Who's behind that web site?

SSL, certificates, and detecting phishers

John Locke

Let's talk about phishing. *Phishing* is just like fishing, only your identity is the fish and the bait is an email that looks like it came from your bank, or eBay, or Paypal, or any other legitimate place. The goal is to get you to follow a link to a site owned by the phisher, and trick you into divulging some private information, such as your bank account number, pin, passwords, or social security number.

Some phishing emails look completely legitimate, using logos, links, and text from the real business. Many try to warn you about fraud being committed with your account – the truth is, the senders of the email are the ones trying to commit fraud with your account, if they can trick you into divulging it. These types of emails are almost always fake. When you follow the link in such an email, you'll usually get taken to a web site that looks exactly like the real web site. But it's not.

How do you tell the difference between a real and a fake web site?

Stop! Wait! Before clicking any links, go update your web browser. In the past few months, many of the vulnerabilities in both Internet Explorer and Mozilla Firefox have been related to making it harder to tell what web site you're really visiting.

A malicious web site could use some carefully crafted web addresses that could lead you to believe you're really at a legitimate site. What's worse is that there have been some vulnerabilities in Internet Explorer that could allow these malicious sites to surreptitiously download programs to your computer. This is one of the prime ways you get infected with "spyware".

Figure 1. When Firefox has an update available, it shows a red arrow icon in the upper right corner

For Internet Explorer, go to Windows Update by following the link on your start menu. For Mozilla Firefox, look for a little red triangle in the upper right corner of your window, as highlighted in Figure 1.

> One of the biggest tricks of the phishermen is to add more symbols after a normal looking domain name, which makes your browser go to a completely different place than you're expecting

Okay, now that your browser is up-to-date, you're a bit more protected online. There are two basic things to pay attention to, when you're deciding whether to trust a web site:

1. The web address, or URL. URL stands for Uniform Resource Locater, but it basically just means the web address of the page you're viewing.

2. The security certificate. Also called the SSL or TLS certificate. We'll get into this shortly.

By applying what you know about web addresses and security certificates, you're better armed to detect a fraudulent site. So what do you need to look for?

Anatomy of a web address

Bear with me. We're about to get slightly technical here – but this is basic information you need to know to web surf safely. By now you've probably seen thousands of web addresses. There are some very strict rules for web addresses – how they are put together, what each part does, etc. For this discussion, we need to look at two of these parts: the protocol, and the domain name.

The protocol

The protocol is the very first part of the web address. It's the *http://* or *https://* part. You generally don't need to type it in when you go to a web site, because most browsers will add it for you. These are two different protocols used for web pages and several other types of data. A protocol, in this context, is a description of how to introduce yourself properly, the browser equivalent of knowing what words to say to introduce yourself, and ask for a book from a librarian or a fried banana from a Thai street vendor. In the web world, the dominant protocol is *Hyper-Text Transfer Protocol*, or HTTP. The Hyper-Text is your web page, the transfer protocol tells your browser how to get it. HTTPS is *Hyper-Text Transfer Protocol over SSL*.

SSL stands for Secure Sockets Layer, and lately it's been renamed to Transport Layer Security, or TLS. SSL/TLS is an entire framework that provides two things: encryption and authentication. Encryption hides your data between the web site and your browser, preventing anyone from intercepting it along the way. Authentication lets you verify that the server you're visiting is really the server it says it is, and not some other server taking its place. I'll get more into this in a moment.

So the first thing to check in the web address is, are you visiting a page that is protected by SSL/TLS? And you can tell this by looking at the protocol in the address bar of your browser – is it https, or http?

The domain name

The other part of the web address to look at is the domain name. The domain name is everything after the https:// part up to the next forward slash ("/"). And that means everything. One of the biggest tricks of the phishermen is to add more symbols after a normal looking domain name, that makes your browser go to a completely different place than you're expecting.

