Hands-on with Oracle WebLogic Server

Deep dive into various configurations with Oracle Weblogic Server

August 18, 2015

Steps to create a self-signed certificate using OpenSSL

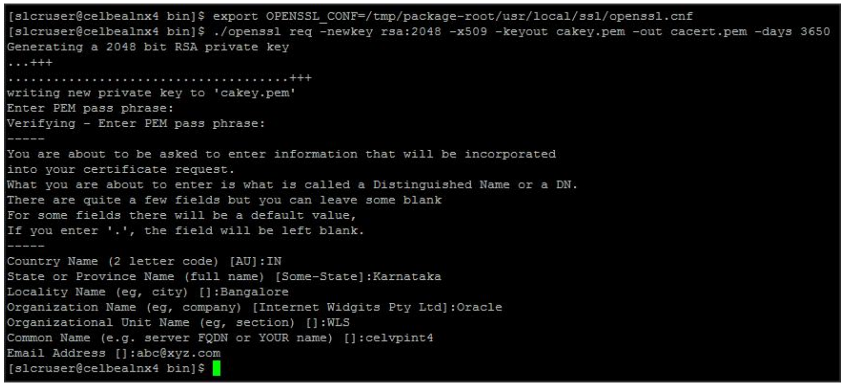
Puneeth Prakash
PRINCIPAL SOFTWARE ENGINEER

Below are the steps to create a self-signed certificate using OpenSSL :

STEP 1 :

Create a private key and public certificate using the following command :

Command : openssl req -newkey rsa:2048 -x509 -keyout cakey.pem -out cacert.pem -days 3650



```
[slcruser@celbealnx4 bin]$ export OPENSSL_CONF=/tmp/package-root/usr/local/ssl/openssl.cnf
[slcruser@celbealnx4 bin]$ ./openssl req -newkey rsa:2048 -x509 -keyout cakey.pem -out cacert.pem -days 3650
Generating a 2048 bit RSA private key
...+++
....................................+++
writing new private key to 'cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Karnataka
Locality Name (eg, city) []:Bangalore
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Oracle
Organizational Unit Name (eg, section) []:WLS
Common Name (e.g. server FQDN or YOUR name) []:celvpint4
Email Address []:abc@xyz.com
[slcruser@celbealnx4 bin]$
```

In the above command :

- If you add "-nodes" then your private key will not be encrypted.

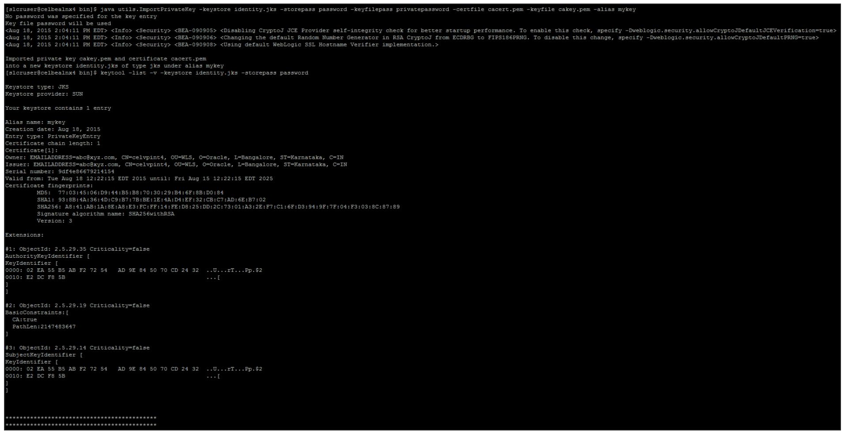
- cakey.pem is the private key

- cacert.pem is the public certificate

STEP 2 :

Use the following java utility to create a JKS keystore :

Command : java utils.ImportPrivateKey -keystore identity.jks -storepass password -keyfilepass privatepassword -certfile cacert.pem -keyfile cakey.pem -alias mykey



