IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Two Way Checkpoint Browser Extension to Combat Phishing

Amisha K[1], M Reetika[2], Sahil Jain[3]

[1, 2, 3]**Department of Information Technology, Misrimal Navajee Munoth Jain Engineering College, Chennai**

Abstract

In the Internet world, the password of the users can be easily tricked through the spoofed web sites. Phishing is a form of online identity theft that aims to steal sensitive information such as online banking passwords and credit card information from users. Our paper aims at creating two check points which is a combination of AntiPhishing and PwdHash toolbar techniques. AntiPhish tracks the sensitive information of a user and generates warnings whenever the user attempts to give away this information to a web site that is untrusted. We then describe a browser extension, PwdHash, that transparently produces a different password for each site, improving web password security and defending against password phishing and other attacks. Hence client can freely access the low or high security website with full confidentiality because we propose the concept of security at client end.

Keywords :*AntiPhishing , PwdHash , Spoof.*

1. Introduction

A password is a word or string of characters used for user authentication to prove identity or access approval to gain access to a resource (example: an access code is a type of password), which should be kept secret from those not allowed access.

But there are two main problems related to passwords namely:

- Password phishing problem
- Common password problem

1.1 Password Phishing Problem

This is a very familiar problem. A user arrives at an official-looking web site, perhaps after clicking a link in a fraudulent email or mistyping a web address. The user thinks she's at a bank site, but actually she's giving his password to the fake site controlled by a malicious criminal. Once the user gives away his password, the attacker can now log in to the real bank with the user's account. This type of attack is called phishing attack.

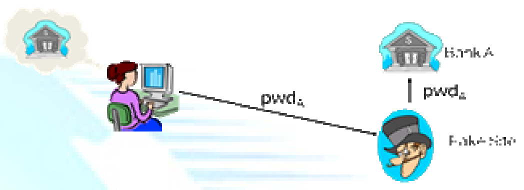

Fig. 1 Password given to the fake site

Phishing attacks are of two types:

- Spoofing e-mails and websites
- Exploit based phishing attack

1.1.1 Spoofing E-Mails And Websites

This attack dates back to the mid 90's. The attack involves sending spoofed mails to the users where the attacker tries to persuade the users to send back their passwords and other sensitive information. The success rate of this attack has been reduced today, because many users have learned not to send sensitive information via email.

1.1.2. Exploit Based Phishing Attack

These are more sophisticated attacks that make use of vulnerabilities in the web browser. For eg, a key logger may be installed in a software to capture key strokes. To mitigate such attacks browser manufactures must create bug free software and the users must be up to date about the latest security fixes.

1.2 Common Password Problem

Here's another problem of password attack. One user has online accounts at many different sites, and because it is difficult to remember a unique password for every site, she uses the same password at her bank and at her newspaper's login page.

Maybe the newspaper login page isn't protected. Maybe a successful phishing attack was executed on the newspaper site, In any case, the security of the bank site is compromised by the low-security site, because the user has a common password at both sites.So we have a situation where the security of the user's password is only as strong as the least secure site where the password is used.

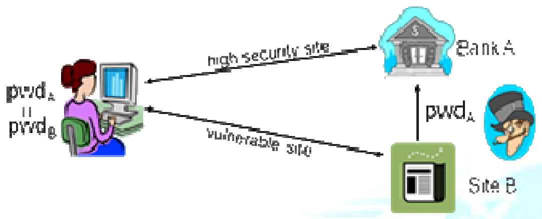

Fig. 2 Common password problem

2. Survey on Existing Antiphising Solution

Anti Phishing techniques are divided into three categories:

- Spam filters
- Anti Phishing toolbars
- Password protection mechanisms

2.1 Spam Filters

A spam filter is a program that is used to detect unsolicited and unwanted email and prevent those messages from getting to a user's inbox. Like other types of filtering programs, a spam filter looks for certain criteria on which it bases judgments. For example, the simplest and earliest versions (such as the one available with Microsoft's Hotmail) can be set to watch for particular words in the subject line of messages and to exclude these from the user's inbox. This method is not especially effective, too often omitting perfectly legitimate messages (these are called false positives) and letting actual spam through. More sophisticated programs, such as Bayesian filters or other heuristic filters, attempt to identify spam through suspicious word patterns or word frequency.

