelists (WWLs), and it is difficult to quantify the degree of a website's vulnerability against phishing attacks. In this paper, we present a quantitative approach for evaluating the phishing possibility of a given website using the refined security risk elements for domain and web page. Design and implementation of the website risk assessment system for antiphishing are also included. It can detect suspicious websites containing phishing attack and abnormal behavior and generates a warning if website is judged untrustworthy. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS

phishing; pharming; website security risk; website blacklist; website whitelist; risk analysis

***Correspondence**

Sungdeok Cha, Department of Computer Science and Engineering, College of Information and Communication, Korea University, 1, 5-ga, Anam-dong, Sungbuk-gu, 136-701 Seoul, Korea.
E-mail: scha@korea.ac.kr

1. INTRODUCTION

Phishing and pharming attacks pose serious threat to Internet security. The former attempts to steal confidential user information such as credit card numbers or passwords and social engineering and spoofing techniques are frequently used. The latter (e.g., hijacking connection to well-known sites) is another type of phishing attack in which connections to a "genuine" website are secretly diverted to a forged website. DNS (Domain Name System) hijacking or crimeware (e.g., Trojan keylogger spyware [1,2]) are often used to launch pharming attacks. Antiphishing working group [1] reported in its report that more than 210 000 unique phishing sites have been found in early 2009 [3]. In practice, the actual number of such "harmful" sites is most likely to be much higher, and strategies used in phishing attacks continue to become more advanced and sophisticated [4].

Various approaches have been proposed to protect innocent users from dangers of hostile and ruthless cyber attacks, and Internet service providers and enterprise security solution vendors rely on the notions of blacklists (e.g., denoted WBLs and known to be malicious sites) versus whitelists (e.g., referring to safe and genuine sites and called WWLs) associated with E-mail or IP addresses. Unfortunately, existing approaches have several weaknesses [5–8]: First, validity and currency of WBLs is questionable, because the typical lifetime of phishing websites is extremely short (e.g., just over 72 h according to Ref. [9]) because attacks use hit-and-run strategy to avoid detection. Second, it is impossible to discriminate between legitimate and forged websites until phishing attacks are reported and entries created in WBLs. Third, WBL and WWL entries may include errors, and possibilities of false negatives for WBLs and false positives for WWLs must be addressed. In other words, a phishing site may go undetected, and a legitimate site may be accused of being a malicious site. Furthermore, there are too many WWLs entries to include for the list to become complete. In addition, WBLs and WWLs must be kept current at all times to maintain validity. Finally, defense mechanisms based on "known" information is useless to offer protection against attacks that use unknown and/or bypass strategies.

In this paper, we extend our past proposals [5,6] and present a quantitative approach for evaluating the phishing possibility of a given website using the refined security risk elements for domain and web page. Design and implementation of the website risk assessment system (WRAS) for antiphishing is also included. WRAS uses a combination of the WWLs and the self-learning phishing filtering techniques to achieve high accuracy and wide coverage. It can detect suspicious websites containing phishing attack and abnormal behavior and generates a warning if website is judged untrustworthy.

The remainder of the paper is organized as follows. We briefly introduce the background and related work in Section 2. In Section 3, security risk elements to evaluate the security risk of the website are described. Section 4 shows the processes of approach to estimate the security risk of a website. In Section 5, we describe our solution, WRAS,, and provide details about its implementation and experimental result. Finally, we conclude the paper in Section 6.

2. RELATED WORK

Many techniques have been proposed to defeat phishing attacks. Examples include filtering or verifying phishing E-mail, evaluating visual similarity to distinguish legitimate websites from phishing websites, and antiphishing toolbars. They can be generally classified as either server-side or client-side approaches as shown below.

Server-side approaches require that server authentication is required to defend against phishing attacks. Ease of generating fake E-mails is one of the root causes of phishing attacks. E-mail filters are generally quite effective, and Microsoft's Sender ID Framework [10] or Yahoo's DomainKey [11] are such examples. Although the industry is developing a standard, domainkeys identified mail [12], users must be aware of the phishing threat and check for signs of a site being spoofed [13]. Ma *et al*. [14] presented an approach to detect phishing E-mails using hybrid features (e.g., content, orthographic, and derived features) and a feature selection method.

Abu-Nimeh *et al*. [15] presented distributed client–server architecture to detect phishing attacks in mobile environment. At the server side, Bayesian additive regression tree is applied to classify the majority of the E-mails. At the client side, lighter machine learning approaches, which can improve their predictive accuracy and eliminate the overhead of variable selection, are used to classify phishing E-mails.

Another authentication approach is to share a secret, such as a password and an image, between the server and client. Dhamija and Tygar [16,17] proposed dynamic security skins, which allow that users visually verify whether the image from the server matches its corresponding local images. Fu *et al*. [18,19] proposed a visual similarity assessment-based antiphishing strategy, which uses visual characteristics to identify potential phishing websites and measure a suspicious web pages' similarity to actual sites registered with the system. Lam *et al*. [20] and Chen *et al*. [21] proposed a phishing page detection mechanism based on layout similarity analysis. They analyzed the layout of web pages rather than the HTML codes, colors, or content and then computed the similarity degree of the suspicious and authentic web page through image-processing techniques.

