Protecting Ecommerce Against The Man-In-The-Middle

Rolf Oppliger, Ralf Hauser and David Basin

Here's why SSL and TLS are not as secure as you might think.

Editor's note: See the accompanying box for References shown in brackets.

Most ecommerce applications employ the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols [DR06] to authenticate the server to the client and to cryptographically protect the communication channel between them. It is widely and wrongly believed that these protocols are sufficient to protect Web-based ecommerce applications against man-in-the-middle (MITM) attacks.

In an MITM attack, a third party typically "spoofs" or pretends to be the server, to fool the client. End users can be taken in by well-designed emails (phishing) and websites that look authentic (visual spoofing). Theft or forgery can result.

One of the most prominent examples of a successful MITM attack occurred in July 2006, when a third party was able to defeat the two-factor user authentication at Citibank's Citibusiness site. The attackers collected the usernames and password combinations for users, as well as their one-time passwords (OTP). (The story is available at http://blog.washingtonpost.com/security-fix/2006/07/citibank_phish_spoofs_2factor_1.html)

More recently, an AT&T DSL vendor site was hacked by phishers, who sent email to DSL customers saying their credit cards couldn't be charged because their bank didn't have sufficient information. The emails were very believable because they included the customers' order numbers, the last 4 digits of their credit card numbers and other personal information. Recipients were directed to a fake website where they were supposed to "confirm" their order by "updating" their credit card info by supplying more information, including their Social Security number and birthdate. (The story is available at http://sfgate.com/cgi-bin/article.cgi?f=/c/a/2006/09/01/BUGVBKSUIE1.DTL)

Rolf Oppliger, PhD, is the founder and owner of eSECURITY Technologies (www.esecurity.ch), a Swiss-based company that provides information security consulting, education, and engineering services. He can be reached at rolf.oppliger@esecurity.ch.

Ralf Hauser, PhD, is the founder and lead architect of PrivaSphere (www.privasphere.com), a Swiss-based company that provides email and ecommerce security services. He can be reached at hauser@privasphere.com.

David Basin, PhD, is a full professor and has the chair for Information Security at the Department of Computer Science at ETH Zurich (www.inf.ethz.ch). He is also the director of the Zurich Information Security Center (ZISC). He can be reached at basin@inf.ethz.ch.

FIGURE 1 A Typical Man-In-The-Middle Attack

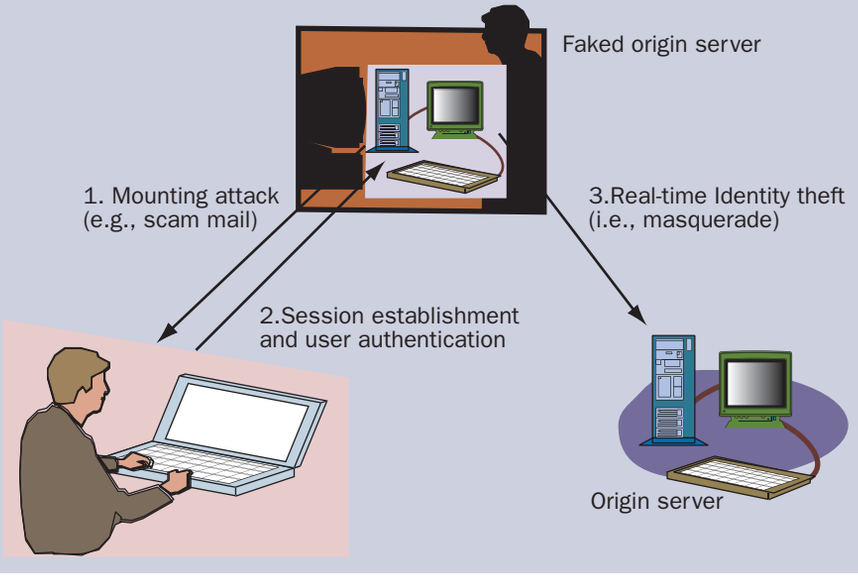
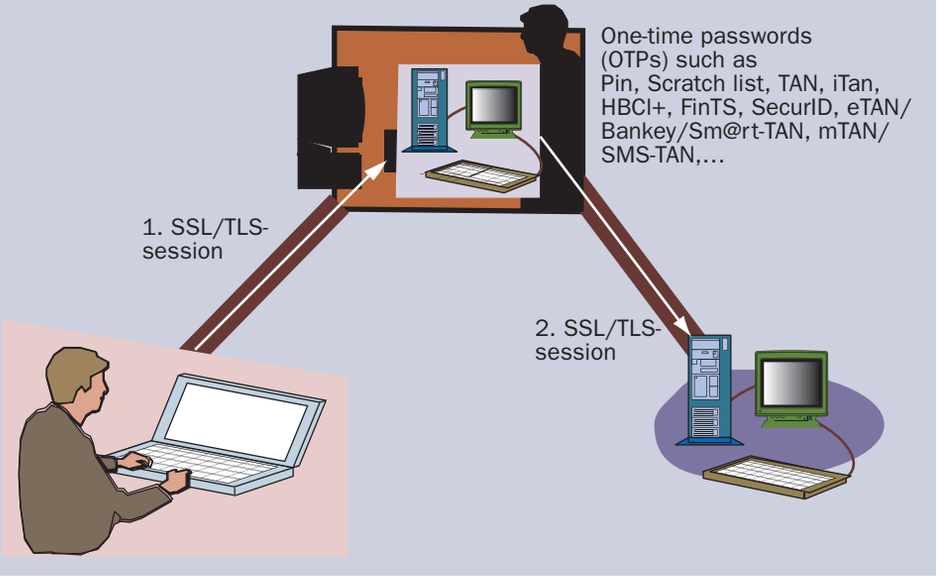


