E-Leader, Hong Kong 2007

How to Defend against Identity Theft, Phishing, and Web Spoofing

Mario A. Garcia, Michael Dirks, and Matt Weatherston
Texas A&M University-Corpus Christi
mario.garcia@tamucc.edu

Abstract

Identity theft is an attractive activity for cyber-criminals. Every day, many people are victim of one of the many cyber-traps resulting in millions of dollars lost. People victim of these crimes suffer terrible consequences trying to clear their names. Hundreds or even thousands of dollars as well as time and energy could be spent in dealing with credit reporting bureaus, financial institutions, law enforcement and other sources to recover from this crime. In this paper, some of the most recent strategies and tools developed to defend against these type of attacks are also discussed. Among the most important tools are Spoofguard, PwdHash, and SafeCache developed at Stanford University. In addition, WebWallet and Security Toolbars implemented at MIT are also described.

Keywords

Identity Theft, Phishing, Spyware, Web spoofing, Information Assurance,
Abuse and Crime involving Computers, Invasive Software.

Introduction

Identity theft, Web spoofing, identity fraud are terms used to refer to crimes in which a cyber-criminal wrongfully obtains and uses another person's personal information in some way that involves fraud or deception. Identity theft is becoming one of the most common and attractive forms of theft in the world. It has affected millions of people in recent years. Cyber-criminals can gain access to personal information in many different ways. Some are as simple as “dumpster diving”, using social engineering, or scamming people into giving personal information to them. Cyber-criminals can hack into databases that contain people’s information, abuse the access that they have been given through their employer, or even illegally collect personal information that is entered into a computer by the user. One of the most common attacks used by cyber-criminals is to scam people into giving up their personal information by deceiving individuals that they work for a legitimate business.

Defense against Identity Theft

A defense is to provide two-phased authentication to users in e-mail and on the corporate website. An example of this would include the normal username/password combination combined with an authenticating image and/or phrase chosen by the user. Any time the corporation sends out an e-mail to a user, they would include this personal phrase and image in the e-mail. Since a phisher would not know this information, the e-mail would be validated to the customer. By the same token, including the image and phrase on the corporation website as part of the login process would also validate the website to the user as well. This can help prevent phishing attacks by providing authentication of communication between a user and a corporation (Geer, 2005).

One of the few current approaches that can possibly target the root of the problem is to watch corporations’ web logs for users downloading their images. Creators of phishing websites usually use the actual images from the corporate website to make the phishing website more believable. If users are downloading images on to their personal computer, then their IP address will show up in the server logs. The Corillian Fraud Detection System (CFDS) is a commercial server that looks for such a behavior in web logs. It then investigates further to find the phishing site that is illegally using those images. Corillian then notifies the administrator of the compromised server and the authorities (Geer,2005).

Trustbar

The "Trustbar" proposal is a third party certification solution, where websites logos are certified. The authors suggest creating a "trusted credentials area" as a fixed part of the browser window [Herzberg & Gbara, 2004). This area can be used to present credentials from the website, such as logos, icons and seals of the brand, that have been certified by trusted certificate authorities or by peers using a PGP "web of trust".

AOL Passcode

America Online's Passcode has been proposed as a phishing defense. This program distributes RSA SecurID devices to AOL members (RSA, 2004). The device generates and displays a unique six-digit numeric code every 60 seconds, which can be used as a secondary password during login to the AOL website. This scheme reduces the value of collecting passwords for attackers because the passwords can not be used for another transaction.

Passmark and Verified by Visa

Shared-secret schemes have been proposed as one simple approach to help users identify known servers. For example in proposals such as Passmark and Verified by Visa , the user provides the server with a shared secret, such as and image and/or passphrase, in addition to his regular password. The server presents the user with this shared secret, and the user is asked to recognize it before providing the server with his password. In the Passmark scheme, the bank server places a secure cookie on user machine, which must be presented at login. This prevents a classic man-in-the middle (MITM) attack where an attacker interposes himself between the client and the bank.

SRD

Ye and Smith proposed "Synchronized Random Dynamic Boundaries" to secure the path from users to their browser (Ye,& Smith, 2002). This scheme uses a random number generator to set a bit that determines whether the browser border is inset or outset. The browser border alternates between inset and outset at a certain frequency in concert with a reference window. A strength of this solution is that it recognizes the "general purpose graphics" problem. In this scheme, rogue servers can not predict the random number that is chosen by the browser, and therefore it is difficult to create spoof windows that blink at the correct frequency.

