A Brief Tour of Apache

The RPM packages are a little different from the standard Apache tarball. For example, if you were using the standard tarball, you would find that all configuration directives are contained inside `httpd.conf`. For the RPM package, you would find that SSL configuration exists as a separate configuration file, `ssl.conf`.

In my opinion, this is a good thing because it promotes modularity and is easier to find the appropriate places where you would add directives or make changes. The downside of this is, of course, that instructions in existing HOW-TOs and books dealing with Apache would need to be adapted to the different file layout.

The Apache configuration remains in `httpd.conf`, and the directory where you can find it is `/etc/httpd/conf`. I will not be covering the directives for this file, but you can find out more about them from the Apache manual or other books and documentation on this subject. Additional configuration files, if you are installing modules, can be found in `/etc/httpd/conf.d` directory.

The "executable file" is `httpd` and it can be found inside `/usr/sbin`. There is also an `apachectl` script in the same directory, for those who are more accustomed to the traditional way of starting and stopping Apache.

The RPM package automatically installs a startup script in `/etc/init.d`, but Apache does not start on boot by default. You will need to use `chkconfig` to enable this.

Log files can be found in `/etc/httpd/logs`. There are 2 log files created : `access_log` and `error_log`.

Your web pages should go into `/var/www/html directory`. This is the default DocumentRoot. Interestingly though, for a default RPM installation, there are no files in that directory. The default page that is served up after install is `/var/www/error/noindex.html`. The configuration file that controls this is `/etc/httpd/conf.d/welcome.conf`.

Enabling SSL on Apache

A lot of people think that SSL is something that "automagically" protects your server. Some people expect a login prompt when a server is configured for SSL operation. SSL is more complex than that, and if you are one of these people, please refer to the links in the References section before continuing.

Getting SSL to work on Apache is surprisingly easy for the Apache RPM packages because of the supplied scripts. I will not be covering the theory behind SSL and web server security in this document. Instead, I will merely give the steps I took to get SSL installed and working for my Apache installation.

But first, let me explain the scenario I will be covering :

There are actually several different ways SSL can be configured. You could order and pay for a CA certificate, server certificate and server private key from a trusted CA (Certificate Authority) institution, such as Verisign. If you do not want to spend the money, you could set yourself up as a Certificate Authority. (Note : If you do not understand what certificates are, and how they relate to SSL, I strongly recommend that you read the links in the References section) If you are going to be running a public website, it would be a good idea to purchase your certificates from Verisign, especially if you are handling transactions. If you are running a private intranet or extranet server, you could get by with just setting yourself up as a CA, but you really should consider the Verisign option.

For this section, I shall be setting myself up as a Certificate Authority (CA), and generating my own server key and certificate.

1. Ensure that all necessary components are installed
2. Edit or create an OpenSSL template
3. Create a new CA certificate
4. Create a Certificate Signing Request (CSR)
5. Sign the CSR
6. Store certificates in a directory
7. Edit ssl.conf
8. Test SSL
9. Disabling the passphrase on startup (Optional)

Ensure that all necessary components are installed

You will need the following components installed along with your base Apache RPM packages

- mod_ssl
- openssl
- openssl-devel

You can check this by executing following commands,

[root@fc4two ssl]# rpm -q mod_ssl

mod_ssl-2.0.54-10

[root@fc4two ssl]# rpm -q openssl

openssl-0.9.7f-7

[root@fc4two ssl]# rpm -q openssl-devel

openssl-devel-0.9.7f-7

Edit or create an OpenSSL template

Look inside the directory `/usr/share/ssl/ or` /etc/pki/tls/ for a file named `openssl.cnf` and open it in your favorite editor. You will need to look for and change the following values in the file:

Edit the openssl.cnf file

[root@fc4two init.d]# find / -name openssl.cnf
/etc/pki/tls/openssl.cnf

[2]+ Stopped find / -name openssl.cnf
[root@fc4two init.d]# vim /etc/pki/tls/openssl.cnf

---

countryName_default: put the name of your country

stateOrProvinceName_default: put the name of your state or province

localityName_default: put the name of your locality (street? region?)