Zhang *et al*. [22] proposed a content-based phishing detection method, CANTINA. It extracts keywords from a suspicious web page, inputs them into a search engine to identify the original legitimate website, and compares the suspicious website with the original website. Wenyin *et al*. [7] proposed a method for identifying a phishing web page and discovering its phishing target by calculating and reasoning defined association relations (e.g., link relation, search relation, and text relation) on its semantic link network. If the web page is identified as phishing, its phishing target can be discovered based on the proposed strategies. Aburrous *et al*. [23] proposed a system for detecting e-banking phishing websites. The proposed system is based on fuzzy logic and data mining algorithms to assess e-banking phishing website risk on the characteristics of phishing websites. Finally, Pamunuwa *et al*. [24] used an intrusion detection system for identifying phishing E-mail and evaluating the identified websites. Upon the detection of suspicious E-mails, web crawlers visit the websites indicated by the E-mail and recursively follow all links from the potentially phishing sites.

Client-side approaches are typically implemented as toolbars that alert different types of security messages to help users detect phishing websites [25,26]. Chou *et al*. [27] proposed a framework for client-side defense using a browser plug-in called SpoofGuard, which examined web pages and warned the user when a request for data might be part of a spoof attack. It uses domain names, invalid links, URL obfuscation, and images to measure the similarity between a given page and the pages in the caches.

Wu *et al*. [28] presented the Web Wallet, which prevents phishing attacks by forcing users to compare, and then confirmed it before going to a website instead of just confirming. Cook *et al*. [29,30] proposed an antiphishing filter, phishwish, which identified phishing E-mails using 11 rules to determine the veracity of an incoming E-mail. It does not depend on centralized white or blacklists, nor does it need to be trained. Crain *et al*. [31] proposed an E-mail verification system, *Trust Email*, to distinguish legitimate E-mail from spam and phishing attempts by combining automatic and transparent E-mail signing with an E-mail client plug-in.

Many toolbars, such as NetCraft [32], EarthLink [33], TrustBar [34], AntiPhish [35], and MS SmartScreen Filter [36], are designed to detect and prevent phishing attacks. They generally warn users if they visit a suspected phishing website. Most of them use WBL and WWL, which depend on phishing reports. As long as a phishing website has not been reported, phishers may steal personal data from the visitors to the website.

Zhang *et al*. [37] developed an automated test bed for testing antiphishing tools. They analyzed 10 antiphishing toolbars and report that only one tool (i.e., SpoofGuard) was able to consistently identify more than 90% of phishing URLs correctly; however, it had a very high false-positive rate (e.g., 42%). A more comprehensive survey of antiphishing solutions can be found in Refs [38,39].

3. SECURITY RISK ELEMENTS FOR THE SECURITY RISK OF WEBSITE

The security risk element, in this paper, denotes a condition or a situation that can cause security attacks such as phishing and pharming. It is used to calculate the degree of the website's security risk. As depicted in Figure 1, the security elements are proposed by dividing them into two categories: domain-related and web page-related security risk elements.

3.1. Domain-related security risk elements

Many kinds of elements are related to a website or a web page. In this paper, we propose the refined six security risk elements that are especially applicable elements related to an occurrence of the phishing attacks. These elements also are implemented as WWL database (DB) in WRAS.

Server-Name This element is used to judge whether the domain name of a website coincides with the registered IP address. Generally, the domain name in URL or resolved from the IP address of the phishing website does not match its claimed identity. Furthermore, most pharming attacks misdirect users to the phishing website through DNS hijacking using crimeware, such as a Trojan. Therefore, pharming attacks may be detected by checking this element.

Domain-Country This element checks if the domain country information requested by a user coincides with the IP address assigned by an organization of the real country to detect suspicious websites considered as phishing routes or phishing websites.

Domain-Life This element denotes a period from the registration date to the expiration date of a domain. This element considers the characteristic of a phishing website, which has a short lifespan. That is, it is considered that websites with a longer Domain-Life span are safer. This can be calculated by performing a WHOIS query on the domain name or IP address in the link. A WHOIS query provides much information such as the domain name, registrant, registrant's address, domain's creation, last update, expiration dates, and so on. The Domain-Life can be calculated using formula (1).

$$\textit{Domain-Life} = \textit{Expiration Date} - \textit{Registered Date} \quad (1)$$

Domain-Age This element denotes a period from the registration date of a website to the current time. This element complements the false-positive feature, which is the characteristic of the Domain-Life. Like the *Domain-Life* element, this can be calculated by performing a WHOIS query. The Domain-Age can be calculated using formula (2)

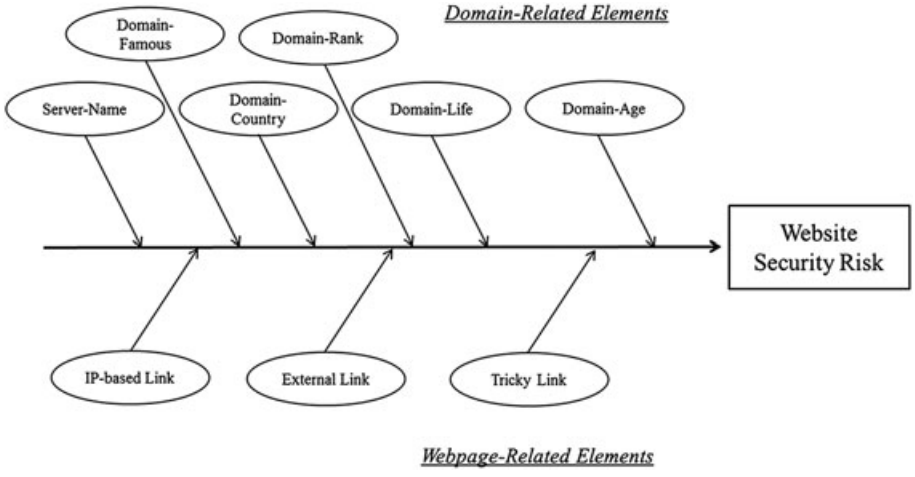

Figure 1. Security risk elements for evaluating the security risk of website.

$$Domain\text{-}Age = Current\ Date - Registered\ Date \quad (2)$$

Domain-Famous This element denotes the eminence of the website that is the number of search results related to the website, evaluated by a search engine, such as Google or Yahoo. If the search result of the website is high, many people use the website information, and it is secure. Conversely, if the search result is low, it is possible that the website is newly created or a phishing website.