2.2 Anti – Phishing Toolbars

Anti-phishing software consists of computer programs that attempt to identify phishing content contained in websites and e-mail. It is often integrated with web browsers and email clients as a toolbar that displays the real domain name for the website the viewer is visiting, in an attempt to prevent fraudulent websites from masquerading as other legitimate web sites. Some of the toolbars used are:

2.2.1 eBay Toolbar

The eBay Toolbar uses a combination of heuristics and blacklists. The toolbar also gives users the ability to report phishing sites, which will then be verified before being blacklisted.

2.2.2 GeoTrustTrustWatch Toolbar

GeoTrust's web site provides no information about how TrustWatch determines if a site is fraudulent; however, it is suspect that the company compiles a blacklist that includes sites reported by users through a button provided on the toolbar.

2.2.3 Google Safe Browsing

Google provides the source code for the Safe Browsing feature and says that it checks URLs against a blacklist.

2.2.4 AntiPhish

AntiPhish is an academic solution which keeps track of where sensitive information is being submitted to.

2.2.5 Netcraft extension

Traps suspicious URLs containing characters which have no common purpose other than to deceive.

Enforces display of browser navigational controls (toolbar & address bar) in all windows, to defend against pop up windows which attempt to hide the navigational controls (Firefox only).

Clearly displays sites' hosting location, including country, helping you to evaluate fraudulent urls (e.g. the real citibank.com or barclays.co.uk sites are unlikely to be hosted in the former Soviet Union).

2.2.6 FirePhish

FirePhish is an Anti-Phishing Toolbar which utilizes the Open Phishing Database to provide the user with information and tools to protect against phishing

2.2.7 Snorkel toolbar

The Snorkel Toolbar is a unique, PKI-based, anti-phising solution that helps your customers to transact with only trusted websites, and alerts them when they visit phishing websites.

2.2.8 Spoofguard

Spoofguard does not use white lists or blacklists. Instead, the toolbar employs a series of heuristics to identify phishing pages.

2.3 Password Protection Mechanisms

2.3.1 Password Composer

Create unique Passwords for every Website and Web-App by only remembering one master password. This extension puts a tiny red icon to the left of a password entry field. If one clicks on this icon, the password field is overlaid with a replacement input, where one can supply a single, secure password (Master Password).

2.3.2 Magic Password generator

This extension [26] combines master password and the domain name of the site to make another unique password for that site. For advanced users, with a catchall address at a domain, just put

"@example.com" (whatever one’s domain is) for the address, and MPWGen will make a different email for every site too. Alternately, use "foo+@…" and the value will be inserted after the + sign, for email accounts that support this feature, like gmail.

2.3.3 Password generator

Password Generator gets the hostname from the page's URL and mixes it together with one’s personal master password using a little cryptographic magic MD5. It always gets the same result if given that hostname and master password, but will never get that result if either changes.

2.3.4 Hassapass

Hasspass automatically generates strong passwords from a master password and a parameter like domain name. The password generation is performed inside this very browser window in JavaScript

2.3.5 Genpass

GenPass is a JavaScript/MD5 bookmarklet-based password generator. GenPassis no longer being updated. Presently consider using SuperGenPass; however, note that SuperGenPass is not compatible with GenPass—given the same input, they generate different passwords.

2.3.6 Password Hasher

When the master key is given to Password Hasher and it enters the hash word into the site's password field. A hash word is the result of scrambling the master key with a site tag. Click on a # marker next to a password field or press the Control-F6 key combination when in a password field or choose Password Hasher from either the Tools menu or the right-click popup menu on a password field to enter the master key.

2.3.7 Pwdhash

Pwdhash is a browser extension that transparently converts a user's password into a domain-specific password. The user can activate this hashing by choosing passwords that start with a special prefix (@@) or by pressing a special password key (F2). Pwdhash automatically replaces the contents of these password fields with a one-way hash of the pair (password, domain-name).

3. Our Solution

In the proposed extension we protect the user and his password from phishing sites by using two check points namely,

- Protecting user’s sensitive information(Antiphish)
- Hashing the password before it is sent to the website server.(Pwdhash)

3.1 Overview

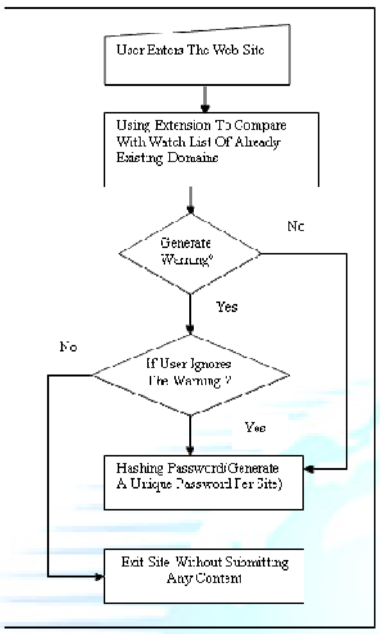
User Enters The Web Site
Using Extension To Compare With Watch List Of Already Existing Domains
Generate Warning?
No
Yes
No
If User Ignores The Warning ?
Yes
Hashing Password/Generate A Unique Password For Site)
Exit Site Without Submitting Any Content

