



Защита Web-сервера Apache с помощью цифрового сертификата *thawte*

ПОШАГОВОЕ РУКОВОДСТВО по тестированию,
установке и использованию цифрового сертификата
thawte на Web-сервере Apache...

1. Обзор
2. Требования к системе
3. Создание секретного ключа
4. Создание своего запроса на подпись сертификата (CSR)
5. Использование тестового сертификата
6. Запрос доверительного сертификата
7. Настройка SSL на сервере Apache
8. Установка своего сертификата
9. Защита виртуальных хостов
10. Полезные адреса URL
11. О роли *thawte*
12. Значение аутентификации
13. Способы связи с *thawte*
14. Глоссарий терминов



1. Обзор

В данном руководстве содержатся сведения по тестированию, приобретению, установке и использованию цифрового сертификата *thawte* на Web-сервере Apache. Основное внимание в данном документе уделяется оптимальным методикам настройки, которые помогут обеспечить эффективное текущее управление ключами шифрования и цифровыми сертификатами.

Мы также остановимся на роли *thawte* как надежной третьей стороны, а также на деловых преимуществах цифровых сертификатов *thawte* за счет ориентированности на решение уникальных проблем сетевой защиты при обеспечении конфиденциальности для клиента.

2. Требования к системе

Перед установкой сертификата SSL на Web-сервер Apache требуется установить обязательные компоненты SSL. Потребуется установить **OpenSSL**, а также либо **ModSSL**, либо **Apache-SSL**. OpenSSL и соответствующие криптографические библиотеки реализуют прикладную часть SSL, а ModSSL или Apache-SSL обеспечивают интерфейс между Apache и OpenSSL. ModSSL отличается от Apache-SSL. ModSSL является полным пакетом SSL и имеет более подробную документацию по сравнению с альтернативным компонентом Apache-SSL, который также базируется на Openssl. Решение о том, какой компонент использовать, остается за Вами - *thawte* не дает рекомендаций по выбору какого-либо из этих компонентов.

В этом руководстве предполагается, что Apache используется вместе с установленным компонентом ModSSL.

ПОЛЕЗНЫЕ WEB-УЗЛЫ:

www.apache.org

www.modssl.org

www.apache-ssl.org

www.openssl.org

3. Создание секретного ключа

Для генерирования секретного ключа используется двоичный файл OpenSSL. Этот ключ будет храниться на Вашем Web-сервере, поэтому рекомендуется использовать наиболее оптимальный способ его защиты с помощью следующей команды:

```
“openssl genrsa –des3 –out 1024
```

Эта команда предписывает компоненту OpenSSL создать секретный ключ RSA длиной 1024 битов, зашифровать этот файл, используя тройной шифр DES, и выдать результат в файл с именем .

Будет отображен запрос на ввод пароля сообщения повышенной секретности (PEM) при генерировании файла секретного ключа, а также запрос на повторный ввод пароля для проверки правильности его назначения.

Зашифрованный секретный ключ защищается паролем - мы рекомендуем задать этот параметр. После перезагрузки компьютера, на котором используется этот ключ, или перезапуска Apache отображается запрос на ввод этого пароля.

Важное замечание

СОЗДАЙТЕ РЕЗЕРВНУЮ КОПИЮ ЭТОГО ФАЙЛА КЛЮЧА, А ТАКЖЕ ЕГО ПАРОЛЯ!

Затруднения, возникающие у пользователей при выполнении этого процесса, чаще всего связаны с секретными ключами. В случае утраты или отсутствия доступа к секретному ключу, либо если невозможно вспомнить пароль PEM, заданный в файле секретного ключа, выданный нами сертификат использовать невозможно. Во избежание таких ситуаций рекомендуется создать резервную копию файла секретного ключа, а также памятку с паролем PEM, который служит для защиты файла секретного ключа.

