Deep Dive into SSL implementation Scenarios for Oracle Applications E-Business Suite

Subhajit Das Chaudhuri
Pythian
Ottawa, Canada

Keywords: SSL, PKI, CSR, E-Business, Public and Private Keys

Introduction

This paper will help us deep dive into the essential concepts of SSL and how it has evolved over time. It will also discuss how security can be strengthened in the various tech stack layers of Oracle Applications E-Business Suite.

What is SSL?

SSL (Secure Sockets Layer) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser; or a mail server and a mail client (e.g., Outlook).

Public Key Infrastructure (PKI)

The Public Key Infrastructure (PKI) is the software management system and database system that allows us to sign certificates, keeping a list of revoked certificates, distributing public keys.

For securing individual applications, you can use any well-known commercial PKI as their root CA certificate is most likely to be inside your browser/application.

PKI is a framework and not a specific technology.

Public and Private Keys

The Public and Private key pair comprise of two uniquely related cryptographic keys. The Public Key is what its name suggests - Public. It is made available to everyone via a publicly accessible repository or directory. On the other hand, the Private Key must remain confidential to its respective owner. Because the key pair is mathematically related, whatever is encrypted with a Public Key may only be decrypted by its corresponding Private Key and vice versa. Public Key Cryptography can therefore achieve Confidentiality. However another important aspect of Public Key Cryptography is its ability to create a Digital Signature.

Digital Signature

It is a One-way hash (mathematical value) that has been encrypted with signer’s private key. To validate the integrity of the data, the receiver first uses the signer's public key to decrypt the hash. If the two hashes match, the data is not tampered since it was signed.

What is inside a digital Certificate?

Illustration 1: A Digital SSL Certificate

It is an electronic document that contains information about:

- Who issued the certificate?
- Who the certificate is issued to?
- Valid till
- Subject's Public Key value
- Digital Signature

Certificate Authority (CA)

It is Responsible for issuing, revoking and distributing certificates. It is often a trusted third-party organization like:

- Digicert
- Verisign
- Companies or organizations can have an in-house CA (eg. Microsoft Certificate Services)
- Stores the public key in a directory that is available to anyone that wants to verify your certificate.

Certificates installed in your PC

You can use Certificate Manager to see details about your certificates installed in your client PC. Open Certificate Manager by clicking the Start button, typing certmgr.msc into the Search box, and then

pressing ENTER. Administrator permission is required if you are prompted for an administrator password or confirmation.

SSL Certificate

SSL Certificates have a key pair: a public and a private key. These keys work together to establish an encrypted connection. The certificate also contains what is called the "subject," which is the identity of the certificate/website owner.

To get a certificate, you must create a Certificate Signing Request (CSR) on your server. This process creates a private key and public key on your server. The CSR data file that you send to the SSL Certificate issuer (called a Certificate Authority or CA) contains the public key. The CA uses the CSR data file to create a data structure to match your private key without compromising the key itself. The CA never sees the private key.

Once you receive the SSL Certificate, you install it on your server. You also install an intermediate certificate that establishes the credibility of your SSL Certificate by tying it to your CA's root certificate.

Certificate Signing Request (CSR)

A Certificate Signing Request (CSR) is a PKCS10 request which is an unsigned copy of your certificate. Entrust Certificate Services will use the Certificate Signing Request (CSR) to generate your signed digital x509 V3 SSL

It contains the following:

- Information about the organization (organization name, country, etc...)
- Web Server's public key
- A unique mathematical match to server's private key

Signed vs Self-Signed Certificate

What it comes down to is trust. When you use a self-signed certificate, you are saying to your customers "trust me - I am who I say I am." When you use a certificate signed by a CA, you are saying, "Trust me - Verisign agrees I am who I say I am."

Code Signing Certificate

A code signing certificate from a Trusted CA is required to sign your Java content securely. It allows you to deliver signed code from your server (e.g. jar files) to client desktops and verify you as the publisher and trusted provider of that code. It also verifies that the code has not been altered.

Example: https://www.digicert.com/code-signing/purchase-code-signing-certificate.htm

Signing of EBS JAR files

JRE 7u40 and JRE 7u45 are now both past their expiration date; forms will no longer launch on a High security setting when using a self-signed certificate.

Running unsigned applications like this will be blocked in a future release because it is potentially unsafe and a security risk

Please refer to: Enhanced Signing of Oracle E-Business Suite JAR Files (Note 1591073.1)

SSL Handshake

- Browser connects to a web server (website) secured with SSL (https). Browser requests that the server identify itself.
- Server sends a copy of its SSL Certificate, including the server's public key.
- Browser checks the certificate root against a list of trusted CAs and that the certificate is unexpired, unrevoked, and that its common name is valid for the website that it is connecting to. If the browser trusts the certificate, it creates, encrypts, and sends back a symmetric session key using the server's public key.
- Server decrypts the symmetric session key using its private key and sends back an acknowledgement encrypted with the session key to start the encrypted session.
- Server and Browser now encrypt all transmitted data with the session key.

Setting up SSL in EBS Application tier (Doc ID 376700.1)

The main steps for setting up SSL on the Application Tier are:

- Set Your Environment.
- Create a wallet.
- Create a Certificate Request.
- Submit the Certificate Request to a Certifying Authority.
- Import your Server Certificate to the Wallet.
- Copy the Apache Wallet to the OPMN Wallet.
- Update the JDK Cacerts File.
- Update the Context File.

- Run Autoconfig.
- Restart the application tier services.

Switch to TLS from SSL for EBS

Oracle E-Business Suite customers can migrate to TLS and mitigate the effects of POODLE attacks by following:

CVE-2014-3566 - Instructions to Mitigate the SSLv3 Vulnerability ("POODLE Attack") in Oracle E-Business Suite (Note 1937646.1)

Contact address:

Name: Subhajit Das Chaudhuri

Company: Pythian

Address: 319 McRae Ave, Suite 700

Ottawa, Ontario, Canada K1Z 0B9

Phone: +1-613-565-8696

Email chaudhuri@pythian.com

Twitter : @DCSubhajit