eSMART®

Авторизация по сертификату на веб-сайте

Настройка сервера

Содержание

1. Общая информация

В руководстве описана настройка SSL-аутентификации на базе трех основных веб-серверов Apache, Nginx и IIS.

Тип сервера	Операционная система	Доменное имя	IP адрес
Сервер IIS 7	Windows Server 2008 R2	myiis.local	10.1.1.5
Сервер Apache 2.2 или выше	Ubuntu 11.4	myapache.local	10.1.1.10
Сервер Nginx	Ubuntu 11.4	mynginx.local	10.1.1.11

В качестве рекомендуемой конфигурации рекомендуется использовать двухуровневую систему, в которой внешним интерфейсом (называемым frontend) выступает nGinx, который полностью берет на себя проверку сертификатов клиентов. Использование nGinx в качестве прокси-сервера позволит добавить SSL-аутентификацию в работающий проект практически без изменения текущей конфигурации.

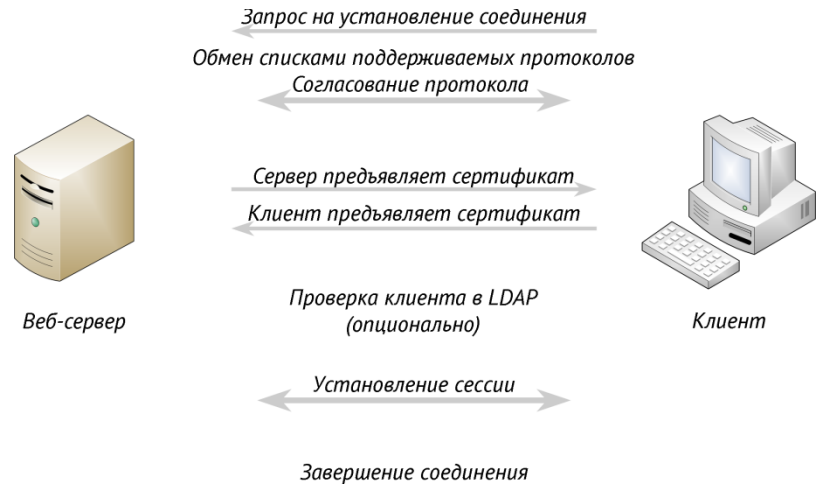
При работе с Windows Server 2008 требуются права администратора домена (группа Domain Admin) или администратора предприятия (группа Enterprise Admin).

Для работы с Linux требуются права root.

2. Протокол SSL

Криптографический протокол SSL использует ассиметричную криптографию для обеспечения безопасности связи. SSL-соединение по умолчанию использует порт 443.

В руководстве рассматривается следующий алгоритм работы авторизации по сертификату:



3. Сертификат сервера

3.1 Способы получения сертификата сервера

Для получения сертификата веб-сервера существуют следующие возможности:

- *Покупка SSL-сертификата (Verisign, Thawte и др.);*
- *Корпоративный центр сертификации на базе Windows Server;*
- *Центр сертификации OpenSSL.*

Каждый способ имеет свои преимущества и недостатки.

Для раздела сайта компании, который предназначен для посетителей, необходим сертификат от удостоверяющего центра, специализирующегося на выдаче SSL-сертификатов, например Thawte, Verisign, GeoTrust и др. Корневой сертификат такого удостоверяющего центра уже по умолчанию присутствует в операционной системе или браузере (Mozilla Firefox поддерживает собственный список корневых доверенных сертификатов).

Корпоративный центр сертификации на базе Windows Server предлагает гибкие настройки и удобный интерфейс управления пользователями и их сертификатами. Центр сертификации на базе Windows Server является наиболее перспективным решением для компаний, которые активно используют Active Directory.

Бесплатное приложение OpenSSL с открытым исходным кодом также может использоваться в качестве центра сертификации. OpenSSL позволяет создать самоподписанный сертификат корневой центра сертификации, сертификат веб-сервера, сертификаты пользователей и список отозванных сертификатов. Данный вариант подходит для тестирования и проектов, в которых не используется Microsoft Active Directory.

Оптимальным может стать вариант, когда сертификат сервера приобретен у специализированого удостоверяющего центра, а проверка сертификата клиента осуществляется внутренним центром на базе Windows Server или OpenSSL.

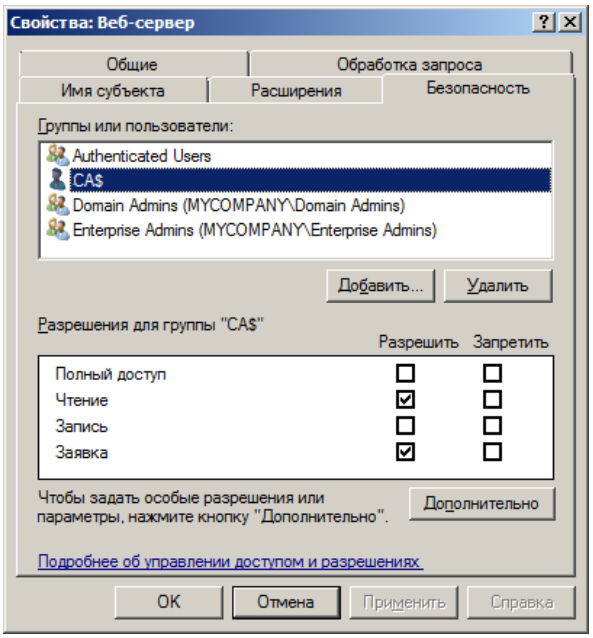
В руководстве в качестве примера описано получение сертификата веб-сервера в центре сертификации Windows Server 2008. Интеграция центра сертификации mina Enterprise с Active Directory позволяет провести дополнительную проверку учетной записи пользователя по протоколу LDAP.

