Website Risk Assessment System for Anti-Phishing

Young-Gab Kim and Sungdeok Cha

College of Information and Communication, Korea University,
1, 5-ga, Anam-dong, SungBuk-gu, 136-701, Seoul, Korea
{always,scha}@korea.ac.kr

Abstract. Phishing attacks steal a user's identity data and financial account credentials using social engineering and technical spoofing techniques. Many counter measures have been developed to protect user's sensitive information from phishing attacks. Although most approaches use both website black lists (WBLs) and website white lists (WWLs), these approaches have several weakneksses. This paper presents a novel anti-phishing Website Risk Assessment System (WRAS). WRAS computes a security risk index of website and generates warnings as to the website trustworthiness. Therefore, it can protect inexperienced users against spoofed website-based phishing attacks and exploit-based phishing attempts that may occur from legitimate web pages.

Keywords: Phishing, Pharming, Anti-Phishing, Website Risk Assessment.

1 Introduction

Phishing is a way of attempting to steal user identity data and financial account credentials using social engineering and technical spoofing techniques in an electronic communication. Pharming (e.g., hijacking brand names of banks) is a more advanced phishing attack in that it redirect a website's traffic to another, forged website or proxy servers typically through DNS hijacking, often using crimeware, such as Trojan keylogger spyware [1]. Phishing and pharming attacks frequently occur and diversify, as Internet technologies (e.g., e-commerce, Internet banking, and social networking) evolve. Therefore, preventing a phishing attacker from stealing user's sensitive information (e.g., social security number, credit card number, account username, and password) is a major challenge in Internet security. Much research has been undertaken and many solutions have been proposed to protect user's economic loss and privacy against these phishing and pharming attacks. Most use website blacklists (WBLs) (list of e-mail addresses or IP addresses that originate from known true website) and website white lists (WWLs) (list of known safe website). Although many Internet Service Providers (ISPs) and enterprise security solution vendors use both WBLs and WWLs, these approaches have several weaknesses [2]: First, validity of WBLs is low, because the life cycle of phishing websites is short. Second, it is impossible to discriminate between legitimate and forged websites until the phishing attacks are detected and recorded in WBLs. Furthermore, the WBL- and WWL-based approaches hardly counter the new generation of sophisticated malware phishing attacks, pharming attacks, designed to target certain services. In addition, many of the

J.J. Park, L.T. Yang, and C. Lee (Eds.): FutureTech 2011, Part II, CCIS 185, pp. 131–138, 2011.

existing WWL-based approach use only a website URL or IP address to distinguish between legitimate and forged websites. Therefore, the previous work [2] proposed a novel approach, which can quantitatively estimate the security risk of websites based on WWLs to overcome the limitation of WBL and WWL approaches. The proposed method in previous work defined the security risk elements and steps to quantitatively calculate the security risk index of a website.

In this paper, we present a novel Website Risk Assessment System (WRAS) for anti-phishing. WRAS employs a combination of the WWLs and the self-learning phishing filtering techniques that deliver high accuracy and wide coverage of websites. This can detect suspicious websites containing phishing attack and abnormal behavior. It generates a warning if the website is considered untrustworthy. Finally, it can protect inexperienced users against spoofed website-based phishing attacks and exploit-based phishing attempts that may occur from legitimate web pages.

The remainder of the paper is organized as follows. We briefly introduce background and related work in Section 2. We describe our solution, Website Risk Assessment System (WRAS), and provide details about its implementation in Section 3. Finally, we conclude the paper in Section 4.

2 Background and Related Works

Anti-Phishing Working Group (APWG) [1] is the global industry working group to eliminate fraud and identity theft that result from phishing, pharming and email spoofing. APWG is composed of many organizations and security companies. It provides diverse information, such as phishing reports, research data, and resources related to phishing, pharming, and crimeware.

Several research efforts have been made to prevent phishing attacks. In this section, we briefly review some typical approaches, dividing them into two parts: Server-side approach and Client-side approach.

Server-Side Approach. In the server-side approach, server authentication is required to defend against phishing attacks. One of the main reasons why phishing attacks are possible is because e-mails can be spoofed easily. Although spam filters work quite well today, they cannot guarantee that all phishing e-mails are intercepted. As example solutions, which authenticate the sender's e-mail, and prevent the phisher using a hijacked mail address, Microsoft presents the Sender ID Framework [3], and Yahoo uses its own technique called DomainKey [4]. Currently, Yahoo and other industry leaders are in the process of standardizing a technique called DKIM (DomainKeys Identified Mail) [5]. Another authentication approach is to share a secret, such as a password and an image, between server and client. Dhamija et al [6, 7] proposes Dynamic Security Skins, which allows that users visually verify whether the image from the server matches its corresponding local images. Finally, Fu et al. [8, 9] proposes a visual similarity assessment-based antiphising strategy, which uses visual characteristics to identify potential phishing websites and measure a suspicious web pages' similarity to actual sites registered with the system.

