Journal of Network and Computer Applications 34 (2011) 1292–1301

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Using one-time passwords to prevent password phishing attacks

Chun-Ying Huang [a],*, Shang-Pin Ma [a], Kuan-Ta Chen [b]

[a] Department of Computer Science and Engineering, National Taiwan Ocean University, No. 2 Pei-Ning Road, Keelung 202, Taiwan
[b] Institute of Information Science, Academia Sinica, No. 128 Academia Road, Section 2, Nankang, Taipei 115, Taiwan

ARTICLE INFO

Article history:
Received 13 July 2010
Received in revised form
20 December 2010
Accepted 22 February 2011
Available online 2 March 2011

Keywords:
Anti-phishing
Identity management
In-band password delivery
One-time password
Web security

ABSTRACT

Phishing is now a serious threat to the security of Internet users' confidential information. Basically, an attacker (phisher) tricks people into divulging sensitive information by sending fake messages to a large number of users at random. Unsuspecting users who follow the instruction in the messages are directed to well-built spoofed web pages and asked to provide sensitive information, which the phisher then steals. Based on our observations, more than 70% of phishing activities are designed to steal users' account names and passwords. With such information, an attacker can retrieve more valuable information from the compromised accounts. Statistics published by the anti-phishing working group (APWG) show that, at the end of Q2 in 2008, the number of malicious web pages designed to steal users' passwords had increased by 258% over the same period in 2007. Therefore, protecting users from phishing attacks is extremely important. A naïve way to prevent the theft of passwords is to *avoid using passwords*. This raises the following question: *Is it possible to authenticate a user without a preset password?*

In this paper, we propose a practical authentication service that eliminates the need for preset user passwords during the authentication process. By leveraging existing communication infrastructures on the Internet, i.e., the instant messaging service, it is only necessary to deploy the proposed scheme on the server side. We also show that the proposed solution can be seamlessly integrated with the OpenID service so that websites supporting OpenID benefit directly from the proposed solution. The proposed solution can be deployed incrementally, and it does not require client-side scripts, plug-ins, nor external devices. We believe that the number of phishing attacks could be reduced substantially if users were not required to provide their own passwords when accessing web pages.

1. Introduction

Phishing is a malicious activity whereby an attacker (phisher) tries to trick Internet users into providing confidential information (Dhamija et al., 2006). It is a serious problem because phishers can steal sensitive information, such as users' bank account details, social security numbers, and credit card numbers. To achieve this goal, a phisher first sets up a fake website that looks almost the same as the legitimate target website. The URL of the fake website is then sent to a large number of users at random via e-mails or instant messages. Unsuspecting users who click on the link are directed to the fake website, where they are asked to input their personal information. Although the process of setting up a fake website sounds complicated, reports show that it is much easier than before as there are now "phishing kits" (McMillan, 2006; Danchev, 2008) that can create a phishing site in a very short time. Users believe that responsible enterprises should protect them from phishing attacks; thus, in addition to the risk of personal information leakage, successful phishing attacks can seriously damage business enterprises, especially a company's brand reputation (McDonnell, 2006; O'Brien, 2006).

1.1. Anti-phishing techniques

As phishing is a serious threat to both users and enterprises, several anti-phishing techniques have been developed. In general, the techniques can be classified as either list-based or heuristic-based technologies. List-based techniques maintain a black list or a white list, or both. Many anti-phishing mechanisms use a black list to prevent users from accessing phishing sites. However, the effectiveness of black list filtering depends on the coverage, freshness, and accuracy of the list. The URLs are usually reported by Internet users or collected by web crawlers, and list maintainers are responsible for verifying whether or not the listed URLs are really phishing sites. Though a well maintained black list can filter most well-known phishing sites, it obviously cannot filter unreported, uncollected, or unanalyzed URLs. No list can

* Corresponding author.
E-mail addresses: chuang@ntou.edu.tw, huangant@gmail.com (C.-Y. Huang), albert@ntou.edu.tw (S.-P. Ma), swc@iis.sinica.edu.tw (K.-T. Chen).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.02.004

guarantee 100% coverage and up-to-date freshness; and list-based filtering techniques often generate false negatives.

Some anti-phishing mechanisms use white lists that contain the names of trusted domains. If a user visits an unlisted website, a white-list-based filter may block the site immediately or require the user to make decisions on the fly. The drawback of this method is that the user may become annoyed if some sites are blocked or if the system constantly requests confirmation. Sometimes, it may even be difficult for the user to make a decision. In the end, the user may loose patience with having to validate unlisted sites and decide to disable the filter mechanism.

Heuristic-based mechanisms employ several criteria to determine whether a website is a phishing site. The following are some frequently used criteria.

