OpenSSL

Introduction

- OpenSSL is an open source tool for using the Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocols for Web authentication.
- **OpenSSL** is a software library for applications that secure communications over computer networks against eavesdrop or need to identify the party at the other end. It is widely used by Internet servers, including the majority of HTTPS websites.
- OpenSSL is written in the C programming language and relies on different ciphers and algorithms to provide encryption.

- OpenSSL is licensed under an Apache-style license, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions. OpenSSL is all about its command lines.

Use of OpenSSL

OpenSSL can be used to secure data, manage digital certificates, and debug SSL communications.

History

Various successive versions of OpenSSL have been developed since 1998, when the product was first revealed. The most recent set of OpenSSL versions including 1.0.1 through 1.0.1f involve a dramatic security flaw discovered in April of 2014.

Digital certificate

A digital certificate is an electronic “drivers license” that is used to prove the identity of a client or server.

- Certificate Authorities (CAs) are responsible for issuing digital certificates, and proving the identity of the entity requesting the certificate.
- Digital certificates contain several pieces of information, including: a certificate version, a serial number to uniquely identify the certificate, an attribute (Issuer) to identity the organization who issued the certificate, a range of dates the certificate is valid, an attribute (Subject) to identify the site the certificate has been issued to, and a digital signature.

- When an organization wants to request a digital certificate from a certificate authority, they will need to submit a certificate signing request (CSR).
- The certificate signing request contains a public key, a common name (e.g., www.example.com) to uniquely identify the site, and locality information to identify the organization.

The following example shows how to generate a certificate signing request:

```
$ openssl req -new -outform PEM -keyform PEM -keyout
secret.key -out cert.csr -newkey rsa:1024
```

Generating a 1024 bit RSA private key

- During the request generation process, two 1024-bit RSA keys are generated, and various pieces of information are gathered.
- The openssl utility will prompt for a pass-phrase, which is used to encrypt the contents of the private key. Once the keys are generated, the private key will be PEM encoded and placed in the file secret.key.
- The certificate signing request is placed in the file cert.csr. This file contains the public key, locality information, and a common name to uniquely identify the site. You can print the contents of the certificate signing request with the req command:

$ openssl -req -in cert.csr –text

- Once you have verified the certificate signing request, you can submit the contents to your favorite Certificate Authority.
- The certificate authority will use the contents of this file along with their private key to generate a digital signature.
- The certificate authority will also assign an expiration date, and incorporate additional attributes to uniquely identify the certificate authority.

Displaying the contents of a digital certificate

Digital certificates can be stored in several formats. Two of the most common formats are PEM (Privacy Enhanced Mail) and DER (Definite Encoding Rules).

OpenSSL can print the contents of both certificate formats with the x509 commands.

The following example will print the contents of the PEM encoded certificate cert.crt.pem:

```
$ openssl x509 -in cert.crt.pem -inform PEM -text -noout
```

Converting between certificate types

- As mentioned above, digital certificates can be stored in a variety of formats. This can cause problems when a certificate needs to be migrated between heterogeneous web servers, or distributed between application components.
- OpenSSL provides the x509 option to convert between PEM and DER encoded certificates.

The following example will convert a PEM encoded certificate to DER format:

```
$ openssl x509 -in cert.crt.pem -inform PEM -out cert.crt.der -outform DER
```

https://prefetch.net/articles/realworldssl.html

https://www.openssl.org/

https://en.wikipedia.org/wiki/Indian_Computer_Emergency_Response_Team

https://www.cert-in.org.in/


Thank You