DNS-Rank This element means that the DNS query ranking information is supported by specific companies, such as Alexa [40] or Google Page Rank Update [41]. This element also considers the characteristic of the phishing website that has extremely low age rank score with little user contact to this website.

3.2. Web page-related security risk elements

Although a domain is safe, serious attacks can occur within web pages. For example, although a famous portal website's domain (e.g., google.com, yahoo.com) is safe, phishers can insert forged web pages into the same domain or the subdomains. Therefore, in order to deal with this kind of phishing attacks, web page-related security risk elements for evaluating the security risk based on web page source code are proposed. We analyze the HTML source code of the web page. The web page's HTML tags related with phishing attacks can be classified into three types: direct insertion, page forwarding, and user interface types.

Direct Insertion Type A set of tag that is capable of inputting data from user such as <form> tag. HTML forms are one of the techniques used to gather information from users.

Page Forwarding Type A set of tag that is used to forward web pages containing the form tags such as <a>, <frame>, and <iframe> tags.

User Interface Type A set of tag (e.g., <img> and <link> tags) that is used to decorate a forged web page like a legitimate web page.

Each type may contain the following three kinds of the elements that are used to calculate the security risk of web page.

IP-based Link This element denotes presence of IP-based address in web pages. IP-based links in a web page make it difficult for users to identify exactly where they want to visit the target website or web page. A forged website usually has an IP-based URL because phishers usually use some IP-based zombie system to host phishing websites.

External Link This element denotes presence of external links in web pages. The external link is a hyperlink that points at an external domain other than the domain the link exists on internal domain. A forged website usually contains a number of external links.

Tricky Link This element denotes presence of tricky links in web pages. The tricky link means that users do not identify domain-related information in a link because the link is encoded. URL encoding normally is used when the browser sends from data to a web server. However, this method is also used to hide domain-related information by a phisher.

4. THE PROPOSED PROCESS FOR EVALUATING THE SECURITY RISK OF A WEBSITE

In this section, processes for quantitative analysis of the website's security risk using the security elements are detailed. As depicted in Figure 2, the security risk of a website calculated via three phases: domain security risk evaluation, web page security risk evaluation, and website security risk evaluation. A more detailed description of each process will be presented in the following.

4.1. Process 1: Select security risk elements

Many kinds of security risk elements are related to a domain or a web page. In this paper, we propose six security elements (e.g., Server-Name, Domain-Country, Domain-Life, Domain-Age, Domain-Famous, and Domain Rank) for the domain and three security risk elements (e.g., IP-based Link, External Link, and Tricky Link) for the web page that are especially applicable elements related to an occurrence of the phishing attacks. As mentioned previously, these elements are implemented as WWL DB.

4.2. Process 2: Weight between security risk elements

In this process, weight is given to the security risk elements defined in the previous process according to their impact on the security risk of the website. In a relative security risk

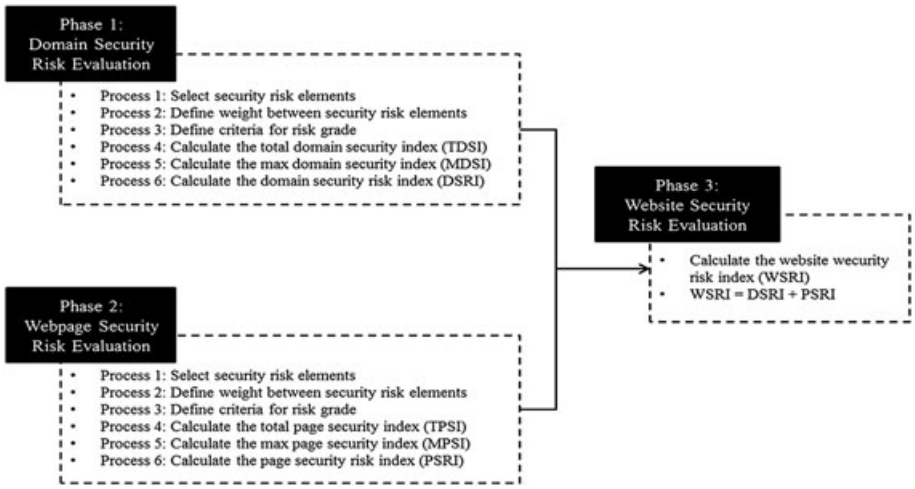

Figure 2. Proposed processes.

evaluation (or an absolute security risk evaluation), for example, the weight value ranges from 1 to 3. A higher weight value represents a more important element, which can increase security risk of a website. In this paper, the security risk elements are assigned the weight value as depicted in Tables I and II.

4.3. Process 3: Define criteria for risk grade

In Process 3, the risk grade for the each security risk element is defined. At this point, the risk grade denotes a possibility of causing the threat occurrence. In this paper, the risk grade can be from 0 to 4; grade "4" is the highest value to cause the risk occurrence. Each risk grade is defined by the statistical information related to the security risk elements, as presented in Table III.

Table I. Weight criteria for domain-related elements.

Weight	Element(s)
1	Domain-Country, DNS-Rank
2	Domain-Life, Domain-Famous
3	Server-Name, Domain-Age

Table II. Weight criteria for web page-related elements.