0.organizationName_default: put the default organization name

organizationalUnitName_default: put your organization unit (OU) name

countryName_default: put the name of your country

stateOrProvinceName_default: put the name of your state or province

localityName_default: put the name of your locality (street? region?)

0.organizationName_default: put the default organization name

organizationalUnitName_default: put your organization unit (OU) name

---

Create a new CA certificate

There is a supplied `CA` script inside the directory `/usr/share/ssl/misc/` or `/etc/pki/tls/misc/` that you can use to generate your certificate. To begin generating your certificate, simply execute the following commands:

```
[root@fc4two init.d]#
[root@fc4two init.d]# cd /etc/pki/tls/
[root@fc4two tls]# ls
cert.pem  certs  misc  openssl.cnf  private
[root@fc4two tls]# cd misc/
[root@fc4two misc]# ls
CA  c_hash  c_info  c_issuer  c_name
[root@fc4two misc]# ./CA –newca
```

Press ENTER to create the new certificate and you will be prompted to key-in a passphrase. You will need to use this passphrase later, so you should remember what you keyed in here. Then you will be prompted for the particulars of your organization, etc. If you edited the openssl.cnf file properly in the previous step, you should be able to just hit ENTER for all the options except your server host name. (I put server host name as localhost.localdomain) A sample session is shown below. In this session, I am generating a CA certificate for my local Fedora Core 4 workstation:

```
mkdir: cannot create directory `../../CA': File exists
mkdir: cannot create directory `../../CA/private': File exists
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
...........++++++
......++++++
writing new private key to '../../CA/private/./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) [SL]:
State or Province Name (full name) [Western]:
Locality Name (eg, city) [Kohuwala]:
Organization Name (eg, company) [ceylonlinux]:
Organizational Unit Name (eg, section) [HelaTechBooks]:
Common Name (eg, your name or your server's hostname) []:localhost.localdomain
Email Address []:suranga@ceylonlinux.com

Create a Certificate Signing Request (CSR)

To create a CSR, we will use the same `CA` script, but with a different switch.

```
[root@fc4two misc]# ./CA -newreq
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'newreq.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) [SL]:
State or Province Name (full name) [Western]:
Locality Name (eg, city) [Kohuwala]:
Organization Name (eg, company) [ceylonlinux]:
Organizational Unit Name (eg, section) [HelaTechBooks]:
Common Name (eg, your name or your server's hostname) []:localhost.localdomain
Email Address []:suranga@ceylonlinux.com
```

You see how useful creating or editing the template file is ? If you did not set the defaults, you'd have to key in the same information all over again.

You will be prompted for extra attributes, a challenge password and an optional company name. If you don't need any of this, you can safely ignore these messages and just hit ENTER.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem

Note the last line, which states that your CSR has been created, and is called `newreq.pem` in the current directory.

Sign the CSR

If you have gotten this far without any errors, you can now sign the CSR. To do that, we will use the `CA` script again, but, again, with a different switch.

You will be prompted for your passphrase, and then information about your certificate will spew out on the screen. You should see something like what is shown below.

```
[root@fc4two misc]# ./CA -sign
Using configuration from /etc/pki/tls/openssl.cnf
Enter pass phrase for ../../CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Feb 18 19:22:13 2006 GMT
            Not After : Feb 18 19:22:13 2007 GMT
        Subject:
            countryName               = SL
            stateOrProvinceName       = Western
            localityName              = Kohuwala
            organizationName          = ceylonlinux
            organizationalUnitName    = HelaTechBooks
            commonName                = localhost.localdomain
            emailAddress              = suranga@ceylonlinux.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                FF:FB:1B:8E:97:40:60:8F:11:6B:AC:1B:A3:BA:52:DC:1B:63:C4:B7
            X509v3 Authority Key Identifier:
                keyid:8C:82:7B:76:95:6C:ED:86:4C:5E:BA:9C:93:99:D9:2E:99:B1:28:3E
```

```
DirName:/C=SL/ST=Western/L=Kohuwala/O=ceylonlinux/OU=HelaTechBooks/CN=lo
calhost.localdomain/emailAddress=suranga@ceylonlinux.com
            serial:8B:60:B8:70:AD:54:E2:90
```

You will then be asked to sign the certificate and commit the changes.

