Improving the Usability of Web Browser Security

Haidong Xia and José Carlos Brustoloni
Department of Computer Science, University of Pittsburgh
{hdxia,jcb}@cs.pitt.edu

ABSTRACT

Existing Web browsers handle security errors in a manner that often confuses users. In particular, when a user visits a secure site whose certificate the browser cannot verify, the browser typically allows the user to view and install the certificate and connect to the site despite the verification failure. However, few users understand the risk of man-in-the-middle attacks and the principles behind certificate-based authentication. We propose context-sensitive certificate verification (CSCV), whereby the browser interrogates the user about the context in which a certificate verification error occurs. Considering the context, the browser then guides the user in handling and possibly overcoming the security error. We also propose specific password warnings (SPW) when users are about to send passwords in a form vulnerable to eavesdropping. We performed user studies to evaluate CSCV and SPW. Our results suggest that CSCV and SPW can greatly improve Web browsing security and are easy to use even without training. Moreover, CSCV had greater impact than did staged security training.

1. INTRODUCTION

The technology for securing Web applications is generally thought to be well understood. Secure Web sites use HTTPS, which layers HTTP over SSL or its standard equivalent, TLS. SSL in turn uses cryptographic algorithms, such as SHA-1 and AES, for guaranteeing communication authenticity and confidentiality.

The usability of this technology has, however, received surprisingly little attention in the literature. We study in this paper three related questions. First, how likely is it that an attack against computer-literate users of existing browsers will succeed, in representative security-sensitive Web applications? We consider only attacks facilitated by tools that can be freely downloaded from the Internet, e.g. eavesdropping (`ethereal`) and man-in-the-middle (MITM) attacks (`arpspoof, dnsspoof, webmitm`). Second, is it possible to make Web browsers more foolproof, such that untrained users employ them more securely? Third, can user education improve how securely users employ existing browsers?

We performed a user study to answer the first question. Our results show that, with current users and Web browsers, the mentioned attacks are alarmingly likely to succeed. More often than not, users' behavior defeats the existing Web security mechanisms.

In response to the second question, we contribute two novel user interface techniques for Web browsers, CSCV (Context-Sensitive Certificate Verification) and SPW (Specific Password Warnings). CSCV's goal is to thwart MITM attacks against HTTPS and other protocols that use certificates to authenticate servers. SPW cautions users against sending passwords in a form vulnerable to eavesdropping. We implemented CSCV and SPW in a Web browser and evaluated them in a second user study, involving the same users and attacks as the first study. CSCV blocked MITM attacks against HTTPS-based applications completely. SPW greatly reduced the insecure transmission of passwords in an HTTP-based application. Although untrained, users had little trouble using CSCV and SPW. These results suggest that, at least for some security tasks, it is indeed possible to design user interfaces that are less error-prone for untrained users.

To answer the third question, we trained users from the first study on security principles, attacks, and tools. We then repeated the experiment using unmodified browsers. Our results show that education can indeed greatly improve how securely users behave. However, security education had significantly less impact than did CSCV and had about the same impact as did SPW.

2. CERTIFICATE VERIFICATION IN THEORY AND IN PRACTICE

In theory, eavesdropping and MITM attacks against secure Web servers would not be possible. Such servers use HTTPS. HTTPS employs a certificate to authenticate the Web server to the client's browser. Browsers typically come preconfigured with public keys of major certifying authorities (CAs, e.g., Verisign). Such keys enable browsers to authenticate certificates issued by those CAs, thwarting MITM attacks. After server authentication, HTTPS can use strong algorithms for authenticating and encrypting data packets sent between client and server.

However, the current state of public-key infrastructure (PKI) deployment is such that browsers frequently encounter certificates that they cannot verify. In such circumstances, browsers typically display a warning to the user, asking if the user wants to continue anyway. By giving users this override ability, browsers enable MITM attacks, despite HTTPS.

Certificate verification can fail for a variety of reasons. First, the browser may not know the public key of the CA that issued a server's certificate. If the accessed server is intended only for members of the organization that owns the server, this failure is very common and not indicative of an attack: many organizations have *private* CAs that issue certificates for internal servers. Such certificates are easier and less costly to obtain than are those issued by major CAs, but they require the public key of the private CA to be installed in all clients – a chore that is often neglected. On the other hand, for servers open to the public, this type of failure could very well be result of a MITM attack. Second, the certificate may have expired. This failure may result from inattention and is not suggestive of a MITM attack. Third, the certificate may be for a server whose name differs from that which the user wishes to visit. Discrepancies at the subdomain level may result from simple server reorganization, and not an attack. On the other hand, if the domains differ, the possibility of a MITM attack is high.