Type	Element	Weight
Direct insertion	IP-based link	2
	External link	2
	Tricky link	2
Page forwarding	IP-based link	1
	External link	1
	Tricky link	1
User interface	IP-based link	1
	External link	1
	Tricky link	1

4.4. Process 4: Calculate the total security index

Process 4 is a step to calculate the total security index (TSI) (e.g., Total Domain Security Index and Total Page Security Index), which is a score that represents a security risk degree embedded in a website. As depicted in formulas (3) and (4), TSI is calculated as a sum of the security risk index (SRI) that is a multiplication of the risk grade and the weight for each security risk element.

$$TSI = \sum_{i=1}^{n} SRI_i \tag{3}$$

where n is the number of the security risk elements and i is the specific security risk element.

$$SRI = \text{Risk} \quad \text{Grade} \times \text{Weight} \tag{4}$$

4.5. Process 5: Calculate the maximum security index

In Process 5, the maximum security index (MSI) (e.g., Max Domain Security Index and Max Page Security Index), which denotes the maximum security risk degree, is calculated. The MSI is the theoretically maximum risk value of the website, as in formula (5). That is, in this paper, the MSI is the total value of the SRI when the all-risk grades for each security risk element have a maximum risk grade of value 4.

$$MSI = \sum_{i=1}^{n} SRI_i = \sum_{i=1}^{n} \left(Risk\,Grade_{\max} \times Weight\right)_i \tag{5}$$

where n is the number of the security risk elements and i is the specific security risk element.

Table III. Criteria for risk grade of security elements.

	Risk grade				
Element	4	3	2	1	0
Server-Name	No match	–	–	–	Match
Domain-Country	No match	–	–	–	Match
Domain-Life	Under 6 months	Under 1 year	Under 2 years	Under 3 years	Over 3 years
Domain-Age	Under 1 year	Under 2 years	Under 4 years	Under 6 years	Over 6 year
Domain-Famous	Under 1000 pages	Under 5000 pages	Under 10 000 pages	Under 20 000 pages	Over 20 000 pages
DNS-Rank	Under 10%	Under 20%	Under 40%	Under 80%	Over 80%
IP-based Link	–	–	–	Exist	None
External Link	–	–	–	Exist	None
Tricky Link	–	–	–	Exist	None

4.6. Process 6: Calculate the security risk index for domain and web page.

In Process 6, the SRI for the domain (DSRI) and the page (PSRI) are calculated using SRI and MSI calculated in processes 4 and 5, respectively, as formulas (6) and (7).

$$DSRI = \frac{TDSI}{MDSI} \times 100 \tag{6}$$

$$PSRI = \frac{TPSI}{MPSI} \times T \tag{7}$$

where T is the maximum risk value for the web page.

Finally, the website SRI (WSRI), which is the security risk degree representing phishing websites, is calculated using DSRI and PSRI respectively, as formula (8).

$$WSRI = DSRI + PSRI \tag{8}$$

The value of WSRI ranges from 0 to 100. A website with a higher value denotes a more suspicious website as a phishing website.

5. WEBSITE RISK ASSESSMENT SYSTEM

In this section, in order to illustrate the motivation of our research, we present our design and implementation of the WRAS for antiphishing.

5.1. Overview and requirements of website risk assessment system

One of the popular methods is using add-in toolbars for the browser. The proposed WRAS also is a system integrated into the Internet Explorer (IE) web browser. It checks and evaluates the security risk of a website domain and its web pages before a user visits a website. That is, WRAS verifies the website using WWLs and real-time analysis of web pages. A website qualified list (WQL), which contains candidate websites for the WWLs, is designed in this paper to complement WWLs. Each candidate website has a score dynamically calculated by submissions from users. Although the existing WWL-based solutions only maintain domain-specific information, such as IP address and URL, our approach uses domain-specific information defined in Section 3.1 and web page-specific scores to reduce false alarms. When the security risk score of a candidate website is below the threshold, the website information is moved from the WQLs to the WWLs and vice versa.

The WRAS proposed in this paper follows typical client–server architecture. The system should satisfy the following requirements to efficiently estimate the security risk of a website between the client and server.

Client-Side Requirements. First, the client-side service in a system should obtain all information about a website the user wishes to navigate. This information includes web page URL, DNS-related information (e.g., Domain-Country, Domain-Life, Domain-Age, and DNS-Ranking), and web page analysis data that will be used to evaluate vulnerabilities. Second, the system should provide a user-friendly feedback system to efficiently maintain WWLs and WQLs in the client. Third, antiphishing solutions must consider both security and usability constraints [8]. Fourth, the alert should always appear at the right time with the right warning message [39]. Finally, the client service should not affect the performance of the client system.

Server-Side Requirements. The server-side service in a system has the same problem as the existing WBL-based solutions. The main problem is the server overload caused by the heavy transaction of the database in the server side. That is, the heavy transaction of WWL and WQL can compromise system performance. Furthermore, high security of the server-side system is required.

5.2. Architecture

The core idea of our approach WRAS is to evaluate a website at the client and server side. Our approach combines both a domain analysis as well as a web page content analysis using the information provided by WWL and WQL servers.

Website risk assessment system uses typical client–server architecture, as illustrated in Figure 3. A mutual

feedback mechanism is deployed between the server and the client. The client side system, called WRAS client, is implemented as a plug-in to the IE web browser. When a user tries to visit a website, the WRAS client makes the first decision whether the website is a phishing website by checking a WWL cache in the client or a WWL DB in the server. If the website is not in the lists, the WRAS client calculates the security risk of the website using WQL cache in the client or WQL DB in the server.

The main goal of the WRAS server is to maintain and update WWLs and WQLs by analyzing the security risk of the websites. That is, the server computes the SRI for the domain based on the method proposed in Section 3.1. If DSRI is below a threshold, the server registers the website in the WWLs, and it transfers the website information to the WRAS client when the WRAS client requests website information. The following subsections present a more detailed description of WRAS client and server.

