Securing your Apache Web Server with a thawte Digital Certificate *with a thawte Digital Certificate A STEP-BY-STEP GUIDE to test, install and use a thawte Digital Certificate on your Apache Web Server...*

1. OVERVIEW

In this guide you will find out how to test, purchase, install and use a *thawte* Digital Certificate on your Apache web server. Throughout, best practices for set-up are highlighted to help you ensure efficient ongoing management of your encryption keys and digital certificates.

We will also touch on the role of *thawte* as a trusted third party and how using a *thawte* digital certificate can benefit your business by addressing unique online security issues to build customer confidence.

2. SYSTEM REQUIREMENTS

Before you can install an SSL certificate on your Apache web server you must have installed the required SSL components. You will need to install **OpenSSL**, as well as either **ModSSL** or **Apache-SSL**. OpenSSL and its cryptographic libraries provide the SSL back-end, while ModSSL or Apache-SSL provides the interface between Apache and OpenSSL. ModSSL and Apache-SSL are fairly similar and it is up to you to decide on which one to use - *thawte* makes no recommendation between the two.
In this guide we will assume that you are using Apache with ModSSL installed.

USEFUL WEBSITES:
www.apache.org
www.modssl.org
www.apache-ssl.org
www.openssl.org

3. GENERATE YOUR PRIVATE KEY

Use the OpenSSL binary to generate your private key. This key will be kept on your web server so we recommend that you follow the best practice of securing it with cryptographic protection using the following command:

“openssl genrsa –des3 1024 –out www.mydomain.com.key” This will tell OpenSSL to generate an RSA private key, 1024 bits in length and to encrypt this file using the Triple DES cipher and to pipe the output to a file called www.mydomain.com.key.

You will be prompted to enter a Privacy Enhanced Message (PEM) pass phrase when generating the Private Key file as well as to enter it a second time to verify the pass phrase set. An encrypted private key is secured with a pass phrase, and we recommend that this option be specified. Whenever the machine using this key is rebooted, or Apache is restarted, you will be prompted to enter this pass phrase.

Important Note

MAKE A BACKUP COPY OF THIS KEY FILE AND ITS PASS PHRASE! By far the most common problem users have when going through this process is related to Private Keys. If you lose or cannot access a Private Key or cannot remember the PEM pass phrase set on the Private Key file, you cannot use the Certificate we issue to you.To ensure this never happens, we advise that a backup of the Private Key file is made and that a note is made of the PEM pass phrase that is used to protect the Private Key file.

To copy the file to another location (in this case your a:\drive) use the following command:
“cp www.mydomain.com.key path-to-removable-disk”
If you get stuck or need further help, please go to:
“openssl genrsa --help”

4. GENERATING YOUR CERTIFICATE SIGNING REQUEST (CSR)

The next step is to create a CSR (Certificate Signing Request) which you will need to provide *thawte* before your certificate can be issued. To generate the CSR, use OpenSSL and your private key created in the previous step as follows:
“openssl req –new –key –out www.mydomain.com.csr”
This step creates a CSR that has the same “modulus” as the private key.
You will be prompted to input the following information when generating the CSR:
Country Name (2 letter code) [GB]: US
State or Province Name (full name) [Berkshire]: Texas
Locality Name (eg, city) [Newbury]: Dallas
Organization Name (eg, company) [My Company Ltd]: Widgets Inc.
Organizational Unit Name (eg, section) []: Widgets
Common Name (eg, your name or your server's hostname) []: www.mydomain.csr
E-Mail Address [Optional]:
These are the details that *thawte* will verify, so please make sure the details in your CSR match those of your company EXACTLY.

Note: You will be asked to enter the Privacy Enhanced Message (PEM) pass phrase. (This is the pass phrase that you set on the Private Key file generated in the previous step).

Important Note

The term "common name" is X.509 speak for the name that distinguishes the Certificate best, and ties it to your Organization. In the case of SSL Certificates, enter your exact host and domain name that you wish to secure.

