

UNIZETO



GENERAL  
CERTIFICATION AUTHORITY



Руководство пользователя

# APACHE 2.0 + SSL – Linux

Использование сертификатов в программном обеспечении  
APACHE 2.0 + SSL – Linux

версия 1.3

# Содержание

<b>1. ВСТУПЛЕНИЕ .....</b>	<b>3</b>
<b>2. СОЗДАНИЕ СЕРТИФИКАТА.....</b>	<b>3</b>
2.1. ФОРМИРОВАНИЕ ЗАПРОСА НА СЕРТИФИКАТ (CSR).....	3
2.2. СОЗДАНИЕ СЕРТИФИКАТА НА ОСНОВАНИИ СОЗДАННОГО ЗАПРОСА (CSR).....	5
2.3. ИМПОРТ СЕРТИФИКАТОВ.....	7
<b>3. ИНСТАЛЛИРОВАНИЕ КЛЮЧЕЙ И СЕРТИФИКАТОВ.....</b>	<b>9</b>
3.1. ИНСТАЛЛИРОВАНИЕ СЕРТИФИКАТОВ CERTUM .....	9
3.2. ИНСТАЛЛИРОВАНИЕ ЗАКРЫТОГО КЛЮЧА.....	9
3.3. ИНСТАЛЛИРОВАНИЕ СЕРТИФИКАТА СЕРВЕРА .....	10
<b>4. АВТОРИЗАЦИЯ СЕРВЕРА НА ОСНОВАНИИ СЕРТИФИКАТА .....</b>	<b>10</b>
<b>5. ОБСЛУЖИВАНИЕ БРАУЗЕРОВ ДЛЯ МНОГОЗНАЧНЫХ АДРЕСОВ .....</b>	<b>11</b>

## 1. Вступление

Apache является наиболее развитым сервером WWW и можно его скачать в виде исходного кода. Этот сервер, благодаря модулю modssl, имеет поддержку сильной криптографии.

Доля Apache в мировом рынке серверов WWW решительно преобладает, а его популярность, благодаря доступности для разных платформ, постоянно растёт.

Чтобы конфигурировать Apache поддерживаемый SSL, необходимы следующие компоненты:

1. Apache - <http://httpd.apache.org>
2. OpenSSL - <http://www.openssl.org/>
3. mod\_ssl – <http://www.modssl.org/>

Если Ваш дистрибутив Linuxa не имеет вышеуказанных компонентов, скачайте и загрузите их.

**ВНИМАНИЕ:** В Apache 1.3 компонент mod\_ssl необходимо установить как отдельный пакет. А Apache 2.0 может быть интегрирован с mod\_ssl.

## 2. Создание сертификата

### 2.1. Формирование запроса на сертификат (CSR)

Чтобы генерировать ключи для Apache'a, используется утилита – Openssl, которую можно скачать с сайта: <http://openssl.org>.

1. После инсталляции библиотеки Openssl, формируем требование:

```
openssl genrsa -des3 -out server.key 2048
```

Это требование вызовет генерацию закрытого ключа для нашего сервера под названием *server.key*. Этот ключ будет иметь длину 1024 бита и будет зашифрован симметрическим алгоритмом 3des. Во время генерации ключа программа попросит нас вписать пароль, который предохранит компонент.

```
debian:~# openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.+++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
```

Файл CSR вместе с закрытым ключом *server.key* необходимо сохранить на дискете или другом носителе.

2. После успешной генерации закрытого ключа, формируем требование:

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

Результатом этого является запрос на сертификат сервера CSR, которое будет сохранен в файле *server.csr*. Следует помнить, об указании файла с закрытым ключом *server.key*. Во время

формирования запроса CSR необходимо ввести пароль, предохраняющий закрытый ключ, данные, связанные с нашей фирмой и сайт www:

**Country (C)** – символ страны, состоящий из 2 букв (PL). Необходимо использовать код ISO, напр. правильным кодом Польши является PL (большие буквы), а не pl или RP.