5.3. Website risk assessment system client

As mentioned earlier, the WRAS client is implemented as a plug-in in the form of a toolbar in IE, because a majority of Internet users use this web browser, and the IE interface is an intimate environment for the user. The WRAS client can analyze the request web page of users by developing a browser helper module (BHO) on Microsoft windows. BHO is the most popular technique to integrate a plug-in component into IE. The developer has access to the event mechanism of IE using BHOs and can create user interface elements, such as toolbars [13]. Figure 4 shows the WRAS client module as a plug-in to the IE.

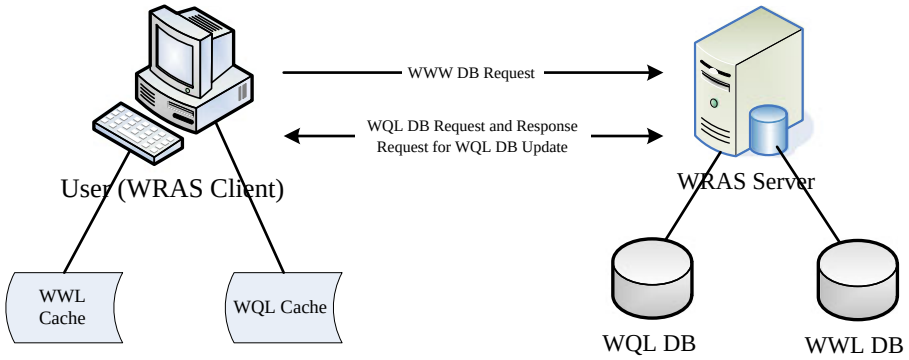

Figure 3. Overview of the website risk assessment system architecture.

Figure 4. Website risk assessment system client module as a plug-in to the Internet Explorer.

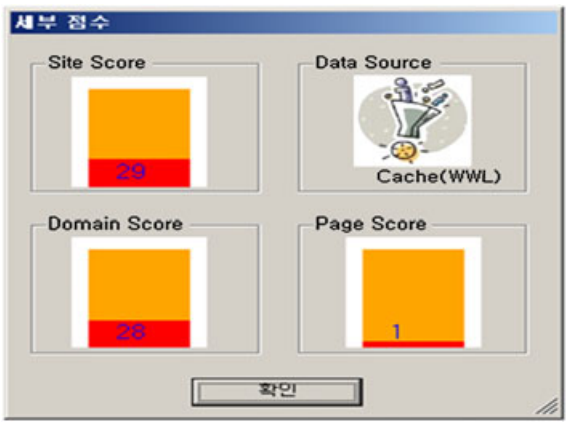

Figure 5. Example of website security risk index detailed information.

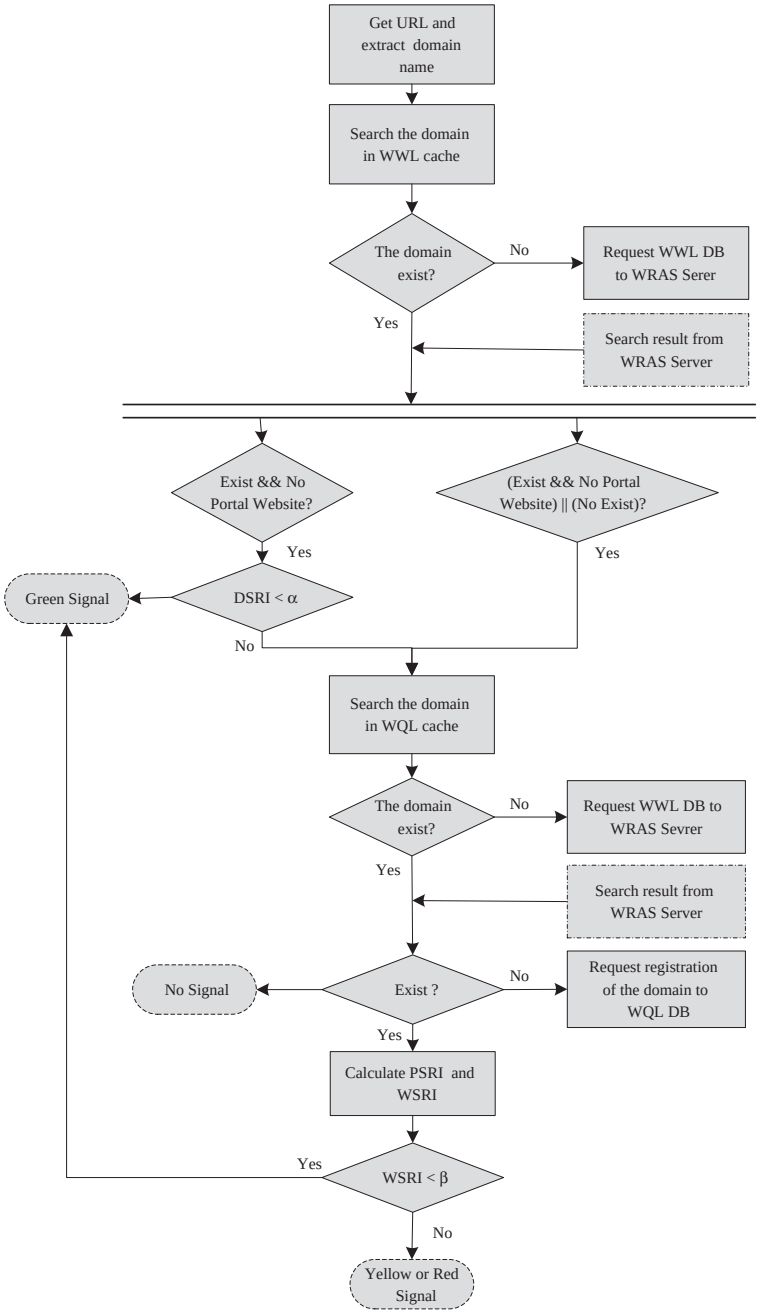

Figure 6. Operational flow chart of the website risk assessment system client.