Для копирования файла в другое местоположение (в данном случае на дисковод a:\) используйте следующую команду:

```
“cp www.mydomain.com.key path-to-removable-disk”
```

В случае затруднений или необходимости в дополнительной справке введите следующую команду:

```
“openssl genrsa --help”
```



4. Создание своего запроса на подпись сертификата (CSR)

Следующим шагом является создание CSR (запроса на подпись сертификата), передача которого в *thawte* необходима для выдачи Вам сертификата. Для создания CSR используйте компонент OpenSSL и свой секретный ключ, созданный на предыдущем шаге:

```
"openssl req -new -key -out www.mydomain.com.csr"
```

На этом шаге создается CSR, который имеет такой же "модуль", что и секретный ключ. При создании CSR будет выдан запрос на ввод следующей информации:

Название страны (код из 2 букв) [GB]: US
 Название штата или области (полностью) [Беркшир]: Техас
 Название населенного пункта (например, города) [Ньюбери]: Даллас
 Название организации (например, фирмы) [Моя компания, ООО]: Widgets Inc.
 Название организационного подразделения (например, отдела) []: Widgets
 Общее имя (например, Ваше имя или имя хоста сервера) []: www.mydomain.com
 Адрес электронной почты [дополнительно]:

Эти сведения будут проверяться *thawte*, поэтому следует проконтролировать, что сведения в CSR В ТОЧНОСТИ совпадают с соответствующими сведения для Вашей компании.

Примечание. Будет отображен запрос на ввод пароля сообщения повышенной секретности (PEM). (Того пароля, который Вы задали для файла секретного ключа, созданного на предыдущем шаге).

Важное замечание

Термин "общее имя" применительно для X.509 употребляется для обозначения имени, которое служит наилучшим отличительным признаком сертификата и связывает его с Вашей организацией. В случае сертификата SSL введите точное имя своего хоста (т.е., www) и имя домена (т.е. mydomain.com), защиту которого требуется обеспечить.

Одной из наиболее распространенных ошибок является указание в поле 'Общее имя' в файле CSR неверного имени домена. Сертификат привязывается к имени хоста и домена в этом файле, поэтому следует убедиться в том, что в это поле введено точное и полное имя домена и имя хоста, которые используются для доступа к защищенной области Вашего Web-узла. **НЕ СЛЕДУЕТ** вводить часть 'http://' адреса URL, а также какие-либо каталоги, которые содержатся в этом домене. Например, если доступ к странице проверки выполняется по адресу <https://secure.mydomain.com/checkout>, в поле 'Общее имя' следует указать только `include secure.mydomain.com`.

Файл CSR, созданный как указано выше, передается в файл с именем `www.mydomain.csr`, и при просмотре этот файл будет иметь приблизительно такой вид:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB2TCCAUICAQAwwZgx CzAJBgNVBAYTAiVTMRAwDgYDVQQIEwdHZW9yZ2IhMREwDwYDVQQQ
HEwhDb2x1bWJ1czEbmBkGB1UEChMSQUZMQUMgSW5jb3Jwb3JhdGVkMQswCQYDVQQLEwJJV
DEYMBYGA1UEAxMPd3d3LmFmbGFjbkuY29tMSAwHgYJKoZIhvcNAQkBFhFKR2FybW9uQGFmb
GFjLmNvbTCBnzANBggqhkiG9w0BAQEFAAOBjQAwGyKCYEAsRqHZCLlrlxqqh8qs6hCC0KR9qEPX
2buwmA6GxegIcKpOi/IYY5+Fx3KZWXmta794nTPShh2lmRdn3iwxwQRKyqYKmp7wHCwtNm2taCRV
oboCQOuyZjS+DG9mj+bOrMK9rLME+9wz1f8l0FuArWhedDBnl2smOKQID45mWwB0hkCAwEAAaAA
MA0GCSqGSIb3DQEBAUAA4GBAJNixhOiv9P8cDjMsqyM0WXXxXWgagdRaGoa8tv8R/UOuBOS8/H
qu73umaB9vj6VHY7d9RKqDEIFc/xlXeDwoXNif8quTm43pmY0Wcqnl1JZDGHMQkzGtg502CLTHM
EIUGTdKpAK6rJKucP0DKKEJKcmTySSnvgUu7m
-----END CERTIFICATE REQUEST-----
```

Для просмотра файла CSR используйте на выбор одну из следующих команд:

```
cat www.mydomain.csr
vi www.mydomain.csr
```

Вы завершили выполнение трех основных шагов, которые позволяют передать запрос на получение сертификата SSL из *thawte*.

5. Использование тестового сертификата

Для знакомства с алгоритмами работы сертификата *thawte* на сервере Apache можно установить на сервере тестовый сертификат, используя тестовый сертификат *thawte*.

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

Хотя такие сертификаты предназначены только для тестирования и оценки работы, они обеспечивают шифрование (однако в начале каждого сеанса SSL с Вашим сервером при установленном тестовом сертификате отображается сообщение с предупреждением). Это сообщение извещает устанавливающего соединение пользователя о том, что этот сертификат не является доверительным сертификатом, а потому не может гарантировать неприкосновенность Web-узла.

Такие сертификаты предназначены для проверки конфигурации сервера перед приобретением доверительного сертификата в центре сертификации (CA).

Они генерируют ошибки в браузерах, для которых не был вручную введен требуемый корневой сертификат.

Для браузера можно указать, что данный тестовый сертификат является доверительным, вручную введя требуемый корневой сертификат в браузере. Откройте страницу по указанному адресу и следуйте инструкциям в мастере по установке тестового корневого сертификата CA *thawte*:
<http://www.thawte.com/roots/index.html>

Наши тестовые сертификаты действительны в течение 21 дня, причем данная услуга предоставляется БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ!

Тестовый сертификат *thawte* можно заказать по адресу в Интернет:
<http://www.thawte.com/ucgi/gothawte.cgi?a=w46840165337049000>

Вам будет предложено скопировать и вставить свой запрос на подпись сертификата (CSR) в текстовую область, предусмотренную на странице Test Certificate System (Система тестовых сертификатов).

Таким образом, Вы создали три файла:

www.mydomain.key
-секретный ключ RSA

www.mydomain.csr
-запрос на подпись сертификата

www.mydomain.crt
-файл тестового сертификата *thawte*

Примечание. Скопировать и вставить CSR следует полностью - вместе с тире и полными строками операторов BEGIN и END.

На основании переданного CSR незамедлительно генерируется тестовый сертификат, который можно просмотреть на следующей странице. Сохраните тестовый сертификат в файле с именем www.mydomain.com.crt

6. Запрос доверительного сертификата

Для запроса SSL-сертификата *thawte* заполните интерактивную форму на странице Интернет:

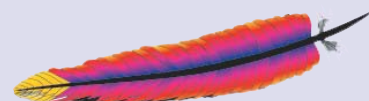
В процессе запроса сертификата Вам будет предложено скопировать свой запрос на подпись сертификата (CSR) в текстовую область в интерактивной форме запроса.

Примечание. Скопировать и вставить CSR следует полностью - вместе с тире и полными строками операторов BEGIN и END.

Важное замечание

В том случае, если сгенерировано несколько CSR, убедитесь в том, что скопирован требуемый CSR. Проверить CSR позволяет следующая команда:

```
“openssl req -text -noout -in csrfilename.csr”
```



В процессе заполнения запроса требуется указать всю запрашиваемую информацию и передать нам документы, подтверждающие Вашу подлинность (например, регистрационное свидетельство компании). С подробными инструкциями по получению SSL-сертификата *thawte* можно ознакомиться на следующей странице:

После завершения процесса заполнения интерактивного запроса *thawte* инициирует выполнение ряда шагов для проверки Вашей подлинности и сведений, которые Вы указали в CSR. Перед выдачей сертификата *thawte* подвергает серьезной проверке предоставленную информацию. Вследствие этого для проверки подлинности и сведений о компании перед выдачей сертификата может потребоваться несколько дней.

В период проверки Вы можете отслеживать рассмотрение своего запроса на личной странице состояния по адресу:

При возникновении вопросов в этот период можно обратиться к назначенному Вам представителю службы поддержки заказчиков. Сведения об этом представителе указаны на Вашей странице состояния по указанному выше адресу в разделе “*thawte* Contact Person” (лицо для контактов с *thawte*).

7. Настройка SSL на сервере Apache

Перед установкой тестовых или “доверительных” сертификатов требуется настроить Web-сервер Apache.

Для точного задания поведения Apache в определенных условиях, начиная от обработки определенного содержимого и заканчивая указанием для Apache имени своего сервера, используются директивы.

Mod_ssl предоставляет директивы, которые служат для настройки поддержки SSL для Apache. Наиболее часто используемые директивы перечислены ниже:

SSLCACertificateFile- задает путь к файлу, который содержит корневые сертификаты CA.

SSLCertificateFile- задает местоположение SSL-сертификата, который должен использоваться определенным компьютером.

SSLCertificateKeyFile- путь к секретному ключу, который соответствует файлу, указанному в предыдущей директиве.

SSLEngine- эта директива задает, ‘включен’ ли SSL для определенного виртуального хоста/сервера.

Mod_ssl предоставляет полный набор директив, которые позволяют настроить сервер в соответствии с конкретными требованиями. Полный список директив SSL, которые предоставляет Mod_ssl, содержится в документации Mod_ssl по адресу:

Для настройки Apache под использование SSL требуется обновить файл “**httpd.conf**” таким образом, чтобы выполнялся поиск нового сертификата. Откройте файл конфигурации “**httpd.conf**” и проверьте, что для директив “**SSLCertificateFile**” и “**SSLCertificateKeyFile**” правильно назначены пути к файлам.

Например, если сертификат содержится в каталоге “**/usr/local/ssl/certs/**”, а секретный ключ находится в каталоге “**/usr/local/ssl/private/**”, в файле “**httpd.conf**” будет содержаться следующий текст:

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

Следует также проверить, что сервер Apache, а также брандмауэр или маршрутизаторы, которые используются вместо него, прослушивают порт 443 и для них “включен” SSL с помощью директивы “**SSLEngine on**” или директивы **SSLEnable** в ModSSL или Apache-SSL соответственно.

8. Установка своего сертификата

После выдачи сертификата его можно загрузить со своей страницы состояния, нажав кнопку “Fetch Certificate” (Извлечь сертификат) (появляется только после выдачи сертификата). Файл сертификата можно скопировать в любой каталог по Вашему выбору. Для более удобного управления рекомендуется создать каталог Certificates, в котором сохранять все сертификаты. Это упростит настройку файла ‘http.conf’.

Для единообразия рекомендуется сохранить его в файле с именем “www.mydomain.com.crt”. Сертификат хранится в базе данных *thawte* неограниченно долго и может загружаться повторно на любом этапе.

Если Web-сервер уже был настроен, вносить изменения в файл конфигурации не потребуется. Можно просто скопировать файл настоящего (доверительного) сертификата поверх тестового сертификата.

После выдачи Вашего сертификата можно открыть файл конфигурации Apache и установить свой сертификат, а также настроить среду SSL.

Приблизительный вид сертификата представлен ниже.