```
Certificate is to be certified until Feb 18 19:22:13 2007 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=SL, ST=Western, L=Kohuwala, O=ceylonlinux, OU=HelaTechBooks,
CN=localhost.localdomain/emailAddress=suranga@ceylonlinux.com
        Validity
            Not Before: Feb 18 19:22:13 2006 GMT
            Not After : Feb 18 19:22:13 2007 GMT
        Subject: C=SL, ST=Western, L=Kohuwala, O=ceylonlinux, OU=HelaTechBooks,
CN=localhost.localdomain/emailAddress=suranga@ceylonlinux.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c3:6f:ff:65:6c:21:d8:78:e4:5a:62:0c:43:13:
                    3d:bd:16:2c:4b:19:cc:1a:74:fb:81:35:8e:0d:20:
                    d8:b2:70:e9:2e:35:ed:0a:1b:82:a2:bd:4a:14:f4:
                    41:04:4f:a8:db:f9:e4:cb:6b:0a:3f:1b:09:ad:49:
                    fb:bb:0c:57:8e:a2:54:51:e5:f1:db:d9:22:2b:2e:
                    45:95:ef:0a:2c:55:31:64:55:a5:b0:ff:6c:51:21:
                    c4:77:95:1a:d6:21:37:f2:47:13:ec:68:3f:e4:c1:
                    cd:45:20:68:cc:41:48:1b:ed:5f:b3:ee:47:35:9f:
                    0d:eb:87:4b:d7:4a:4b:11:27
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
```

FF:FB:1B:8E:97:40:60:8F:11:6B:AC:1B:A3:BA:52:DC:1B:63:C4:B7
X509v3 Authority Key Identifier:
keyid:8C:82:7B:76:95:6C:ED:86:4C:5E:BA:9C:93:99:D9:2E:99:B1:28:3E

DirName:/C=SL/ST=Western/L=Kohuwala/O=ceylonlinux/OU=HelaTechBooks/CN=localhost.localdomain/emailAddress=suranga@ceylonlinux.com
serial:8B:60:B8:70:AD:54:E2:90

Signature Algorithm: md5WithRSAEncryption
9e:ab:cb:99:07:41:8c:e2:5c:e6:ee:5b:f8:10:d8:f8:a3:21:
a6:27:7b:66:e4:e2:93:06:c4:e1:41:14:b3:6b:c3:fb:67:d9:
b5:f0:2c:87:e4:9f:42:58:0c:c2:bd:b7:1c:4d:9d:85:fd:50:
ae:82:0b:21:83:d0:9b:d6:5b:60:6d:21:29:72:ae:48:5a:d9:
91:d1:e0:09:0c:9c:93:67:eb:3c:ad:ad:64:8a:09:57:e4:bd:
ca:30:49:f6:16:db:64:bc:a8:e5:e6:fd:3d:39:6d:14:9f:bf:
ec:ca:8a:6b:81:21:7d:e8:b3:2b:3b:68:88:56:a9:12:a9:79:
3f:d8
-----BEGIN CERTIFICATE-----
MIIEBjCCA2+gAwIBAgIBATANBgkqhkiG9w0BAQQFADCBqDELMAkGA1UEBhMCU0wx
EDAOBgNVBAgTB1dlc3Rlcm4xETAPBgNVBAcTCEtvaHV3YWxhMRQwEgYDVQQKEwtj
ZXlsb25saW51eDEWMBQGA1UECxMNSGVsYVRlY2hCb29rczEeMBwGA1UEAxMVbG9j
YWxob3N0LmxvY2FsZG9tYWluMSYwJAYJKoZIhvcNAQkBFhdzdXJhbmdhQGNleWxv
bmxpbnV4LmNvbTAeFw0wNjAyMTgxOTIyMTNaFw0wNzAyMTgxOTIyMTNaMIGoMQsw
CQYDVQQGEwJTTDEQMA4GA1UECBMHV2VzdGVybjERMA8GA1UEBxMIS29odXdhbGEx
FDASBgNVBAoTC2NleWxvbmxpbnV4MRYwFAYDVQQLEw1IZWxhVGVjaEJvb2tzMR4w
HAYDVQQDExVsb2NhbGhvc3QubG9jYWxkb21haW4xJjAkBgkqhkiG9w0BCQEWF3N1
cmFuZ2FAY2V5bG9ubGludXguY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB
gQDDb/9lbCHYeORaYgxDEz29FixLGcwadPuBNY4NINiycOkuNe0KG4KivUoU9EEE
T6jb+eTLawo/GwmtSfu7DFeOolRR5fHb2SIrLkWV7wosVTFkVaWw/2xRIcR3lRrW
ITfyRxPsaD/kwc1FIGjMQUgb7V+z7kc1nw3rh0vXSksRJwIDAQABo4IBPDCCATgw
CQYDVR0TBAIwADAsBglghkgBhvhCAQ0EHxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2Vy
dGlmaWNhdGUwHQYDVR0OBBYEFP/7G46XQGCPEWusG6O6UtwbY8S3MIHdBgNVHSME
gdUwgdKAFIyCe3aVbO2GTF66nJOZ2S6ZsSg+oYGupIGrMIGoMQswCQYDVQQGEwJT

TDEQMA4GA1UECBMHV2VzdGVybjERMA8GA1UEBxMIS29odXdhbGExFDASB
gNVBAoT
C2NleWxvbmxpbnV4MRYwFAYDVQQLEw1IZWxhVGVjaEJvb2tzMR4wHAYDVQ
QDExVs
b2NhbGhvc3QubG9jYWxkb21haW4xJjAkBgkqhkiG9w0BCQEWF3N1cmFuZ2FAY2V
5
bG9ubGludXguY29tggkAi2C4cK1U4pAwDQYJKoZIhvcNAQEEBQADgYEAnqvLmQ
dB
jOJc5u5b+BDY+KMhpid7ZuTikwbE4UEUs2vD+2fZtfAsh+SfQlgMwr23HE2dhf1Q
roILIYPQm9ZbYG0hKXKuSFrZkdHgCQyck2frPK2tZIoJV+S9yjBJ9hbbZLyo5eb9
PTltFJ+/7MqKa4EhfeizKztoiFapEql5P9g=
-----END CERTIFICATE-----
Signed certificate is in newcert.pem

Information about your signed certificate will then be dumped to screen. Note the validity dates of the certificate.

At the end of the information dump, you will be told that the certificate filename is `newcert.pem`, and can be found in the current directory.

[root@fc4two misc]# ls
CA c_hash c_info c_issuer c_name newcert.pem newreq.pem

Store certificates in a directory

Finally, we will create a directory and copy the newly created certificates to the new directory. Now create a directory called myCA in /var folder and copy servercert.pem, serverkey.pem, cacert.pem into that directory.

[root@fc4two misc]# cd /var
[root@fc4two var]# mkdir myCA
[root@fc4two var]# cd myCA/
[root@fc4two myCA]# cd /etc/pki/tls/misc/
[root@fc4two misc]# ls
CA c_hash c_info c_issuer c_name newcert.pem newreq.pem
[root@fc4two misc]# find / -name cacert.pem
/etc/pki/CA/cacert.pem

[1]+ Stopped find / -name cacert.pem
[root@fc4two misc]# cp /etc/pki/CA/cacert.pem /var/myCA/
[root@fc4two misc]# cd /var/myCA/
[root@fc4two myCA]# cp /etc/pki/tls/misc/newcert.pem ./servercert.pem
[root@fc4two myCA]# cp /etc/pki/tls/misc/newreq.pem ./serverkey.pem
[root@fc4two myCA]# ls
cacert.pem servercert.pem serverkey.pem

[root@fc4two myCA]#

We will now need to copy the certificates and keys to a directory where Apache can access it. For simplicity, we will overwrite the default certificates that come with the mod_ssl RPM package.