Client-Side Approach. In the client-side approach, most solutions are supported as a toolbar, which show different types of security messages to help users detect phishing websites, built-in the web browser [10]. Chou et al. [11] proposes a framework for client-side defense using a browser plug-in called SpoofGuard that examines web pages and warns the user when a request for data may 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. Many toolbars, such as Netcraft [12], EarthLink [13], and MS Phishing Filter [14], are designed to detect and prevent phishing attacks. Most of them use WBL and WWL, which depends on phishing reports. As long as a phishing website has not been reported, phishers may steal personal data from visitors to the website. Wu et al. [15] presents Web Wallet, which prevents phishing attacks by forcing users to compare, then confirm before going to a website instead of just confirming. A more comprehensive survey of anti-phishing solutions can be found in [16].

3 Website Risk Assessment System (WRAS)

3.1 Overview and Requirements of WRAS

WRAS 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 websites using WWLs and real-time analysis of web pages. A Website Qualified Lists (WQLs), 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. While the existing WWL-based solutions only maintain domain-specific information, such as IP address and URL, our approach uses domain-specific information defined in [2] and webpage-specific scores to reduce false alarms. When the webpage score of a candidate website is below the threshold, the website information is moved from the WQLs to the WWLs.

WRAS proposed in this paper follows typical client-server architecture. The system should satisfy the following requirements to efficiently estimate the security risk of website between 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 webpage URL, DNS related information (e.g., Domain-Country, Domain Life, Domain-Age, DNS-Ranking, etc), and webpage 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. Furthermore, 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 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.

3.2 Architecture

WRAS employs typical client-server architecture, as illustrated in Fig. 1. 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 security risk index for the websites based on the method proposed in previous work [2]. If a website security risk index (WSRI) 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.

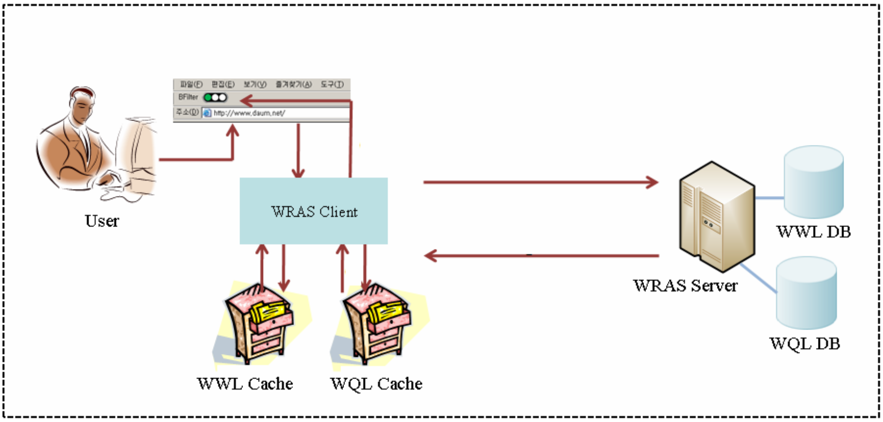

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T)
BFilter
주소(D) http://www.daum.net/
User
WRAS Client
WWL Cache
WQL Cache
WRAS Server
WWL DB
WQL DB

Fig. 1. Overview of WRAS architecture

Fig. 2 depicts the operational flow chart of WRAS: (1) A user inputs a target URL in the web browser. (2) WRAS client checks if the target website's webpage contains an inputtable page. If there is no inputtable page, the WRAS client permits a user to visit the target website. This is based on the premise that the attacker can only steal sensitive information in the inputtable webpage after the user performs a submit. (3) If the target website contains an inputtable webpage, the WRAS client first looks for the requested URL in the WWL cache. The WRAS client feedbacks the security risk index for the target website to the user, if the target URL is located in the WWL cache. (4) If the requested URL is not in the WWL cache, the WRAS client requests the target website information from the WARS server. (5) If the target URL is located 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).