**State / Province (ST)** – название области, напр.: Московская. Нельзя использовать сокращения.

**Locality (L)** – название города или деревни, напр.: Цекиновка, Городок, Иванов.

- **Organization Name (O)** – полное название организации / фирмы, напр.: Моя фирма
- **Organizational Unit (OU)** – если есть такая необходимость, можно заполнить поле, вставляя название отдела напр. Отдел в Моя фирма
- **Common Name (CN)** – очень важное поле. Здесь должно быть полное название DNS (fqdn) сервера напр.: www.test.com.pl pop3.test.net.
- **Email (Email)** – вписать почтовый адрес администратора сервера напр.: adminivanov@firma.ru.

**Помните**, что в поле **Common Name** необходимо вписать адрес нашего веб-сайта

- напр. mojsserver.com, mojdomen.ru, www.mojsajt.com.ru – в случае однозначного адреса:

```
debian:~# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
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]:RU
State or Province Name (full name) [Some-State]:Sankt-Petersburg
Locality Name (eg, city) []:Sankt-Petersburg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Web section
Common Name (eg, YOUR name) []:exampledomain.ru
Email Address []:root.mydomain@gmail.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1111
An optional company name []:Private Company
```

- напр. \*.mojsserver.com, \*.mojdomen.ru, \*.mojsajt.com.ru – в случае многозначного адреса:

```
debian:~# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
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]:RU
State or Province Name (full name) [Some-State]:Sankt-Petersburg
Locality Name (eg, city) []:Sankt-Petersburg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Web section
Common Name (eg, YOUR name) []:*.exampledomain.ru
Email Address []:root.mydomain@gmail.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1111
An optional company name []:Private Company
```



### Купить сертификат Enterprise SSL с периодом действительности (1 год)

#### Запрос сертификата

В нижеследующее поле вставьте запрос сертификата (CSR)\*.

Запрос сертификата можно сгенерировать:

- пользуясь генератором, представленным на сайте CERTUM,
- на сервере домена, для которого хотите получить сертификат.

```
fj
+VdemsApGUbQ5zdwKhQxCd6KMC6NoQUY/62saRsiivFiFNZ9IL5I92D0/G8CK
/Ah
xX7nLr7nu3xUhSiqzIlbv783tBYaQ4IjWFXW/AqkrryHVX4NIQcLceS4AN6i/UJ
G
J5uwbsbm3jNrYNH2pqDPpTpwZHj8dKeqNz5cmH8CAmItB9GXbc0DeiG1w2Q
QScKH
fiPRJgPlofpx0hKfyw+PE/GuFIMQ/rbsh0NNpRePagP/IwXdvN2rDva2BpuFeY
B0
-----END CERTIFICATE REQUEST-----
```

#### Адрес e-mail

E-mail\*:

mojadres@mail.ru

#### Верификация домена

Выберите email адрес, на который будет отправлена ссылка. При переходе по ссылке Вы, тем самым, подтвердите, что имеете доступ к аккаунту, указанному в CSR запросе. Ниже находится список доступных для администратора аккаунтов. Выберете адрес к которому имеете доступ.

Верификация e-mail\*:

admin@domain.ru

На этом шаге происходит верификация прав доступа к серверу, доступному в домене, указанном в CSR запросе. Можете выбрать один из двух методов верификации: используя вставку META tag в содержании главной страницы или сохранив сгенерированный файл в корневой папке сервера, доступного под доменным именем, указанным в CSR запросе.

Верификация домена\*:

Запиши файл HTML

#### Заявление

ПЕРЕД ПОДАЧЕЙ ЗАЯВКИ НА ВЫПУСК СЕРТИФИКАТА, ЕГО ПОДТВЕРЖДЕНИЯ ИЛИ ИСПОЛЬЗОВАНИЯ ДЛЯ ПЕРВОЙ ПОДПИСИ – ВЫ ДОЛЖНЫ ПРОЧИТАТЬ ТЕКСТ НАСТОЯЩЕГО ЗАЯВЛЕНИЯ. ЕСЛИ ВЫ НЕ СОГЛАСНЫ С УСЛОВИЯМИ НАСТОЯЩЕГО ЗАЯВЛЕНИЯ, НЕ ПОДАВАЙТЕ ЗАЯВКУ НА ВЫПУСК СЕРТИФИКАТА, НЕ ПОДТВЕРЖДАЙТЕ И НЕ ИСПОЛЬЗУЙТЕ ЕГО.