One of the most common mistakes is to put the incorrect domain name in the Common Name Field in the CSR file. A certificate is tied to the domain name in it, so make sure that this field is populated with the exact fully qualified domain name you will use to access the secure portion of your site.

You **MUST NOT** include the ‘http://’ portion in the URL, or any directories that fall underneath this domain. For example, if your checkout is accessed via https://secure.mydomain.com/checkout, you will only include secure.mydomain.com in the Common Name field.

The CSR file created above is piped to a file called www.mydomain.csr and if you view the file it should look similar to this:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB2TCCAUICAQAwgZgxCzAJBgNVBAYTAlVTMRAwDgYDVQQIEwdHZW9yZ2lhMREwDwYDVQQHEwhDb2x1bWJ1
czEbMBkGB1UEChMSQUZMQUMgSW5jb3Jwb3JhdGVkMQswCQYDVQQLEwJJVDEYMBYGA1UEAxMPd3d3LmFmb
GFjbnkuY29tMSAwHgYJKoZIhvcNAQkBFhFKR2FybW9uQGFmbGFjLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjqAWg
YkCgYEAsRqHZCLIrlxqqh8qs6hCC0KR9qEPX2buwmA6GxegICKpOi/IYY5+Fx3KZWXmta794nTPShh2lmRdn3iwxwQR
KyqYKmP7wHCwtNm2taCRVoboCQOuyZjS+DG9mj+bOrMK9rLME+9wz1f8l0FuArWhedDBnI2smOKQID45mWwB0hkC
AwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBAJNlxhOiv9P8cDjMsqyM0WXxXWgagdRaGoa8tv8R/UOuBOS8/H
qu73umaB9vj6VHY7d9RKqDElFc/xlXeDwoXNiF8quTm43pmY0WcqnL1JZDGHMQkzzGtg502CLTHM
ElUGTdKpAK6rJCkucP0DKKEJKcmTySSnvgUu7m
-----END CERTIFICATE REQUEST-----
```

To view the CSR file, use either of the following commands:
cat www.mydomain.csr
vi www.mydomain.csr
You have just completed the three basic steps that will allow you to request an SSL certificate from *thawte*.

5. USING A TEST CERTIFICATE

This step assumes that SSL has been configured in Apache. If not, please refer to section 7 to set up configuration before proceeding.

To familiarize yourself with the workings of a *thawte* certificate on an Apache server, you can set up a test certificate on yourserver using a *thawte* test certificate.

While these certificates are for testing and evaluation only, they will provide encryption, but whenever an SSL session is established to your server with a test certificate installed, a warning message will be displayed.

This message informs the user connecting that the certificate is not Trusted, and as such the integrity of the site cannot be guaranteed.

These certificates are intended for you to test your server configuration before you buy a Trusted certificate from a CA (Certification Authority).

They will generate errors with browsers that have not manually inserted the required root certificate.

You can get your browser to Trust that test certificate by manually inserting the required root into your browser. Follow the instructions provided in the Wizard for installing the thawte Test CA Root Certificate by clicking on: http://www.thawte.com/roots/index.html

Our test certificates are valid for 21 days and this service comes with ABSOLUTELY NO WARRANTY!

You can request a thawte test certificate online from:
http://www.thawte.com/ucgi/gothawte.cgi?a=w46840165337049000

So far, you have created three files:
www.mydomain.key
-RSA private key
www.mydomain.csr
-Certificate Signing Request
www.mydomain.crt
-thawte Test Certificate File

You will be asked to copy and paste your CSR (Certificate Signing Request) into the text area provided on the Test Certificate System page.

Note: you will need to copy and paste the CSR, including the dashes and the full BEGIN and END line statements.

The Test Certificate will be generated immediately based on the CSR provided and you will be able to see it on the next page. Save the test certificate to a file called:
www.mydomain.com.crt

6. REQUEST A TRUSTED CERTIFICATE

thawte SSL certificates are requested online from:
https://www.thawte.com/buy/

During the certificate request process, you will be asked to copy anD paste your CSR (Certificate Signing Request) into a text area on theonline enrollment form.

