Impersonating secure web servers

BREACH OF TRUST

Find out why you can't trust your web browser or certificate authorities.

BY KURT SEIFRIED

If you're reading this magazine, a large portion of your life is likely handled through a web browser. Online banking, shopping, bill payment, social networks, news – you name it and chances are you can do it with a web browser. To do these things safely (especially the things like banking), you need to trust your web browser. However, your web browser can't magically verify that yourbank.com is actually your bank, which is why SSL was invented.

One of the main goals of SSL was to make online shopping possible, and one critical component was ensuring that the website you were on was actually the website you meant to be on and not some attacker spoofing a merchant or your bank. But, of course, a piece of software can't check and verify this, so trusted third parties, called Certificate Authorities (CAs) were used. The theory goes like this: You trust your web browser. Your web browser trusts that the CAs do a really good job verifying that when someone tries to buy a certificate for yourbank.com that they are indeed from yourbank.com and have legitimate authority to use this certificate. If the CAs do their job properly, then we can all browse the Internet and bank online with that little lock icon that lets us know we are safe.

Why Trust?

Trust is critical because the modern world simply cannot function without it, especially trust that can be transferred. I can't verify that the food I eat is what it claims to be without a significant amount of laboratory equipment and expertise. When I eat in a restaurant, I can't easily verify that the food has been handled and prepared properly. I depend on various food safety agencies to inspect and license entities to make and serve food. Businesses selling things on credit rely on third parties like Equifax and TransUnion to give a risk rating on how likely a person is to pay for what they are buying on credit.

To address the problem of online commerce, Netscape released the SSL 2.0 specification almost exactly 15 years ago (followed a year later by the 3.0 specification, which fixed a number of serious problems). Back then, there were only a handful of CAs, and they tended to be large companies with a lot at stake (a few bad certificates could ruin their reputation). A few years ago, when I purchased a certificate for seifried.org, the process required several email and phone exchanges and a copy of my passport to prove that I had a legitimate interest in the domain seifried.org. This is no longer the case.

The CA market is hyper-competitive (with some CAs even giving away free personal certificates), which has driven prices down and forced CAs to issue certificates more quickly and with fewer hassles. The result is that the procedures used to verify that you are who you say you are, and that you legitimately control a domain for which you are buying a certificate, have largely been thrown out the window. The very core of what CAs are supposed to be doing – deciding whether someone can be trusted – is no longer what they do. Of course, you could, buy an EV certificate (Extended Verification), which ironically costs about what a regular certificate used to cost, but most sites don't use them.

With all this in mind, I started researching exactly how to get a root certificate into Firefox, what the requirements are, and what protections ensure that users are not put at risk from bad CAs and other malicious parties. I chose Firefox because, not only is it open source, but, unlike commercial browsers, a large amount of Mozilla Foundation business is

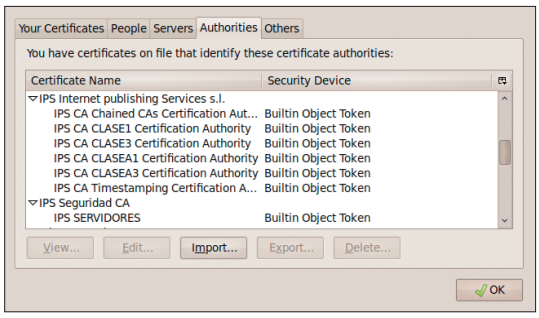

Figure 1: View your certificates in Preferences.

done in public on mailing lists, making it much easier to research. It is important to note that the problems I found in Firefox are generally also present in the other web browsers (e.g., Internet Explorer, Safari, Chrome, Opera, etc.).

What I found was surprising and very worrying. The first issue deals with the process of getting a root certificate into Firefox. To do this, you basically need to pass several audits for various technical standards and post the results online where the Mozilla people can view them. Then after a two-week comment period, if no one objects, you're in. Essentially, it is a fail-open system, and you have to mess up pretty badly to fail it. As a result, Firefox currently has a large number of very questionable CAs in its root store. IPS, for example, a Spanish firm with seven certificates out of 159 total (Figure 1), is reportedly in the process of being removed [1] [2].