```
[root@localhost myCA]# cd /var/myCA
[root@localhost myCA]# cp servercert.pem
/etc/httpd/conf/ssl.crt/server.crt
cp: overwrite `/etc/httpd/conf/ssl.crt/server.crt'? y
[root@localhost myCA]# cp serverkey.pem
/etc/httpd/conf/ssl.key/server.key
cp: overwrite `/etc/httpd/conf/ssl.key/server.key'? y
```

[root@fc4two myCA]# find / -name server.crt

If you cannot find server.crt file, you should make this file as follows,

[root@fc4two share]# find / -name ssl.conf
/etc/httpd/conf.d/ssl.conf

[2]+ Stopped find / -name ssl.conf
[root@fc4two share]#
[root@fc4two share]#
[root@fc4two share]# vim /etc/httpd/conf.d/ssl.conf
[root@fc4two share]#
[root@fc4two share]# vim /etc/httpd/conf.d/ssl.conf

Here I edited ssl.conf file as follows (Bolded lines are edited)

####################edited ssl.conf file#########################

LoadModule ssl_module modules/mod_ssl.so
Listen 443
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl
SSLPassPhraseDialog builtin
SSLSessionCache shmcb:/var/cache/mod_ssl/scache(512000)
SSLSessionCacheTimeout 300
SSLMutex default
SSLRandomSeed startup file:/dev/urandom 256
SSLRandomSeed connect builtin
SSLCryptoDevice builtin
<VirtualHost _default_:443>
DocumentRoot "/var/www/ssl"