Note: you will need to copy and paste the CSR, including the dashes and the full BEGIN and END line statements.

Important Note

Please ensure that you are submitting the correct CSR, if you have generated more than one. You can check your CSR with the following command:

"openssl req –text –noout –in csrfilename.csr"

You will have to provide all the requested information during the enrollment process, and send us documentation proving your, or your company's identity (a company registration certificate for instance).

You can view detailed instructions for obtaining a *thawte* SSL certificate at:
http://www.thawte.com/support/docs.html

Once you have completed the online request process, *thawte* will initiate a number of steps to verify your identity and the details you provided in the CSR. *thawte* performs a considerable amount of background checking before it issues a certificate. As a result, it may take a few days to verify your company identity and details, and issue the certificate.

During this verification period, you can track the progress of your request on your personal Status Page at: http://www.thawte.com/cgi/server/status.exe

Should you have any queries during this period, you may contact the Customer Service Representative assigned to your request. The details of the representative can be found on your Status Page at the URL above, under "*thawte* Contact Person".

7. SSL CONFIGURATION IN APACHE

Prior to installing test or “trusted” certificates, you will need to configure your Apache web server. ‘Directives’ are used to tell Apache exactly how it should behave in certain conditions, from how certain content is handled to telling Apache what your server’s name is.

Mod_ssl provides the directives used to configure SSL support on Apache, and the directives dealt with most often are listed below:

SSLCACertificateFile- specifies the path to a file which contains CA root certificates.
SSLCertificateFile- specifies the location of the SSL certificate to be used by a particular machine.
SSLCertificateKeyFile- path to the private key which corresponds to the file mentioned in the previous directive. **SSLEngine-** this directive controls whether or not SSL is switched on’ for a particular Virtual Host/Server.

Mod_ssl provides a whole host of directives that allow you to configure your server according to your particular needs. For a complete listing of the SSL directives Mod_ssl provides please consult the Mod_ssl documentation:
http://www.modssl.org/docs/2.2/ssl_reference.html

To configure Apache for SSL you will need to update your “httpd.conf” file to look for a new certificate. Open the “httpd.conf” configuration file and make sure that you have the ”SSLCertificateFile” and “SSLCertificateKeyFile” directives associated with the correct file paths.

For example, if you have your certificate in the “/usr/local/ssl/certs/” directory and your private key in the “/usr/local/ssl/private/” directory, then you will have the following in your “httpd.conf” file:

SSLCertificateFile: /usr/local/certs/www.mydomain.com.crt
SSLCertificateKeyFile: /usr/local/ssl/private/www.mydomain.com.key

You will also need to make sure your Apache Server as well as any firewall or routers that are in place are listening on Port 443 and “switch on” SSL with the “SSLEngine on” or SSLEnable directives in ModSSL or Apache-SSL respectively.

8. INSTALL YOUR CERTIFICATE

Once the certificate has been issued, you will be able to download it from your Status Page by clicking on the “Fetch Certificate” button (which only appears once the certificate has been issued). Save it within the appropriate directory in your “httpd.conf” file, this willmake key and certificate management easier.

For consistency, it is advisable to save it to a file called “www.mydomain.com.crt”. The certificate is stored in *thawte’s* database indefinitely, and can be downloaded again at any stage. Assuming you configured you web server, then you do not need to make any changes to your configuration file. You can simply copy the real (Trusted) certificate file over the test certificate. Once your certificate is issued you can open up Apache’s configuration file and install your certificate, as well as configure your SSL environment.