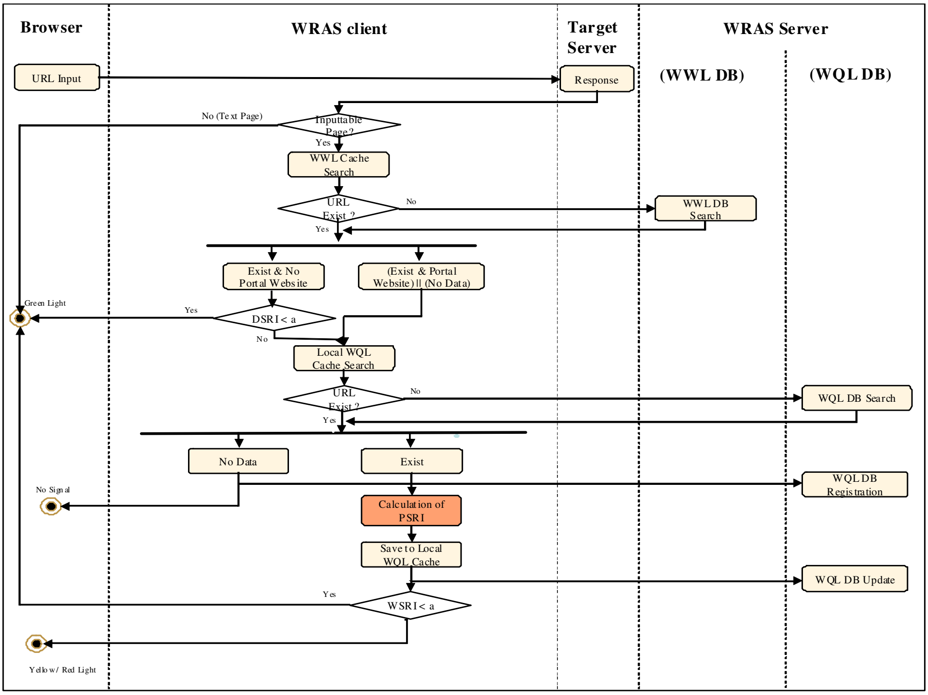


Fig. 2. WRAS Operational flowchart

(6) In the previous step, if the requested URL 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 security risk index based on the information in the WQL cache if the target URL is located in the WQL cache. (7) If the request URL is not in the WWL cache, the WRAS client requests WQL DB from the WRAS server. If the target URL is located in the WQL cache or WQL DB, the WRAS client calculates a Page Security Risk Index (PSRI). It saves it to the WQL cache and updates WQL DB in the server. (8) Finally, if WSRI (e.g., WSRI is the value of DSRI+PSRI) is under the threshold, the WRAS client permits a user to visit the target website (e.g., green light in the toolbar).

3.3 WRAS Client

The WRAS client is implemented as a plug-in in the form of a toolbar in Internet Explorer (IE), since a majority of Internet users use this web browser, and the IE interface is an intimate environment for the user. Furthermore, the WRAS client can analyze the request webpage 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 [10]. Fig.3 shows the WRAS client module plugged-in to IE.

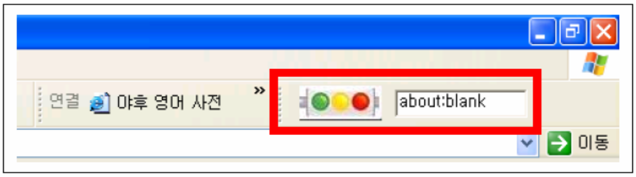


Fig. 3. WRAS client module plugged-in to IE

As depicted in Fig. 3, the WRAS client module is represented with signal lights (e.g., green, yellow, and red light). The signal light is activated by the webpage's security index. That is, the signal light reports the security risk index of the target website calculated by the proposed algorithm depicted in Fig. 2. Furthermore, the WRAS client supports more detailed security index information, as illustrated in Fig. 4.

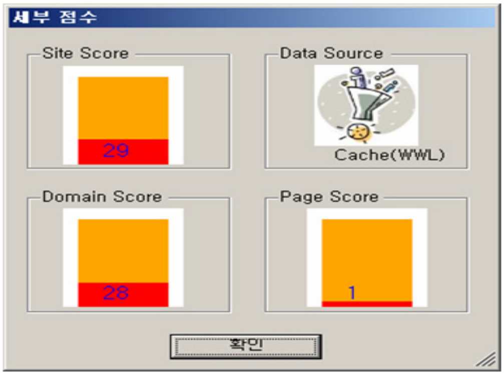


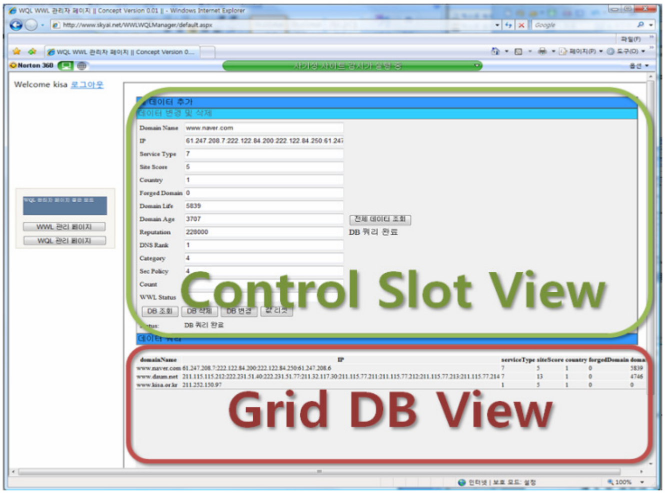
Fig. 4. Example of WSRI detailed information

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 [2]). 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, as explained in Section 3.2.