```
ServerName localhost.localdomain:443
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn
SSLEngine on
SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/private/server.key
<Files ~ "\.(cgi|shtml|phtml|php3?)$">
   SSLOptions +StdEnvVars
</Files>
<Directory "/var/www/cgi-bin">
   SSLOptions +StdEnvVars
</Directory>
SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
CustomLog logs/ssl_request_log \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
```

##

Here I am creating server.crt file in /etc/pki/tls/certs/ directory from servercert.pem that I have created in /var/myCA/ folder

```
[root@fc4two share]# cd /etc/pki/tls/certs/
[root@fc4two certs]# ls
ca-bundle.crt  localhost.crt  make-dummy-cert  Makefile
[root@fc4two certs]# cp /var/myCA/servercert.pem server.crt
[root@fc4two certs]# ls
ca-bundle.crt  localhost.crt  make-dummy-cert  Makefile  server.crt
[root@fc4two certs]# vim /etc/httpd/conf.d/ssl.conf
```

Now I am creating server.key file in /etc/pki/tls/private/ directory from /var/myCA/serverkey.pem

```
[root@fc4two certs]# cd /etc/pki/tls/private/
[root@fc4two private]# ls
localhost.key
[root@fc4two private]# cp /var/myCA/serverkey.pem server.key
[root@fc4two private]# ls
localhost.key  server.key
```

