

The Role of Authentication Tokens in Preventing Man-in-the-Middle Attacks

Security threats and potential breaches can stem from a wide variety of vulnerabilities, ranging from simple password theft or spyware to Trojan horses, keyword sniffers and more. But the tactic that combines high levels of deception, great potential risk of loss and broad distribution is a new form of “man-in-the-middle” attack—real-time phishing. Man-in-the-middle attacks are not new—they’ve been used in various contexts (such as password snatching) for years. However, combined with social engineering attacks and web spyware, the man-in-the-middle attack is now a more powerful type of phishing.

CONTENTS

I. THE MAN IN THE MIDDLE IS NOT HERE TO HELP YOU	PAGE 1
II. THE EMERGENCE OF ONLINE PHISHING	PAGE 1
III. SIGNING—NOT THE ANSWER	PAGE 2
IV. CONTEXTUAL OVERWRITING	PAGE 2
V. CONTEXTUAL REORDERING	PAGE 3
VI. THE NEED FOR SITE VERIFICATION	PAGE 3
Host Authentication	PAGE 3
Browser Plug-In	PAGE 3
Trusted Path	PAGE 3
CONCLUSION	PAGE 3

I. THE MAN IN THE MIDDLE IS NOT HERE TO HELP YOU.

With so-called offline phishing, the attacker sends out a realistic-looking e-mail to thousands of people that purports to be from a legitimate, established enterprise (such as a financial-services provider or online auction site). That e-mail typically asks recipients to click on a link to update personal information.

However, the link in the e-mail doesn't direct the user to a site hosted by the legitimate enterprise. It uses embedded links to surreptitiously direct the user to a site that is instead merely a bogus site that asks for and collects passwords, credit-card numbers and other identity information that the attacker then sells or uses to execute fraudulent transactions.

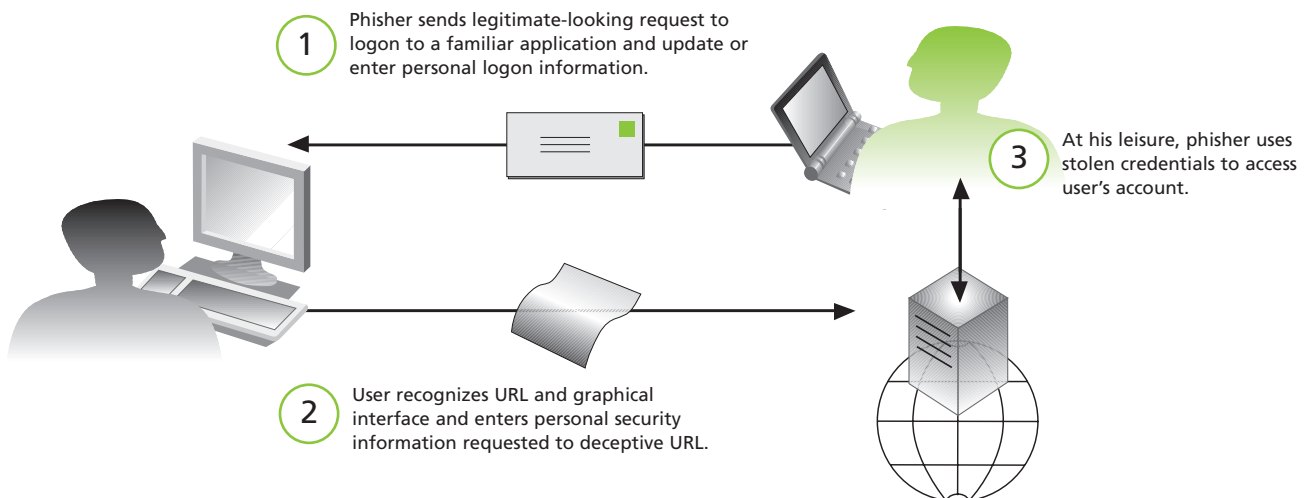
Today, offline phishing is the most common form of man-in-the-middle attack. It's effective in instances where it can exploit a victim's use of static codes (such as passwords and PINs). However, devices that can generate one-time passwords (such as hardware tokens or smart cards) effectively neutralize the threat of this type of attack because the password codes are typically valid for a very limited time window (often as little as one minute) or can only be used once. By the time the attacker receives the OTP, the window for its use has closed.