YURL Petnames

In the YURL proposal, the user's browser maintains a mapping of a public key hash to petname. When a user visits a page identified by a YURL, the browser displays the petname that the user previously associated with the public key hash (Waterken). An untrusted site can be recognized by the absence of a corresponding petname displayed in the browser. This is a very simple scheme that requires a small degree of personalization for each website.

eBay Toolbar

The eBay Toolbar is a browser plug-in that eBay offers to its customers to help keep track of auction sites (eBay). The toolbar has a feature, called AccountGuard, which monitors web pages that users visit and provides a warning in the form of a colored tab on the toolbar. The tab is usually grey, but turns green if the user is on an eBay or PayPal site or red if the user is on a site that is known to be a spoof by eBay. The toolbar also allows users to submit suspected spoof sites to eBay.

Spoofstick

Spoofstick is a toolbar extension for Internet Explorer and Mozilla Firefox that provides basic information about the domain name of the website. For example, if the user is visiting Ebay, the toolbar will display "You're on ebay.com". If the user is at a spoofed site, the toolbar might instead display "You're on

10.19.32.4". This toolbar can help users to detect attacks where the rogue website has a domain name that syntactically or semantically similar to a legitimate site.

Spoofguard (Chou et al, 2004).

Web spoofing is an internet fraud situation when a website fools a user into giving up private information. An attack usually starts as a mass emailing to victims, claiming that the victims need to visit the spoof page to resolve an issue with their account. Once users are tricked into visiting the site and giving up their login information, the spoofer uses the information to engage in fraud. SpoofGuard is an Internet Explorer plug-in that warns users of potential spoofing activity. This plug-in monitors internet activity, computes a 'spoof index' indicating the probability that a given page is a spoof, and warns of spoofing if the index is above a minimum set by the user.

Attributes such as domain name, url, link, and images are used by SpoofGuard to compute the spoof index. The history is also examined, as well as referring pages; if the referring page is an email service, it is more likely to be a spoof. SpoofGuard will detect user entered passwords and warn if the password is not going to the place it should be going. In this way it can prevent information leakage. The evaluation of SpoofGuard is based on how well it catches spoofers, how many false alarms there are, and how it affects performance.

To detect phishing attacks, Spoofguard uses three groups of tests: stateless methods which evaluate a downloaded page, stateful methods that evaluate a page with respect to user activity, and post data from the page. Based off of how the page scores with these methods, the total spoof score is calculated as a standard aggregation function, summing products of pairs, triples and larger subsets as well as individual test results, because certain combinations of attributes make a page drastically more suspicious. Stateless page evaluations include a URL check to make sure that it is not misleading or too close to an honest site without actually being the honest site, an image check to make sure that any logos found on the page do not match with logos from honest sites that are kept in a database by the plug-in in the form of an image hash, a link check to make sure links pass the previously stated URL test, and a password check to see if the page is asking for a password. Stateful page evaluation uses a domain check to see if the domain is similar to a previously visited domain indicating that it may be part of a spoof, checking the referring page to see if the user was reading email right before clicking the link, and image-domain associations check similar to before built on an image history file kept by the plug-in. Post data is intercepted by SpoofGuard for checking, and if found suspicious, blocked. SpoofGuard here checks if there is a previously used password in the data, if the logo for the site where that password is valid is present, and all other data is checked in case fields are not labeled as password fields in an identifiable way to the plug-in. Search engines are an exception to these checks because they might contain any of this data legitimately.

PwdHash (Ross et al. 2005)

PwdHash is a browser extension developed at Stanford University that can stop phishing attacks by hashing passwords entered into the browser. The idea behind password hashing is that passwords are harder to steal, and when stolen, are only useful at one site. The method boils down to, rather than sending the password a user enters, a hash of that password is sent. The cleartext password is hashed with the domain name of the login site. The pseudo random function used by PwdHash would yield the proper password when hashed with the correct domain. If a phisher does not have the right domain name, then the password he receives will be useless to him.