Заявление является обязательным с момента подачи заявки на выпуск сертификата до CERTUM – Открытого Удостоверяющего Центра. Подавая заявку на выпуск сертификата, Вы требуете от выдающего органа рассмотреть заявку и выпустить сертификат:

Подтверждаю заявление\*

\* - обязательное поле

Заказываю



(размещение сертификата в нашем репозитории, доступным на сайтах www):

Инсталляционное ID сертификата: f00d56ad1edb4fd3d00bbe237fbe9772d66ab0cf

Введите ID на сайте:

<https://www.certum.eu/ru/install/>

Коллектив ANK и Certum

[SSL@ank-pki.ru](mailto:SSL@ank-pki.ru)

Входим на сайт, вклеиваем ID и, нажимая Далее, активируем сертификат:

### Инсталлирование сертификата

Впишите ID инсталляции сертификата, который Вы получили в сообщении e-mail от CERTUM:

**Внимание!**

В случае сертификатов e-mail инсталляция подписи должна происходить на том же самом компьютере и с помощью того же самого браузера, который Вы использовали указывая адрес e-mail.

Появится сайт, с которого сможем скачать наш сертификат в бинарном или текстовом виде. Нажимаем на *Записать в текстовом виде*:

### Инсталлирование сертификата

<b>Enterprise SSL</b>	действителен до: 30.04.2010
Субъект: exampledomain.ru Email: root.mydomain@gmail.com Номер: 0x493DC	
<input type="button" value="Записать бинарно"/>	<input type="button" value="Записать текстово"/>

## 3. Установка ключей и сертификатов

### 3.1. Установка сертификатов CERTUM

Кроме нашего сертификата, необходимо еще установить на сервере сертификаты CERTUM (сертификаты CERTUM в одном пакете находятся по адресу <http://www.certum.pl/keys/ca-bundle.crt>). В пакете находятся все сертификаты CERTUM: все промежуточные сертификаты (в очередности от Level I до Level IV), и root CA.

Чтобы установить сертификаты root CA и промежуточные сертификаты сохраняем (с уровня Midnight Commandera или командной строки) файл с нашим пакетом *ca-bundle.crt* в папку, где будем его хранить, напр. в:

```
/usr/share/ssl/certs/ca-bundle.crt
```

Запись в файле *ssl.conf* будет выглядеть следующим образом:

```
SSLCACertificateFile /usr/share/ssl/certs/ca-bundle.crt
```

После введения изменений перезагружаем сервер *www* с помощью напр. такой команды:

```
#httpd restart
```

Установка сертификата *root CA* и промежуточных сертификатов закончена успешно.

Для нашего удобства в файле *ca-bundle.crt* можно поместить в начале файла сертификат нашего сервера (сохраняем содержание с файла *Nr\_certyfikatu.pem* и вставляем в начало файла *ca-bundle.crt*).

### 3.2. Установка закрытого ключа

Чтобы установить закрытый ключ на сервере, необходимо сохранить (с уровня Midnight Commandera или командной строки) файл с закрытым ключом *server.key* в папку, в которой его будем хранить, напр.:

```
/etc/httpd/conf/ssl.key/server.key
```

Запись в файле *ssl.conf* будет выглядеть следующим образом:

```
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server.key
```

Снимаем пароль с закрытого ключа, (чтобы при каждой перезагрузке Apache не спрашивал нас пароль):

```
openssl rsa -in server.key -out server.key
OpenSSL> rsa -in server.key -out server.key
Enter pass phrase for server.key:
writing RSA key
OpenSSL>
```

Защищаем ключ перед считыванием:

```
#chmod 400 /etc/httpd/conf/ssl.key/server.key
```

После введения изменений перезагружаем сервер *www* с помощью напр. такой команды:

```
#httpd restart
```

Установка закрытого ключа закончена успешно.

### 3.3. Инсталлирование сертификата сервера

После того, как вклеите ID на наших сайтах, нам будет возвращен сертификат нашего сервера www. Необходимо его сохранить мышкой в произвольный текстовый редактор и записать его как напр. server.crt.