FIGURE 2 MITM Over SSL/TLS



Users ignore pop-up security warnings, they answer phishing emails, and they visit fake websites

The aspect of SSL and TLS that could protect against MITM is the use of the public key infrastructure (PKI), but PKI is expensive and complex and PKI pop-up windows warning of certificate issues are routinely ignored on the client side. Apart from online banking, few institutions have fully recognized the seriousness of MITM threats. Those who acknowledge the danger tend to declare that client-side certificates will be their "strategic" security mechanism. Often, however, the cost and complexity of PKI end up preventing successful, large-scale deployments.

In lieu of PKI, most SSL/TLS-based ecommerce applications employ traditional user authentication mechanisms—passwords, personal identification numbers (PINs), transaction authorization numbers (TANs), and scratch lists, as well as more sophisticated authentication systems, such as OTP or challenge-response (C/R) systems. With respect to MITM attacks, these mechanisms are less secure than PKI.

Sadly, even when the server side uses PKI certificates—and the SSL/TLS protocol requires a certificate on the server side—the end users on the client side don't play their parts. They ignore pop-up warnings about the status of the server's PKI certificates, they respond to phishing emails and they click on and interact with websites that are obviously fakes.

In this article, we elaborate on crucial aspects of MITM attacks and we survey the proposed countermeasures, including our preferred

References

[ANN03]—Asokan, N., Niemi, V., and K. Nyberg, "Man-in-the-Middle in Tunneled Authentication Protocols," *Proc. International Workshop on Security Protocols*, 2003, pp. 15–24.

[ASS03]—Alkassar, A., Stüble, C., and A.-R. Sadeghi, "Secure Object Identification—or: Solving The Chess Grandmaster Problem," *Proc. Workshop on New Security Paradigms*, ACM Press, NY, 2003, pp. 77–85.

[BH06]—Badra, M., and I. Hajjeh, "Key-Exchange Authentication Using Shared Secrets," IEEE Computer, Vol. 39, Issue 3, March 2006, pp. 58–66.

[BM94]—Bellovin, S.M., and M. Merritt, "An Attack on the Interlock Protocol When Used for Authentication," *IEEE Transactions on Information Theory*, Vol. 40, No. 1, January 1994.

[Bur02]—Burkholder, P., "SSL Man-in-the-Middle Attacks," *SANS Reading Room*, February 2002.

[DR06]—Dierks, T., and E. Rescorla, The TLS Protocol Version 1.1, RFC 4346, April 2006.

[ET+05]—Eronen, P., and H. Tschofenig (Eds.), "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)," RFC 4279, December 2005

[O+07]—Oppliger, R., et al., "A Proof of Concept Implementation of SSL/TLS Session-Aware User Authentication," *Proc. Kommunikation in Verteilten Systemen* (KiVS 2007), Springer-Verlag, 2007 (to appear).

[OG05]—Oppliger, R., and S. Gajek, "Effective Protection Against Phishing and Web Spoofing," *Proc. Conference on Communications and Multimedia Security* (CMS 2005), Springer-Verlag, LNCS 3677, September 2005, pp. 32–41.

[OHB06]—Oppliger, R., Hauser, R., and D. Basin, "SSL/TLS Session-Aware User Authentication—Or How to Effectively Thwart the Man-in-the-Middle," *Computer Communications*, Vol. 29, Issue 12, August 2006, pp. 2,238–2,246.

