Effective Protection Against Phishing and Web Spoofing

Rolf Oppliger[1] and Sebastian Gajek[2]

[1] eSECURITY Technologies, Gümligen, Switzerland
[2] Horst Görtz Institute for IT-Security, Ruhr University Bochum, Germany

Abstract. Phishing and Web spoofing have proliferated and become a major nuisance on the Internet. The attacks are difficult to protect against, mainly because they target non-cryptographic components, such as the user or the user-browser interface. This means that cryptographic security protocols, such as the SSL/TLS protocol, do not provide a complete solution to tackle the attacks and must be complemented by additional protection mechanisms. In this paper, we summarize, discuss, and evaluate the effectiveness of such mechanisms against (large-scale) phishing and Web spoofing attacks.

Keywords: SSL/TLS, phishing, Web spoofing, visual spoofing.

1 Introduction

There are many technologies to protect e-commerce applications. Most importantly, the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols are used to authenticate the Web server and to establish a cryptographically secure channel between the browser acting on behalf of the user and the Web server. The user is typically authenticated with one or several of the following mechanisms: user identification (ID) and password, transaction authentication number (TAN), one-time password, challenge-response mechanism, or public key certificate. The corresponding authentication information is transmitted over the SSL/TLS channel. While the SSL/TLS protocol is—with some minor theoretical vulnerabilities—reasonably secure [1], the way it interacts with the user and the way it is employed in e-commerce applications typically are not [2]. In fact, spoofing attacks—like phishing and Web spoofing—show that currently deployed authentication mechanisms for Web applications are insufficient to protect users against fraudulent Web sites. When spoofing attacks on the Web were first introduced and discussed in [3], they were purely theoretical and only a few incidents really occurred in practice. This has changed dramatically, and contemporary e-commerce applications are severely threatened by phishing and Web spoofing attacks, which have proliferated and recently become a major nuisance on the Internet and WWW. Consequently, there is considerably strong pressure to protect users against these types of attacks.

One could argue that a cryptographically secure end-to-end mutual authentication protocol is all that is needed to protect users against phishing and Web

J. Dittmann, S. Katzenbeisser, and A. Uhl (Eds.): CMS 2005, LNCS 3677, pp. 32–41, 2005.

spoofing attacks. There are several protocols that can be used for this purpose. Even though these protocols are cryptographically sound, the user behavior may induce some security problems. An unsophisticated user may simply recognize a window that pops up and requests to enter his credentials (i.e., authentication information). As long as an adversary is capable of imitating the window, the user can not reliably verify whether the window is legitimate and originated by the browser or whether it is spoofed by malware. Consequently, the user may provide his credentials assuming that they are handled by a secure protocol, whereas in reality they are sent to the adversary. Recently, the term *doppelganger window attack* was coined to refer to this type of attack [4]. The ease of mounting these attacks is worrisome, and the underlying problem is the interface between user and the implementation of the cryptographic system. This problem cannot be solved with another layer of cryptography—the user would have to be interfaced to it again. Against this background, we argue (i) that phishing and Web spoofing is threatening because the corresponding attacks target non-cryptographic components, such as the user or the user-browser interface, (ii) that the implementations of existing cryptographic security protocols, such as the SSL/TLS protocol, do not provide a complete solution, and (iii) that these protocols must be complemented by additional protection mechanisms. This it what this paper is all about. In Section 2, we introduce the relevant attacks and distinguish between five attack levels. In Section 3, we summarize, discuss, and evaluate the effectiveness of corresponding protection mechanisms. In Section 4, we summarize our major findings, and in Section 5, we identify some open research challenges. Finally, we provide some conclusions in Section 6.

2 Relevant Attacks

In *native phishing*, an attacker sends an e-mail to the victim, requesting the victim to reveal and send back (by e-mail) his or her password. Today, more complex and sophisticated phishing and Web spoofing attacks take place. They typically consist of two stages:

1. The attacker directs the victim to a Web site he controls. According to [5], we use the term *mounting attack* for this stage.
2. The attacker uses his Web site to spoof a legitimate Web site of an arbitrary company or organization. We use the term *spoofing attack* for this stage.

