The Most Common OpenSSL Commands

One of the most versatile SSL tools is OpenSSL which is an open source implementation of the SSL protocol. There are versions of OpenSSL for nearly every platform, including Windows, Linux, and Mac OS X. OpenSSL is commonly used to create the CSR and private key for many different platforms, including Apache. However, it also has hundreds of different functions that allow you to view the details of a CSR or certificate, compare an MD5 hash of the certificate and private key to make sure they match, verify that a certificate is installed properly on any website, and convert the certificate to a different format.

A compiled version of OpenSSL for Windows can be found here.

General OpenSSL Commands

These commands allow you to generate CSRs, Certificates, and Private Keys, and do other miscellaneous tasks.

- **Generate a new private key and Certificate Signing Request**

  ```
  openssl req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout privateKey.key
  ```

- **Generate a self-signed certificate (see How to Create and Install an Apache Self Signed Certificate for more info)**

  ```
  openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout privateKey.key -out certificate.crt
  ```

- **Generate a certificate signing request (CSR) for an existing private key**

  ```
  openssl req -out CSR.csr -key privateKey.key -new
  ```

- **Generate a certificate signing request based on an existing certificate**

  ```
  openssl x509 -x509toreq -in certificate.crt -out CSR.csr -signkey privateKey.key
  ```

- **Remove a passphrase from a private key**

  ```
  openssl rsa -in privateKey.pem -out newPrivateKey.pem
  ```

Checking Using OpenSSL

If you need to check the information within a Certificate, CSR or Private Key, use these commands. You can also check CSRs and check certificates using the online tools linked here.

- **Check a Certificate Signing Request (CSR)**

  ```
  openssl req -text -noout -verify -in CSR.csr
  ```

- **Check a private key**

  ```
  openssl rsa -in privateKey.key -check
  ```

- **Check a certificate**

  ```
  openssl x509 -in certificate.crt -text -noout
  ```

- **Check a PKCS#12 file (.pfx or .p12)**

  ```
  openssl pkcs12 -info -in keyStore.p12
  ```

Debugging Using OpenSSL

If you are receiving an error that the private doesn't match the certificate or that a certificate that you installed to a site is not trusted, try one of these commands. If you are trying to verify that an SSL certificate is installed correctly, be sure to check out the SSL Checker.

- **Check an MD5 hash of the public key to ensure that it matches with what is in a CSR or private key**

  ```
  openssl x509 -noout -modulus -in certificate.crt | openssl md5
  openssl rsa -noout -modulus -in privateKey.key | openssl md5
  openssl req -noout -modulus -in CSR.csr | openssl md5
  ```

- **Check an SSL connection. All the certificates (including Intermediates) should be displayed**

  ```
  openssl s_client -connect www.paypal.com:443
  ```

Converting Using OpenSSL

These commands allow you to convert certificates and keys to different formats to make them compatible with specific types of servers or software. For example, you can convert a normal PEM file that would work with Apache to a PFX (PKCS#12) file and use it with Tomcat or IIS. You can use an SSL Converter to convert certificates without using OpenSSL.

- **Convert a DER file (.crt .cer .der) to PEM**

  ```
  openssl x509 -inform der -in certificate.cer -out certificate.pem
  ```

- **Convert a PEM file to DER**

  ```
  openssl x509 -outform der -in certificate.pem -out certificate.der
  ```

- **Convert a PKCS#12 file (.pfx .p12) containing a private key and certificates to PEM**

  ```
  openssl pkcs12 -in keyStore.pfx -out keyStore.pem -nodes
  ```

  You can add -nocerts to only output the private key or add -nokeys to only output the certificates.

- **Convert a PEM certificate file and a private key to PKCS#12 (.pfx .p12)**

  ```
  openssl pkcs12 -export -out certificate.pfx -inkey privateKey.key -in
  certificate.crt -certfile CACert.crt
  ```