Fig. 3 Overview Of Proposed Solution

The proposed system works as follows:

3.1.1 Registering On A Site

- When first time a user enters the website and fills the form, the password entered by the user is obtained by capturing the user's keystrokes.
- The keystrokes are captured using embedded HTML in the extension and not using any JavaScript codes.
- This information is then stored as the sensitive information in the watch list of the extension.
- Along with the password also the domain information of the site is captured and stored.
- After the user leaves the password field (i.e) focus is lost from the password field, the extension then hashes the password.
- Finally when the user clicks submit this hashed password is sent to the web server.

3.1.2 Logging Into A Site

- When the user enters the website and enters login details the extension captures the password and compares this with the watch list of sensitive information.
- The extension checks if the given password and domain information match:
    - If they match then the site is considered as trusted site and hashed password is sent to the web server.
    - Else the site generates a warning message telling the user that the site may be a phishing site.
- Upon receiving the warning the user may,
    - Ignore the warning, if he believes that the site is a trusted site or if the user wants to use a common password across multiple sites. In this case even if the site is a phishing site as the password is hashed and sent, the attacker cannot use the password at a genuine site.
    - Else if user upon receiving the warning feels that the site is a phishing site, he may leave the site immediately.
- Thus, the user's password is protected from any kind of phishing attack.

3.2 Working Of Check Point 1: Creation And Comparison With Watch List

- User enters text in the password field.
- In order to identify that the text being entered by the user is a password we use password prefix(@@ is used to identify a password)
- The password is obtained by capturing the keystrokes of the user.
- The password content is then mapped with the watch list already available.
    - If the password is not found in the list then it is considered as registering on a new site, and stores the password along with the domain information into the watch list.
    - Else if the password is found in the list then the corresponding domain details are checked,
        - If they match the user is allowed to continue normally.
        - Else a warning is generated.
- If the user would like to use the same piece of sensitive information (e.g., the same password) on multiple web sites, this information has to be captured by AntiPhish for all sites where it is being used. This is typically done by first deactivating AntiPhish from the menu in order to prevent it from generating false phishing alerts.

3.3 Working Of Checkpoint 2: Hashing The password

The following figure clearly shows how a watch list is created and used by our solution.

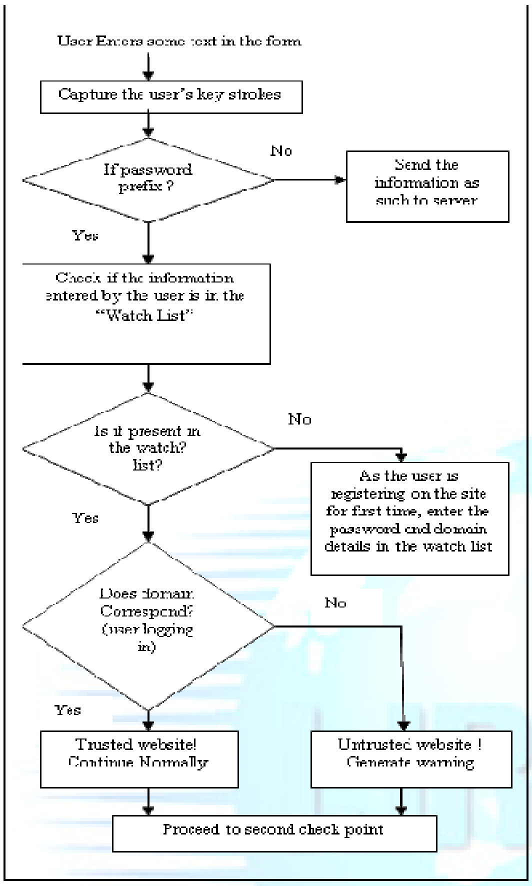
User Enters some text in the form
Capture the user's key strokes
If password prefix ?
No
Send the information as such to server
Yes
Check if the information entered by the user is in the "Watch List"
Is it present in the watch? list?
No
As the user is registering on the site for first time, enter the password and domain details in the watch list
Yes
Does domain Correspond? (user logging in)
No
Yes
Trusted website! Continue Normally
Untrusted website ! Generate warning
Proceed to second check point