```
-----BEGIN CERTIFICATE-----
MIIDBTCCAm6gAwIBAgIDcxV2MA0GCSqGSIb3DQEBAUAMIGHMQswCQYDVQQGEwJaQTEi
MCAGA1UECBMZk9SIFRFU1RJTkcqUFVSUUE9TRVMgT05MWTEdMBsGA1UEChMUUVGhhd3RlI
ENlcnRpZmljYXRpb24xFzAVBgNVBAsTDIRFU1QgVEVTVCBURVNUMRwwGgYDVQQDEExNUaG
F3dGUgVGVzdCBDQSBSb290MB4XDTAyMTEwNTemJhDUzMloXDTAyMTEyNjE0MDUzMlowgd8
xFDASBgNVBAMTC3d3dy5jcmRlLmJlMRswGQYDVQQQLHhIAQwBPAEwAVABAAarfJsdQBTAfQx
czBxBgNVBAoeagBDAGEAcAAgAEcAZQBtAGkAbgBpACAAVABIAGwAZQBjAG8AbQAQAE0BZq
BkAGkAYQAQACyAIABoAGUAAdAB3AG8AcgBrAHMAIABCAGUAbABnAGkAdQB
BkRpZWdlbTeCxMBUGA1UECBMOVmxhYW1zIEJyYWJhbnQxQzAJBgNVBAYTAkFJFMIGfMA0GC
SqGSIb3DQEBAQUAA4GNADCBiQKBgQDaNm3HPzG6Rbk5Am0HI6JFHODQku2/YmVMGbzK5A
HeR13QxIP7Uva08/k8qR3B7B0mfbxaNlxdwV9c7c1z4mZYQRfAeryoW4sU2jh1OHc4Cin+i9UarkH
m8WnUnIcVZEnJrySdfLZNuxtbnXBNkca8rk6tnlbXodD3gEQJBMJtQIDAQABoyUwIzAT
-----END CERTIFICATE-----
```

При просмотре с помощью следующей команды OpenSSL:
 “openssl req -text -noout -in ” файл сертификата содержит следующие сведения:

Сертификат:

Данные:

Версия: 3 (0x2)

Серийный номер: 645099 (0x9d7eb)

Алгоритм подписи: md5WithRSAEncryption

Выдан: C=ZA, ST=Western Cape, L=Cape Town, O=thawte Consulting cc, OU=Certification Services Division, CN=thawte Server CA/Email=servercerts@thawte.com

Срок действия

не ранее: 11 декабря 2002 г 12:34:19 GMT

не позднее: 11 декабря 2003 г 12:34:19 GMT

Субъект: C=US, ST=Texas, L=Dallas, O=Widgets Inc., OU=Widgets, CN=www.widgets.com

Информация об открытом ключе субъекта:

Алгоритм личного ключа: rsaEncryption

Открытый ключ RSA: (1024 разрядов)

Modulus (1024 разрядов):

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

Порядок: 65537 (0x10001)

расширения X509v3:

использование расширенных ключей X509v3:

Аутентификация Web-сервера TLS

Основные ограничения X509v3: критические

CA:FALSE

Алгоритм подписи: md5WithRSAEncryption

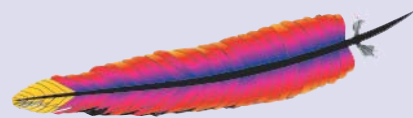
97:48:b9:78:ca:66:f5:33:b9:3b:62:c2:52:26:

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



Этот сертификат привязывается к секретному ключу, который был создан ранее (`www.mydomain.com.key`), и может быть ‘прикреплен’ только к этому ключу. В случае утери секретного ключа, к которому привязан сертификат, этот сертификат становится непригодным для использования.

Теперь следует указать с помощью директивы `SSLCertificateFile` местоположение, которое было выбрано для сохранения этого файла - обычно оно совпадает с местоположением файла `“httpd.conf”`, т.е. каталогом `/etc/apache` или каким-либо другим каталогом.

`SSLCertificateFile: /etc/apache/www.mydomain.com.crt`

Требуется также указать серверу Apache, какой из файлов ключей использовать для этого сертификата, поэтому задайте с помощью директивы `SSLCertificateKeyFile` ссылку на секретный ключ для этого сертификата:

`SSLCertificateKeyFile: /etc/apache/www.mydomain.com.key`

Центры сертификации (CA) подписывают свои сертификаты с помощью корневых сертификатов верхнего уровня, и любое приложение, которое должно проверять сертификаты конечных пользователей, должно иметь возможность перекрестной проверки сертификатов пользователей на соответствие корневому сертификату, выданному CA. В ModSSL с этой целью используется `SSLCACertificateFile`, и эта директива включена в ModSSL.

Изменять содержимое этого файла не требуется. Эта директива используется при установке сертификата SSL123 или SGC SuperCert и ссылается на промежуточный сертификат, с помощью которого подписан выданный сертификат.

`SSLCACertificateFile: /etc/apache/cacertificate.crt`

Теперь, после настройки всех этих директив, SSL должен заработать, верно? Не верно. Существует еще одна директива, которую следует упомянуть - `SSLEngine`. Эта директива имеет 2 аргумента: `“on”` (включено) и `“off”` (выключено). Очевидно, что Вам требуется включить SSL:

SSLEngine on

Указанная выше директива может использоваться как в глобальном контексте сервера, так и внутри контейнера `<VirtualHost>`.

Предположим, что сертификат используется для надлежащим образом настроенного виртуального хоста. Тогда конфигурация должна выглядеть примерно так:

```
<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/cacertificate.crt
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
</VirtualHost>
```

Обратите внимание на то, что в контейнере `<VirtualHost>` указан определенный номер порта, 443. Этот порт является портом SSL по умолчанию и настраивается с помощью глобальной директивы `‘Listen’`. По умолчанию в файле `“httpd.conf”` содержится директива `‘Listen 80’`. Теперь требуется лишь добавить директиву `‘Listen 443’` в новой строке. Рекомендуется группировать похожие директивы в одном месте файла.

9. Защита виртуальных хостов

Для защиты виртуальных хостов каждый такой хост должен иметь собственный IP-адрес, т.к. SSL не поддерживает виртуальные хосты с обращением по именам.

SSL невозможно настроить для VirtualHosts с доступом по именам, если только эти VirtualHosts не используют разные порты SSL.

Следует помнить, что приведенная выше конфигурация содержит только базовые сведения, и возможно включение многих других директив SSL, которые позволяют настроить среду SSL.

После установки сертификата и надлежащей настройки SSL требуется полностью перезагрузить сервер, а не только данную программу-демон. Перезагрузка обеспечит вступление установки в силу. Местоположение сценариев, обеспечивающих запуск Apache, отличается для различных установок Linux, поэтому предположим, что в каталоге `/etc/init.d/` существует сценарий с именем `'apache'`, который вызывает сценарий с именем `'apachectl'` из каталога `/usr/sbin/`.

