Security
May 4, 2009 7:31 AM PDT

Inventor: SSL not to blame for security woes

by Vivian Yeo

Font size
Print
E-mail
Share

Yahoo! Buzz

At the **RSA Conference** last month in San Francisco, **Taher Elgamal** was conferred the Lifetime Achievement Award--only the third recipient of the award since its inception in 2004.


Taher Elgamals

The **chief security officer of Axway** has more than 25 years of experience in the security industry, starting out as a cryptographic expert. Egypt-born Elgamal has been **credited as an inventor of SSL (Secure Sockets Layer)**, having joined Netscape in early 1995 to release the protocol, which later came under the oversight of the Internet Engineering Task Force.

In a phone interview with ZDNet Asia, Elgamal shares his concern that "SSL gets blamed for all the stuff" and explains what needs to be done to boost security on the Internet.

Q: We've heard about SSL man-in-the-middle attacks and the ability to intercept session cookies. Has the sophistication of attacks grown too rapidly for Internet security standards?
Elgamal: First, it's important to identify the pieces of the solution, and who's responsible for which pieces. SSL is the protocol between two points, usually browser and server. The weaknesses in the system usually are due to the browser, not the protocol. The protocol says (servers) would identify themselves to each other, and it's up to both sites to accept whether this is a good site or not. Unfortunately, the browser

trust model...allows end users to accept things without actually understanding what they are accepting, unrelated to the protocol as it stands.

I think we need to send Web site and software developers to cookie design school so that they can design cookies correctly.

Man-in-the-middle attacks are not actually part of SSL; (they) are network design issues where somebody designs the network and puts in a proxy that makes the browser believe that the server is a different place and then substitutes a different certificate to both sides.

That's a trust issue, actually, and not a man-in-the-middle attack. Because the trust model and the browser are not designed correctly, you can convince the browser that this is the right certificate and convince the server something else, and then look like you actually broke the protocol. You actually did not break the protocol; you terminated the protocol at the wrong point because the browser trust model is broken.

I think all of these problems have to do with browser design rather than security or protocol. It's interesting because SSL gets blamed for all the stuff, but (they are) actually not even related to SSL. (The issue is) which certificate the browser should trust or should not trust.

The cookie (incident) has nothing to do with SSL. The cookie is something that is associated with an HTTP session--it's actually a Web standard. The cookie idea was invented to make sure that you can have a long session on the Web, before SSL (came into the picture).

It also turns out that the secure sessions also use same cookie design to maintain sessions. Some cookies are well-designed, and people cannot hijack the sessions. Some cookies are really badly designed. This has nothing to do with the SSL protocol at all.

I think we need to send Web site and software developers to cookie design school so that they can design cookies correctly. We know very well (which) cookies are good and which cookies are bad, and there are ways to design cookies so that people cannot actually hijack the session.

A security researcher has also pointed out that users still log on to sites that have

expired SSL certificates, and that poses a problem. Accepting the expired certificate is a browser problem.

We had this fight early on in the Internet days: What do we tell the user to do when there is an expired certificate? Security professionals always struggle with the general public because usability always wins. When you get an expired certificate, the site owner or organization would always prefer to allow the user to do things rather than disallow. This is just an unfortunate fact.

Unrelated to what the protocol really is, or whether something is good or bad, the browser allows the end user to say "Yes, I want to accept this anyway." That, in my opinion as a security professional, is the wrong thing to do. I think this is something that the browser makers need to consider better. Of course, (Microsoft Internet Explorer has) 80 percent (share) of the browsers, and then we have (Mozilla) **Firefox** and Apple (Safari). Again, there's no security issue to deal with, as far as the encryption or SSL protocol itself--I think the (browser makers) need to convey these messages better to the end users.

But I know for a fact that Microsoft would never turn off a site because the certificate has expired. Because maybe it expired, and (the owners) are working on getting an extension...you turn the site off, and they lose half a million dollars. There is a commercial issue here that is just hard to deal with.

Security professionals always struggle with the general public because usability always wins.

From a technical standpoint, (however), it should be the case that the certificate would warn the Web server owner that (it will) expire in seven days (and to) go and get the certificate renewed. There should be a process to do that better, but the automation hasn't happened yet.