```
[root@fc4two private]# service httpd restart
Stopping httpd:                                [FAILED]
Starting httpd: Warning: DocumentRoot [/var/www/ssl] does not exist
Apache/2.0.54 mod_ssl/2.0.54 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server localhost.localdomain:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
                                               [  OK  ]
```

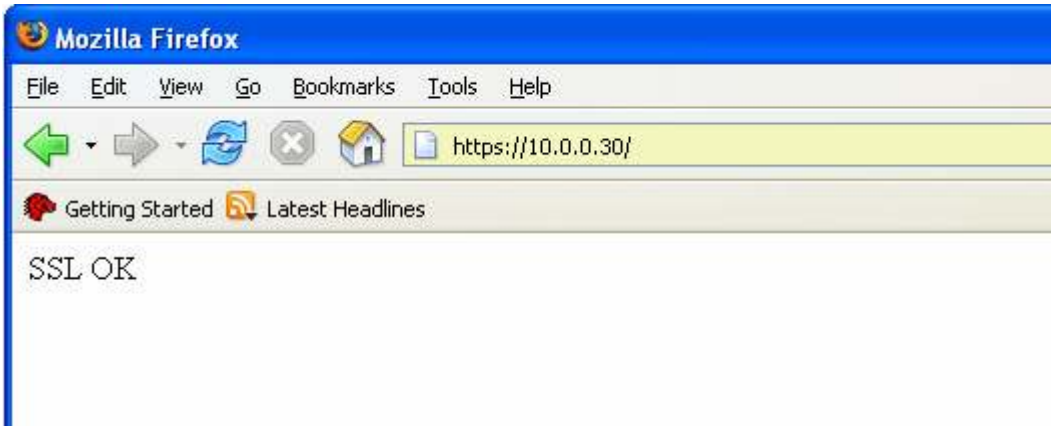
Sorry I forgot to create /var/www/ssl directory. This directory holds all the files under https connection. We can change this directory by editing DocumentRoot of ssl.conf

```
[root@fc4two private]# mkdir /var/www/ssl
[root@fc4two private]#
[root@fc4two private]# service httpd restart
Stopping httpd:                                [  OK  ]
Starting httpd: Apache/2.0.54 mod_ssl/2.0.54 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server localhost.localdomain:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
                                               [  OK  ]
[root@fc4two private]# vim /var/www/ssl/index.html
[root@fc4two private]#
```

I have created the webpage index.html as follows in the /var/www/ssl directory

```
<html>
<body>
SSL OK
</body>
</html>
```

Here we go, I typed https://<ipaddress-of-ssl-enabled-webserver> and got the following output.



Disabling the passphrase on startup (Optional)

Sometimes, the passphrase prompt can be inconvenient, especially when you want Apache to startup automatically on boot, without user intervention. We can disable the passphrase prompt by simply de-crypting the server key. To do this, we begin by making a copy of the server keyfile, then run the following command to decrypt it:

```
# cd /etc/pki/tls/private/
# cp server.key server.bak
# openssl rsa -in server.bak -out server.key
```

• If you find the access got refused. By examining the /etc/httpd/logs/ssl_error_log, we found the apache complains that it does not read in pass phrase.
[Tue Feb 15 13:31:40 2005] [error] Init: Unable to read pass phrase [Hint: key introduced or changed before restart?]
• The reason is that the server.key is actually protected by the password phrase. It contains the following content:

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,E0CD27BB7E7391A8

/B3JCpLAAL+Nx7UhWcFLtFSvZQYAAWTJKdn0PXkd4yww/m4hSzvAnIae8hs/RC3a
qVsjuuVL0KGOLwCvbTpKxC6NUCWgMjsxqzwwCOEOe839uarM1kfL1sszpqtTDMF
e
CZwP8yEBBkbe72+Txllq7TVkaKPwVFTmey4T8Bk2iWc5XqYQjefTBRvBsASOlTbq
7x0nWcsLbTuAYLZbWPMweWXkODfg7uXRHOH5OLaaeiI9wFXy8PRyZFaUtKr/IoI
y
e0ZVvveU3lBO+56y1KyGkys1ggqCwLvMGddVRpx7qoSL9epzjwQgaS3LndpwzIi0
HF1zaFOe3C66rkENMlh1Iz+bl2YD6bYIzYr0g3A0UZFIM70deuXH9OOioNt4+slb

Ck2zD1iqi8aIUz/CUgY3K5vTiTTuqeMCpvs5LmDOwmau3GfkwEG0Z6VAkifLmhrc
HbzLTk6q4KRCYuHhTq2snXbrxsICqvgSUJG90owgV/B3aQVpS59pi9tiuGK13xji
sH9UHF8NbTKTugApajWK7fUrVIF4duSDlilEgx+VG5KnbPErUWlJcP67emdRHk9Z
GucYpbyoNA5v1BXdcU2lmoBDQm9dyU6+eX0x6ZNopLYELM8rINr/zcFkJ4+xIz9B
eZg+EvrTyHfWRAeYVoj7yZKBLPZrnESC1Pztwx1xSZsPCkJ02H4BzzStZ0v/a3zh
KvfywfHl9hPVfUELfSaKQJaUsqAokEJeEhMlSqIhnjg7DKPqBeR7fw38fw/QUeKO
OSgfBEXAMMS+GA974rw1AvQrDlJOdpmFtB9XfkF+of2pStrcWJcuow==
-----END RSA PRIVATE KEY-----

The first two lines indicated the private key is encrypted with 3DEC encryption method and E0CD27BB7E7391A8 is the encrypted password.

if we use serverPrivateKey.pem as server.key directly, the apache web server will ask for pass phrase during the configuration time and fails if we use apachectl script.

- To avoid the problem of entering the pass phrase, we can remove the pass phrase protection on the server private key with the following openssl command.

[root@fc3c ssl.key]# copy server.key server.key.new
[root@fc3c ssl.key]# openssl rsa -in server.key.new -out server.key
Enter pass phrase for server.key.new:
writing RSA key
[root@fc3c ssl.key]# vi server.key

- -----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCjiktsmyUL31UmfktdO82doP1zCnsAgdTCfjQBCKLhip7s0l6d
XGEpBe+G4R3m0Z4nhSrofBOOls2+A9DQIykoz5fqOLdFWoqOxvwPzUWozoLlJ5LR
HMZ6UTD0SlorHWD4yCHyDg0gnb5tvtsfgGalM+cKuqx4yKVb6Qf7g8BZEQIDAQAB
AoGAF8N5sD9fA7mhEuXZJ6QP/a6uBXBpbSpwcw6JmfjaSjGtZDYxX2ZUC/T72DqP
8MFW3OFB2eRlpxuMq+a8CfKCAVupNh8rmEgIGn6qgjasledp4ifFOAwsekC7E3WY
hp1k1X8AUzZ6pS9c849zObhgWZ6vGCBvay00Se23ex7mQAECQQDO1jAA7CT8fjqA
FodWnK0K0UblKa8H1Vm4+4+x4+YV+UZUo11QCEUPkgg3QDki+zIQL9RwMrKK5rF7
eDOhpvwBAkEAymmPMi7EVRzmw8PFq2iVuCPhJt+HEWf9RAEg8RD/mBmwCUKueU/J
n+l9GDzOqlrdDgeDuXwfoop07pzL0x2dEQJAN0MW1EhDoYqAStS6GDQIL8m2bWfz
sd4Y+MmNnPPM97YASoDTX5y2BvD3bPulyGjg+V4uHkQNW/tDFEALW3doAQJBAMnv
zCvNmpQrdFJkc0XR3mTKburgYJlN/M+mrJ20TrsJDaX/f5+JqWa/g8z1hV1Rr246
swEPQ2Re68/OYE7sIXECQQCA5jLmCophfT54L3toKwWzRrwiLXdG/3x5qPQfsNv9
Q+jyRqVIQ46ecavqS803BYRUa2nId7Hx92sWAyhLBaJi
-----END RSA PRIVATE KEY-----

- Restart the web server with "apachectl restart"
- Start the firefox web browser
- When we type in the https://testsrv.ceylonlinux.com/, we should see the prompt for allowing us to exmine the server certificate.