II. THE EMERGENCE OF ONLINE PHISHING

Ultimately, phishers are seeking to manipulate context and relationships—not transactions. The attacker wants to manipulate the user's feelings and attitudes about the context of the interaction and trick the victim into doing the right thing at the wrong time. The victim makes a decision to mistakenly trust an attacker because client authentication interfaces have been corrupted and (false) authentication success indicators are presented.

In offline phishing, hardware tokens can help the victim make proper judgments about the veracity of authentication processes and prevent him from placing trust improperly. To defeat hardware tokens, a new class of man-in-the-middle attacks—online phishing—have the potential to emerge and circumvent the protections that hardware tokens and one-time passwords can provide.

In offline phishing, the objective is simply to collect information and use it at a later time for any manner of fraudulent transactions. By contrast, online phishing is a real-time attack, where the attacker stands in the middle of the transaction stream between the victim and the legitimate enterprise—possibly using a small piece of malware on the victim's PC.



OFFLINE MAN-IN-THE-MIDDLE ATTACK

With online phishing, the victim receives the bogus e-mail and clicks through to the falsified Web site. However, instead of merely collecting rapidly changing passwords and contact information, the attacker now inserts himself in the middle of an online transaction stream. The attacker asks for and intercepts the user's short-time-window, one-time password and stealthily initiates a session with the legitimate site, posing as the victim and using the victim's just-intercepted ID and OTP.

By carefully mimicking the transactions of the intended site—for example by re-transmitting the results of the successful login back to the user—the attacker can convincingly deceive the victim into believing he is logged into the legitimate site. Once logged into a valid session, the attacker is then free to conduct fraudulent transactions.

III. SIGNING—NOT THE ANSWER

One strategy that site hosts have implemented is “signing” of transactions. Signing requires a customer to “sign,” that is, enter a series of number displayed on the hardware token. To sign securely, the user must sign a full transaction document—which requires that the user sign a hash value (a lengthy cryptographic series of numbers and digits) to authenticate the transaction. This signing authenticates that the user is, indeed, the proper user or account holder. There are several issues to consider with signing.

- The entry of this hash needn't be manual. The hash could be transmitted automatically to the signing device -- as is done with smart cards. However smart cards that lack displays fail to provide assurances of binding a signature to a transaction. The user will not know what's being signed, which means co-resident malware could corrupt the authentication.
- Today's tokens accept only manual user input. Full signing an entire transaction document with a secure hash value (of 25-50 characters) is too inconvenient for user entry. What's more, there's no way to ensure that a particular hash is bound to the specific document being displayed.
- If the bank signs the transaction, it doesn't provide any additional assurance of the user/customer's intentions.

To respond to these issues, some financial institutions are considering “partial signing.” With partial signing, the victim uses his hardware token to, for example, verify his account number—not the details of the transaction, such as the amount of a transfer or the destination account of a transfer.

Partial signing is easier on the user—because there are fewer pieces of data that must be signed. However, partial signing authenticates only portions—certain numbers, such as an account number—not an entire transaction. Unless the token can provide a very narrow context for partial signing (providing a description of the authentication / transaction), there is data associated with the transaction that remains unsigned and therefore vulnerable to manipulation. Furthermore, even a careful user might not understand all of the contextual information or detect anomalies. As a result, partial signing offers little additional protection to the user. Following are some explanations of how man-in-the-middle attackers can take advantage of partial signing to trick users and execute fraudulent transactions.

IV. CONTEXTUAL OVERWRITING

Once the victim sees the re-transmitted login confirmation from the attacker, he thinks he's engaged in a legitimate session. Let's assume the victim intends to purchase a \$10 book from an e-commerce site. The request is intercepted by the logged-in attacker who changes the request from a \$10 purchase with a destination of the online bookstore to a \$10,000 purchase with a destination of the attacker's account.

The victim never sees that contextual overwriting. Instead, the attacker presents a bogus transaction confirming a \$10 book purchase that never actually happened. This all happens in real-time.