There are many mounting attacks. They can be categorized into those operating on the network layer and those operating on the application layer. Examples of network layer mounting attacks are ARP, IP, and DNS spoofing. Examples of application layer mounting attacks are e-mail and URL spoofing or cross-site-scripting. For the purpose of this paper, we assume that there are so many types of current and future mounting attacks, that it is infeasible to protect against all of them. So we only consider spoofing attacks.

Again, there are many possibilities to realize a spoofing attack. For example, in the simplest case, the attacker simulates the look-and-feel of the spoofed site—this is simple, because the attacker can reuse the images and icons from the spoofed site. Although common browsers have a set of indicators that provide information about the connection, two practical concerns remain: on the one hand users are notoriously bad (or untrained) at verifying and validating these indicators and, on the other hand, there are still many possibilities to either manipulate or overwrite them. However, if the browser and server communicate over an SSL/TLS channel, then it is somehow more challenging for the adversary to mount a spoofing attack. This is because there are several *browser's secure connection indicators (BSCIs)* [5]:

- The icon that indicates the use of the SSL/TLS protocol (e.g., the padlock icon in the case of the Microsoft Internet Explorer).
- The certificate dialog that displays information about the server's certificate and the current status of the SSL/TLS connection.
- The location bar that displays the URL (including, for example, the prefix `https` standing for HTTP over SSL/TLS).
- A few menu items that can be used to display information about the status of the SSL/TLS connection.

Following [5], we use the term *visual spoofing* to refer to attacks that tamper with the BSCIs to fool the users about the status of their SSL/TLS connections (e.g., [3,6,7,8,9]). We consider visual spoofing as a powerful and very threatening attack that jeopardizes the security of most of today's Web applications. The details of these attacks are beyond the scope of this paper (a proof-of-concept can be found in [5]). Instead, we focus on mechanisms that can be employed to protect users against phishing and Web spoofing attacks.

To better understand why protection against phishing and Web spoofing is difficult, we look at the entities and components involved in a Web transaction.

- The *user* is the human entity who initiates the transaction.
- The *platform* is the client-side computer system employed by the user to initiate and perform the transaction. It consists of hardware and software (e.g., an operating system).
- The *browser* is the application software that is executed on the platform on behalf of the user to initiate and perform the transaction.
- The *Web server* (or *server* in short) is the server-side computer system that hosts the site and the resources that are requested by the browser.

In this setting, we are ultimately interested in a secure (i.e., authentic and private) channel between the user and the Web server. Such a channel may protect the user against phishing and Web spoofing. Unfortunately, all we have today is a supposedly secure channel between the browser and the Web server—using the SSL/TLS protocol. Note that this is an entirely different situation. Before the user can initiate this channel, he must convince himself (i) that the Web server is authentic, (ii) that the browser is authentic and not compromised in a way that it may leak secret information (e.g., authentication information), and (iii) that the platform is not compromised (otherwise it is not possible to establish a secure channel to the browser in the first place). The last point is the most critical one, and for all practical purposes one must assume that the platform is not compromised, has not been tampered with, and operates soundly.

To evaluate protection mechanisms against phishing and Web spoofing attacks, we need a classification of such attacks. We distinguish between five classes of attacks and corresponding attack levels. In either case, we assume an adversary who is powerful enough to passively and actively attack network communications. We define the following five attack levels:

Level 0: Attacks that implement native phishing as mentioned above.

Level 1: Attacks that implement classical phishing as reported in the media, i.e., the victim is directed to a Web site the adversary controls and the user is asked to reveal his credentials. Note that the adversary does not try to spoof an official Web site. Level 1 attacks are the most popular ones.

Level 2: Similar to attack level 1, except that the adversary tries to spoof an official Web server. The case is an aggravation of the prior one. The adversary imitates the official site's look-and-feel to mislead the user about the real connection.