**ВНИМАНИЕ:** Чтобы вклеить сертификат с сайта необходимо копировать фрагмент текста от линии "--BEGIN CERTIFICATE --" до "--END CERTIFICATE--", используя для этого текстовый редактор напр. Notepad и мышку.  
**Не используйте для этой операции Word, или другой текстовый процессор!**

В случае, если бы файл с сертификатом сервера был утрачен, необходимо помнить, что он находится на первом месте в файле *ca-bundle.crt* (откуда можно будет его просто копировать). Альтернативно – вы можете найти интересующий вас сертификат в хранилище на нашем сайте.

Чтобы инсталлировать сертификат сервера сохраняем (с уровня Midnight Commandera или командной строки) файл с сертификатом в папку, в которой будем его хранить, напр. в:

```
/etc/httpd/conf/ssl.crt/server.crt
```

Запись в файле *ssl.conf* будет следующая:

```
SSLCertificateFile /etc/httpd/conf/ssl.crt/server.crt
```

После введения изменений перезагружаем сервер www с помощью напр. такой команды:

```
#httpd restart
```

Инсталлирование сертификата сервера закончено успешно.

После конфигурации сервера DNS к обслуживанию нашего домена (или субдомена) www, сервер будет обслуживать сертификаты, как однозначных, так и многозначных адресов (если добавим виртуальные хосты – описание в п. 5).

Если Вы не имеете собственного сервера DNS контактируйте со своим провайдером и представьте ситуацию.

**ВНИМАНИЕ:** Ключи и сертификаты могут также храниться в одном файле. Для этого необходимо к файлу *ca-bundle.crt* добавить закрытый ключ и соответственно изменить запись в конфигурационном файле *ssl.conf*.

```
SSLCertificateFile /sciezka_do_pliku/ca-bundle.crt
```

```
SSLCACertificateFile /sciezka_do_pliku/ca-bundle.crt
```

```
SSLCertificateKeyFile /sciezka_do_pliku/ca-bundle.crt
```

## 4. Авторизация сервера на основании сертификата

Чтобы принудить клиента к использованию сертификата в файле *SSL.conf* дописываем две строки:

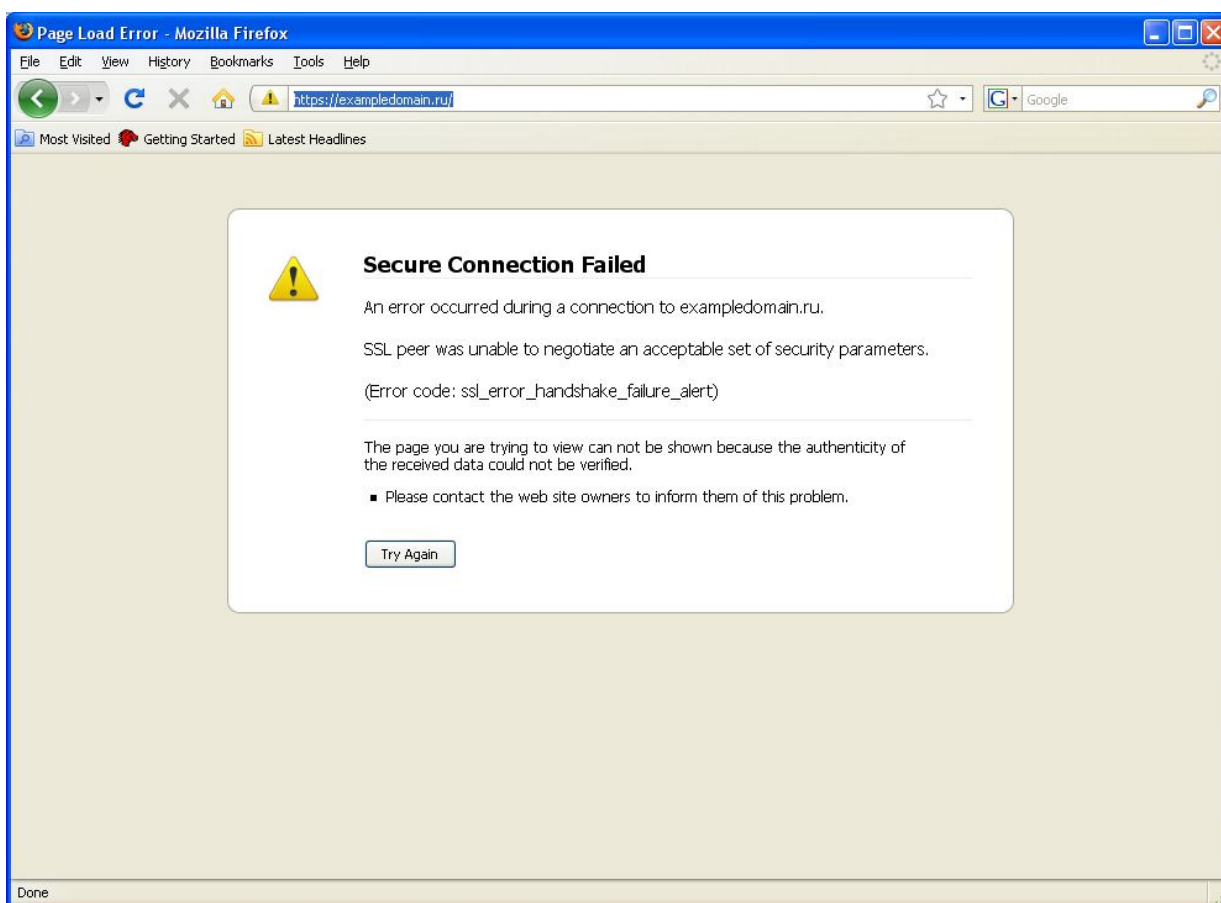
```
SSLVerifyClient require (требование обязательного использования сертификата клиентом)
```

