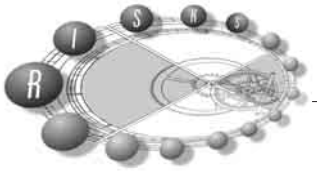

How Secure Is Secure Web Browsing?

Security is of particular importance when sensitive information is sent through the Web. Users must rely on the security of the browser's Secure Socket Layer (SSL) protocol. Although the closed-padlock icon in a browser window depicts a secure connection, it does not imply a totally risk-free secure connection. Whenever the padlock is snapped or a security-related message pops up, you should be alerted and scrutinize the security of that connection.

During the handshake of a secure connection, the server sends a public-key certificate to identify itself. You assume you have a secure connection to the entity identified in the certificate, but that entity may not be who you think it is. So, what is the critical issue in verifying a server certificate? It is in the root certification authority's (CA's) self-signed certificate that the verification starts. We trust root CAs (assuming they don't issue certificates to copycat servers) because our browser developer trusts them. An initial list of root CA certificates comes with browsers. Depending on their trust in browser developers, users may assume all root CAs that come with browsers are robust. However, authenticity is an important concern for other root CA certificates installed after the browser. An attacker can introduce bogus certificates for installation automated via a Visual Basic script. The client sees only a final approval screen that may easily be ignored by clicking on the "yes" button.

Consider the following possible scenario. Suppose you've connected to your bank, www.xyzbank.com. Using network-spoofing techniques, an attacker reroutes this traffic to a counterfeit site and imitates a well-known root CA as the issuer for a fake certificate created for xyzbank. The attacker creates a second imitation certificate: a self-signed root CA certificate for the same well-known root CA. When these imitation certificates are used for a secure connection, you, as a client, will see a warning saying the root CA is not to be trusted. Taking a closer look at the certificate details is of no help, even harmful, because your favorite root CA seems to be the issuer. You might easily prefer to continue and maybe install the imitation certificate assuming there is a bug in your system. Because the well-known root CA's name appears on the final approval screen, it is easy to be fooled by this scheme.

PAUL WATSON

The only authentication guarantee provided by a closed padlock is that the URL in the certificate is the same as the one in the address bar of your browser. A closed padlock does not indicate the server's commercial identity; browsers tell nothing about the certificate it just used to snap the padlock. You must examine the certificate details to ascertain commercial identity. For example, when you connect to www.delta.com, you can't be certain you're connected to Delta Airlines just by the closed padlock; you have to scan the certificate details by clicking on the padlock. If www.delta.com was, say, Delta Foods, you would have seen a closed padlock even if the Web page looks like the airline's.

Certificate examination highlights the dilemma of server identification: the certificate contains the formal name and URL, but the average user needs to see something easily recognizable from previous experience such as the brand name, logo, or current slogan.

Furthermore, to take advantage of URL control, you always have to be aware of the URL you're browsing by checking the address bar. Some secure applications pop up browser windows with the address bars and toolbars removed, in an attempt to restrict the customers to just the buttons provided. In other cases, address bars exist, but due to the copious information in the address bar, the address bar is scrolled left and the URL part is not visible without scrolling right.

A closed/open padlock indicates whether the just-completed transfer was secured or not; it doesn't give any security information about the next connection, which might involve password transfer by clicking on the "sign-in" button. Therefore, whether you enter your password in a secured or an unsecured Web page, that password may go unencrypted. In either case, you should examine the source code of the current Web page to see if the next connection is secured or not.

Secure Web browsing requires a careful and questioning user. Checking certificate details and controlling the root certificate store definitely helps. Root certificate installations should be avoided. Also, pay particular attention to the address bar. Don't bury your head in the sand by merely trusting a closed-padlock icon. C

ALBERT LEVI (levi@sabanciuniv.edu) is an assistant professor of computer science and engineering at Sabanci University, Istanbul, Turkey.