As depicted in Figure 4, the WRAS client module is represented with signal lights (e.g., green, yellow, and red light). The signal light is activated by the web page's security index. The signal light reports the SRI of the target website calculated by the proposed algorithm depicted in Figure 6. Furthermore, the WRAS client supports more detailed security index information, as illustrated in Figure 5.

The WRAS client, as well as the server, maintains WWL and WQL information (e.g., WSRI, Server-Name, Domain-Country, Domain-Life, Domain-Age, Domain-Famous, and DNS Ranking defined in Section 3). Whenever a user tries to navigate to a website, the WRAS client checks the domain name (or IP address) in a WWL cache. If the domain name is not in the WWLs, the client proceeds to check the safety of the website. Furthermore, the WRAS client can provide the value of the website security risk directly to the user. Besides, the performance is more effective than other approaches, such as an application or java applet, because it is a component in the web browser.

Figure 6 depicts the operational flow chart of WRAS client:(i) If a user inputs a target URL in the web browser,

(a) WWL request format (b) WWL reply format (c) WWL reply result

Figure 7. XML formats of WWL data.

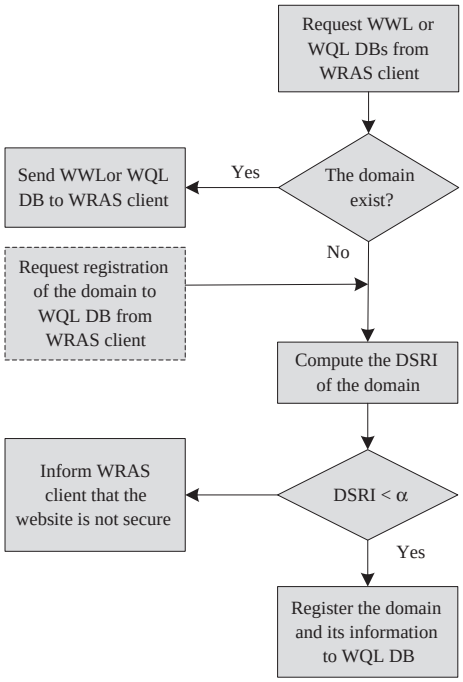

Figure 8. Operational flow chart of the website risk assessment system server.

WRAS client gets the URL and extracts the domain name. (ii) The WRAS client first looks for the requested domain in the WWL cache. The WRAS client feedbacks the SRI for the target website to the user, if the target URL is in the WWL cache. (iii) If the requested URL is not in the WWL cache, the WRAS client requests the target website information from the WARS server. (iv) If the target URL is in the WWL DB at the WRAS server, the WRAS server sends the target website information; then, the WRAS client analyzes the information. If the target URL is in the WWL cache or WWL DB, and the target website is not a portal website, the WRAS client permits a user to visit the target website (e.g., green light in the toolbar). (v) In the previous step, if the requested domain is not in the WWL DB at the WRAS server, the WRAS client looks for the request URL in the WQL cache. Like the previous steps, the WRAS client calculates the SRI based on the information in the WQL cache if the target URL is in the WQL cache. (vi) If the request URL is not in the WWL cache, the WRAS client requests WQL DB from the WRAS server. (vii) If the target URL is in the WQL cache or WQL DB, the WRAS client calculates a PSRI and a WSRI. It saves it to the WQL cache and updates WQL DB in the server. (viii) Finally, if WSRI is under the threshold β, the WRAS client permits a user to visit the target website (e.g., green light in the toolbar).

5.4. Website risk assessment system server

The main functionality of the WRAS server is to maintain WWL and WQL DB by frequently analyzing the security risk of websites. The WRAS server supports the domain-specific information for the target website by an XML Web service whenever the client requests the target information. Furthermore, it helps a website manager control (e.g., register, edit, and delete) WWL and WQL DB.

In this paper, the XML Web service with Oracle DB is used to support effective processing and protection of communication data from external access. A popular relational database, such as Oracle or MS SQL, guarantees the system stability and security, because they have provided the service to users for a long time. Furthermore, XML Web service guarantees compatibility between other products, because it operates in existing web environments. Figure 7 shows XML formats of WWL data SOAP (Simple Object Access Protocol) and its example result implemented in this research.

Figure 8 depicts the operational flow chart of WRAS server:(1) If WRAS client requests WWL DB or WQL DB corresponded with the target domain name, WRAS server searches the domain name in WWL DB and WQL DB. (2) If the domain name is in the WWL DB or WQL DB, the WRAS server feedbacks the SRI for the target website to the WRAS client. (3) If the requested domain name is not in the WWL DB and WQL DB, or the WRAS client requests registration of the target domain to WQL DB, the WRAS server computes the DSRI of the target domain. (4) If DSRI is under the threshold α, the WRAS server registers the target domain and its information to WQL

DB. Finally, DSRI is not under the threshold, and the WRAS server informs WRAS client that the website is not secure.

Figure 9 depicts WRAS server manager.

5.5. Experimental result

We conducted an experiment to show the effectiveness of the detection and to show the potential of the work. We have used nine website URLs including two phishing websites collected from real phishing attack cases. We empirically set the parameters weight and risk grade for security risk elements as described in Section 4 and $T = 12$, $\alpha = 30$, and $\beta = 60$ in these experiments. Figure 10 gives experimental results obtained from our WRAS.