- *Domain name*. A phishing site may register a similar domain name to that of the target site. For example, `paypal.com` and `paypa1.com` look the same, but the latter is actually `paypa` plus the numeral `1`. Several metrics can be applied to measure the "distance" between two strings, e.g., the Levenshtein (1965) distance, the Needleman–Wunch (1970) distance, and the Smith–Waterman (1981) distance. The measured distances can be used as an index to identify possible phishing sites.
- *URL*. A phisher may attempt to mislead users by including the `@` symbol in a URL. Browsers treat the text before the `@` symbol as the name used to access a website. This allows a phisher to redirect users to a fake site using a URL like www.paypal.com@123.123.123.123. The user thinks he is visiting `www.paypal.com`, but he is actually being redirected to another site with the IP address `123.123.123.123`. Thus, it is important to check whether a URL contains special symbols.
- *Image similarities*. Some methods try to detect phishing sites by checking image and visual similarities. There are various ways to do this. For example, it can be done strictly by comparing the hash values of image files (Venkatesan et al., 2000), or loosely by comparing the distribution of colors in images or web pages (Fu et al., 2006; Chen et al., 2009; Huang et al., 2010). A website that looks the same or similar to a well-known site but uses a different domain name may be a phishing site.
- *Specific input fields*. As phishing sites usually require users to input their personal information, a phishing site has some input fields for personal information, such as passwords, social security numbers, and credit card numbers. Thus, if a web page has such fields, it may be a phishing site.
- *Keywords*. Keywords can be used to distinguish between phishing and non-phishing activities. Zhang et al. (2007) propose a technique to identify frequently used terms on a web page by computing the term frequency and inverse document frequency (TF-IDF) (Salton and McGill, 1986). The identified terms are then submitted to a search engine to obtain a list of matching sites. The checked page may be a phishing site if its URL is not included in the site list returned by the search engine.

Heuristic-based mechanisms may use only one criterion to assess web sites. For example, the basic CANTINA filter (Zhang et al., 2007) only calculates the TF-IDF score. In contrast, the advanced CANTINA filter and the SpoofGuard filter (Chou et al., 2004) use a weighted score based on several criteria. Given a set of predefined weights for each criterion, the overall score used to evaluate a site is calculated by

$$s = \sum w_i P_i, \tag{1}$$

where w_i and P_i are, respectively, the weight and the probability of a given criterion i.

List-based and heuristic-based methods can not detect and block all phishing sites (Cranor et al., 2007). Moreover, users are not always aware of alerts displayed by anti-phishing toolbars (Wu et al., 2006a). Therefore, to prevent password phishing, we believe *it would be better to develop methods that tackle the root of the problem*, i.e., methods that authenticate a user on the Web.

1.2. Motivation

Lists of active phishing sites provided by PhishTank (2009) show that 70% of the sites are designed to obtain users' login names and passwords. Once a phisher obtains a valid username and password, the phisher can login to the phished account and retrieve further valuable information about the user. As phishers target users' account names and passwords, a naïve way to reduce the number of such attacks is to authenticate a user without having to use a preset password. Instead of using a preset password, a user is given a new password every time the user wishes to utilize the web service. However, to do this, *we need a reliable secondary channel to deliver the password.*

The rationale behind the proposed solution is quite simple. We propose that users can be authenticated with one-time passwords (OTPs) delivered via a reliable secondary communication channel on demand. The user database at the server side matches a user's login name with its corresponding identity on another secondary channel. When a user wishes to access a web site, the server sends an OTP to the user through the secondary channel. On receipt of the OTP, the user can login before the password expires. The proposed solution provides three levels of protection. A phishing attack can only succeed if the attacker knows (1) the user's account name; (2) the identity of the secondary channel through which the user receives the one-time password; and (3) the password used to access the secondary channel. These constraints complicate the phishing attack process. Moreover, as preset passwords are not used, phishers can only obtain users' login names.

There are several kinds of secondary communication channels, for example, e-mails, short message services, and instant message services. We consider that *an instant messaging service is the best secondary communication channel for our solution.* In addition to delivering messages in real-time, such *services are almost ubiquitous and the cost of using them is negligible*. Although an instant messaging service is not totally secure, with proper design and configuration, it can be a good medium for delivering passwords. The infrastructure for instant messaging services is already available on the Internet; and downloading the client side program and obtaining a user account are free of charge. Therefore, to utilize such services, a website only needs to set up an identity management database and run an instant messaging bot program to send one-time passwords.

1.3. Scope and limitations

It is probably impossible to prevent phishing attacks completely. Thus, the purpose of the proposed solution is to *reduce the number of password phishing attacks*. That is, a website that applies our solution can reduce the probability that password phishing attacks will be successful. However, as we leverage instant messaging services to deliver one-time passwords, those services may also become targets for phishers. We discuss this issue further in Section 3. Apart from phishing, other types of attacks try to steal users' account names or sensitive information. Many websites allow a user who has logged in to maintain his/her on-line status for a period of time by storing persistent cookies; however, the cookies might be stolen via cross-site script (XSS)