Level 3: Similar to attack level 2. In this case, however, the adversary additionally employs visual spoofing to hide the attack. This means that the adversary can compromise the browser in some meaningful way. As mentioned above, this includes the capability of tampering an SSL/TLS-connection.

Level 4: Similar to attack level 3. In this case, however, the adversary can compromise the platform on which the browser executes. These attacks are very powerful, because they allow an adversary to install and execute key loggers, Trojan horses, and any other malicious software.

Attacks of levels 0 and 1 are comparably simple to detect even for casual users. Consequently, awareness and education programs may help to have users protect themselves against these attacks. Contrary to that, attacks of levels 2 and 3 are much more difficult to detect (sometimes even for the experienced and well-educated user). This is even more true for attacks of level 4.

3 Protection Mechanisms

In the past, several mechanisms have been developed that can be employed to partially protect users against phishing and Web spoofing attacks. The mechanisms may address the user, the platform, the browser, the Web server, or combinations thereof. For example, the user may be educated not to trust (and click on) anything received over the Internet and to use a different password for every Web site (if possible). Furthermore, the platform may be secured using best practices. This includes, for example, the use of firewalls and intrusion detection systems. For the purpose of this paper, we only address protection mechanisms that can be implemented on the browser or server side, or that can be implemented as a(n additional) interaction between the browser and the server. In the second case, the interaction must be specified and implemented as a protocol—that is either independent from other protocols or enhances an already existing protocol, such as the SSL/TLS protocol.

We mention that several authors have proposed heuristics to detect phishing and Web spoofing attacks (e.g., [10,5,11]). Heuristic protection mechanisms can be as simple as blacklists of known phishing sites (e.g., EarthLink's Toolbar[1] and GeoTrust's TrustWatch[2]) or take into account multiple rules (e.g., SpoofGuard[3]). Due to space limitations they are not addressed in this paper.

3.1 Browser-Side Protection Mechanisms

There are several mechanisms that can be employed on the browser side to (partially) protect users against phishing and Web spoofing attacks.

- In [12], the author proposes three modifications of the browser: First, the domain names of the Web sites being visited may be hashed, and the resulting hash value (or a prefix thereof) may be displayed by the browser in some appropriate way (e.g., as a two-character symbol). The aim is to make lexically close or homographic domain names look significantly different. Second, the browser may keep track of visited Web sites and notify the user if a new site is being visited (using, for example, some graphical warning sign). To defeat privacy concerns, the domain names of the visited sites may be concatenated with some user-specific random string (before they are hashed). Third, the browser may use heuristics to determine whether a site is suspicious (not addressed in this paper). The first two proposals are simple and effective to protect users against phishing and Web spoofing attacks up to level 2.
- In [3], the authors propose to configure the browser in a way that active Web scripting and programming languages (e.g., Java, JavaScript, and ActiveX) are deactivated. This proposal is effective to protect against phishing and Web spoofing attacks up to level 3; it is, however, neither complete nor practical.
- A user can protect himself against phishing and Web spoofing attacks, if he can properly authenticate the Web server. In theory, he can authenticate the server by verifying its SSL/TLS certificate. More specifically, the user must ensure that the certificate is valid, meaning that it is issued by a trusted certification authority (CA) and has not been revoked. Furthermore, he must compare the certificate's fingerprint with a reference value that is received out-of-band (e.g., published in print media). This is not a trivial task, and we propose that it can be simplified considerably by representing the hash value visually (e.g., [13,14,15]).
- Before the user provides his credentials, he must be sure that the browser is authentic in order to protect himself against visual spoofing attacks (level 3). If the platform supports trusted computing, then user has some certainty

[1] http://www.earthlink.net

[2] http://www.trustwatch.com/

[3] http://crypto.stanford.edu/SpoofGuard/

that the browser is indeed authentic. In all other cases, the user has to make sure in one way or another that the browser or components of its GUI (e.g., the BSCIs) have not been tampered with. There are a couple of proposals that protect users against attacks up to level 3.