A domain name is broken up into domain "parts." The further right you go in the domain name, the more significant the part is. For example, looking at the domain name *www.freelock.com*, the most significant part is *.com*. It's significant for technical reasons, not just for the phrase "dot-com bomb".(should this be "dot-com boom")

Ready for another acronym? **.com** is called the *Top-Level Domain*, or TLD, of this domain name. Each country has its own two-letter TLD, such as .us, .uk, .au, .tv, and then there are the three-letter TLDs we're all familiar with: .com, .net, .org, .edu, .gov, .biz and a few other less common ones. These top-level domains are controlled by a designated registry, and copied into what are called the *Root Domain Name Servers*. There are 14 of them, scattered around the world.

Moving to the left in the domain name, the actual domain we're looking at is *freelock.com*. When your browser asks for www.freelock.com, it first goes to the Root Domain Name Servers to ask for where to find the directory for the domain freelock.com. The Root Domain Name Servers tell your browser to go to the IP address 69.55.225.251, which happens to be my Domain Name Server. Your browser then asks my name server where to find **www**, which could be an actual computer, or it could be another domain part.

If you're paying attention here, you might ask: what's so special about the "www" part? The answer is, absolutely nothing. It's just another domain name part, and usually takes you to exactly the same place as you would get if you left it out. A very small percentage of sites expect you to type www as the first part of the domain name; the vast majority take you to the same place without it. Hint for the lazy: don't bother typing www.

Bad domain names

So how do the phishers trick you? By putting something between the .com part and the first slash. Here are some

bogus web addresses I see in my quarantine, all targeting Paypal, a popular e-commerce site:

- *http://www.paypal.com-cgi-bin.biz/ppverify.php?cmd=_login-run&mail=&?motd=account_verify* – this link appears in an email apparently from Paypal. See the first slash after the http:// part? The TLD is .biz, and the primary domain part is com-cgi-bin.biz. Remember, anybody can get a domain name, and if you think about it, why would Paypal have a domain name called com-cgi-bin.biz? It certainly doesn't inspire trust.
- *http://www.paypalllaa.biz/ppverify.php?cmd=_login-run&mail=&?motd=account_verify* – Another phisher, registering a domain name like Paypal, hoping to trick you into visiting.
- *http://211.220.195.70/paypal/login.html* – Never, ever, ever, trust an IP address in a financial mailing. Four numbers instead of a domain name takes you to an otherwise anonymous machine on the internet. This is okay for your web developer, to show you a development version of your web site, but if anybody else gives you an address like this, you'd better have a good reason to trust them.

You can see these web addresses in the status bar (the very bottom strip of a program window) in most email readers. I say most, because there are a couple very popular email programs by a certain vendor that until recently thought it would be too confusing to show you the real link in the status bar. In these programs, which will remain nameless, it's very difficult to tell where a link really points. Yet another reason to switch to an open source email program, like Mozilla Thunderbird or Evolution.

But even if the address looks completely legitimate, you still have to be aware – there have been flaws in pretty much every mail reader that make it possible to hide the real destination of a link. The phishers use a couple of technical tricks that involve either embedding backspaces into the address, so that the real address gets overwritten by a fake one, or by using international characters that look just like the plain English text characters.

Worse, in March 2005, there were a series of attacks on the internet that were called "DNS Cache Poisoning." What happened was that many vulnerable name servers in use all over the world were fed bogus information about where to find the Root Domain Name Servers. Essentially, these name servers were hijacked. Imagine calling 411 to get a phone number, but instead of getting an operator at a legitimate phone center, you got a fake one that gave you the wrong phone number, on purpose, for the person you were trying to reach. By doing this, the bad guys could send you anywhere they wanted, and pull all sorts of tricks to make you believe you were really sent to your bank's web site.

This is where SSL authentication comes in.

Figure 2. Firefox warns you if there's a problem authenticating the SSL certificate

SSL/TLS authentication

As I said before, SSL/TLS does two things: encrypts traffic, and authenticates the server at the other end. The encryption part is simple, from a user point of view – if you see the lock icon in the bottom right corner of the window, the connection is encrypted. But without authentication, you could just be talking very privately with the garage-based scam artist posing as your bank.

That's why authentication is important. Authentication provides some assurance that you're at the real Paypal.com, and not a scam site. It works by checking something called a certificate that the web server presents to your browser, before sending any data. Your browser does a number of checks on the certificate, and if it appears to be valid, and matches the domain name in the address bar, it shows the lock icon in the status bar and gets the page. If it detects anything wrong, it gives you an authentication warning.