[PKP06]—Parno, B., Kuo, C., and A. Perrig, "Phoolproof Phishing Prevention," *Proc. Financial Cryptography and Data Securit*, Springer-Verlag, 2006.

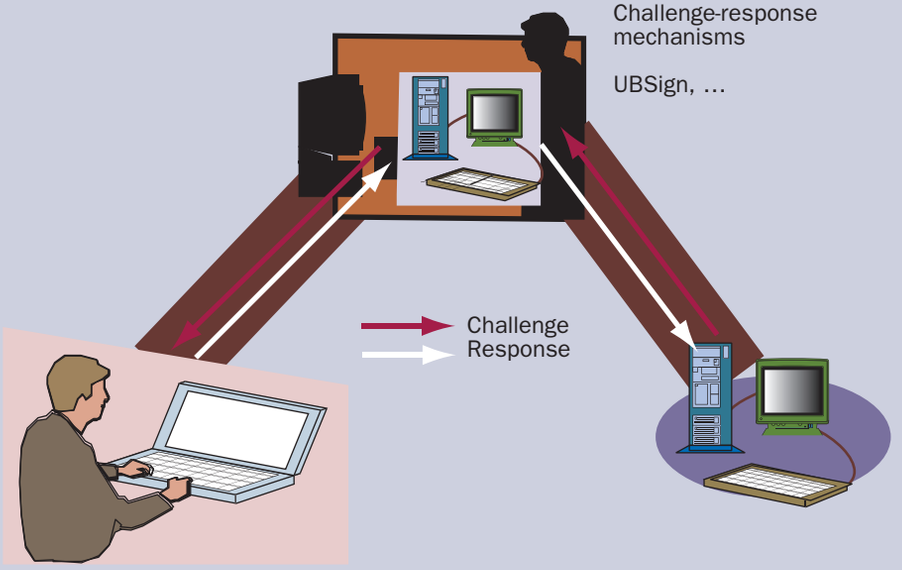
[RS84]—Rivest, R.L., and A. Shamir, "How to Expose an Eavesdropper," *Communications of the ACM*, Vol. 27, No. 4, 1984, pp. 393–395.

[S+01]—Steiner, M., et al., "Secure Password-Based Cipher Suite for TLS," *ACM Transactions on Information and System Security* (TISSEC), Vol. 4, No. 2, May 2001, pp. 134–157.

[T+05]—Taylor, D., *et al.*, "Using SRP for TLS Authentication," work in progress, October 2005□

If users properly authenticated the servers, they would be protected

FIGURE 3 MITM And Challenge/Response



solution, session-aware user authentication (TLS-SA). We do not sell TLS-SA equipment or software at this time, but we have coordinated several proof-of-concept deployments, and we have proposed TLS-SA to the World Wide Web Consortium (W3C). We will also be presenting TLS-SA [O+07] at the Kommunikation in Verteilten Systemen conference (KiVS '07) to be held in Berne, Switzerland during February/March 2007. (Further information about the conference is available at http://kivs07.unibe.ch/).

MITM Attacks

MITM attacks are active attacks that target the association between communicating entities (rather than the entities themselves or the communication channel). In a typical setting, the MITM places himself between the user and the server such that he can talk to the user and the server separately, while the user and the server think that they are talking directly with each other. Examples are shown in Figures 1, 2 and 3.

There are many possibilities for mounting MITM attacks. A user may be directed to the MITM using phishing techniques, as discussed above, or by various so-called pharming or cache poisoning techniques. In these cases, erroneous information is implanted in the end user's machine or on the Internet's DNS servers so that users are directed to fake websites even when they type in the correct addresses.

MITM attackers can function like any other SSL/TLS proxy server; neither the user nor the server is aware of the proxy. Once the MITM is in the loop, cryptography doesn't make any difference, since the MITM can decrypt and reencrypt all messages on the fly. In fact, many firewalls and content screening proxies work this way (e.g., Cybertrust, Microdasys and WatchGuard).

MITM attacks are very powerful and may have devastating effects. For example, if a user authenticates himself to an application server, then he also inadvertently reveals his credentials to the MITM. Afterwards, the MITM can misuse the credentials to impersonate (or spoof) the user. If the user employs a one-time password system, the MITM can grab the OTP (which is typically valid for at least a few seconds) and reuse.

If the user employs a challenge/response system, again, the MITM can simply send back and forth the challenge and response messages. Even zero-knowledge authentication protocols cannot alone provide protection against MITM attacks—they can only protect against the leakage of the user's secret. (Wikipedia defines zero knowledge authentication as an interactive method for one party, the prover, to prove to another party, the verifier, that it knows a value for a password, without revealing anything to the verifier other than the fact that the prover knows that password.)