3.2 Пример получения сертификата сервера

Сертификат веб-сервера в центре сертификации на базе Windows Server 2008 можно получить любым из следующих способов:

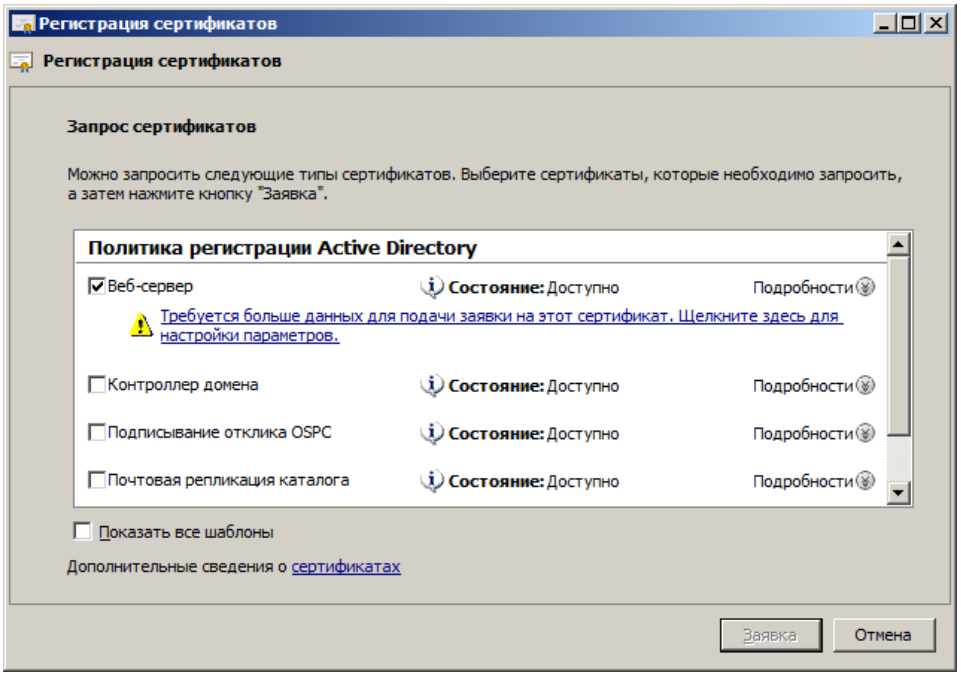
- *через консоль mmc с оснасткой для компьютера;*
- *через веб-интерфейс;*
- *через панель управления сервером IIS.*

***Важно:** шаблон сертификата веб-сервер (Web Server) может быть недоступен для заявки через консоль mmc или панель управления сервером IIS. Необходимо проверить, имеет ли право на запрос сертификата та машина, с которой выполняется запрос.*



Рассмотрим процесс получения сертификата с извлекаемым закрытым ключом для тестового доменного имени mynginx.local через консоль mmc. Этот способ позволяет получить наиболее гибкие параметры настройки при выдаче сертификата.

Добавьте в консоль mmc оснастку Сертификаты (Certificates) для локального компьютера. В папке личное в контекстном меню выполните ***Все задачи › Запросить новый сертификат*** *(All tasks › Request new certificate).*

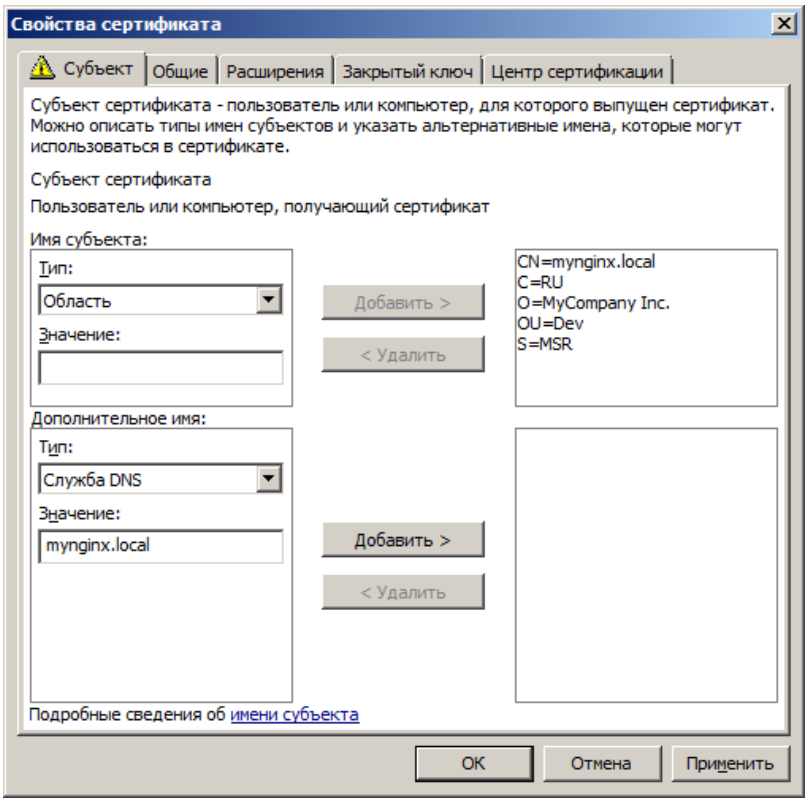


Данные для сертификата веб-сервера не могут быть получены из Active Directory, поэтому их необходимо ввести вручную. Нажмите на ссылку с предупреждением, чтобы перейти к редактированию параметров.

