Install Apache2 SSL protocol (https)

Installation

First you need to enable ssl.conf and ssl.load in /etc/apache2/mods-enabled. Enter the following command to check wether the module is already loaded:

```
a2enmod ssl
```

Certificates and Security

To set up your secure server, use public key cryptography to create a public and private key pair. In most cases, you send your certificate request (including your public key), proof of your company's identity, and payment to a Certificate Authority (CA). The CA verifies the certificate request and your identity, and then sends back a certificate for your secure server.

Alternatively, you can create your own self-signed certificate. Note, however, that self-signed certificates should not be used in most production environments. Self-signed certificates are not automatically accepted by a user's browser. Users are prompted by the browser to accept the certificate and create the secure connection.

Once you have a self-signed certificate or a signed certificate from the CA of your choice, you need to install it on your secure server. Types of Certificates

Most Web browsers that support SSL have a list of CAs whose certificates they automatically accept. If a browser encounters a certificate whose authorizing CA is not in the list, the browser asks the user to either accept or decline the connection. A self-signed certificate is not automatically recognized by most Web browsers.

Certificate Signing Request

To generate the Certificate Signing Request (CSR), you should create your own key. You can run the following command from a terminal prompt to create the key:

```
openssl genrsa -des3 -out server.key 1024
```

The minimum length when specifying -des3 is four characters, if you omit this option then no passphrase is asked. Next, run:

```
openssl req -new -key server.key -out server.csr
```

It will prompt you enter the passphrase. If you enter the correct passphrase, it will prompt you to enter Company Name, Site Name, Email Id, etc. Once you enter all these details, your CSR will be created and it will be stored in the server.csr file. You can submit this CSR file to a CA for processing. The CAN will use this CSR file and issue the certificate. On the other hand, you can create self-signed

certificate using this CSR.

Note: enter your domain name www.mydomain.com when asked for the common name.

Creating a Self-Signed Certificate

To create the self-signed certificate, run the following command at a terminal prompt:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out
server.crt
```

The above command will prompt you to enter the passphrase. Once you enter the correct passphrase, your certificate will be created and it will be stored in the server.crt file. Copy the generated key files as follows:

```
cp server.crt /etc/ssl/certs
cp server.key /etc/ssl/private
```

Apache2 configuration

You should add the following four lines to the /etc/apache2/sites-available/default file or the configuration file for your secure virtual host. You should place them in the VirtualHost section. They should be placed under the DocumentRoot line:

```
SSLEngine on
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

HTTPS should listen on port number 443. You should add the following line to the /etc/apache2/ports.conf file:

```
Listen 443
```

Accessing the Server

Once you install your certificate, you need to restart your web server. You can run the following command at a terminal prompt to restart your web server:

```
/etc/init.d/apache2 restart
```

Trouble shooting

Enter the following command to test the secure server working properly:

```
/usr/bin/openssl s_client -connect localhost:443
```

Output:

```
CONNECTED(00000003)
depth=0 /CN=Test-Only Certificate
verify error:num=18:self signed certificate
verify return:1
depth=0 /CN=Test-Only Certificate
verify return:1
---
Certificate chain
 0 s:/CN=Test-Only Certificate
   i:/CN=Test-Only Certificate
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICLzCCAZigAwIBAgIBADANBgkqhkiG9w0BAQQFADAgMR4wHAYDVQQDExVUZXN0
LU9ubHkgQ2VydGlmaWNhdGUwHhcNMDQxMTIyMTg0ODUxWhcNMDQxMjIyMTg0ODUx
WjAgMR4wHAYDVQQDExVUZXN0LU9ubHkgQ2VydGlmaWNhdGUwgZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAMEttnihJ7JpksdToPi5ZVGcssUbHn/G+4G43OiLhP0i
KvYuqNxBkSqqM1AanR0BFVEtVCSuq8KS9LLRdQLJ/B1UTMOGz1Pb14WGsVJS+38D
LdLEFaCyfkjNKnUgeKMyzsdhZ52pF9febB+d8cLmvXFve28sTIxLCUK7l4rjT3Xl
```

```
AgMBAAGjeTB3MB0GA1UdDgQWBBQ50isUEV6uFPZ0L4RbRm41+i1CpTBIBgNVHSME
QTA/gBQ50isUEV6uFPZ0L4RbRm41+i1CpaEkpCIwIDEeMBwGA1UEAxMVVGVzdC1P
bmx5IENlcnRpZmljYXRlggEAMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEEBQAD
gYEAThyofbK3hg8AJXbAUD6w6+mz6dwsBmcTWLvYtLQUh86B0zWnVxzSLDmwgdUB
NxfJ7yfo0PkqNnjHfvnb5W07GcfGgLx5/U3iUROOObYlwKlr6tQzMoysNQ/YtN3pp
52sGsqaOOWpYlAGOaM8j57Nv/eXogQnDRT0txXqoVEbunmM=
-----END CERTIFICATE-----
subject=/CN=Test-Only Certificate
issuer=/CN=Test-Only Certificate
---
No client certificate CA names sent
---
SSL handshake has read 1143 bytes and written 362 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES256-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : SSLv3
    Cipher    : DHE-RSA-AES256-SHA
    Session-ID:
56EA68A5750511917CC42A1B134A8F218C27C9C0241C35C53977A2A8BBB9986A
    Session-ID-ctx:
    Master-Key:
303B60D625B020280F5F346AB00F8A61A7C4BEA707DFA0ED8D2F52371F8C4F087FB6EFFC02CE
3B48F912D2C8929DB5BE
    Key-Arg   : None
    Start Time: 1101164382
```

```
    Timeout   : 300 (sec)
    Verify return code: 18 (self signed certificate)
---
GET / HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Mon, 22 Nov 2004 22:59:56 GMT
Server: Apache
Last-Modified: Mon, 22 Nov 2004 17:24:56 GMT
ETag: "5c911-46-229c0a00"
Accept-Ranges: bytes
Content-Length: 70
Connection: close
Content-Type: text/html
```

```
<html><head><title>Test</title></head><body>Test works.</body></html>
closed
```

From:
https://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
https://wiki.condrau.com/wserver:helssl

Last update: **2009/04/07 10:32**