When are self-signed certificates acceptable?

Source : https://www.sslshopper.com/article-when-are-self-signed-certificates-acceptable.html

This question was posted on Slashdot and it solicited many different responses. UltraLoser posed the question this way:

> *When is it acceptable to encourage users to accept a self-signed SSL cert? Recently the staff of a certain Web site turned on optional SSL with a self-signed and domain-mismatched certificate for its users and encourages them to add an exception for this certificate. Their defense is that it is just as secure as one signed by a commercial CA; and because their site exists for the distribution of copyrighted material the staff do not want to have their personal information in the hands of a CA. In their situation* ***is it acceptable to encourage users to trust this certificate*** *or is this giving users a false sense of security?*

There were hundreds of different responses but many people displayed a mistaken understanding of the purpose of SSL certificates. This is expressed in the first poster's response:

> ***SSL certificates provide one thing, and one thing only: Encryption*** *between the two ends using the certificate.*
>
> *They do not, and never been able to, provide any verification of who is on either end. This is because literally one second after they are issued, regardless of the level of effort that goes into validating who is doing the buying, someone else can be in control of the certificate, legitimately or otherwise.*
>
> *Now, I understand perfectly well that Verisign and its brethren have made a huge industry out of scamming consumers into thinking that identification is indeed something that a certificate provides; but that is marketing illusion and nothing more. Hokum and hand-waving.*

This is common perception of SSL certificates. **It is also completely wrong!** It is easy to see why server administrators think this. They think, "I need to get an SSL certificate to secure my server." But the certificate doesn't secure anything. The web server (IIS, Apache, etc.) simply requires a certificate so that **it** can do the encryption. It could do it all automatically if it wanted. But there is a reason for the SSL certificate. It is required by the server to enable encryption because it is an essential to establishing a trusted/secure connection.

It's All About Trust

A self-signed certificate is like a fake drivers license. Who would accept a fake drivers license? Most people wouldn't. But Internet communication is very different from real-life communication. You have little idea who is sending the information on the other end. **The biggest problem with a self-signed certificate, is a man-in-the-middle attack**. Even if you are 100% sure that you are on the correct website and you completely trust the site (your email server for example), you could have someone intercept the connection and present you with their own self-signed certificate. You would think that you are using a secure connection with your email server but you are really using a secure connection to an attacker's email server. Oh, and they now have your login credentials and anything else you gave them.

JSBiff explains it well:

> *It all comes down to, can you determine that you are using the same crypto key that the server is? The reason for signing certificates and the like is to try to detect when you are being hit with a man-in-the-middle attack. In a nutshell, that attack is when you try to open a connection to your 'known' IP address, say, 123.45.6.7. Even though you are connecting to a 'known' IP address of a server you trust, doesn't mean you can necessarily trust traffic from that IP address. Why not? Because the Internet works by passing data from router to router until your data gets to its destination.* ***Every router in between is an opportunity for malicious code on that router to re-write your packet****, and you'd never know the difference, unless you have some way to *verify* that the packet is from the trusted server.*
>
> *A crypto key, if you have the *correct* key, can verify for you that the data hasn't been tampered with. The problem is, however, that* ***before you can begin encrypted communications, you must do an *unencrypted* key exchange****, where the server gives you its crypto key. Here's where the man-in-the-middle has an opportunity. If your traffic is going through my router, I can intercept the self-signed key from the server, and* ***generate a new self-signed key with the same server name****, etc in it, so that it *looks* like the self-signed key from your server, but which allows me to decrypt the communications between you and the server. My router then establishes a connection to the server using the *correct* key, and as data passes between you and the server, I unencrypt the data using the real key, then re-encrypt it using the 'fake' key. So, the data is encrypted between me and the server, and between me and you, but gets unencrypted in my router, giving me the opportunity to spy on your data, or even alter if I want.*
>
> ***The point of a CA-signed certificate is to give slightly stronger verification that you are actually using the key that belongs to the server you are trying to connect to.***
>
> *Yes, self-signed keys have some uses - in particular if you happen to know the real key's fingerprint (a fingerprint is a numeric or hex string which identifies a cryptographic key), so that you can verify yourself that you are using the correct key for SSL. If you don't happen to know the fingerprint, it's probably still fine to use self-signed certs on a LAN, where you control all the equipment, so don't have to worry so much about a man-in-the-middle (although, arguably, on a LAN you might not even need encryption).*
>
> *So, in summary, yes, SSL adds security to the connection, but ONLY if you can verify that the correct SSL key for your server is being used, and not a different key that a hostile router has injected.*

Got it? Don't use self-signed certificates for sensitive, public connections. If you don't want to buy an SSL certificate, at least set-up your own certificate authority with its own root certificate. This will still give an error message to visitors unless you or they import the root certificate into the browser, but there is far less of a chance of a man-in-the-middle-attack.

When Is a Self-Signed SSL Certificate Acceptable? - [Slashdot]