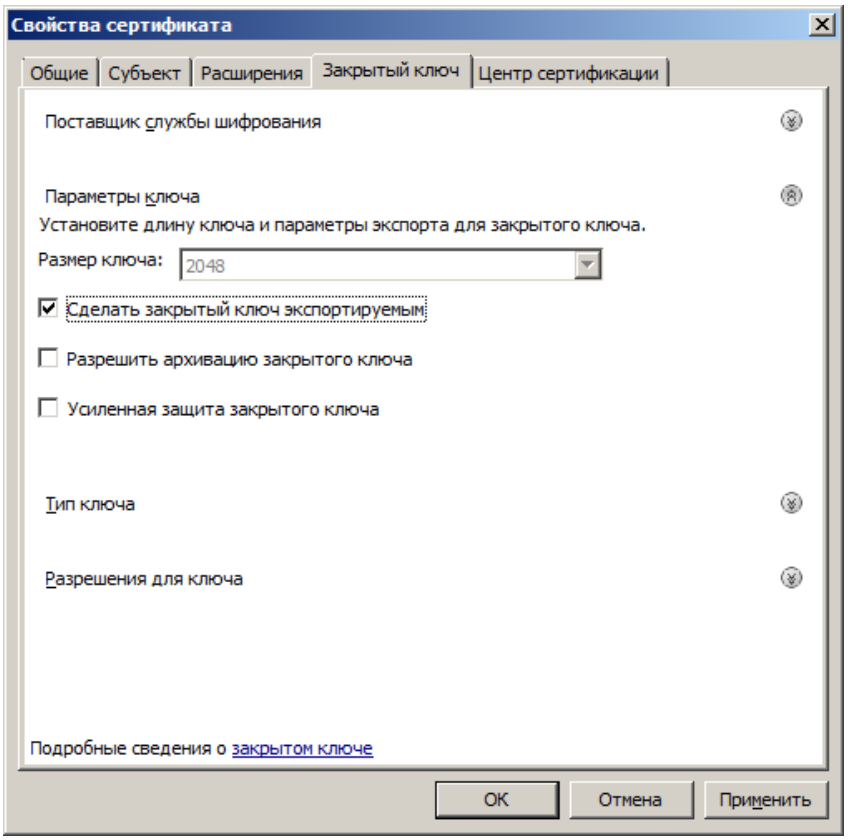
В появившейся форме заполните данные имени субъекта сертификата. С помощью выпадающих списков заполните параметры субъекта.

Внимание! *В качестве значения Общее имя (Common name) следует ввести доменное имя сайта. Кроме того, доменное имя можно добавить в качестве альтернативного имени субъекта (SAN – Subject Alternative Name). Для этого в списке параметров альтернативного имени выберите* ***Служба DNS*** *(DNS Service) и впишите доменное имя.*

Если параметр SAN содержит доменное имя сайта, параметр обязательного имени ***Общее имя*** *(Common name) можно оставить пустым. Все современные браузеры корректно интерпретируют доменное имя в параметре SAN. Тем не менее, для совместимости со старыми браузерами рекомендуется вписать доменное имя в поле* ***Общее имя*** *(Common name).*



Чтобы перенести сертификат в формате PFX (PKCS#12) на другой сервер, на котором будет работать nGinx, закрытый ключ сертификата можно сделать извлекаемым. Для этого перейдите во вкладку ***Закрытый ключ*** *(Private Key) и отметьте опцию* ***Сделать закрытый ключ экспортируемым****.*



*Сохраните изменения. В окне выбора сертификатов нажмите **Заявка** (Enroll) и дождитесь выполнения операции. Проверьте наличие сертификата с помощью оснастки mmc.*

Если сертификат будет использоваться на другой машине, экспортируйте сертификат в файл .pfx с надежным паролем.

4. Подготовительные этапы

4.1 Сертификат пользователя

Оптимальным вариантом при авторизации по SSL-сертификату является запись сертификата пользователя на смарт-карту.

*Приложение ESMART PKI Client позволяет упростить процесс выдачи клиентских сертификатов. Процедура записи сертификата пользователя на смарт-карту описана в руководстве **ESMART PKI Client – Руководство Администратора**.*

Администратор, который занимается выдачей сертификатов, должен иметь сертификат агента подачи заявок. Получение сертификата по данному шаблону описано в руководствах по развертыванию центра сертификации на базе Windows Server 2003 или 2008.

4.2 Установка ESMART PKI Client

Для работы со смарт-картами и USB-ключами ESMART Token рекомендуется использовать графическое приложение ESMART PKI Client. Установка ESMART PKI Client описана в руководстве администратора к данному приложению.

4.3 Настройка Mozilla Firefox

*Веб-браузер Mozilla Firefox в Windows использует для работы со смарт-картами открытый стандарт PKCS#11. Чтобы авторизоваться на сайте по сертификату на смарт-карте, требуется дополнительно настроить браузер Mozilla Firefox. См. руководство администратора **Настройка пользовательских приложений**. Для браузеров Internet Explorer или Google Chrome дополнительных действий не требуется.*

4.4 Модуль php5 для работы с LDAP

В руководстве описана возможность дополнительной проверки учетной записи пользователя в Active Directory по протоколу LDAP. Обычно LDAP принимает входящие соединения на порт 389.

Для работы web-сервера по протоколу LDAP требуется модуль php5_ldap и предложенные зависимые модули. Для установки модуля можно использовать следующую команду:

```
sudo apt-get install php5-ldap
```

5. Скрипт для обновления CRL-файла

Сервер IIS может получать список отозванных сертификатов для проверки сертификатов клиентов непосредственно из Active Directory.

nGinx и Apache требуется представить список отозванных сертификатов в локальном файле.

Задача обновления списка отозванных сертификатов выполняется запуском скрипта по расписанию (cron). Скрипт получает список отозванных сертификатов по http, конвертирует в требуемый формат и сохраняет в файл. Для перевода двоичного сертификата в сертификат в формате PEM требуется OpenSSL.

5.1 Пример скрипта загрузки списка CRL

```
#!/bin/sh
      ### Подставьте свои значения переменных ###
      # Адрес публикации списка отозванных сертификатов
CRL_URL="http://mycompany.local/Mycompany-Root-Ent-CA.crl"
      # Локальный путь на веб-сервере к папке, в которой хранятся
      # и обрабатываются сертификаты
CRL_PATH=/etc/ssl/certs/
      # Название файла в двоичной форме
CRL_NAME_DER=mycompany.crl
      # Название файла в кодировке base64
CRL_NAME_PEM=mycompany.crl.pem
      # Команда на перезагрузку сервера
      # для nGinx: опция reload позволяет перезагрузить сервер
      # без прерывания его работы
WEB_SERVER_RELOAD="/etc/init.d/nginx reload"
      # Получение текущего значения MD5
LAST_MD5=`/usr/bin/md5sum $CRL_PATH/$CRL_NAME_DER`
      # Скачивание последней версии списка отозванных сертификатов
/usr/bin/wget -o /dev/null -O "$CRL_PATH/$CRL_NAME_DER" "$CRL_URL"
      # Вычисление нового значения MD5 и сравнение со старым значением
NEW_MD5=`/usr/bin/md5sum $CRL_PATH/$CRL_NAME_DER`
if [ "$LAST_MD5" != "$NEW_MD5" ]
then
# Преобразование двоичного файла в PEM (base64) при помощи OpenSSL
/usr/bin/openssl crl -in "$CRL_PATH/$CRL_NAME_DER" -inform DER -out
"$CRL_PATH/$CRL_NAME_PEM" -outform PEM
      # Перезапуск веб-сервера
/bin/sh $WEB_SERVER_RELOAD
fi
```

Чтобы адаптировать данный файл для использования с Apache, следует заменить путь к сертификатам и команду перезапуска сервера:

```
WEB_SERVER_RELOAD="/etc/init.d/apache2 restart"
```

Регулярность выполнения скрипта определяет то, насколько часто центр сертификации публикует списки отозванных сертификатов.