What is the solution then? How can browser makers keep users and protect them?

There needs to be another control in the browser (in which), for important sites--banking or payment--it refuses to let the users do something, if the certificate is not valid. For simple sites, maybe you give the users the control to continue. We don't do that differentiation these days--there is no difference between an important site...and a

site (where you are) looking for information.

When users walk into a bank branch, they assume that it's trusted. And they make the same assumption when they go to the online bank branch.

Microsoft (and the other browser makers have) the notion of security zones--there is a differentiation between different kinds of sites--but it is really very hard to do from a user standpoint. Most end users don't understand what a security zone means. End users are not very security-savvy, unfortunately. When users walk into a bank branch, they assume that it's trusted. And they make the same assumption when they go to the online bank branch.

There are things that the ecosystem needs to do to help the users not be in a situation where they are compromised. I'm sure there will be solutions that come up...because the Internet itself needs to fix that.

With the development in browser technology, we haven't achieved such a stage yet?

Because usability always wins. Being in security for such a long time, I knew that it was going to be a problem. I had that discussion inside of Netscape for a while, and I had that discussion with Microsoft people--we had that discussion at various times for a very long time. What do you do if the certificate is expired? What do you do if the certificate is wrong? (The latter is) actually a more (serious) problem.

The browser does certain checks when the certificate comes in--(it) will check (whether) the name of the certificate and the URL matches or not. The checks are not enough, as there are certain cases where somebody can fool the browser into thinking that this is the right URL. You can design sessions where that check is very tight--where the connection will not happen--but the general browser basically allows the user to trust things. And the user doesn't understand what that means, of course, so the user will always say yes.

The current security issues are finally bringing up things that we knew about in the security world a long time ago...because (now) the size of the economy of the Internet is growing. The industry needs to deal with this in a better way.

SSL was invented over a decade ago. How different do you think it would be if it had been invented in the current security landscape?
Actually, I honestly do not think it would be different. If 15 years ago, we knew what it would look like (today), we would change the design of the client--the browser. But the protocol itself is actually quite good. The protocol allows the client and the server to agree on which algorithms should be used in a particular session, and it's intentionally done this way.

Twenty years from now, we will find out that different protocols are no longer considered secure, and we should not use them, but we cannot design that protocol to use only a particular set of security algorithms, because I would not know, really, 20 years from now, what would be secure and what would not be secure. Fifteen years ago, certain algorithms were considered good, and we used them in the early Internet days, and then a few years later, we found out that (they) were not secure and should not be used.

All security protocols allow the use of multiple algorithms because we have to (design) the protocol (for use) over a long period of time. The (SSL) protocol is pretty solid...changes in the protocol have been minimum (over the years).

What are you most dissatisfied about in the current security landscape?
The biggest issue with Internet security today is that there are databases with a lot of important info that are available from the Internet, from the outside. Designing secure networks has not been progressed enough. Most of the security problems that you see today (occur) because hackers or insiders are able to access information that they are not authorized to get access to. This is the reality of what today's security environment looks like.

There are attempts (at control)--for example, Visa and MasterCard will force merchants to go through the PCI DSS (Payment Card Industry Data Security Standards) regulations. These are useful--they force Web site owners to go through particular security testing and design to make the site better. There needs to be a more collaborative effort that, whenever a site looks like it has a security deficiency, the Internet tries to help. Whether that's from governments, partners, industry, or associations--it almost doesn't matter--I think a collaborative effort is really important.

That's really the only way to fix a large network like the Internet from a security standpoint.

*Vivian Yeo of **ZDNet Asia** reported from Singapore.*

Topics: **Privacy & data protection**, **Vulnerabilities & attacks**

Tags: **Taher Elgamals**, **SSL**, **browser security**, **Axway**

Share: Digg Del.icio.us Reddit Yahoo! Buzz Facebook

Related

From CNET

Prediction: Apple will recommend security software

The Cold War moves to cyberspace

People are still the biggest security vulnerability

From around the web

Cybersecurity's Twitter-Fast Shifts Forbes.com

Safari, Opera Users Lag Behind in Securi... Washington Post Blogs - Securi...

More related posts powered by Sphere