Countermeasures

There are only a few countermeasures to thwart MITM attacks against SSL/TLS-based ecommerce applications. Most of them were developed, as was SSL/TLS, before the advent of phishing, pharming and poisoning techniques. For example, security specialists have assumed for years that two-factor user authentication would prevent MITM (among other problems), but recent attacks have proved otherwise.

The truth is that vulnerability to MITM attacks is not really a user authentication problem—it is a server authentication problem. In other words: If users properly authenticated the server with which they establish an SSL/TLS session, then they would be protected against MITM attacks.

Unfortunately, this is not the case and it is

questionable whether it is possible at all. An MITM can employ many tricks to give the user the impression of being connected to the right server (e.g., visual spoofing). In the most extreme case, an MITM could even control the graphical user interface of the user's browser.

To make things worse, we have seen phishing sites that use valid certificates. Consider the case in which a phisher employed a valid certificate for www.mountain-america.com and www.mountain-america.net to spoof the website of the Mountain America Credit Union (www.mtnamerica.org). In such a setting, most users are unable to recognize that they are subject to an attack.

Against this background, it is important to emphasize that the SSL/TLS protocol can protect against MITM attacks, but only if clients and servers are equipped with public key certificates, and if they actually authenticate to one another using those certs.

People have been exploring ways to relax this requirement, by extending the SSL/TLS protocols with some alternative client authentication methods. One solution, verifying a preshared secret, can be easily automated and is much more intuitive for the user. The IETF is working to specify ciphersuites for the TLS protocol that will support authentication based on pre-shared secret keys (e.g., [S+01, ET+05, T+05, BH06]). We think that these efforts are very important, but that there is still a long way to go before these TLS extensions are available, implemented and widely deployed.

In practice, a growing number of ecommerce applications—especially in Europe—authenticate users by sending out SMS messages that contain verification codes that users must enter when they login. Sending out SMS messages is an example of using two communication channels for two-factor authentication (the mobile phone being the second factor). While it is sometimes argued that this mechanism protects against MITM attacks, unfortunately, this is not the case. If an MITM is located between the user and the server, then he need not eavesdrop on the SMS messages; all he needs to do to spoof the user is to forward the verification code submitted by the user on the SSL/TLS session to the origin server.

If one wanted to use verification codes distributed via SMS messages, then it would be better to use them on a per-transaction basis. For every transaction submitted by the user, a summary must be returned to the user, together with an appropriate code in an SMS message. To confirm the transaction, the user must then enter the code.

This approach has several downsides, notably its expense. Moreover, it is not particularly user-friendly, and it is not even completely secure—an MITM can still attack the parts of a transaction that are not part of the verification code. Furthermore, care must be taken so that an adversary cannot substitute the user's mobile phone number with one that he controls.

A few cryptographic techniques and protocols also have been proposed over the years to protect against MITM attacks:

- Rivest and Shamir proposed the Interlock protocol [RS84], although it was later shown to be vulnerable when used for authentication [BM94].
- Jakobsson and Myers proposed a technique called delayed password disclosure (DPD, described at www.informatics.indiana.edu/markus/stealth-attacks.htm). DPD can be used to complement a password-based authentication and key exchange protocol to protect against a special form of MITM attack—called the doppelganger window attack. It does work for MITM attacks that employ simple pop-up windows.

Unfortunately, DPD requires a password-based authentication and key exchange protocol and does not protect against an MITM that controls the browser's user interface to some extent (using, for example, visual spoofing).

- Kaliski and Nyström proposed the use of a password protection module (PPM, described at www.rsasecurity.com/rsalabs/staff/bios/bkaliski/publications/other/kaliski-authentication-risk-readiness-bits-2004.ppt) to provide protection against MITM attacks. The PPM is a trusted piece of software that uses password hashing to generate a passcode that is unique for a user and an application in question. Again, PPMs do not provide protection against an MITM that controls the browser's user interface. Moreover, PPMs appear difficult to deploy and manage in practice.
- Parno *et al*, proposed the Phoolproof anti-phishing system that works by employing a trusted device, such as a mobile phone, to thwart MITM attacks [PKP06].
- Asokan *et al*, proposed protection mechanisms to secure extensible authentication protocol (EAP) tunneled authentication protocols against MITM attacks in wireless networks [ANN03].

Some researchers hold out hope that, if the quality of the PKI certificate experience were improved, or if the end user were more definitely notified of a variance between the address they are seeking and the one being offered, then users would not visit fake websites or otherwise participate in MITM attacks. But experience shows that users are simply not willing to pay any more attention to addresses or pop-up windows than they must in order to get to where (they think) they are going on the Web.