```
widget@mydomain-pc/etc/init.d/apachectl startssl
```

Теперь имеется возможность безопасного доступа к данному компьютеру и просмотра сведений сертификата. На установление сеанса SSL будет указывать символ в виде золотого висячего замка, отображаемый в нижней строке инструментов браузера. При двойном щелчке мышью на этом значке отображаются сведения сертификата.

10. Полезные адреса URL

Распространенные ошибки, связанные с Apache-SSL и Apache ModSSL, рассматриваются в нашем разделе “Часто задаваемые вопросы”:
<http://www.thawte.com/support/keygen/index.html>

Руководство по созданию ключей для Apache-SSL см. на следующей странице:
<http://www.thawte.com/support/keygen/index.html>

Руководство по созданию ключей для Apache ModSSL см. на следующей странице:
<http://www.thawte.com/support/keygen/index.html>

Процесс передачи запроса на выдачу сертификата для сертификатов Web-серверов и 128-разрядных сертификатов SuperCerts инициируется на следующей странице:
<https://www.thawte.com/buy/>

Инструкции по созданию тестового сертификата:
<http://www.thawte.com/ucgi/gothawte.cgi?a=w46840165337049000>

Страница, с которой можно загрузить тестовый корневой сертификат CA *thawte*:

11. О роли *thawte*

thawte Technologies является центром сертификации (CA), который выдает сертификаты SSL для Web-серверов организациям и частным лицам по всему миру. *thawte* проверяет, что заказавшая сертификат компания является зарегистрированной организацией и что лицо, заказавшее сертификат от имени этой компании, имеет соответствующие полномочия.

thawte также проверяет, что данная компания владеет соответствующим доменом. Цифровые сертификаты *thawte* полностью совместимы с серверами Apache и новейшим программным обеспечением Microsoft и Netscape, поэтому приобретение цифрового сертификата *thawte* позволяет завоевать доверие заказчиков к Вашей системе и решает проблему неприкосновенности данных – при обращении к Вам по сети обеспечивается высокая степень защищенности.

12. Значение аутентификации

Информация является критически важным компонентом жизнедеятельности предприятия. Для обеспечения неприкосновенности и защиты данных важно точно знать, с кем Вы имеете дело, а также быть уверенным в том, что полученные данные являются подлинными. Аутентификация помогает установить надежные отношения между сторонами, участвующими во всех типах транзакций, позволяя обнаруживать целый ряд злоупотреблений, в том числе:

Доступ путем обмана:

Низкая стоимость проектирования Web-узлов и простота копирования существующих страниц позволяет легко создавать незаконные Web-узлы, которые выглядят как созданные известными организациями. В действительности искусные аферисты незаконно узнают номера кредитных карт, создавая электронные витрины, внешне имитирующие легальные коммерческие предприятия.

Несанкционированные действия:

Конкуренты или недовольные покупатели могут изменить Ваш Web-узел таким образом, что он будет неверно функционировать или отказывать в обслуживании потенциальным заказчикам.

Несанкционированное разглашение:

При передаче информации о транзакции “открытым текстом” хакеры могут перехватить передаваемые данные для получения от Ваших заказчиков важной информации.

Подмена данных:

Содержимое транзакции может быть перехвачено и изменено на пути передачи как намеренно, так и случайно. Имена пользователей, номера кредитных карт и данные о денежных суммах, передаваемые “открытым текстом”, находятся под угрозой изменения.

13. Способы связи с *thawte*

С вопросами по содержанию этого руководства или продуктам и услугам *thawte* обращайтесь к консультанту по продажам:

Электронная почта: sales@thawte.com
Тел.: +27 21 937 8902

14. Глоссарий терминов

Apache

Apache является широко известным проектом Apache Software Foundation, который имеет своей целью создание защищенного, эффективного и расширяемого Web-сервера, предоставляющего услуги HTTP в соответствии с текущими стандартами HTTP.

jakarta.apache.org

Асимметричное шифрование

Метод шифрования, в котором для шифрования и дешифрования сообщений используется пара из открытого и секретного ключа. Для передачи зашифрованного сообщения пользователь шифрует сообщение с помощью открытого ключа получателя. После получения сообщения оно дешифруется с помощью секретного ключа получателя.

Функции шифрования и дешифрования, использующие для шифрования и дешифрования разные ключи, называются защитной однонаправленной функцией. Т.е. открытый ключ используется для шифрования сообщения, но не может использоваться для дешифрования этого сообщения. Не зная секретный ключ, практически невозможно раскодировать информацию при использовании современных мощных алгоритмов шифрования.

Центр сертификации

Центр сертификации (CA) - это организация (например, *thawte*), которая выдает реквизиты защиты и открытые ключи для шифрования сообщений, а также заведует этими данными.

Запрос на подпись сертификата (CSR)

CSR представляет собой открытый ключ, который Вы создаете на своем сервере, и который проверяет подлинность относящейся к компьютеру информации о Вашем Web-сервере и организации при выполнении запроса сертификата у *thawte*.

Mod_ssl

Apache является модульным приложением, и одной из его сильных сторон является высокая степень настройки под пользователя путем включения расширяющих его возможности модулей сторонних разработчиков. Одним из наиболее распространенных (и важных для электронной коммерции) модулей, созданных для Apache, является Mod_ssl.

Mod_ssl является модулем, который обеспечивает поддержку SSL для Apache; без Mod_ssl приложение Apache не может обслуживать запросы SSL, поскольку ему не задана процедура их обработки.

OpenSSL

OpenSSL является набором инструментов для криптографии, в котором реализованы сетевые протоколы защищенных сокетов (SSL v2/v3) и защиты транспортного уровня (TLS v1), а также необходимые для этих протоколов соответствующие криптографические стандарты.

В основном OpenSSL обеспечивает платформу, на которой выполняется Mod_ssl, и его следует устанавливать на компьютер, на котором будут использоваться Apache и Mod_ssl. Без OpenSSL модуль Mod_ssl практически бесполезен. Любая служебная программа/приложение, где требуются возможности шифрования, использует криптографические библиотеки OpenSSL.

OpenSSL

OpenSSL является набором инструментов для криптографии, в котором реализованы сетевые протоколы защищенных сокетов (SSL v2/v3) и защиты транспортного уровня (TLS v1), а также необходимые для этих протоколов соответствующие криптографические стандарты.

В основном OpenSSL обеспечивает платформу, на которой выполняется Mod_ssl, и его следует устанавливать на компьютер, на котором будут использоваться Apache и Mod_ssl. Без OpenSSL модуль Mod_ssl практически бесполезен. Любая служебная программа/приложение, где требуются возможности шифрования, использует криптографические библиотеки OpenSSL.

Секретный ключ

Секретный ключ - это цифровой код, используемый для дешифрования сообщений, зашифрованных соответствующим уникальным открытым ключом. Неприкосновенность зашифрованных данных обеспечивается секретным ключом, который не разглашается.

Открытый ключ

Открытый ключ - это цифровой код, который позволяет шифровать сообщения, передаваемые держателю соответствующего уникального секретного ключа. Открытый ключ можно легко вычислить без ущерба для безопасности шифрования, и в то же время он повышает эффективность и удобство связи с использованием шифрования.

Симметричная криптография

Метод шифрования, при котором для шифрования и дешифрования используется один и тот же ключ. Этот подход имеет тот недостаток, что создает угрозу безопасности при распространении ключа, поскольку он должен быть передан и известен и отправителю, и получателю, но не должен раскрываться третьей стороне.