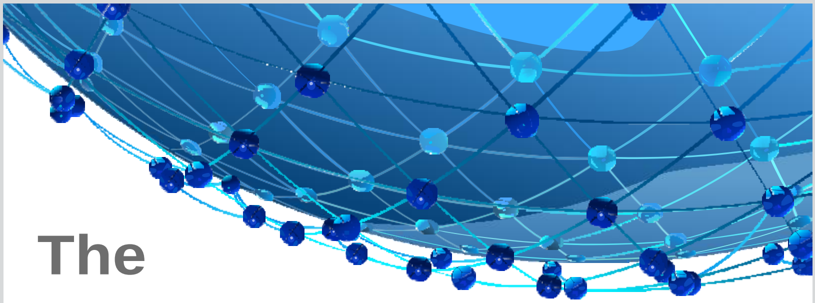

The Top 10

TLS/SSL Client, Server, and Application Best Practices

TLS/SSL certificates are something that we take for granted. Some of us know that they provide identification and encryption services, but most of us don't spend much time ensuring that we are using them in the safest and most secure manner. In this top ten, we look at 10 TLS/SSL client, server, and application best practices:

1 **Ensure that security software and settings on your web server are up-to-date.** When the Heartbleed vulnerability became public a year ago, systems administrators were encouraged to upgrade OpenSSL to a version where the vulnerability had been addressed. A recent survey found that only 16% of the web servers of the largest 2000 organizations in the world that were vulnerable to Heartbleed had been fully repaired in the year since the vulnerability was made public.[1] It's important to be up-to-date because the SSL 3.0 protocol has security vulnerabilities and organizations should only be using the secure successor protocol to SSL 3.0, called TLS 1.2.

2 **Ensure that your Internet browser and client are up-to-date.** Certificate Authorities can become compromised, meaning that bad actors can issue certificates that appear legitimate.[2] Browser vendors often place blocks on certificates issued from certificate authorities that have been found to have problems. Newer Internet browsers also won't use the vulnerable SSL protocol, but instead will only use the secure TLS 1.2 protocol.

3 **Check that vendor-installed software running on your computers doesn't intercept TLS/SSL traffic.** The Superfish and PrivDog[3] software, installed on some computers by the vendor, were able to intercept web traffic. What made the problem worse was that security researchers were able to discover the private key that the software used, which would allow malware to generate certificates that the client would trust.[4]

4 **Ensure that your client computers are free of malware.** Malware can intercept traffic and make it appear that communication is secure when it is instead being fed to nefarious third parties.

5 **If you are creating applications, ensure that your application performs a revocation check on the TLS/SSL certificate.** Applications should only use valid certificates and should check to determine if the certificate they have remains valid and is not revoked. If an application cannot check a certificate's validity, it should reject the certificate.

6 **Ensure that your application is encrypting traffic when communicating.** One Australian bank released a mobile app where the developer had disabled encrypted communication during testing and forgotten to enable it when the application was published to an App store. You can check this with an application like Fiddler.

7 **Ensure that you keep an eye on the expiry date of your organization's TLS/SSL certificates and replace them in a timely manner.** Even big organizations like Google and Microsoft have had certificates expire, causing disruptions to their customers until the certificates were replaced.

8 **Ensure that your organization obtains its SSL certificate from a trusted Certificate Authority.** Getting a certificate by a vendor that's trusted by all major operating systems and application vendors is a much better proposition than getting a certificate from a cheap unknown vendor.

9 **If you are managing a CA, ensure that your CRL distribution points are accessible to anyone who might use a certificate your organization has issued.** By default, CRL distribution points may not be configured to allow everyone that needs access to have access. Ensure that CRL distribution points are set correctly before you start issuing certificates.

10 **If you are managing a CA, ensure that the necessary clients are configured to trust certificates issued from the server.** You can use Active Directory or other configuration management products to ensure that the computers and devices that you manage trust certificates issued by the CA that you manage.

---

[1] http://itsecuritynews.info/2015/04/07/still-bleeding-one-year-later-heartbleed-2015-research/
[2] http://techcrunch.com/2015/04/01/google-cnnic/
[3] http://www.pcworld.com/article/2887632/secure-advertising-tool-privdog-compromises-https-security.html
[4] http://www.gizmodo.com.au/2015/02/lenovo-joins-the-malevolent-side-of-the-online-advertising-industry/