Authentication tokens have little effect on the transaction flow. If the online merchant (or financial-services institution) issues a challenge for a one-time password from the customer, the attacker simply retransmits that request to the victim. The victim supplies the OTP, which the attacker then relays back to the merchant.

V. CONTEXTUAL REORDERING

Suppose a vendor/financial institution requires signing of all transaction numbers. As unlikely as this may be for common transactions—the mechanics would likely be far too cumbersome to be adopted by most users—it would nonetheless still present exploitable vulnerabilities.

In contextual reordering, the attacker simply re-sequences the labels for signed fields. For example, the victim might see a bogus screen (presented by the man-in-the-middle attacker) and believe he's authorizing an inbound transfer. In actuality, the attacker has switched the order of accounts and created an outbound transfer—with his account as the destination.

VI. THE NEED FOR SITE VERIFICATION

The proper course is for the computer industry to create a comprehensive method and infrastructure for site verification—mutual authentication by both site host and user. Most authentication is about knowing who the user is—but the user wants the same level of assurance that he's dealing with the right/trusted site. Site verification creates a two-way authentication process. Different security advocates have proposed a couple of alternatives to achieve site verification.

HOST AUTHENTICATION

In this method, the legitimate site host presents a value on-screen. The user must compare that value to what's displayed on the token and ensure it matches. (In other words, the site host says, in effect, "Customer, you should see the following number on your token.") Advocates of this host-authentication mechanism suggest it ensures the user knows he's talking to a legitimate Web site. However, this approach does nothing to defeat man-in-the-middle attacks because the attacker can simply retransmit the displayed value from the legitimate site. This method doesn't even prevent offline phishing attacks because the attacker could simply suppress the host-authentication dialog on the bogus site. No one but the most vigilant user/customer would likely notice the omission. The danger with host authentication: the false sense of security.

BROWSER PLUG-IN

With this method, a locally resident browser plug-in cryptographically binds the one-time password (or challenge-response) to the legitimate site—i.e., the actual URL, rather than the claimed site name. This means that the password is good only for the legitimate site being visited. This is an implicit level of site verification and is a far better approach than token-based host authentication and can prevent man-in-the-middle attacks. There are drawbacks and vulnerabilities, however. First, a browser plug-in presents all of the attendant issues of client software: it must be successfully loaded by the customer and updated, supported, and maintained by the site host. And, if the PC has been compromised through some form of co-resident malware, it remains vulnerable to subsequent exploitation.

TRUSTED PATH

Although this is not a near-term solution, ideally, the answer lies in creating a trusted path that the user can rely on to ensure he is interacting with the legitimate, intended site. Trusted path computing is a framework that assures the user that "what he sees is what he gets." It uses hardware-supported security assurances to establish a secure data transfer from the keyboard/monitor through the actual local processing, into the network and all the way into the intended site and into all of the back-end portions of the site. A trusted path ensures that no one can intercept or modify the bidirectional flow of data or use it in any attack or fraud.

CONCLUSION

Clearly, protection from phishing is required today. The threats posed by a combination of weaknesses in PC architectures, browsers, and Internet security; the openness of online transactions; sophisticated malware; and other sources require a multifaceted integration of tools and mechanisms that are not easily solvable by simple-minded transaction signing.

However, there is no panacea to protect from every possible attack that exists today that would provide an acceptable user experience. Instead, security professionals must make a measured assessment of the level of risk vs. the impact on user experience and then make appropriate and pragmatic business decisions.

In the end, trust is a human affair and the right technology foundations can create a much stronger basis for forming that trusted relationship. As consumers and vendors continue to respond to new and emerging threats to identity theft, it will be essential for them to bear these principles in mind.



RSA Security Inc.
www.rsasecurity.com

RSA Security Ireland Limited
www.rsasecurity.ie

RSA, RSA Security and *Confidence Inspired* are registered trademarks or trademarks of RSA Security Inc. in the United States and/or other countries. All other products or services mentioned are trademarks of their respective owners.
©2005 RSA Security Inc. All rights reserved.

MANIM WP 1005