PwdHash uses the current site domain as the hashing salt in order to prevent possible password reflection attacks. It does not use SSL certificates because authenticity can be questionable, they are hard to manually replicate when roaming, and not all sites have SSL certificates. For most sites, the domain of the login page and the domain of the destination page are the same so there is no difference, but sometimes there is a difference so these are special cases that are dealt with via a short rule sequence containing a regular expression for the domain name, a salt-rule, and an encoding algorithm to use. These special case rules can determine when to use a different salt, what rules to use to make sure a workable domain is used for the salt, and which of the five hashing algorithms to use to create the most appropriate compatible hash. These rules would need to be updated to handle new websites.

For the Internet Explorer implementation, PwdHash uses a low-level windows keyboard hook to intercept the password before the web page does. The functionality is just as previously described, with the addition of a small 'traffic light' toolbar to indicate password security. Internet Explorer version does not submit the hashed password until form submission. The BeforeNavigate2 handler is intercepted and replaced with a modified one that has the hashed password. The Mozilla FireFox version differs only in that it uses a lock icon instead of a traffic light, and it replaces the password with the hashed value right when focus leaves the password field. To keep the user from mingling hashed and unhashed characters when editing, the field is automatically cleared when it gets focus.

SafeCache (Jackson et al, 2006)

SafeCache is a Firefox browser extension developed at Stanford University that overrides the original caching policy and asserts its own policies. No toolbar feature or other visible change is involved, just a modification of Firefox via downloadable software. The software allows the user to set cookie settings and then enforces them.

The long-term state of a browser is necessarily kept for many web features, but hiding this state is necessary for prevented privacy attacks. To keep privacy contained a "same-origin" principle is applied. This principle says that sites from different domains cannot interact with each other except in limited ways. The principle, however, must be adapted to the persistent browser state situation because of the privacy leaks that can happen due to the state. In particular, the web feature of caching stores information that is not hidden from other sites, and visited link differentiation can tell a web site what other sites have been visited by a given computer.

The most frequent technique to track users is to store identifiers into the browser state. "Only the site that stores some information in the browser may later read or modify that information." In practice, this means that whenever two or more sites jointly observe the storing of information in the browser state, each site is only able to read back that portion which was written by that site. The same principle with the read event applies. In this way, unwanted tracking can be reduced.

Cache timing is one way that internet users can be tracked. By measuring how long a page takes to load, a privacy attacker can determine if the images etc. were already in the cache. DNS cache timing is similar, though less reliable: the time taken for a DNS lookup will give away whether a particular domain has been recently visited.

Caching behavior can be changed to follow the same-origin policy to prevent non-cooperative sites. The browser would identify both the embedding site and the host site, allowing only the host site full knowledge of information caching while the embedding site would only know that something was cached. If a different site embeds the same content, the cached content can't be used so that it remains unknown whether the information is already there or not.

Jackson et al, present a plug-in that will enforce this same-origin caching policy. This extension will install itself in between the default caching service and the browser, and will use the cookie preferences to determine whether or not to block caching for a given site, i.e. if the user allows cookies for a site then caching will be allowed and so on.

The other kind of visited link tracking, is a based off of another feature that uses persistent client-side state. The knowledge of whether or not a user has clicked a link can be used to track a user, and can be obtained by reading the color of a link with javascript.

Cache control is a major issue when it comes to tracking and popups. SafeCache provides a better means of handling cookies and cache according to the users settings then does the original methods provides by Firefox and Internet Explorer. By eliminating semi-cooperative tracking and non-cooperative tracking the users experience on the internet is safer. On the other hand cooperative tracking does allow business to provide more customized user interfaces. Depending on the users desires SafeCache can make the internet experience markedly safer.

The Web Wallet (Wu et al., 2006)

WebWallet, developed at MIT, is the implementation of an anti-phishing solution, incarnated as a browser sidebar used to securely submit personal information across the internet. Currently, web sites are easy to imitate, and this is effective because users base their decision on whether to trust a site on its appearance.

The WebWallet should be opened whenever sensitive information is to be entered by pressing the F2 key. On opening, the WebWallet will either retrieve stored data on a login card for the site that is being visited or allow the user to fill out a new login card if the site is new to WebWallet and if WebWallet deems the site good enough, or allow the user to indicate intention beforehand if the site is not trustworthy. As long as the user always presses the security key (F2), WebWallet will protect against phishing attacks, effectively preventing phishing in all attacks where it was used. It is dependant upon users being vigilant in always using the security key, however, so it is susceptible to a fake WebWallet attack.