- In [8], the authors recommend to prevent that the status bar is being deactivated by active Web languages. This proposal is simple and effective at protecting the user against some visual spoofing attacks (as it prevents deactivation of the padlock indicating a trustworthy SSL/TLS connection). It was recently implemented in Windows XP Service Pack 2, for instance.
- In [9], the authors propose to add a secure and tamper-resistant component called TrustBar[4] to the browser to visualize information about the Web site and the CA that issued the corresponding Web site's certificate. According to RFC 3709 [16], it is possible to include logotypes (commonly known as a *logo*, which is the graphical representation of a trademark or brand) in X.509 public key and attribute certificates. Consequently, the TrustBar renders the corresponding logotypes (of the Web site and CA) or display textual representations thereof. A fake site is divulged when a user does not recognize his corresponding visualization on the TrustBar. In addition, for unprotected Web sites, TrustBar displays a highly visible warning. We argue that if future browsers included a TrustBar (or something similar), then users would have a better way to judge and argue about the trustworthiness of Web sites.
- In [17], the authors introduce and propose the notion of synchronized random dynamic (SRD) boundaries. The idea is to distinguish between authentic parts of a browser's graphical user interface (GUI) and rendered content dynamically received from a Web server, and to make this distinction obvious by changing the boundary colour of the real GUI between two colors, blinking in synchrony with a trusted reference window. SRD boundaries are simple and effective, and they do not require any user interaction. However, they do not allow for modular verification of portions of a Web page. Furthermore, the modification of the browser required to implement SRD boundaries are not trivial. This is also true for the modification proposed in [8].
- In [5], the authors adopt an idea of [18] and suggest to authenticate the browser (or the BSCIs, respectively) by applying the concept of personalization with individually chosen background bitmaps.

The first, second, and fourth proposal are useful and worth considering (the third proposal is more complex than the fourth proposal, but achieves more or less the same level of protection).

In summary, our analysis implies that client-side protection mechanisms can effectively protect users against phishing and Web spoofing attacks up to level

[4] http://TrustBar.Mozdev.org

3. This is particularly true for protection mechanisms that employ tamper-resistant or personalized browser BSCIs. We recommend that these countermeasures should be adapted in future Web applications, which fail in authentically presenting the connection identifiers. As the visualization of these identifiers is not tamper-proof, the user is unable to distinguish between real and faked components of his user interface and, thus, susceptible to visual spoofing attacks.

3.2 Server-Side Protection Mechanisms

In addition to a proper authentication of the server during the execution of the SSL/TLS protocol, there are a few proposals and corresponding mechanisms that can be used on the server side to (partially) protect the users.

- If the login process is separated into two phases, then the user can enter his user ID in the first phase and his credentials in the second phase. Furthermore, the user can be taught to enter his credentials if and only if the second window is personalized in some meaningful way (e.g., by showing an image selected by the user).[5] We argue that such a mechanism protects the user against phishing and Web spoofing attacks up to level 2 because the adversary's Web site is unable to personalize the second window. However, we judge this mechanism to be ineffective to counteract attacks of higher level. The personalized window can be retrieved by anybody (simply by entering the appropriate user ID) and (mis)used to mount a visual spoofing attack. The only advantage we see is that an adversary must personalize the attack. Since this personalization can be automated, we don't see any real benefit.
- Another idea is implemented in GeoTrust's True Site seal. In short, the seal is a dynamically created "smart icon" that is placed on a Web site to make sure that the site is legitimate and authentic. The browser renders the seal and the user must actively validate it via a trusted party. This mechanism looks promising to protect users against phishing and Web spoofing attacks up to level 2. The major drawback is that the mechanism is passive (meaning that its validation must be initiated by the user), and hence the seal itself may be subject to spoofing attacks.

In summary, we conclude that server-side protection mechanisms are inappropriate for protecting users against this attack generation at all. The reason is lack of content protection or content secrecy, i.e., the adversary is generally able to perceive the same content as the user does. As a result, the adversary is able to imitate the content, i.e., he can copy and paste the Web site's look-and-feel (e.g., brand marks, logos and layout) and camouflage his attack by tampering with all BSCIs. There is no possibility to verify the authenticity and to realize its true origin with means standard Web browsers provide.