`SSLVerifyDepth 10` (определяет максимальную длину цепочки сертификации)

Чтобы ограничить доступ к сайту для конкретных пользователей, напр. имеющих сертификат Certum Level III, с серийным номером 02F110 добавляем к файлу `ssl.conf` в секции `Location` запись:

```
<Location />  
SSLRequire ( %{SSL_CLIENT_I_DN_CN} eq "Certum Level III" and  
             %{SSL_CLIENT_M_SERIAL} eq "02F128")  
</Location>
```

В случае, если клиент не имеет прав к веб-сайту, то появится следующий сайт:



Сохраняем файл `SSL.conf` и перезагружаем сервер:

```
#httpd restart
```

Детальную информацию на эту тему Вы найдете на: <http://httpd.apache.org/>

## 5. Обслуживание браузеров для многозначных адресов

Чтобы на одном сервере запустить много виртуальных субдоменов, обслуживаемых нашими сертификатами Wildcard, необходимо внести несколько изменений в файле `ssl.conf`. Для этого открываем файл в редакторе и добавляем запись:

```
NameVirtualHost adres_ip_naszego_serwera:443
```

- Секция VirtualHost для первого виртуального хоста:

```
<VirtualHost adres_ip_naszego_serwera:443>
```

в этой папке размещаем наши файлы www:

```
DocumentRoot /var/www/html1
```

название DNS виртуального хоста с сертификатом \*.mojserver.ru:

```
ServerName poddomena1.mojserver.ru
```

обслуживание шифрованных сессий:

```
SSLEnable
```

остальное остается без изменений:

```
...  
</VirtualHost>
```

- 2. Секция VirtualHost для второго виртуального хоста:

```
<VirtualHost adres_ip_naszego_serwera:443>
```

```
DocumentRoot /var/www/html2
```

```
ServerName poddomen2.mojserver.ru
```

```
SSLEnable
```

```
...  
</VirtualHost>
```

Записываем изменения и перезагружаем сервер :

```
#httpd restart
```

После конфигурации сервера DNS к обслуживанию нашего домена (если не имеете собственного сервера DNS контактируйте со своим провайдером и представьте ситуацию), сервер будет подготовлен к обслуживанию сертификатов для многозначных адресов.

Чтобы проверить действие виртуальных серверов запускаем сервер:

```
#httpd start
```

и вписываем в браузере:

- <https://poddomena1.mojserver.ru>
- <https://poddomena2.mojserver.ru>

Появление характерного замка внизу экрана:



обозначает шифрованную сессию.

В случае проблем, стоит определить проблему, используя специальные инструменты типа *ntop*, *ps*, *netstat* или *openssl s\_client*.