5.2 Подключение по LDAP

Модуль php5_ldap используется для проверки учетной записи пользователя сертификата в Active Directory по протоколу LDAP для проектов, использующих PHP5.

Для проверки настроек в коде страницы можно вызвать функцию phpinfo(). Требуемая информация будет выведена в разделе LDAP.

Перезапустите веб-сервер.

Для проверки доступности ldap на Linux можно использовать пакет ldap utils:

```
apt-get install ldap-utils
```

Команда для проверки доступности Active Directory по протоколу LDAP:

```
ldapsearch -D "Dmadmin@mycompany.local" -x -W -b "dc=mycompany,dc=local" -h
server-ca.mycompany.local
```

6. OpenSSL

Для работы с сертификатами может использоваться бесплатное кроссплатформенное приложение OpenSSL. Основные способы использования OpenSSL:

- *Создание ключевой пары RSA;*
- *Создание сертификатов формата x509, запросов на сертификаты;*
- *Извлечение закрытого ключа из формата .pfx (PKCS#12);*
- *Конвертация форматов сертификатов (двоичный и base64);*
- *Вычисление хэш-сумм;*
- *Проверка работы серверов и клиентов HTTPS посредством* `s_client`.

6.1 Работа с PFX-файлами

Если файл сертификата получен в формате PKCS#12 (формат файла .pfx), потребуется разделить его на 2 отдельных файла:

- *файл закрытого ключа;*
- *файл сертификата с открытым ключом.*

Извлечение сертификата, содержащего открытый ключ:

```
openssl pkcs12 -in domain.pfx -clcerts -nokeys -out domain.cer
```

Извлечение закрытого ключа в защищенном виде:

```
openssl pkcs12 -in domain.pfx -nocerts -out domain_encrypted.key
```

Веб-сервер может использовать зашифрованные ключи, но при каждой перезагрузке придется вводить пароль. Снятие защиты с зашифрованного файла закрытого ключа:

```
openssl rsa -in domain_encrypted.key -nodes -out domain.key
```

6.2 Конвертация PEM - DER

Для перекодировки сертификата, цепочки сертификатов или списка отозванных сертификатов используются следующие команды OpenSSL

PEM-->*DER*

```
openssl x509 -inform PEM -in cert.pem -outform DER -out cert.cer
```

DER-->PEM

```
openssl x509 -inform DER -in cert.cer -outform PEM -out cert.pem
```

Для цепочки сертификатов

```
openssl pkcs7 -inform DER -outform PEM -in certificate.p7b -out
certificate.pem
```

Для списка отозванных сертификатов

```
OPENSSL crl -inform DER -in ca.crl -outform PEM -out ca.crl
```

Чтобы хранить в одной директории файлы одновременно файлы в двоичном и текстовом виде с одинаковым названием, к расширению текстового файла можно добавить .pem, например, usercert.cer.pem

6.3 Проверка работы HTTPS

Для проверки SSL-соединения можно воспользоваться командой OpenSSL

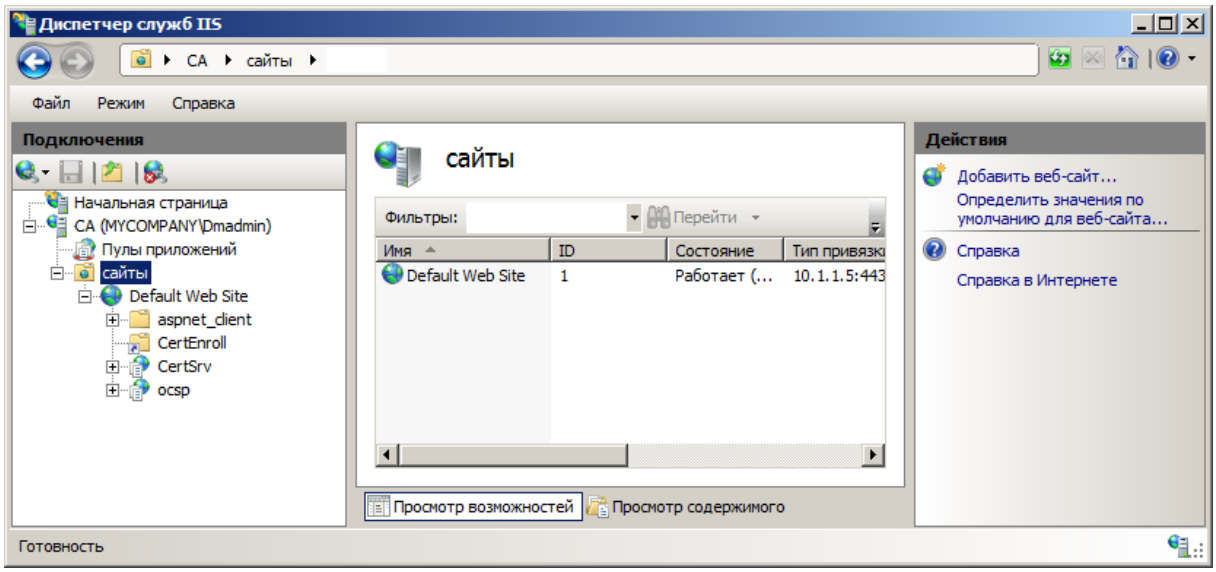
openssl s_client -connect example.com:443