The certificate will look something like this:

-----BEGIN CERTIFICATE-----
MIIDBTCCAm6gAwIBAgIDcxV2MA0GCSqGSIb3DQEBBAUAMIGHMQswCQYDVQQGEwJaQTEiMCAGA1UECBMZRk9
SIFRFU1RJTkcgUFVSUE9TRVMgT05MWTEdMBsGA1UEChMUVGhhd3RlIENlcnRpZmljYXRpb24xFzAVBgNVBAsTDlR
FU1QgVEVTVCBURVNUMRwwGgYDVQQDExNUaGF3dGUgVGVzdCBDQSBSb290MB4XDTAyMTEwNTemJhDUzMlo
XDTAyMTEyNjE0MDUzMlowgd8
xFDASBgNVBAMTC3d3dy5jcmRiLmJlMRswGQYDVQQLHhIAQwBPAEwAVABAAarfJsdQBTAFQxczBxBgNVBAoeagB
DAGEAcAAgAEcAZQBtAGkAbgBpACAAVABlAGwAZQBjAG8AbQAgAE0BZqBkAGkAYQAgACYAIABOAGUAdAB3AG8
AcgBrAHMAIABCAGUAbABnAGkAdQBBkRpZWdlbTeCxMBUGA1UECBMOVmxhYW1zIEJyYWJhbnQxCzAJBgNVBAY
TAkJFMIGfMA0GCSqGsIb3DQEBAQUAA4GNADCBiQKBgQDaNm3HPzG6Rbk5Am0HI6JFHoDQku2/YmVMGbZk5AHe
R13QxIP7UVa08/k8qR3B7B0mfbxaNIxdwV9c7c1z4mZYQRfAeryoW4sU2jh1OHc4Cin+i9UarkHm8WnUnlcVZEnJrySdfL
ZNuxtbknXBNkca8rk6tnlbXodD3gEQJBMJtQIDAQABoyUwIzAT
-----END CERTIFICATE-----

When viewed with OpenSSL, using the following command:
“openssl req –text –noout –in www.mydomain.com.crt”
the certificate file contains the following details:

Certificate:
Data:
Version: 3 (0x2)
Serial Number: 645099 (0x9d7eb)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=thawte Consulting
cc, OU=Certification Services Division, CN=thawte Server CA/Email=servercerts@
thawte.com
Validity
Not Before: Dec 11 12:34:19 2002 GMT
Not After : Dec 11 12:34:19 2003 GMT
Subject: C=US, ST=Texas, L=Dallas, O=Widgets Inc., OU=Widgets,
CN=www.widgets.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (1024 bit)
Modulus (1024 bit):
00:b5:89:6c:cb:bb:9c:56:32:5f:77:5d:3d:9c:9c:
81:41:3d:8a:37:bc:4d:10:26:03:8c:f4:27:07:74:

88:a5:3a:d5:32:82:ab:1b:42:12:2a:bf:65:ad:b8:
b3:c7:f1:b0:ea:66:94:5e:82:ca:55:6e:26:c4:7f:
b0:5b:e5:22:b1:39:12:fd:a0:0d:cd:ef:59:56:95:
d3:33:14:da:f6:b8:c1:f8:d7:c1:05:32:d7:2d:90:
83:e6:91:f0:70:b1:d9:88:29:06:6a:45:02:17:aa:
df:1d:4b:56:d8:8d:ff:02:fc:22:20:e2:be:63:e5:
4e:09:e1:9c:97:24:91:ef:b1
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 Basic Constraints: critical
CA:FALSE
Signature Algorithm: md5WithRSAEncryption
97:48:b9:78:ca:66:f5:33:b9:3b:62:c2:52:26:04:8d:3f:e9:
32:ec:c9:e4:a2:fa:a5:b0:f8:df:10:5b:11:8b:36:97:62:e3:
82:63:20:93:7b:84:08:03:de:9e:a1:37:e3:12:e5:03:87:33:
f5:74:7e:84:9e:bb:52:bb:e3:8a:c1:a8:68:87:ad:8a:a4:95:
0d:61:98:4e:cd:da:13:fe:8c:0c:87:d4:7f:e6:18:3e:36:a4:
d1:ad:23:13:07:fc:bf:8c:bd:8a:42:32:e3:22:af:1b:7c:fb:
5e:d3:1a:94:f9:24:3c:4b:bd:3e:e9:f2:c6:9c:56:e4:b6:e2:
1e:6d

This certificate is tied to the private key that you created earlier (www.mydomain.com.key) and can only be ‘attached’ to this key. If you lose the private key to which a certificate is tied, your certificate is unusable. What you need to do now is point the SSLCertificateFile directive to the location where you have chosen to save this file, usually in the same directory as your “httpd.conf” file, /etc/apache or similar:

SSLCertificateFile: /etc/apache/www.mydomain.com.crt You also need to tell Apache which key file to use for this certificate, so don’t forget to point the SSLCertificateKeyFile directive to the private key for this cert: SSLCertificateKeyFile: /etc/apache/www.mydomain.com.key Certificate Authorities (CA) sign their certificates with a top level root, and any application wishing to verify an end users certificate needs to be able to cross check the users certificate against the Root certificate used by the CA. ModSSL uses the SSLCACertificateFile to do this, and this is included with ModSSL.

You should not need to customize the contents of this file. SSLCACertificateFile: /etc/apache/ca –bundle.crt So, now that you have configured all of those directives SSL should be working, right? Wrong. There is one more directive that requires some attention - SSLEngine. This directive has 2 arguments, “**on**” or “**off**”.

Obviously you would like SSL on:
SSLEngine on