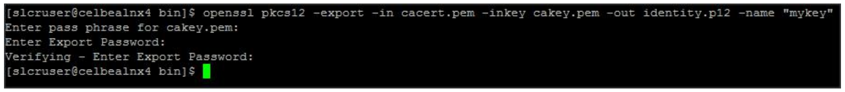
```
[slcruser@celbealnx4 bin]$ java utils.ImportPrivateKey -keystore identity.jks -storepass password -keyfilepass privatepassword -certfile cacert.pem -keyfile cakey.pem -alias mykey
No password was specified for the key entry
Key file password will be used
<Aug 18, 2015 2:04:11 PM EDT> <Info> <Security> <BEA-090905> <Disabling CryptoJ JCE Provider self-integrity check for better startup performance. To enable this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true>
<Aug 18, 2015 2:04:11 PM EDT> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in RSA CryptoJ from ECDRBG to FIPS186PRNG. To disable this change, specify -Dweblogic.security.allowCryptoJDefaultPRNG=true>
<Aug 18, 2015 2:04:11 PM EDT> <Info> <Security> <BEA-090908> <Using default WebLogic SSL Hostname Verifier implementation.>

Imported private key cakey.pem and certificate cacert.pem
into a new keystore identity.jks of type jks under alias mykey
[slcruser@celbealnx4 bin]$ keytool -list -v -keystore identity.jks -storepass password

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: mykey
Creation date: Aug 18, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: EMAILADDRESS=abc@xyz.com, CN=celvpint4, OU=WLS, O=Oracle, L=Bangalore, ST=Karnataka, C=IN
Issuer: EMAILADDRESS=abc@xyz.com, CN=celvpint4, OU=WLS, O=Oracle, L=Bangalore, ST=Karnataka, C=IN
Serial number: 9df4e86679214154
Valid from: Tue Aug 18 12:22:15 EDT 2015 until: Fri Aug 15 12:22:15 EDT 2025
Certificate fingerprints:
         MD5:  77:03:45:06:D9:44:B5:B8:70:30:29:B4:6F:8B:D0:84
         SHA1: 93:8B:4A:36:4D:C9:B7:7B:BE:1E:4A:D4:EF:32:CB:C7:AD:6E:B7:02
         SHA256: A8:41:AB:1A:8E:A8:E3:FC:FF:14:FE:D8:25:DD:2C:73:01:A3:2E:F7:C1:6F:D3:94:9F:7F:04:F3:03:8C:87:89
         Signature algorithm name: SHA256withRSA
         Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 02 EA 55 B5 AB F2 72 54   AD 9E 84 50 70 CD 24 32  ..U...rT...Pp.$2
0010: E2 DC F8 5B                                        ...[
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 02 EA 55 B5 AB F2 72 54   AD 9E 84 50 70 CD 24 32  ..U...rT...Pp.$2
0010: E2 DC F8 5B                                        ...[
]
]



*******************************************
*******************************************
```

Alternatively, you can use the following commands to create a PKCS12 / JKS file :

STEP 2a :

Create a PKCS12 keystore :

Command : openssl pkcs12 -export -in cacert.pem -inkey cakey.pem -out identity.p12 -name "mykey"



```
[slcruser@celbealnx4 bin]$ openssl pkcs12 -export -in cacert.pem -inkey cakey.pem -out identity.p12 -name "mykey"
Enter pass phrase for cakey.pem:
Enter Export Password:
Verifying - Enter Export Password:
[slcruser@celbealnx4 bin]$
```