As depicted in Figure 10, famous websites, such as Wikipedia, Daum, and Weibo, are recognized as safe websites. However, other websites (e.g., chemchallenger.com, tepedelik.com) are estimated to be suspicious websites, such as phishing websites.

6. CONCLUSION

Phishing attacks have become a severe problem of Internet security during the past several years. In this paper, we

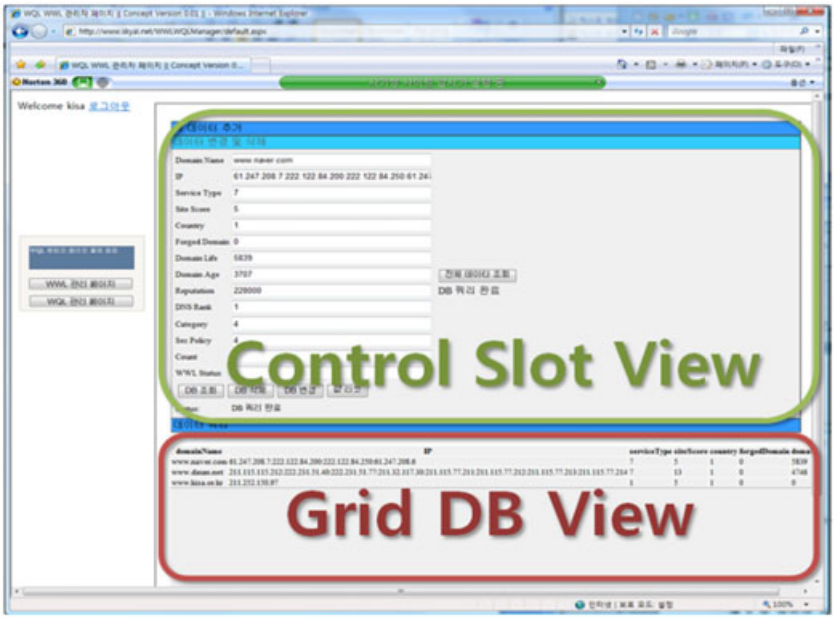

Figure 9. Website risk assessment system server manager.

Website	Server-Name	Domain-Country	Domain-Life	Domain-Age	Domain-Famous	DNS-Rank	DSRI	PSRI	WSRI
Weight	3	1	2	3	2	1	MDSI=48	MPSI=12	
daum.net	0	0	2	0	0	0	8	3	11
			541 days	5,665 days	31,800,000	304			
cyworld.com	0	0	4	0	0	0	17	4	21
			57days	4,323 days	2,430,000	1473			
wikipedia.com	0	0	0	0	0	0	0	1	1
			1,218 days	3,892 days	57,700,000	6			
weibo.com	0	0	0	0	0	0	0	4	4
			3,475 days	4,555 days	13,600,000	29			
google.co.in	0	0	3	0	1	0	17	0	17
			283 days	3,002 days	72,300,000	14			
craigslist.org	0	0	3	0	1	0	17	4	21
			725 days	5,114 days	3,060,000	35			
korea.ac.kr	0	0	0	0	0	0	0	2	2
			1,855 days	6,613 days	1,400,000	70,692			
chemchallenger.com	4	1	3	3	4	4	83	6	89
			493 days	602 days	349	N/A			
tepedelik.com	4	4	4	1	4	4	81	4	85
			177 days	1648 days	1,100	N/A			

Figure 10. Experimental result (conducted in 30 August 2011).

Security Comm. Networks 2012; **5**:1181–1192 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

proposed a novel method for estimating the security risk of a website, including the definition of security risk elements and the processes to evaluate the security risk of the website. The major strength of the proposed approach is its ability to estimate the security risk of websites quantitatively based on the WWL database. This property is particularly useful for browser plug-ins or applications against phishing attacks. Furthermore, we presented a novel WRAS for antiphishing. WRAS alerts users when they are about to contact with the target website. The system protects a user against phishing and pharming attacks. That is, the WRAS system supports a more precise SRI for the target website using both WWLs and self-learning phishing filtering techniques. Its evaluation algorithm is independent of how phishing attacks are implemented. Thereby, it can easily detect sophisticated phishing websites that other techniques find hard to deal with. In addition, inexperienced users can prevent phishing websites from transferring sensitive information.

However, in order to obtain a more precise result, more correct and trusted information about the security risk elements should be required. Furthermore, the algorithm to compute the SRI in the client and the heavy transaction of the database in the server are critical in influencing the effectiveness and efficiency of the system. Although our proposed system protects users from phishing sites, the detection algorithm and its efficiency should be improved.

ACKNOWLEDGEMENT

This work was partially supported by the Defense Acquisition Program Administration and Agency for Defense Development under the contract.