7. Веб-сервер Microsoft IIS

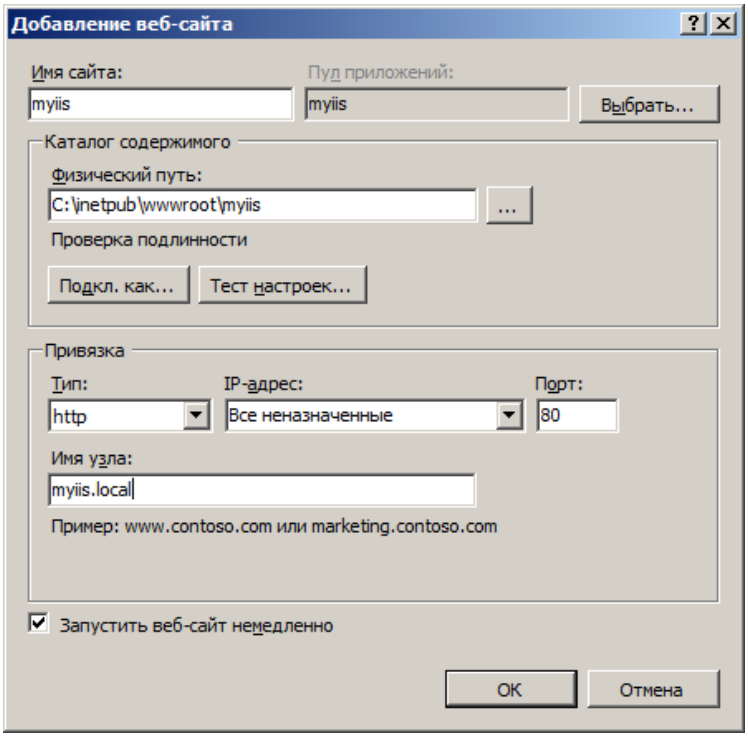
7.1 Создание сайта

Тестовое доменное имя: myiis.local.

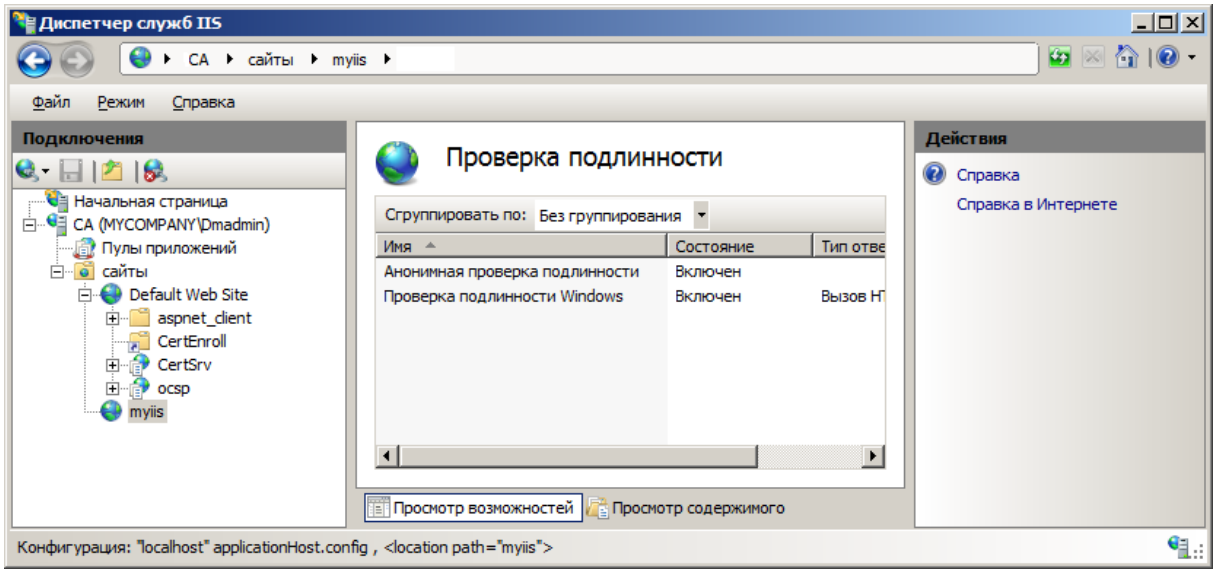
Откройте диспетчер служб IIS. Перейдите в раздел ***Сайты****. В меню справа выберите* ***Добавить веб-сайт****.*



Заполните параметры сайта.



*Проверьте доступность сайта. При ошибке **HTTP Error 401.1 – Unauthorized** необходимо включить анонимную аутентификацию в диспетчере служб IIS.*

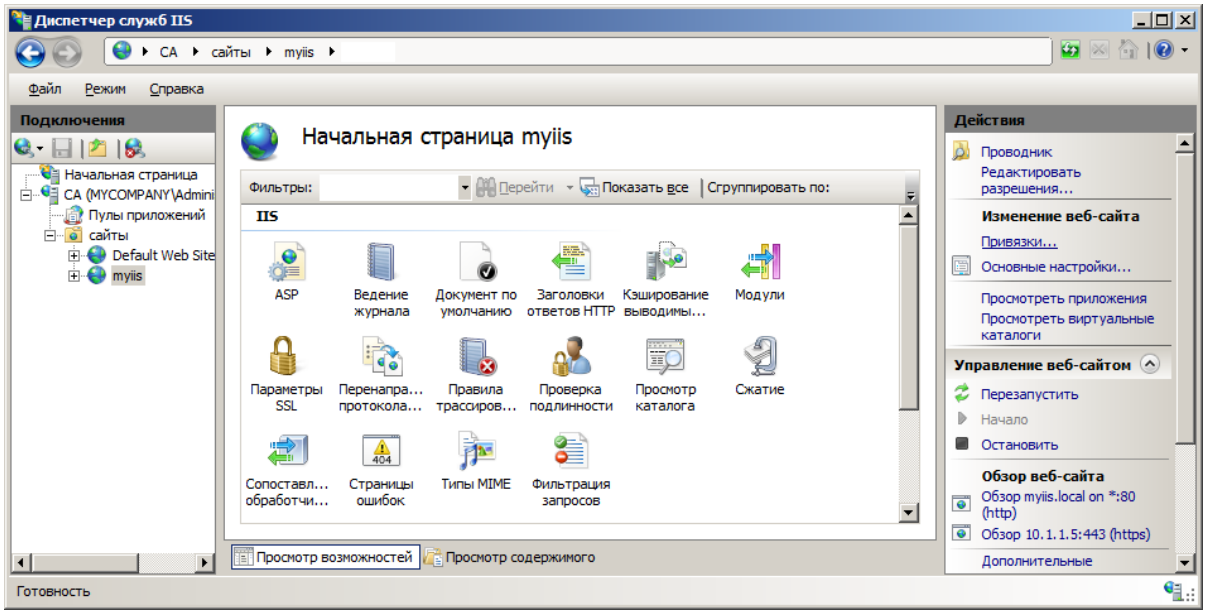


Создайте первую тестовую страницу в каталоге, указанном в качестве физического пути. Проверьте работоспособность сайта через браузер по протоколу HTTP.