		Study 1: 17 users			Study 2: 17 users			Study 3: 12 users		
Browser		Unmodified Internet Explorer			Modified Mozilla Firebird			Unmodified Internet Explorer		
UI methodology		Warn and continue (WC)			CSCV		SPW	Well-in-advance instruction		
Site type		UC/M	UC/C	NC	UC/M	UC/C	NC	UC/M	UC/C	NC
Score frequency	0	17	16	11	0	0	2	6	5	2
	50	0	–	–	1	–	–	2	–	–
	100	0	1	6	16	17	15	4	7	10
Average Score		0	6	35	97	100	88	42	58	83

Table 1: Users' security scores when accessing sites of various types. CSCV and SPW greatly improved security scores of untrained users on sites of all types. The effect of security training was smaller than that of CSCV, but similar to that of SPW. "UC/M" represents site with unverified certificate and belonging to organization user is member of, "UC/C" is site with unverified certificate and belonging to organization user is simply a client of, and "NC" is site without certificate (no SSL).

3. CONTEXT-SENSITIVE CERTIFICATE VERIFICATION

When certificate verification fails because the public key of the certificate's issuer is unknown, CSCV-aware Web browsers ask whether the user has the necessary security information on removable media. CSCV-aware private CAs give to the respective organization's members tokens containing the CA's self-signed certificate with the CA's public key. The tokens are distributed to members securely out-of-band on removable media, such as USB keys.

If the user does not have the CA's certificate, CSCV-aware browsers ask whether the user is an internal member (e.g., student or employee) of the organization that owns the server the user wishes to access. If so, the browser displays the CA's contact information and tells the user how to verify the contact information and the administrator. CSCV-aware private CAs include the CA's contact information in the *issuer alternative name* field of server certificates they issue. This field typically would otherwise be unused. Users thus learn how to obtain, securely and out-of-band, the private CA's self-signed certificate. After installing the certificate, users can authenticate the organization's servers securely, without resorting to overrides.

On the other hand, if the user is simply a client of the organization that owns the server, CSCV-aware browsers warn the user that the situation is unusual, and instruct the user how to (try to) obtain the CA's certificate. The user cannot connect to the server without first obtaining the CA's certificate and authenticating the server.

4. SPECIFIC PASSWORD WARNINGS

Existing Web browsers can be configured to warn users when they are about to send unencrypted data. Because these warnings do not discriminate between data and passwords, they may occur quite often. Therefore, many users disable or ignore such warnings.

On the contrary, SPW-aware browsers detect when a user is about to transmit a password in plaintext, and ask the user whether the password protects an account that the user wouldn't want strangers to access. If so, the browser strongly discourages the user from continuing. The browser informs that such accounts should be accessed securely and explains simply how the user can tell whether a site is being accessed securely. The browser also asks the user to consider whether the current server is an insecure replica of a server that the client normally accesses securely, and explains in plain language that such a replica could be used in a MITM attack. The browser cautions the user that an eavesdropper on the network may be able to capture the user's password and later impersonate the user, with possibly significant financial or privacy loss to the user. Finally, the browser asks whether the user is willing to accept all the mentioned risks.

5. EVALUATION

We performed three user studies to evaluate CSCV and SPW. The users in the first two studies were 17 male Computer Science undergraduates. The first study provides a baseline before any browser modifications or training. In it, students used the Internet Explorer (IE) browser. In the second study, performed back-to-back with the first one, the same users employed a modified version of the Mozilla Firebird browser with CSCV and SPW. No feedback or further information was given to students between the first two studies. The second study served also as the first stage of the third study. Twelve of the original seventeen students then received a month's training on certificates and MITM and eavesdropping attacks. Results are for the study's final stage, where students again used unmodified IE.

In each user study, we asked students to visit three Web sites where students were assigned password-protected accounts. The first site is maintained by the students' university. It allows students to monitor the respective *reward points*. Students earn these points, e.g., by doing well in exams. The second site is maintained by a remote e-merchant that is not affiliated with the university and where students can spend their reward points, e.g., to buy books or CDs. The third site provides access to users' Web email accounts. We configured the first two sites with HTTPS and server certificates issued by private CAs whose public key was unknown by client software. The first site's CA contact information was that of a real person in the same building. The second site's certificate was bogus. We configured the third site with HTTP only (no SSL). We asked students to verify their balance on the first site, spend some of it by ordering something from the second site, and get an order confirmation message on the third site.

We scored how securely users accessed the sites as follows. If a user accessed a site despite lack of security, the user got 0 points. In the first site, if a user simply did not visit the site insecurely, the user got 50 points. If the user also correctly obtained and installed the issuing CA's certificate and thus accessed the server after authenticating the server, the user got 100 points. Lack of security in the second and third sites could not be corrected. Thus, users who simply did not visit each site insecurely got 100 points. The students' security scores are represented on Table 1.

We further discuss these results and related work in the full version of this paper, which is available online [1].

6. REFERENCES

[1] Xia, H. and Brustoloni, J.: Hardening Web Browsers Against Man-in-the-Middle and Eavesdropping Attacks. In *Proc. WWW'2005*, W3C/ACM, May 2005, pp. 489-497. [Online] `http://www2005.org/cdrom/docs/p489.pdf`