The WRAS client can provide the value of the website security risk directly to user. Furthermore, the performance is more effective than other approaches, such as an application or java applet, because it is a component in the web browser. In this paper, we inherit some pre-defined classes (e.g., UserControl, IObejectWithSite, IDeskBand, IDockingWindow, OIeWindow, IIputObject) by Microsoft to implement the WRAS client.

3.4 WRAS 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, as illustrated in Fig. 5.



Domain Name www.naver.com
IP 61.247.208.7:222.122.84.200:222.122.84.250:61.247
Service Type 7
Site Score 5
Country 1
Forged Domain 0
Domain Life 5839
Domain Age 3707
Reputation 228000
DNS Rank 1
Category 4
Sec Policy 4
Count
WWL Status

DB 쿼리 완료

Status: DB 쿼리 완료

domainName	IP	serviceType	siteScore	country	forgedDomain	doma
www.naver.com	61.247.208.7:222.122.84.200:222.122.84.250:61.247.208.6	7	5	1	0	5839
www.daum.net	211.115.115.212:222.231.51.40:222.231.51.77:211.32.117.30:211.115.77.211:211.115.77.212:211.115.77.213:211.115.77.214	7	13	1	0	4746
www.kisa.or.kr	211.252.150.97	1	5	1	0	0

Fig. 5. WRAS server manager

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 RDB (Relational Data Base), such as Oracle or MS SQL, guarantee 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.

4 Conclusion and Future Work

We presented a novel Website Risk Assessment System (WRAS) for anti-phishing. The system protects a user against phishing and pharming attacks. That is, the WRAS system supports a more precise security risk index for the target website using both WWLs and self-learning phishing filtering techniques. Thereby, inexperienced users can prevent phishing websites from transferring sensitive information.

The algorithm to compute the security risk index 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 research was supported by the National IT Industry Promotion Agency (NIPA) under the program of Software Engineering Technologies Development.

References

1. Anti-Phishing Working Group (APWG) (2011), `http://www.antiphishing.org`
2. Kim, Y.-G., Cho, S.-H., Lee, J.-S., Lee, M.-S., Kim, I.H., Kim, S.H.: Method for evaluating the security risk of a website against phishing attacks. In: Yang, C.C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P.S., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K.M., Mao, W., Zhan, J. (eds.) ISI Workshops 2008. LNCS, vol. 5075, pp. 21–31. Springer, Heidelberg (2008)
3. Microsoft, Sender ID Framework Overview (2011), `http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx`
4. Yahoo: Yahoo! Anti-Spam Resource Center (2008), `http://antispam.yahoo.com`
5. Mutual Internet Practices Association, DomainKeys Identified Mail, DKIM (2011), `http://www.dkim.org`
6. Dhamija, R., Tygar, J.D.: The Battle against Phishing: Dynamic Security Skins. In: Proc. of the 2005 Symposium on Usable Privacy and Security (SOUPS 2005), pp. 77–88 (2005)
7. Dhamija, R., Tygar, J.D.: Phish and Hips: Human Interactive Proofs to Detect Phishing Attacks. In: Proc. of the Second International Workshop, pp. 127–141 (2005)
8. Fu, A.Y., 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 3(4), 301–311 (2006)
9. Liu, W., Deng, X., Huang, G., Fu, A.Y.: An Antiphishing Strategy Based on Visual Similarity Assessment. IEEE Internet Computing, 58–65 (2006)
10. Raffetseder, T., Kirda, E., Kruegel, C.: Building Anti-Phishing Browser Plug-Ins: An Experience Report. In: Proc. of third international workshop on Software Engineering for Secure Systems, SESS 2007 (2007)
11. Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., Mitchell, J.C.: Client-side Defense against Web-Based Identity Theft. In: Proc. of 11th Annual Network and Distributed System Security Symposium, NDSS 2004 (2004)
12. NetCraft (2011), `http://www.netcraft.com`
13. EarthLink (2011), `http://www.earthlink.com`
14. Microsoft, Anti-Phishing Technology (2011), `http://www.microsoft.com/mscorp/safety/technologies/antiphishing/`
15. Wu, M., Miller, R.C., Little, G.: Web Wallet: Preventing Phishing Attacks by Revealing User Intentions. In: Proc. of Symposium On Usable Privacy and Security (SOUPS 2006), pp. 102–113. ACM Press, New York (2006)
16. Emigh, A.: Online Identity Theft: Phishing Technology, Chokepoints and Countermeasures. ITTC Report on Online. Identity Theft Technology and Countermeasures (2005)