[5] Such a system has been developed and is being marketed by PassMark Security (cf. http://www.passmarksecurity.com).

3.3 Protection Mechanisms for the Interaction

There are a few proposals that affect the browser, the server, and the way they interact. The proposals are quite complex and are able to protect users against phishing and Web spoofing attacks up to level 2.

- It is useful to restrict the temporal validity of user credentials. This does not completely protect against phishing and Web spoofing attacks, it does, however, make sure that an adversary must operate in real-time.
- In [19], the authors propose a system in which a Web server can enrich its contents with HTML extensions called prooflets. Prooflets in turn can be verified by the browsers using special Web services. In theory, this proposal is appropriate and effective. In practice, however, this proposal has similar drawbacks as GeoTrust's True Site seal.
- In [4], the authors propose a technique called *Delayed Password Disclosure* (DPD) that protects a user executing a password-based mutual authentication protocol against the *doppelganger window attack* mentioned above. The technique is based on augmenting each user password with an easy-to-recognize sequence of images (that are specific to the user, the password, and the Web site). The user enters his password letter by letter, and for every letter he must recognize an image. The technique provides protection against phishing and Web spoofing attacks up to level 3.

In summary, we evaluate these proposals as helpful and effective to counteract level 3 attacks. Taken into account the current situation on the Internet and WWW, it is certainly time to move from user ID and password to more secure (mutual) authentication mechanisms. Particularly, the notion of prooflets and DPD look promising for the future.

4 Summary and Major Findings

As mentioned in Section 2, we are interested in mechanisms that are effective to protect users against phishing and Web spoofing attacks of level 2 and 3 (i.e., without or with visual spoofing attacks). Unless we do not enter the field of trusted computing, we assume that we are not able to protect users against phishing and Web spoofing attacks of level 4.

With regard to level 2, the proposals of [12] are effective and should be implemented on the browser side. Furthermore, we opine that server authentication must be improved. The visual representation of certificate fingerprints is certainly something that should be considered first (since it does not require infrastructural changes). Similarly, systems like TrustBar and True Site are useful to make the notion of a public key certificate and the entire certification process more transparent to the user. However, more research is needed with regard to the usefulness and user acceptance of these mechanisms.

With regard to level 3, the proposal of [3] is by far the most simple and effective protection mechanism. Unfortunately, it is impractical, and hence we must evaluate alternative mechanisms. Making the BSCIs as tamper resistant as possible and personalizing them are certainly good ideas that should be implemented. In particular, the personalization paradigm has been largely ignored by the browser manufacturers. We argue that it is about time to change this and that future versions of browsers should incorporate features that allow users to personalize the BSCIs. Similarly, browser manufacturers should include features that allow users to realize that they have connected to a Web site for the very first time (especially if the connection is secured with the SSL/TLS protocol). The Petname tool[6] for the Firefox browser is a good example. This does not help in the Internet café scenario; it does help, however, in the home PC scenario.

5 Research Challenges

In [20], the author proposes a possibility to improve the security of TAN lists in a way that the user is protected against an adversary reading out the list and misusing the TANs, as well as some simple phishing attacks. The idea is to protect the TAN list in a way that reading a TAN requires a physical act that can be detected easily at some later point in time. One possibility is to use a physical layer that hides the TANs and can easily be rasped away by the user (similar to lots in some lotteries). Furthermore, an authentication code or a prefix thereof can be printed on the physical layer, and the user can be taught to verify the code before he rasps away the physical layer. Alternatively, the code can also be covered by a physical layer that must be rasped away. The corresponding TAN lists are securely delivered, mainly because they are distributed out-of-band (i.e., using an out-of-band distribution channel). This can be simulated, for example, using the short messaging service (SMS) of GSM networks. An interesting research challenge is to find a similar mechanism that does not require an out-of-band distribution mechanism. One possibility is to combine a challenge-response mechanism with a non-trivial redundancy scheme that allows a browser to verify the authenticity of a challenge. Note that this does not protect against adversaries that operate in real-time (i.e., the adversary can simply act as as a relay between the origin server and the user). Consequently, an important research challenge is to find technologies and mechanisms that are able to protect users against adaptive adversaries that operate in real-time. There are many applications (e.g., Internet banking) that could take advantage of such technologies and mechanisms.