You've probably seen authentication warnings in your browser. Figure 2 shows one. There are several things your browser checks to determine if the certificate is legitimate:

1. Does the domain name in the certificate match the domain name in the address bar?

2. Has the certificate expired?
3. Is the certificate signed by a trusted authority?
4. Is the signing authority certificate valid, and current?

I'll get to certificate authorities shortly, but first, I'm going to point out what an SSL certificate DOES NOT DO:

- *A valid certificate does not guarantee that a site is legitimate!!!*

Just like a domain name, anybody can get a certificate. It costs a little more – between $35 and $800 a year, depending on the certificate authority – but it's easy to do.

A valid certificate provides a guarantee that somebody you trust (the certificate authority) has verified that the web site you're visiting really is the one in your browser's address bar.

SSL warnings would alert you if your DNS server has been hijacked, and you're visiting a fake site. It will also tell you if a web master is too cheap to buy a certificate from a trusted authority, or if they've been lazy renewing their certificates.

What's a certificate?

Public Key Encryption made secure electronic communications possible. Public Key Encryption involves two *keys*, or codes used to encrypt or decrypt data. One key is public, shared with the world at large. The other is secret, only stored on your computer. In public key encryption, whatever you encrypt with the public key can only be decrypted by the secret key – you cannot even decrypt it with the public key you used to encrypt it.

This may sound counter-intuitive, but think about the difference between multiplication and division – it's much easier to multiply two large numbers than to divide one from the other. This is the underlying principle that makes public key encryption possible.

In the other direction, you can use a secret key to *digitally sign* a chunk of data, and verify the signature with the public key. This turns out to be a very useful thing to use to guarantee the authenticity of a message.

A *certificate* is a public key, combined with information about the owner of the corresponding secret key, digitally signed by somebody else – the *Certificate Authority*. So when somebody wants to run an encrypted web server, here's what they do:

- Create a brand new pair of keys, public and secret.
- Send the public key, along with name, address, site name, and other details to a certificate authority, such as Verisign, GeoTrust, Thawte, or their local webmaster as a certificate signing request.
- The certificate authority uses their secret key to sign the certificate signing request, and the result is a signed certificate.
- The site owner installs the secret key and the signed certificate on the server.

Now, when you visit the site, here's what happens:

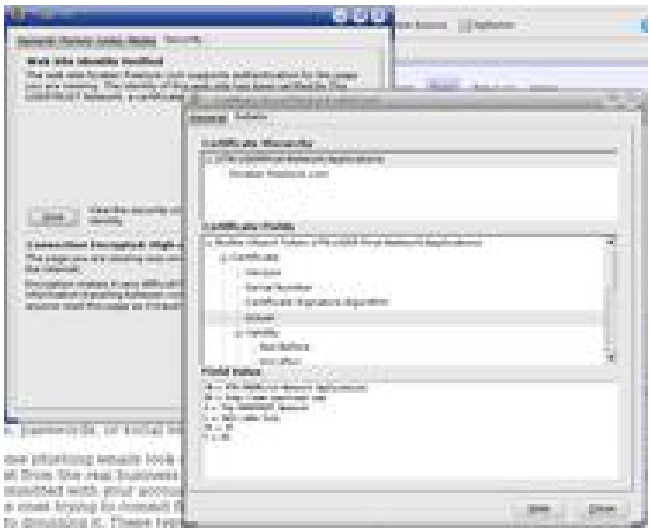
- Your browser connects to the server, and asks for its certificate.
- Your browser verifies the signature of the certificate authority. If it doesn't recognize the certificate authority, or the details in the certificate do not match the web address or have expired, or anything else weird, it pops up a warning.
- If the certificate is properly verified, or if you tell your browser to go ahead even though it couldn't verify the signature, your browser creates a new, random symmetrical key, encrypts it with the public key in the certificate, and sends it back to the server.
- From then on, both the server and your web browser have a big, shared key they use to encrypt all data going back and forth. Your browser will show the lock icon, so that you can easily see that your surfing is protected.

Who do you trust?