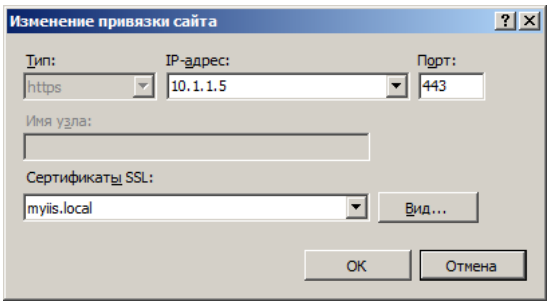
7.2 Настройка привязок

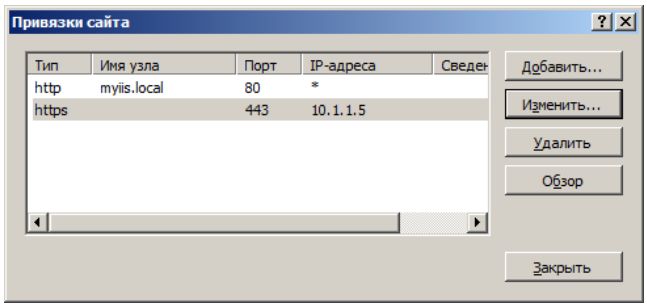
*Создайте сертификат для выбранного домена или импортируйте готовый сертификат. Проверьте наличие сертификата в консоли mmc с оснасткой **Сертификаты** для локального компьютера.*

*Откройте диспетчер служб IIS, выберите созданный сайт и в меню справа выберите **Привязки** (Bindings).*



Добавьте привязку по https на 443 порт. В списке сертификатов выберите сертификат, созданный для данного доменного имени.

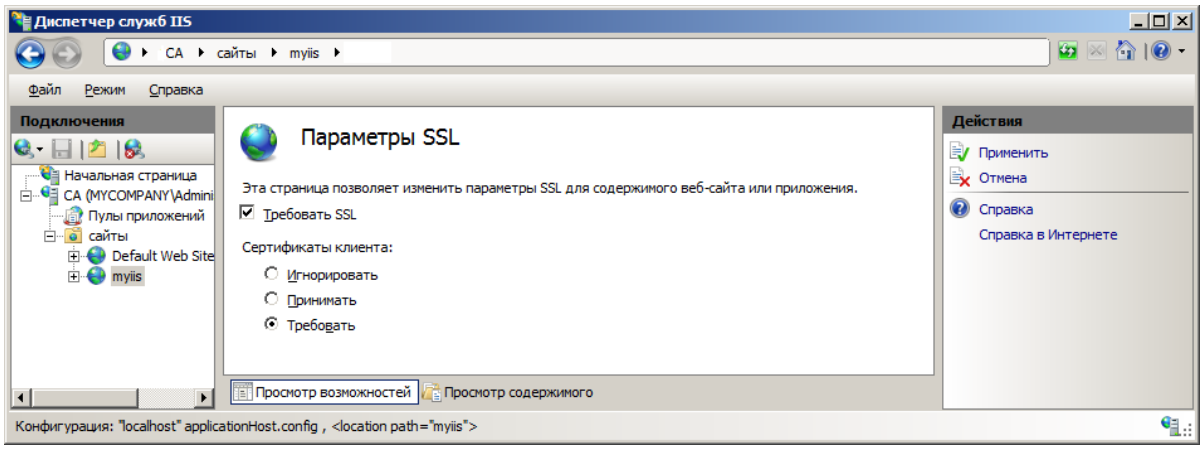




7.3 Авторизация клиентов

*Чтобы использовать авторизацию по сертификату в разделе **Параметры SSL** отметьте **Требовать SLL**. В разделе настроек для сертификатов клиента отметьте:*

- ***Принимать** (клиент МОЖЕТ предъявить сертификат для авторизации) или*
- ***Требовать** (клиент ДОЛЖЕН предъявить сертификат для авторизации).*



Настройка веб-сервера завершена.

8. Веб-сервер Apache

Тестовое доменное имя: myapache.local.

Рекомендуется использовать сервер Apache версии 2.2 или выше. Для проверки сертификатов по протоколу OCSP требуется версия не ниже 2.3.

Далее описана процедура настройки для сервера Apache 2.2 без OCSP.