6 Conclusions

In this paper, we summarized, discussed, and evaluated the effectiveness of mechanisms to protect users against (large-scale) phishing and Web spoofing attacks. Some mechanisms are simple and effective and should be implemented immediately. This is particularly true for browser-side mechanisms. Server-side mechanisms seem to be advantageous, mainly because they are simpler to deploy but

[6] http://www.waterken.com/user/PetnameTool

are also more challenging to design and come up with. In fact, we have found no server-side protection mechanism that can be used to effectively protect users against phishing and Web spoofing attacks. The protection mechanisms for the interaction look promising but still require more analysis. More surprisingly, there is a simple and reasonable secure protection mechanism against many relevant phishing and Web spoofing attacks that employ plain old TAN lists.

References

1. Wagner, D., Schneier, B.: Analysis of the SSL 3.0 Protocol. In: USENIX Security Symposium. (1996) 29–40
2. Clayton, R.: Insecure Real-World Authentication Protocols (or Why Phishing is so Profitable). In: Financial Cryptography. (2005)
3. Felten, W.E., Balfanz, D., Dean, D., Wallach, D.S.: Web Spoofing: An Internet Con Game. Technical Report 540-96, Dept. of Computer Science, Princeton University (1996)
4. Jakobsson, M., Myers, S.: Stealth Attacks and Delayed Password Disclosure. (2005)
5. Adelsbach, A., Gajek, S., Schwenk, J.: Visual Spoofing of SSL Protected Web Sites and Effective Countermeasures. In: Information Security Practice and Experience Conference. (2005)
6. De Paoli, F., DosSantos, A., Kemmerer, R.: Vulnerability of 'Secure' Web Browsers. In: National Information Systems Security Conference. (1997)
7. Lefranc, S., Naccache, D.: Cut-&-Paste Attacks with JAVA. In: ICISC. (2002) 1–15
8. Li, T.Y., Wu, Y.: Trust on Web Browser: Attack vs. Defense. In: ACNS. (2003) 241–253
9. Herzberg, A., Gbara, A.: TrustBar: Protecting (even Naive) Web Users from Spoofing and Phishing Attacks. IACR Cryptology ePrint Archive (2004)
10. Chou, N., Ledesma, R., Teraguchi, Y., Mitchell, J.C.: Client-Side Defense Against Web-Based Identity Theft. In: NDSS. (2004)
11. Jakobbson, M.: Modeling and Preventing Phishing Attacks. In: Financial Cryptography. (2005)
12. Markham, G.: Phishing-Browser-based Defences. (2005) http://www.gerv.net/security/phishing-browser-defences.html#ssl-essential.
13. Perrig, A., Song, D.: Hash visualization: A new technique to improve real-world security. In: Cryptographic Techniques and E-Commerce. (1999)
14. Perrig, A., Dhamija, R.: Déjà Vu: A User Study Using Images for Authentication. In: USENIX Security Symposium. (2000)
15. Dohrmann, S., Ellison, C.: Public key support for collaborative work. In: PKI Research Workshop. (2002)
16. Santesson, S., Housley, R., Freeman, T.: Internet X.509 Public Key Infrastructure: Logotypes in X.509 Certificates. (2004) Request for Comments 3709.
17. Ye, Z.E., Smith, S.: Trusted Paths for Browsers. In: USENIX Security Symposium. (2002) 263–279
18. Tygar, J., Whitten, A.: WWW Electronic Commerce and Trojan Horses. In: USENIX Workshop on Electronic Commerce. (1996)
19. Shin, M., Straub, C., Tamassia, R., Polivy, D.: Authenticating Web content with Prooflets. Technical report, Brown University, Center for Geometric Computing (2002)
20. Oppliger, R.: Sichere Streichlisten. digma **5** (2005) 34–35