REFERENCES

1. Anti-Phishing Working Group (APWG). http://www.antiphishing.org
2. Dhinakaran C, Lee JK, Nagamalai D. Reminder: please update your details: phishing trends. In *Proceedings of 2009 First International Conference on Networks and Communications*. 2009; 295–300.
3. APWG. Phishing Activity Trends Repot, 1st Half, 2009. http://www.antiphishing.org
4. Aaron G. The state of phishing. *Computer Fraud & Security* 2010; **2010**(6): 5–8.
5. Kim Y-G, Cho S, Lee J-S, Lee M-S, Kim IH, Kim SH. Method for evaluating the security risk of a website against phishing attacks. *Lecture Notes in Computer Science* 2008; **5075**: 21–31.
6. Kim Y-G, Cha S. Website risk assessment system for anti-phishing. *Communications in Computer and Information Science* 2011; **185**: 131–138.
7. Wenyin L, Fang N, Quan X, Qiu B, Liu G. Discovering phishing target based on semantic link network. *Future Generation Computer Systems* 2010; **26**: 381–388.
8. FlorCncio D, Herley C. Analysis and improvement of anti-phishing schemes. In *Proceedings of IFIP International Federation for Information Processing*, Vol. **201**. Security and Privacy in Dynamic Environments, 2006.
9. APWG. Phishing activity trends report for the month of Jan. 2008.
10. Microsoft. Sender ID framework overview. http://www.microsoft.com/mscorp/safety/technologies/senderid/defaulat.mspx [2011]
11. Yahoo. Yahoo! anti-spam resource center. http://antispam.yahoo.com [2011]
12. Mutual Internet Practices Association. DomainKeys Identified Mail, DKIM. http://www.dkim.org [2011]
13. Ludl C, McAllister S, Kirda E, Kruegel C. On the effectiveness of techniques to detect phishing sites. *Lecture Notes in Computer Science* 2007; **4579**: 20–39.
14. Ma L, Ofogh B, Watters P, Brown S. Detecting phishing emails using hybrid features. In *Proceedings of Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*. 2009; 493–497.
15. Abu-Nimeh S, Nappa D, Wang X, Nair S. Distributed phishing detection by applying variable selection using Bayesian additive regression trees. In *Proceedings of IEEE International Conference on Communications (ICC)*. 2009; 1–5.
16. Dhamija R, Tygar JD. The battle against phishing: dynamic security skins. In *Proceedings of the 2005 Symposium on Usable Privacy and Security*. 2005; 77–88.
17. Dhamija R, Tygar JD. Phish and hips: human interactive proofs to detect phishing attacks. In *Proceedings of the Second International Workshop*. 2005; 127–141.
18. Fu AY, Wenyin L, Deng X. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). *IEEE Transactions on Dependable and Secure Computing* 2006; **3**(4): 301–311.
19. Liu W, Deng X, Huang G, Fu AY. An antiphishing strategy based on visual similarity assessment, *IEEE Internet Computing* 2006; **10**(2): 58–65.
20. Lam I-F, Xiao W-C, Wang S-C, Chen K-T. Counteracting phishing page polymorphism: an image layout analysis approach. *Lecture Notes in Computer Science* 2009; **5576**: 270–279.
21. Chen K-T, Chen J-Y, Hauang C-R, Chen C-S. Fighting phishing with discriminative keypoint features of webpages. *IEEE Internet Computing* 2009; **13**(3): 56–63.
22. Zhang Y, Hong J, Cranor L. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, 2007; 639–648.

23. Aburrous M, Hossain MA, Dahal K, Thabtah F. Intelligent phishing detection system for e-banking using fuzzy data mining. *Expert Systems with Applications* 2010; **37**: 7913–7921.
24. Pamunuwa H, Wijesekera D, Farkas C. An intrusion detection system for detecting phishing attacks. *Lecture Notes in Computer Science* 2007; **4721**: 181–192.
25. Raffetseder T, Kirda E, Kruegel C. Building anti-phishing browser plug-ins: an experience report. In Proceedings of the Third International Workshop on Software Engineering for Secure Systems, 2007.
26. Huang H, Zhong S, Tan J. Browser-side countermeasures for deceptive phishing attack. In *Proceedings of the Fifth International Conference on Information Assurance and Security (IAS)*, 2009; 352–355.
27. Chou N, Ledesma R, Teraguchi Y, Boneh D, Mitchell JC. Client-side defense against web-based identity theft. In Proceedings of the 11th Annual Network and Distributed System Security Symposium, 2004.
28. Wu M, Miller RC, Little G. Web wallet: preventing phishing attacks by revealing user intentions. In *Proceedings of Symposium On Usable Privacy and Security*. 2006; 102–113.
29. Cook DL, Gurbani VK, Danluk M. Phishwish: a stateless phishing filter using minimal rules. *Lecture Notes in Computer Science* 2008; **5143**: 182–186.
30. Cook DL, Gurbani VK, Danluk M. Phishwish: a simple and stateless phishing filter. *Security and Networks* 2009; **2**(1):29–43.
31. Crain J, Opyrchal L, Prakash A. Fighting phishing with trusted email. In *Proceedings of 2010 International Conference on Availability, Reliability and Security (ARES 2010)*, 2010; 462–467.
32. NetCraft. http://www.netcraft.com [2011]
33. EarthLink. http://www.earthlink.com [2011]
34. Herzberg A, Gbara A. Security and identification indicators for browsers against spoofing and phishing attacks. *ACM Transactions on Internet Technology* 2008; **8**(4):1–36.
35. Kirda E, Kruegel C. Protecting users against phishing attacks with AntiPhish. In *Proceedings of the 29th IEEE International Computer Software and Applications Conference (COMPSAC)*, 2005; 1–8.
36. Microsoft. SmartScreen Filter. http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter [2011]
37. Zhang Y, Egelman S, Cranor L, Hong J. Phinding phish: evaluating anti-phishing tools. In Proceedings of the 14th Annual Networks and Distributed System Security Symposium (NDSS), 2007.
38. Emigh A. Online identity theft: phishing technology, chokepoints and countermeasures. *ITTC Report on Online*, 2005.
39. Dong X, Clark JA, Jacob JL. Defending the weakest link: phishing websites detection by analyzing user behaviours. *Telecommunication System* 2010; **45**: 215–226.
40. Alexa the Web Information Company. http://www.alexa.com [2011]
41. Google. Google Page Rank Update. http://google pagerankupdate.com [2011]