8.1 Добавление модуля

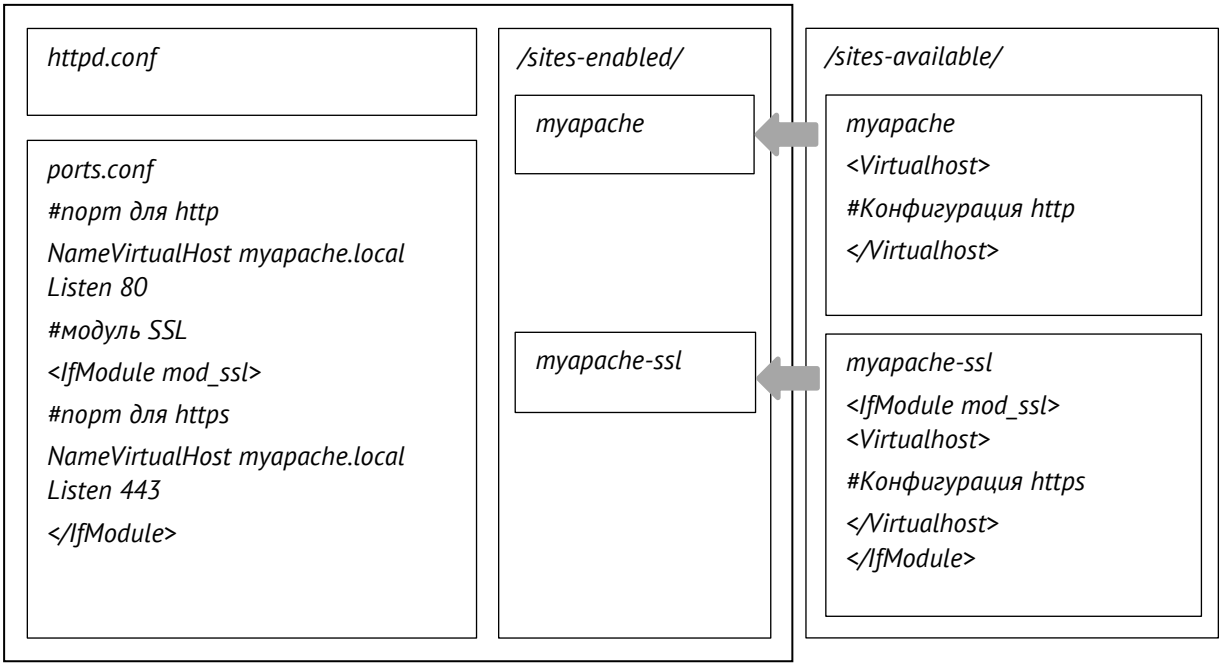
Включите модуль mod.ssl командой:

```
sudo a2enmod ssl
```

8.2 Структура конфигурационных файлов Apache

В примере приведен один из способов организации конфигурационных файлов Apache. В конкретной реализации в основной файл конфигурации apache.conf могут быть включены и другие конфигурационные файлы или целые директории.

apache2.conf



8.3 Примеры конфигурационных файлов

В конфигурационный файл ports.conf помимо директивы NameVirtualHost и Listen для HTTP добавляется директива для модуля Apache mod_ssl. Модуль задает имя виртуального хоста и порт для HTTPS-соединения. После внесения изменений в любой файл конфигурации требуется перезапуск Apache.

ports.conf

```
<IfModule mod_ssl.c>
            NameVirtualHost myapache.local
                  # 443 порт используется по умолчанию
            Listen 443
</IfModule>
```

Пример конфигурационного файла myapache-ssl, который содержит только директивы для HTTPS-соединения.

myapache-ssl

```
<IfModule mod_ssl.c>
<VirtualHost myapache.local:443>
ServerName myapache.local
      # Директива позволяет использовать извлекать данные сертификата
      # и использовать их в качестве переменных окружения
SSLOptions +StdEnvVars +ExportCertData
      # Позволяет устанавливать параметры соединения в безопасном режиме
SSLInsecureRenegotiation on
      DocumentRoot /var/www
      <Directory />
            Options FollowSymLinks
            SSLOptions +StdEnvVars +ExportCertData
            AllowOverride None
      </Directory>
      <Directory /var/www>
            Options Indexes FollowSymLinks MultiViews
            AllowOverride None
            Order allow,deny
            allow from all
      </Directory>

      # Директива SSLEngine разрешает или запрещает
      # использование SSL-модуля
SSLEngine on
      # Способ проверки клиентского сертификата
SSLVerifyClient require
      # Глубина проверки сертификатов в цепочке
SSLVerifyDepth  2

      # Файл сертификата сервера (без закрытого ключа) в формате PEM
SSLCertificateFile /etc/ssl/myapache/certs/myapache.cer
      # Файл закрытого ключа к сертификату сервера в формате PEM
SSLCertificateKeyFile /etc/ssl/myapache/private/myapache.key
      # Файл сертификата корневого центра сертификации
      # в формате PEM и путь к файлу
SSLCACertificatePath /etc/ssl/myapache/certs/
SSLCACertificateFile /etc/ssl/myapache/certs/mycompany_root_ca.cer

      # Файл списка отозванных сертификатов в формате PEM и путь к файлу
SSLCARevocationPath /etc/ssl/myapache/certs/
SSLCARevocationFile /etc/ssl/myapache/certs/mycompany.crl.pem

      </VirtualHost>
</IfModule>
```

Возможные опции директивы `SSLVerifyClient`:

none	*Сертификаты не используются*
optional	*Наличие сертификата не обязательно. Если сертификат обнаружен, пользователю будет предложено авторизоваться по сертификату*
optional_no_ca	*Пользователю будет предложено авторизоваться по сертификату, но сертификат не проверяется. Как правило, используется только для тестирования.*
require	*Пользователь должен представить действительный сертификат, чтобы авторизоваться.*

8.4 Активация сайта

Сделайте сайт доступным, указав конфигурационный файл или файлы.

```
a2ensite myapache myapache-ssl
```

В результате выполнения команды в папке sites-enabled появляются симлинки на соответствующие файлы конфигурации. Если конфигурация для http была активирована ранее, выполните операцию только для конфигурационного файла для https.

8.5 Переменные PHP

В HTTP-заголовка результаты аутентификации передаются в частности следующими переменными:

Переменная	*PHP*	*Описание*
SSL_PROTOCOL	*$_SERVER["SSL_PROTOCOL"]*	*Показывает, действительно ли соединение установлено по HTTPS.*
SSL_CLIENT_M_SERIAL	*$_SERVER["SSL_CLIENT_M_SERIAL"]*	*Серийный номер сертификата клиента*
SSL_CLIENT_S_DN	*$_SERVER["SSL_CLIENT_S_DN"]*	*Имя DN субъекта клиентского сертификата*
SSL_CLIENT_I_DN	*$_SERVER["SSL_CLIENT_I_DN"]*	*Имя DN организации, выпустившей сертификат*
SSL_CLIENT_VERIFY	*$_SERVER["SSL_CLIENT_VERIFY"]*	*Результат проверки сертификата клиента: NONE, SUCCESS, GENEROUS или FAILED:указание_причины*

Полный список доступных переменных опубликован на странице проекта Apache
http://httpd.apache.org/docs/2.2/mod/mod_ssl.html

8.6 Переадресация

Данные, полученные из сертификатов клиентов, можно использовать в директивах модуля переадресации mod_rewrite.