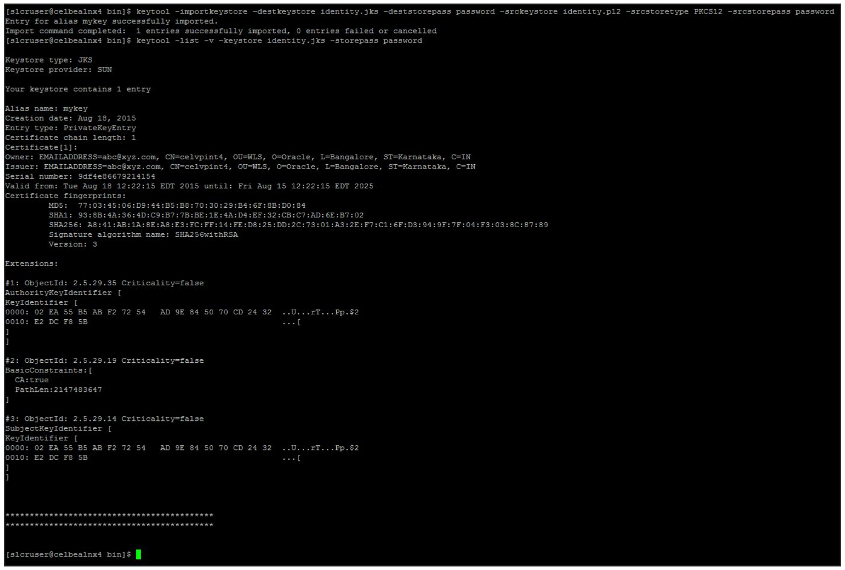
In the above command :

- "-name" is the alias of the private key entry in keystore.

STEP 2b :

Now convert the PKCS12 keystore to JKS keytstore using keytool command :

Command : keytool -importkeystore -destkeystore identity.jks -deststorepass password -srckeystore identity.p12 -srcstoretype PKCS12 -srcstorepass password

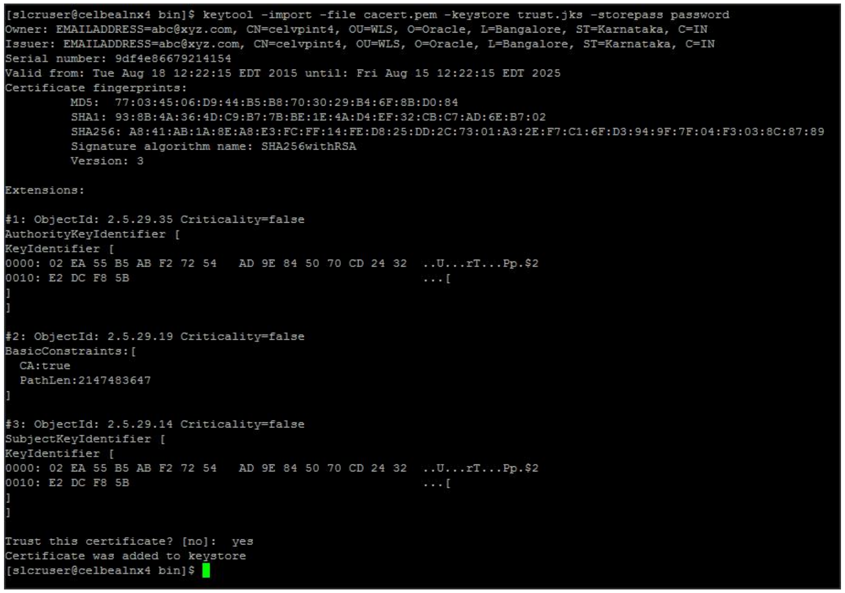


```
[slcruser@celbealnx4 bin]$ keytool -importkeystore -destkeystore identity.jks -deststorepass password -srckeystore identity.p12 -srcstoretype PKCS12 -srcstorepass password
Entry for alias mykey successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
[slcruser@celbealnx4 bin]$ keytool -list -v -keystore identity.jks -storepass password

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: mykey
Creation date: Aug 18, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: EMAILADDRESS=abc@xyz.com, CN=celvpint4, OU=WLS, O=Oracle, L=Bangalore, ST=Karnataka, C=IN
Issuer: EMAILADDRESS=abc@xyz.com, CN=celvpint4, OU=WLS, O=Oracle, L=Bangalore, ST=Karnataka, C=IN
Serial number: 9df4e86679214154
Valid from: Tue Aug 18 12:22:15 EDT 2015 until: Fri Aug 15 12:22:15 EDT 2025
Certificate fingerprints:
         MD5:  77:03:45:06:D9:44:B5:B8:70:30:29:B4:6F:8B:D0:84
         SHA1: 93:8B:4A:36:4D:C9:B7:7B:BE:1E:4A:D4:EF:32:CB:C7:AD:6E:B7:02
         SHA256: A8:41:AB:1A:8E:A8:E3:FC:FF:14:FE:D8:25:DD:2C:73:01:A3:2E:F7:C1:6F:D3:94:9F:7F:04:F3:03:8C:87:89
         Signature algorithm name: SHA256withRSA
         Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 02 EA 55 B5 AB F2 72 54   AD 9E 84 50 70 CD 24 32  ..U...rT...Pp.$2
0010: E2 DC F8 5B                                        ...[
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 02 EA 55 B5 AB F2 72 54   AD 9E 84 50 70 CD 24 32  ..U...rT...Pp.$2
0010: E2 DC F8 5B                                        ...[
]
]



*******************************************
*******************************************


[slcruser@celbealnx4 bin]$
```