The second issue is that only minimal restrictions are placed on a certificate; typically, they can be used for one of three purposes: email, web, or code signing. Who the CA signs certificates for is not restricted. So, for example, government CAs (e.g., Turkey or CNNIC, which is allegedly controlled by the Chinese government) can sign certificates for any domain (as opposed to being restricted to *.tk* and *.cn*, respectively). Trust is rarely an all or nothing proposition. I might trust the Turkish government to sign certificates used on Turkish websites, but I certainly wouldn't trust them to sign the certificate for a website claiming to be my bank.

A Broken Industry

These problems, however, pale compared with what the CAs are doing. The first and most obvious problem is poor verification of customers. In the past (before spam was such a huge problem), the WHOIS service for most domains provided a wealth of correct information for a domain (administrative, technical, and billing contact names, phone numbers, email addresses, physical addresses, etc.). This information was controlled by the domain holder and provided ways to verify easily that someone was who they claimed to be. But now, many domain registrations are private to shield personal information, and some WHOIS providers no longer offer any information publicly at all. Thus, CAs can't rely on WHOIS information to find a valid email address at which to contact you.

Some CAs, such as RapidSSL, really, really want to sell SSL certs cheaply and quickly. RapidSSL has implemented a system whereby if you can receive email at *admin@*, *administrator@*, *hostmaster@*, *info@*, *is@*, *it@*, *mis@*, *postmaster@*, *root@*, *ssladmin@*, *ssladministrator@*, *sslwebmaster@*, *sysadmin@*, or *webmaster@some_domain_name*, you can buy a certificate for it. This system seemed really easy to fool, so I set up an account called *ssladministrator* at a webmail provider and tried to purchase a certificate from RapidSSL; about 30 minutes later, they emailed me the final signed certificate (for the phone verification, you can just use an anonymous prepaid cell and mumble; it's automated and doesn't care). This means that most webmail providers can easily be spoofed by an attacker if they haven't locked down this list of addresses. Sadly, this problem with RapidSSL has been brought up a number of times, and nothing has been done by RapidSSL or by Firefox (e.g., removing the RapidSSL root certificate) to change the process.

This is because the industry-accepted standard for confirming someone is who the say they are and that they control a domain is that "the CA takes reasonable measures to verify," which is very ambiguous at best and meaningless at worst. One CA proposed that customers could fax a signed letter on company letterhead as proof that they controlled a domain (Have they not heard of word processors and image editing programs? Or online fax services?). CAs want to sell as many certificates for as little money as they can; if this puts users at risk but doesn't cost the CA anything, then there is no incentive to fix things.

What to Do

So, how can you protect yourself? As a domain owner, you make sure the above list of email addresses is under strict control. (Note: Other CAs will probably accept other email addresses, good luck finding them all!) Beyond that, there is almost nothing you can do. As a web browser user, I would recommend disabling the trust bits on any certificates for parties that you don't deal with (e.g., the Japanese government) or for which there are known problems (e.g., RapidSSL) [3]. Also, you might want to consider using Certificate Patrol [4], which will alert you when a certificate changes (indicating a possible man-in-the-middle attack by someone with a valid certificate).

I hope by the time this article comes to print that RapidSSL, Firefox [5], and other CAs and web browsers will have addressed these issues, but I doubt it because they have been known for quite some time now and not much has happened. ■

INFO

[1] Firefox-included certificate list: *http://spreadsheets.google.com/pub?key=ttwCVzDVuWzZYaDosdU6e3w&single=true&gid=0&output=html*

[2] Included certificate list: *http://www.mozilla.org/projects/security/certs/included/*

[3] How to override default root certificate settings: *https://wiki.mozilla.org/CA:UserCertDB*

[4] Certificate Patrol: *https://addons.mozilla.org/en-US/firefox/addon/6415*

[5] Root change process: *https://wiki.mozilla.org/CA:Root_Change_Process*

THE AUTHOR

Kurt Seifried is an Information Security Consultant specializing in Linux and networks since 1996. He often wonders how it is that technology works on a large scale but often fails on a small scale.