Fig. 4 Check Point 1

- User enters text in the password field.
- In order to identify that the text being entered by the user is a password we use password prefix(@@ is used to identify a password)
- The password is obtained by capturing the keystrokes of the user.
- The password content is then mapped with the watch list already available.
- If the password is not found in the list then it is considered as registering on a new site, and stores the password along with the domain information into the watch list.
- Else if the password is found in the list then the corresponding domain details are checked,
  - If they match the user is allowed to continue normally.
  - Else a warning is generated.
- If the user would like to use the same piece of sensitive information (e.g., the same password) on multiple web sites, this information has to be captured by AntiPhish for all sites where it is being used. This is typically done by first deactivating AntiPhish from the menu in order to prevent it from generating false phishing alerts.

3.3 Working Of Checkpoint 2: Hashing The Password

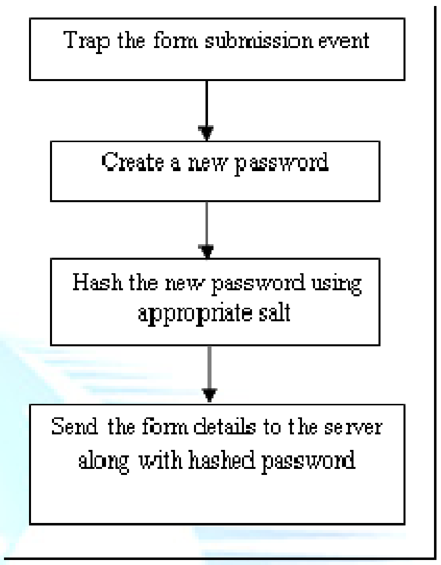
Trap the form submission event
Create a new password
Hash the new password using appropriate salt
Send the form details to the server along with hashed password

Fig. 5 Checkpoint 2

- For hashing to be done, we trap the form submission event.
- By doing so, before the form is submitted the password field is replaced by hashed password.
- A new password is created as follows:
  - The password prefix '@@' is ignored.
- Every letter after the prefix is replaced by another letter starting with Z. Even special characters are hashed in same way.
  - Example:
    - Original pwd:
    - @@abacus11#$
    - New Pwd:
    - ZYZXWVUUTS
- The new password is then hashed by using appropriate salt. There are two possible salts that can be used:
  - Domain of site hosting the current page.
  - Domain name of site that is going to receive the content.
  - Value provided in SSL Certificate.
- But in our extension we use the domain name of site hosting current page because:
  - If target domain is used as salt then it becomes difficult for the server to obtain the original password from hashed password.
  - Also they suffer from Password Reflection attacks.

- Finally the hashed password along with other contents is sent to server.

4. Possible Complexities and Their Solutions

4.1 Keyboard monitoring

In the normal scenario, the attacker can attack via JavaScript functions. These functions can listen to keyboard events sent to the password field and record those keys in some auxiliary hidden field .

```
<form><input type="hidden" name="secret"
value=""><inputtype="password"
name="passwordonKeyPress="this.form.secret.val
ue+=String.fromCharCode(event.keyCode);"></for
m>
```

As a result, the phisher obtains the user’s clear text password.

Solution

This situation can be prevented by antiphishing technique used in our system. The solution we use in AntiPhish is to deactivate Javascript every time the focus is on an HTML text element and to reactivate it whenever the focus is lost. Using this technique, we ensure that the attacker is not able to create hooks,timers and intercept browser events such as key presses while the user is typing information into a text field. At the same time, we ensure that the legitimate Javascript functionality on a page(e.g., such as input validation routines) are preserved.By the time the focus is lost from the text element and Javascript is reactivated, AntiPhish has already determined if the information that was typed into the text element is sensitive.

4.2 Domain Rewriting

When the page is first loaded, the form action attribute can point to a proper banking site. However, when the user hits the “login” button, a JavaScript function changes the form action to point to the phishing site. As a result, in the straightforward implementation, the browser sends the user’s password hashed with the banking domain name to the phisher. The phisher thus obtains the user’s banking password.

```
<form action="http://www.bank.com/">
<input type="password "name="password">
<input type="submit" value="Submit"
onClick=’this.form.action="http://www.phishers.co
m/"’></form>
```

Solution

There are two possible values for the salt(domain name): (1) The domain name of the site hosting the current page(the current

domain), or (2) The domain name that will receive the form data upon form submissions (the target domain). But for security reasons we favor using the current domain name over the target domain name. As shown in the above code

If we hash the password using target address, there is a possibility for the phisher to track the password.