The above directive can be used in a global server context or within the <VirtualHost> container. Assuming you are using the certificate on a correctly configured virtual host, you should have a configuration that looks like this:

<VirtualHost 192.168.1.22:443>
DocumentRoot /var/www/widgets
ServerName www.mydomain.com
ServerAdmin root@mydomain.com
ErrorLog /etc/httpd/logs/error_log
TransferLog /etc/httpd/logs/access_log
SSLEngine on
SSLCertificateFile /etc/httpd/conf/ssl.crt/www.mydomain.com.crt
SSLCertificateKeyFile /etc/apache/www.mydomain.com.key
SSLCACertificateFile /etc/apache/ca_bundle.crt
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
</VirtualHost>
You will note that in the <VirtualHost> container a specific port, port 443, is mentioned. This is the default SSL port and is configured using the global 'Listen' directive. By default 'Listen 80' is configured in the "httpd.conf" file, all you will need to do now is add 'Listen 443' on a new line. It is a good idea to group all similar directives together.

9. SECURING VIRTUAL HOSTS

If you have secure virtual hosts, each will need its own IP, as SSL does not support Name-based virtual hosts. SSL cannot be configured on Name Based VirtualHosts unless these VirtualHosts use different SSL ports.

Bear in mind that the configuration shown above is very basic and you can include many other SSL directives that allow you to customize your SSL environment.

Once the Certificate has been installed and SSL has been correctly configured, you will need to restart the entire server and not just the daemon. This ensures that the installation takes effect. The location of the scripts which will start Apache differ between the various Linux distributions, so we will assume that there is a script called 'apache' in /etc/init.d/ which calls a script in /usr/sbin/ called 'apachectl.'
widget@mydomain-pc/etc/init.d/apache start

You should now be able to access your machine securely and view the certificate details. You will know if the SSL session has been established if a golden padlock appears in the lower toolbar of your browser. Double clicking this icon will bring up the certificate details.

10. USEFUL URLs

Common problems experienced with Apache-SSL and Apache ModSSL are dealt with in our FAQs:

http://www.thawte.com/support/keygen/index.html
Key generation guide for Apache-SSL is available at the following url:
http://www.thawte.com/support/keygen/index.html
Key generation guide for Apache ModSSL is available at the following url:
http://www.thawte.com/support/keygen/index.html
The certificate enrollment process for Web Server Certificates and 128-bit SuperCerts begins at:
https://www.thawte.com/buy/
Instructions for generating a test certificate:
http://www.thawte.com/ucgi/gothawte.cgi?a=46840165337049000
Where to download the *thawte* Test CA Root Certificate:
https://www.thawte.com/roots/index.html

11. WHAT ROLE DOES thawte PLAY?

thawte Technologies is a Certification Authority (CA) which issues SSL Web Server Certificates to organizations and individuals worldwide. *thawte* verifies that the company ordering the certificate is a registered organization and that the person in the company who ordered the certificate is authorized to do so.

thawte also checks that the company in question owns the relevant domain. *thawte* digital certificates interoperate smoothly with Apache and the latest software from Microsoft and Netscape, so you can rest assured that you purchase of a *thawte* Digital Certificate will give your customers confidence in your system and integrity – they will feel secure about transacting online.

12. THE VALUE OF AUTHENTICATION

Information is a critical asset to your business. To ensure the integrity and safety of your information, it is important to identify with whom you are dealing, and the data you are receiving is trustworthy. Authentication can help establish trust between parties involved in all types of transactions by addressing a unique set of security issues including:

Spoofing:
The low cost of website design and the ease with which existing pages can be copied makes it all too easy to create illegitimate websites that appear to be published by established organizations. In fact, con artists have illegally obtained credit card numbers by setting up professional looking storefronts that mimic legitimate businesses.

Unauthorized Action:
A competitor or disgruntled customer can alter your website so that it malfunctions or refuses to service potential clients.

Unauthorized Disclosure:
When transaction information is transmitted “in the clear”, hackers can intercept the transmissions to obtain sensitive information from your customers.

Data Alteration:
The content of a transaction can be intercepted and altered en route, either maliciously or accidentally. User names, credit card numbers and currency amounts sent “in the clear” are all vulnerable to alteration.

13. CONTACT thawte

Should you have any further questions regarding the content of this guide or *thawte* products and services, please contact a Sales Advisor:

E-Mail: sales@thawte.com
Telephone: +27 21 937 8902
Fax: +27 21 937 8967

14. GLOSSARY OF TERMS

Apache
Apache, as it is commonly known, is a project of the Apache Software foundation that aims to produce a secure, efficient and extensible web server that provides HTTP services in sync with the current HTTP standards.
jakarta.apache.org

Asymmetrical Cryptography
A cryptographic method using a combined public and private key pair to encrypt and decrypt messages. To send an encrypted message, a user encrypts a message with the recipient's public key. Upon receipt, the message is decrypted with the recipient's private key.
Using different keys to perform the encryption and decryption functions is known as a trap-door one way function, that is, the public key is used to encrypt a message but it cannot be used to decrypt the same message. Without knowing the private key, it is practically impossible to reverse this function when modern strong encryption is used.

Certification Authority
A certificate authority (CA) is an organization (such as *thawte*) that issues and manages security credentials and public keys for message encryption.

Certificate Signing Request (CSR)
A CSR is a Public Key that you generate on your server that validates the computer-specific information about your web server and Organization when you request a Certificate from *thawte*.

Mod_ssl
As Apache is a modular application one of its strongest features is that it is highly customizable with third party modules that extend its features. One of the most popular (and in the e-commerce world, essential) modules created for Apache is Mod_ssl. Mod_ssl is a module that provides SSL support for Apache; without Mod_ssl Apache cannot serve SSL requests as it would not know what to do with them.

OpenSSL
OpenSSL is a cryptographic toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptographic standards required by them.
www.openssl.org

OpenSSL basically provides the platform upon which Mod_ssl runs, and must be installed on a machine where Apache + Mod_ssl are to be used. Without OpenSSL, Mod_ssl will not be of much use. Any utility/application that requires encryption capabilities will use OpenSSL's cryptographic libraries.

Private Key
A private key is numeric code used to decrypt messages encrypted with a unique corresponding public key. Integrity of encryption depends on the private key being kept secret.

Public Key
A public key is a numeric code which enables encryption of messages sent to the holder of the corresponding unique private key. The public key may be freely circulated without compromising encryption while increasing the efficiency and convenience of enabling encrypted communication.

Symmetric Cryptography
A cryptographic method where the same key is used for both encryption and decryption. This approach is handicapped by the security risks involved in secure distribution of the key since it must be communicated to and known by both sender and receiver without being disclosed to third parties.