Trusted Authority. Certificate Authority. Signing Authority. All these are the same thing–a small set of organizations your browser has been preconfigured to trust. Their certificates are built into your browser or operating system. They're there because the people who created your web browser worked out a deal with a bunch of different organizations to have their certificates pre-installed. These are the companies like Verisign, Geotrust, Thawte, and others.

Anybody can run their own certificate authority, but if the signing certificate isn't built into browsers visiting sites signed with it, you'll get warned when you visit the site.

You can view the details of an SSL certificate, and see the "chain of trust" of signing authorities



Checking a certificate can help you judge whether the site you're visiting is authentic. To check a certificate in Firefox:

1. Double-click the lock icon in the bottom right corner of the browser window. This will open up the page info box, with the Security tab in front.
2. Click the *View* button to view the certificate.
3. On the *General* tab, you can see details about the owner of the certificate, the certificate authority who issued it, and validity dates. Check the very first item, the Common Name (CN). This is what your browser compares to the domain name in the URL. While the URL might be spoofed due to a browser vulnerability, chances are it will appear correctly here. Are there any strange characters in the Common Name? If you see any "^", or "%" symbols, watch out. Does the name actually represent the site you're trying to visit?
4. On the *Details* tab, you can see the whole *Chain of trust* in the Certificate Hierarchy pane. The first item in this list is a certificate authority built into your browser, who you trust to verify web sites.

Each of the items in the Certificate Hierarchy pane is a certificate. The first one is the one you trust. Sometimes there are up to four certificates in the chain. Each certificate has signed the next, vouching for it, telling you that it's legitimate. You can click on any of them to get more information about the particular certificate. The last one in the list is the one belonging to the server you're visiting.

How reliable is this system?

A phisherman can hijack your connection, and send you to a completely bogus web site. But at this point, they cannot forge an SSL certificate.

One of the principles of security is that no security is perfect – you just have different levels of protection. The entire system of public keys, certificates, and certificate authorities is collectively called a *Public Key Infrastructure*. Given that the encryption and authentication details are effective, there are still a few possible ways to defeat the system:

- Attackers could break into a server and steal the secret key. By installing it on a fake server, and using a different hijacking technique, they could possibly spoof the real thing. There's a mechanism called a *Certificate Revocation List* that allows certificate authorities to revoke a certificate, but this isn't fully working. So far, there haven't been many documented cases of this.
- Attackers could trick you into installing the certificate for a malicious certificate authority, and then your browser would trust whatever they told you to trust. Spyware could do this. This is perhaps the easiest way to break the system. Run anti-spyware software and keep your machine up-to-date to prevent this.
- Attackers could break into a real, trusted certificate authority, and sign a bunch of fake certificates. This would be like breaking into Fort Knox.
- Somebody might figure out how to break the encryption system itself, discovering a new mathematical technique. If this happens, I guarantee you'll hear about it all over the news – it would make the entire e-commerce system broken, and would affect everybody.

Seven steps to safer surfing

Based on the risks just listed, here is a summary of what you can do to safely conduct e-commerce on the internet:

1. Keep your system up-to-date with the latest operating system and web browser updates.

2. If you're on Windows, use anti-virus and anti-spyware software to keep your system clean.
3. Look for the lock icon in your browser anytime you're doing anything financial, or passing any kind of sensitive information.
4. If your browser alerts you to an authentication problem, pay close attention, and be extra cautious before doing anything sensitive.
5. Check the URL in the address bar, before typing anything sensitive – are you really where you think you are?
6. Compare the URL in the address bar with the Common Name in the SSL certificate, to be really sure of a site.
7. Listen for news of any break-ins to the sites you visit, or significant changes to e-commerce security in general.

Following these steps will keep you much safer online. But nothing is a substitute for having a questioning, critical mind. Don't trust anything you receive in email – verify it with the real source. Don't follow links blindly. Look in the address bar. The internet is a dangerous place, but armed with a bit of knowledge, you can keep your sensitive information safe. Now, if only the companies you do business with could do the same...

Copyright information

About the author

John Locke is the author of the book *Open Source Solutions for Small Business Problems.* He provides technology strategy and free software implementations for small and growing businesses in the Pacific Northwest through his business, Freelock Computing (`http://freelock.com`).