4.3 Mock Password Field

Phishers can create a text field <input type=“text”> that behaves like a password field. For every keystroke sent to the field, a JavaScript function appends the key to some hidden field and writes an asterisk into this mock password field . Since the field type is text, the PwdHash browser extension leaves it unhashed. As a result, once the form is submitted, the phisher obtains the user’s cleartext password. More generally, phishers can use JavaScript to confuse the user into typing a password in an insecure location, such as a text field or a popup window.

Solution

Finally, if the password-prefix is ever detected while focus is not on a password field, our browser extension reminds the user not to enter a password. Thus, users are protected from mock password field attacks that confuse them into entering a password into an insecure location.

4.4 Dictionary Attacks

PwdHash ensures that phishing sites only learn a hash of the user’s password. Since PwdHash uses a well known hash function, the phishing site could attempt an offline dictionary attack to obtain the user’s cleartext password. Since dictionary attacks succeed 15-20% of the time , this is a potential weakness.

This solution, already implemented in UNIX, increases the computing resources needed to mount a dictionary attack.Extreme versions of this solution, using ultraslowhash functions, are proposed . PwdHash is an ideal application for slow hash functions.

5. Conclusion

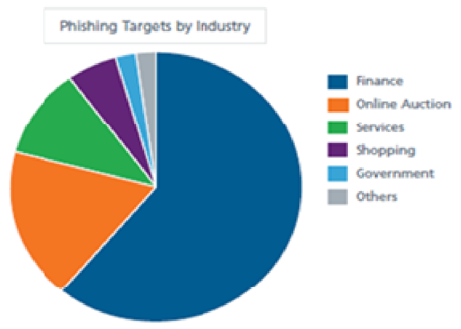

Fig. 6 Phishing targets by industry as per 2013

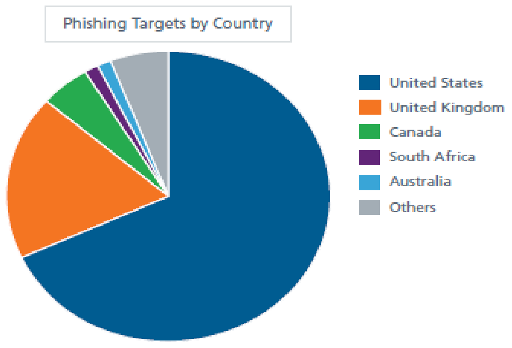

Fig. 7 Phishing targets by country as per 2013

Although phishing scams have received extensive press coverage, phishing attacks are still successful because of many inexperienced and unsophisticated Internet users. Attackers are employing a large number of technical spoofing tricks such as URL obfuscation and hidden elements to make a phishing web site look authentic to the victims.In this paper we provide a novel solution for protecting against such phishing attacks. We provide an extension with two check points by keeping track of the user's sensitive information and password prefix. We not only protect against phishing attacks but also provide a solution for common password problem.

In terms of future work we plan implement our extension in various browsers and also support remote login facility.

References

[1] "Protecting Users Against Phishing Attacks with AntiPhish" ,EnginKirda and Christopher Kruegel Technical University of Vienna.

[2] "Stronger Password Authentication Using Browser Extensions", Blake Ross, Collin Jackson, Nick Miyake, Dan Boneh, John C Mitchell.

[3] "Client Side protection from Phishing attack", Venkata Prasad Reddy et al. / (IJAEST) International Journal Of Advanced Engineering Sciences And Technologies.

[4] "Authenticated Key Exchange Secure Against Dictionary Attacks", MihirBellare, David Pointcheval, and Phillip Rogaway.

[5] "Client-side defense against web-based identity theft",Neil Chou , Robert Ledesma Yuka Teraguchi, Dan Boneh and John CMitchell.

[6] "SpoofGuard. Client-side defense against webbased identity theft." http://crypto.stanford.edu/SpoofGuard

[7] Mozilla Extensions. Home Pagehttp://update.mozilla.org/extensions

[8] News.Com. Netscape readies antiphishing browser. http://news.com.com/21007355 3-5558006.html

[9] Password Hasher. http://wijjo.com/PasswordHasher

[10] Password generator. http://www.angel.net/~nic/passwdlet.html