INFORMATION SECURITY AND PRIVACY

Project Report

Hardik Vasa

Q1. What are the limitations of using self-signed certificates?

Answer: Some of the issues of the self-signed certificate are:

Self-signing of a certificate basically means signing the certificate (and hence validating the application by oneself). Here if a person is self-signing a certificate in a company on behalf of his entire development unit, the unit should trust the user of his loyalty of non-misconduct.

Self-signed certificates may not be authorized by other systems and devices. Those devices do not usually trust non-authorized signing of certificates and hence, they will reject it. Operating systems only allow authorized certificates from trusted vendors. Macintosh is one of the example of strict certificate policy.

Self-signed certificates usually does not meet the modern security standardization and are poorly hashed. This may lead to security loop in the certificate validation. Also they cannot be reversed. Once the certificate is signed, it cannot be de-signed or cryptographically destroyed. One of the example of this is the Android Application. Once the application is self-signed by the application developer, it cannot be de-signed and a clone of the application is to be made to use another certificate.

Most modern browsers and web parsers are also programmed to block the applications with the self-signed certificates. Also, a user is blocked from accessing the access point to connect to the Wi-Fi if the certificates are signed by the user instead of using the network provided certificate for authorization. It is usually not recommended to self-sign the certificate in a globally available application.

[Reference - http://security.stackexchange.com/questions/8110/what-are-the-risks-of-self-signing-a-certificate-for-ssl]

Q2. What are they useful for?

Answer:

Above I mentioned some of the disadvantages of the self-signed certificates. But in some scenarios, they also form to be helpful.

It is relatively easy, hassle free to create a self-signed certificate as compared to the standardize version. And self-signing is cheaper (relatively). This means that if the application is small and if both the entities agree to it, it is better to have the certificate self-signed.

According to the resources, it is easier to customize and go beyond the standard self-signed certificates. One of the example is to have a non-standard size of key or to have an alternate signee and so on.

Self-signed certificates are useful for server-testing applications. This is helpful because the department does not have to bear the burden of payment to the CA just for testing and troubleshooting purposes. Once the application is globally available, the authorization can then shift to the CA root certificates.

This is how sometimes self-signing the certificate is beneficial.

[Reference - http://serverfault.com/questions/1117/what-are-the-benefits-of-a-self-signed-certificate-on-a-live-site]