The WebWallet has some similarities to Microsoft's InfoCard identity metasystem, in that it provides a uniform user authentication interface. However there are many differences, including: Websites must be modified to accept InfoCard submissions whereas WebWallet can be used at sites as is, InfoCard needs support from various identity providers, InfoCard users need to obtain InfoCards from different identity providers and users must authenticate themselves every time they select an InfoCard. Lastly, InfoCard users must make correct security decisions with only site information, something that they generally are not reliable at.

The design of WebWallet is based on two principles: structuring the interface to ensure user intention and integrating security into the task workflow so that it won't be ignored. Since users recognize sites mainly by visual appearance and content, and systems only recognize sites by system properties, neither alone can prevent a phishing attack. The WebWallet will enable users to clearly indicate to the system their intentions so the system can determine if the user's intentions match the system actions. Whenever WebWallet is opened, the user has indicated that secure information needs to be submitted, so this intention is made known. Selecting a specific card lets the system know where the user intends the information to go—if the site is not the right one, WebWallet can alert the user and allow him the option of going to the correct site. If it is a new card that is being filled out for a site, then WebWallet can use heuristics to alert the user that the site is suspicious and help the user find a legitimate site, or allow the user to decide to continue anyway. This respects the user's intentions and provides a way to finish the task without risks. WebWallet does not rely on users to remember to use it in most cases, because it disables sensitive input fields so that the user must go through WebWallet to give the information.

Toolbars

Many proposed anti-phishing solutions use toolbars that show different types of security messages to help users to detect phishing sites. Users are also advised to look at the existing browser security indicators, e.g., the URL displayed in the address bar and the lock icon displayed in the status bar when a connection is SSL-protected. However, controlled user studies have shown that these security indicators are ineffective against high-quality phishing attacks for several reasons: (Wu et al., 2006).

- Warning indicators located in a peripheral area provide a much weaker signal than the centrally displayed web page and can be easily overwhelmed by convincing web content.
- The security-related information shown by the indicators is not really needed for the user's current task. Since security is rarely a user's primary goal, users fail to pay continuous attention to the indicators. Making security a separate task that users are required to remember is not an effective solution.

- Sloppy but common web practices cause some users to rationalize the violation of the security rules that some indicators use to detect phishing attacks. For example, users are told to examine the hostname displayed in the address bar, to make sure that the hostname is the one they are expecting. But some legitimate websites use IP addresses instead of hostnames (e.g., the Google cache) and some sites use domain names that are totally different from their brand names.
- Users are also told to find the SSL lock icon before submitting sensitive information. But many legitimate banks still use unprotected login pages (Herzberg, 2005). Moreover, some indicators deliver warnings without detailed convincing explanations, which makes users think that the software is buggy and not treat the warning seriously.
- Security indicators tend to show that something is wrong and advise users not to proceed, but they do not suggest good alternatives. This may encourage users to risk submitting their information anyway, since they don't see any other way to accomplish their goal.
- Active interruption like the popup warnings is far more effective than the passive warnings displayed in the toolbars. But it's well-known that popup confirmations, used indiscriminately, become less effective over time. It should always appear at the right time with the right warning message.
- Internet companies need to follow some standard practices to better distinguish their sites from malicious phishing attacks. Companies should use a single domain name that matches their brands name rather than using IP addresses or multiple domain names for servers. They should use SSL to encrypt every web page on their sites. SSL certificates should be valid and from widely used CAs.
- Stop phishing at the email level (e.g., Adida et al., 2005), since most current phishing attacks use broadcast email (spam) to lure victims to a phishing website.
- Use security toolbars. The phishing filter in IE7 (Sharif, 2006) is a toolbar approach with more features such as blocking the user's activity with a detected phishing site.
- Visually differentiate the phishing sites from the spoofed legitimate sites. Dynamic Security Skins (Dhamija & Tygar, 2005) proposes to use a randomly generated visual hash to customize the browser window or web form elements to indicate the successfully authenticated sites. PassMark includes a personalized image in a web page to indicate that the user has set up an account with the site. This approach places the burden on users to notice the visual differences between a good site and a phishing site and then correctly infer that a phishing attack is underway. The Web Wallet, by contrast, detects the discrepancy itself, by comparing the user's intention with what the user is actually doing.
- Two-factor authentication, which ensures that the user not only knows a secret but also presents a security token [FDIC, 2004)]. However, this approach is a server-side solution. Phishing can still happen at sites that do not support two-factor authentication. Sensitive information that is not related to a specific site, e.g., credit card information and SSN, cannot be protected by this approach either. The Web
- The PRIME project (Pettersson, 2005) helps users to manage their online identity in a more natural and intuitive way using three UI paradigms. It supports drag-and-drop actions for personal information submission. It does not specifically target the phishing problem but its improved user interface could help users correctly manage their online information. One potential problem with the PRIME interface is its "Just-In-Time-Click-Through Agreements" (JITCTAs) that is used to generate "small agreements that are easier for the user to read and process". Users could still ignore the agreements by directly clicking through the "I Agree" button. On the other hand, the Web Wallet integrates security questions into the user's workflow so that users have to explicitly indicate their intended sites when submitting sensitive information.