Instead of cryptographic techniques and protocols, or modifying the user's browser experience, some researchers have also suggested employing multiple communication channels and channel-hopping to thwart MITM attacks (e.g., [ASS03]). However, establishing multiple channels increases complexity and reduces performance and we think that it is impractical for most Internet-based applications in use today. Most people worry less about MITM than they would worry about a major change in standard Internet protocols.

Two-factor authentication has failed to stop recent MITM attacks

The server's session state could be used in the UAC to thwart MITM attacks

In summary, all countermeasures proposed so far either do not adequately solve the problem or have severe disadvantages. At least that is the conclusion we came to and the motivation behind our proposed solution.

SSL/TLS Session-Aware User Authentication

Recall that most user authentication mechanisms fail to provide protection against MITM attacks, even when they run on top of the SSL/TLS. That is because SSL/TLS itself is not broken—the problem is in the way that SSL/TLS is being employed (or trusted) by the application and the way it is interfaced to the user.

As discussed above, we see two main reasons for this. First, the user is not properly authenticating the server, so when the user "talks" to the MITM, s/he reveals his/her credentials. Second, because the SSL/TLS session establishment is usually decoupled from user authentication, the user's credentials can be reused by the MITM to spoof the user to the origin server.

An effective countermeasure against MITM attacks in an SSL/TLS setting must address these problems either

1.) by enforcing proper server authentication by the user, or

2.) by combining user authentication to the application with SSL/TLS session establishment.

The first possibility we have already discounted above as unworkable, because it requires changes in user behavior and hard-coded server certificates and/or dedicated client software (e.g., actual PKI deployments). In contrast, the second possibility requires modifications to SSL/TLS or to the authentication protocol(s) in use.

Our focus is on the second approach, which we call SSL/TLS session-aware user authentication (TLS-SA). The main idea is to make the user's authentication depend not only on the user's (secret) credentials, but also on state information related to the SSL/TLS session in which the credentials are transferred to the server.

The rationale behind this idea is that the server should be able to determine whether the SSL/TLS session in which it receives the credentials is the same session as the one the user employed when he sent out the credentials in the first place. If the two sessions are the same, then there is probably no MITM involved. If the two sessions are different, then something abnormal is taking place, and it is likely that an MITM is located between the user's client system and the server.

With TLS-SA, the user authenticates himself or herself by providing a user authentication code (UAC) that now depends on both his/her credentials and on information in the SSL/TLS session state, which is cryptographically hard to alter. For example, one may apply a one-way hash function taking both the UAC and state information as input; a number of concrete solutions in this area are given in [OHB06].

The key point is that an MITM who gets hold of the UAC can no longer misuse it by simply retransmitting it, because the server will immediately see the UAC has not been issued for the SSL/TLS session on which it is received. If the UAC is submitted on a different session, then the server can detect this fact and drop the session accordingly.

There are many possibilities to implement TLS-SA as hardware or software tokens. In either case, a token can be personal or impersonal, and it may conform to a cryptographic token interface standard, such as Public Key Cryptography Standard #11 (PKCS #11) or Microsoft Crypto API (MS-CAPI).

Our basic approach employs impersonal authentication hardware tokens that users can plug into their client systems [OHB06]. Such tokens provide simple and straightforward support for TLS-SA (as well as for OTP and C/R systems).

We also have solutions for making all widely deployed SSL/TLS authentication mechanisms session-aware. This includes, for example, the Chip Authentication Program (CAP) implemented on Eurocard, MasterCard, and Visa (EMV) cards. More recently, we have built a proof-of-concept implementation of TLS-SA [O+07]. The implementation employs C/R tokens and a plug-in for Microsoft Internet Explorer on the client side, as well as a modified Web portal infrastructure on the server side.

Conclusions

In conclusion, we believe that TLS-SA solves the problems outlined while leveraging the legacy authentication systems that the masses have become accustomed to using. Most importantly, TLS-SA protects against MITM attacks by providing a lightweight alternative to the use of public key certificates on the client side.

Bear in mind however, that any device which functions above the transport layer will be detected as a man-in-the-middle. Consequently, any technology that protects against MITM attacks also prohibits the use of SSL/TLS proxy servers□

Companies Mentioned In This Article

AT&T (http://att.sbc.com/)

Cybertrust (www.cybertrust.com)

Eurocard (www.mastercard.com)

Mastercard (www.mastercard.com)

Microdasys (www.microdasys.com)

Microsoft (www.microsoft.com)

Mountain America Credit Union (www.mtnamerica.org)

Visa (www.usa.visa.com)

WatchGuard (www.watchguard.com)