STEP 3 :

Create a trust keystore using the following command :

Command : keytool -import -file cacert.pem -keystore trust.jks -storepass password



```
[slcruser@celbealnx4 bin]$ keytool -import -file cacert.pem -keystore trust.jks -storepass password
Owner: EMAILADDRESS=abc@xyz.com, CN=celvpint4, OU=WLS, O=Oracle, L=Bangalore, ST=Karnataka, C=IN
Issuer: EMAILADDRESS=abc@xyz.com, CN=celvpint4, OU=WLS, O=Oracle, L=Bangalore, ST=Karnataka, C=IN
Serial number: 9df4e86679214154
Valid from: Tue Aug 18 12:22:15 EDT 2015 until: Fri Aug 15 12:22:15 EDT 2025
Certificate fingerprints:
         MD5:  77:03:45:06:D9:44:B5:B8:70:30:29:B4:6F:8B:D0:84
         SHA1: 93:8B:4A:36:4D:C9:B7:7B:BE:1E:4A:D4:EF:32:CB:C7:AD:6E:B7:02
         SHA256: A8:41:AB:1A:8E:A8:E3:FC:FF:14:FE:D8:25:DD:2C:73:01:A3:2E:F7:C1:6F:D3:94:9F:7F:04:F3:03:8C:87:89
         Signature algorithm name: SHA256withRSA
         Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 02 EA 55 B5 AB F2 72 54   AD 9E 84 50 70 CD 24 32  ..U...rT...Pp.$2
0010: E2 DC F8 5B                                        ...[
]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 02 EA 55 B5 AB F2 72 54   AD 9E 84 50 70 CD 24 32  ..U...rT...Pp.$2
0010: E2 DC F8 5B                                        ...[
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
[slcruser@celbealnx4 bin]$
```

<Additional Info>

- To view the public certificate :

```
openssl x509 -in cacert.pem -noout -text
```

- To concatenate the private key and public certificate into a pem file (which is required for many web-servers) :

```
cat cakey.pem cacert.pem > server.pem
```

Be the first to comment

Comments (0)

Recent Content

FUSION MIDDLEWARE

Configure WLS Web Server Proxy Plug-In for Internet Information Services 8.5 (IIS)

In this example, I have IIS v8.5 installed on Windows 2012R2 forwarding request to WLS 10.3.6 using WLS Plugin 1.1 NOTE: The steps mentioned...

FUSION MIDDLEWARE

Steps to configure Oracle Identity Cloud Integrator provider with Java Cloud Service(JCS)

Oracle Identity Cloud Integrator provider has to be configured in JCS Admin console if you have a requirement to login to your application...

WEBLOGIC SECURITY

Steps to configure SAM IDCS (Identity Cloud Se Identity Provider and JC Java Cloud Service) as S Provider

Service Provider Configu Last updated: 30-Dec...

Site Map Legal Notices Terms of Use Privacy Cookie Preferences Ad Choices Oracle Content Marketing Login