Conclusion.

The paper presented some of the most frequently strategies used by cyber-criminals to steal identity from people. The next phase of this research will present more technical details about how to help users to protect against this crime. Relevant work has been done at Stanford University where a couple of patches have been released for Mozilla Firefox. Harvard University and MIT have done experiments with students to demonstrate why Identity theft attacks succeed. Hopefully, the information presented in this research will help people to understand this problem and be alert all the time.

References

Chou, N., Ledesma, R., Teraguchi, Y., & Mitchell, J.C. (2004). Client- Side Defense Against Web-Based Identity Theft. *Proceedings of the 11th Annual Network and Distributed System Security Symposium.*

Core Street, *Spoofstick*, http://www.corestreet.com/spoofstick/

Dhamija, R. & Tygar, J.D. (2005). The Battle Against Phishing: Dynamic Security Skins. Symposium on Usable Privacy and Security, pp. 77-88.

eBay, *eBay Toolbar*, http://pages.ebay.com/ebay_toolbar/

FDIC. (2004). Putting an End to Account-Hijacking Identity Theft. http://www.fdic.gov/consumers/consumer/idtheftstudy/ identity_theft.pdf

Herzberg, A. (2005). The 'Unprotected Login' Inter-Net Fraud League (I-NFL) Hall of Shame. http://www.cs.biu.ac.il/~herzbea//shame/

Herzberg, A. & Gbara, A. (2004). TrustBar: Protecting (even Naïve) Web Users from Spoofing and Phishing Attacks. http://www.cs.biu.ac.il/~herzbea/Papers/ ecommerce/spoofing.htm.

Jackson, C., Boneh, D., Bortz, A., & Mitchell, J.C. (2006, May). Protecting Browser State from Web Privacy Attacks, 15th International World Wide Web Conference WWW2006, Edinburgh.

PassMark Security, *Protecting Your Customers from Phishing Attacks- An Introduction to PassMarks*, http://www.passmarksecurity.com/

Ross, B., Jackson C., Miyake, N., Boneh, D., & Mitchell J.C. (2005, August). Stronger Password Authentication Using Browser Extensions, Usenix Security Symposium, Baltimore.

RSA (2004). Security, *Protecting Against Phishing by Implementing Strong Two-Factor Authentication*. 2004, https://www.rsasecurity.com/products/securid/ whitepapers/PHISH_WP_0904.pdf

Waterken Inc., *Waterken YURL Trust Management for Humans*,
http://www.waterken.com/dev/YURL/Name/

Wu, M., Garfinkel, S., & Miller, R. (2004). Secure Web Authentication with Mobile Phones. DIMACS Workshop on Usable Privacy and Security Software, 2004.

Wu, M., Miller, R., & Garfinkel, S. (2006, July). Do Security Toolbars Actually Prevent Phishing Attacks? CHI 2006.

Wu, M., Miller, R., & Little, G. (2006, July). Web Wallet: Preventing Phishing Attacks by Revealing User Intentions, Symposium On Usable Privacy and Security (SOUPS) July 12-14, Pittsburgh, PA, USA.

Ye, Z., & Smith, S. (2002). *Trusted Paths for Browsers.* Proceedings of the 11th Usenix Security Symposium.