Правила могут быть вписаны в файл конфигурации в блоке <Location> или в .htaccess.

Например, можно использовать следующее правило, чтобы пользователи, представившие истекшие или отозванные сертификаты (то есть статус проверки их сертификатов не равен SUCCESS), были переадресованы на страницу missing_cert.html.

```
RewriteEngine On
RewriteCond %{SSL:SSL_CLIENT_VERIFY} !=SUCCESS$
RewriteRule .* /missing_cert.html [L]
```

Аналогичным образом можно ограничивать доступ в соответствии с отделом сотрудника в определенном отделе (SSL_CLIENT_S_DN_OU) или сроком действия сертификата (SSL_CLIENT_V_START и SSL_CLIENT_V_END). Данные параметры могут комбинироваться с другими переменными окружения, например, IP-адресом (REMOTE_ADDR) или днем недели (TIME_WDAY).

8.7 Проверка пользователя в LDAP

Пользователь может быть заблокирован в Active Directory, но сертификат при этом автоматически не отзывается. Можно использовать дополнительную проверку пользователя по протоколу LDAP.

Требуется модуль php_ldap. Для авторизации пользователя используется класс AD_USER.

1. *Проверяются переменные окружения SSL_PROTOCOL и SSL_CLIENT_VERIFY, чтобы убедиться, что используется аутентификация по сертификату. Если значения переменных соответствуют требуемым, переходим к следующему этапу. Если SSL_PROTOCOL пуст, а*

SSL_CLIENT_VERIFY не равна SUCCESS, пользователь должен использовать стандартную авторизацию по логину/паролю.

2. *С помощью класса AD_USER имя DN из сертификата (SSL_CLIENT_S_DN) преобразуется в логин пользователя.*
3. *Осуществляется проверка, не заблокирован ли пользователь*
4. *Если пользователь не заблокирован, устанавливается защищенное соединение.*

Подставьте в файл ad_user.class.php значения следующих переменных:

$ldaphost - IP домен-контроллера (AD)
$username – пользователь домена с административными привилегиями
$password - пароль пользователя
$base_dn - базовый DN

ad_user.class.php

```
<?php
class AD_USER {
	##############################################
	######### Отредактируйте параметры ############
	var $ldaphost = "10.1.1.5";
	var $ldapport = 389;
	var $username = "admin@mycompany.local";
	var $password = "password";
	var $base_dn = "DC=mycompany,DC=local";
	##############################################

	function __construct($invar=array()) {
		if(isset($invar['allowed_ips'])) $this-
>allowed_ips=$invar['allowed_ips'];
		if(isset($invar['ldaphost'])) $this->ldaphost=$invar['ldaphost'];
		if(isset($invar['ldapport'])) $this->ldapport=$invar['ldapport'];
		if(isset($invar['username'])) $this->username=$invar['username'];
		if(isset($invar['password'])) $this->password=$invar['password'];
		if(isset($invar['base_dn']))  $this->base_dn=$invar['base_dn'];
	}

	function checkUserAd() {
		//$r=$this->checkRemoteIP();
		//if(!$r[0]) return $r;

		$source_dn=trim($_SERVER["SSL_CLIENT_S_DN"],"/");
		$dn_array = explode("/", $source_dn);
		$dn_array=preg_grep("/^emailAddress=/i", $dn_array, PREG_GREP_INVERT);
		$dn_array=array_reverse($dn_array);
		$ready_dn=implode(",",$dn_array);

		$filter = "(&(objectCategory=user)(distinguishedName=$ready_dn))";
		$attributes = array("samaccountname", "useraccountcontrol");
		//   LDAP
		if(($ldapconn = ldap_connect($this->ldaphost, $this-
>ldapport))===false)
			return array(false,"Can't connect to $this->ldaphost");

		ldap_set_option($ldapconn, LDAP_OPT_PROTOCOL_VERSION, 3);
		ldap_set_option($ldapconn, LDAP_OPT_REFERRALS, 0);
		$ldapbind = @ldap_bind($ldapconn, $this->username, $this->password);
		if (!$ldapbind) return array(false,"LDAP bind failed...");

		// Поиск в Active Directory
		if (!($search = ldap_search($ldapconn, $this->base_dn, $filter,
$attributes)))
```

```
                return array(false,"Unable to search ldap server");

        $number_returned = ldap_count_entries($ldapconn,$search);
        if ($number_returned == 0)
                return array(false,"No user found ($ready_dn)");

        if ($number_returned > 1)
                return array(false,"too many users found for ($ready_dn)");

        $info = ldap_get_entries($ldapconn, $search);
        $login_enabled=1;
        if (($info[0]["useraccountcontrol"][0]&0x2) > 0) {
                $login_enabled=0;
        }
        $result=array(
                'dn' => $info[0]["dn"],
                'login' => $info[0]["samaccountname"][0],
                'enable' => $login_enabled,
        );
        return array(true,$result);
    }

}
```

Реализация в файле, например login.php

```
<?php
include "includes/ad_user.class.php";
      // Создаем объект
$ad = new AD_USER;
      // Проверяем пользователя
$reslogin=$ad->checkUserAd();
if(!$reslogin[0]) die($reslogin[1]);
if(!$reslogin[1]['enable']) die('Пользователь заблокирован.');
?>
```

9. Веб-сервер nGinx

Для авторизации по сертификатам рекомендуется использовать nGinx. Данный сервер совместим с 32 и 64-битными версиями операционных систем Windows, Linux, Mac OS, FreeBSD и Solaris.

nGinx выполняет всю процедуру по проверке сертификата пользователя. Преимуществом nGinx является возможность использовать нестандартные коды ошибок, связанные с авторизацией по сертификатам.

9.1 Требуемые модули nGinx

Для работы с SSL сертификатами требуется модуль http_ssl.

Проверить, входит ли модуль в текущую конфигурацию nGinx можно командой

```
$ nginx -V
```

Если при выводе команды nginx -V опции `--with-http_ssl_module` не присутствует, nGinx необходимо собрать заново с поддержкой данного модуля. В отличие от Apache активировать модуль в nGInx нельзя.

Пример скрипта для установки nGinx:

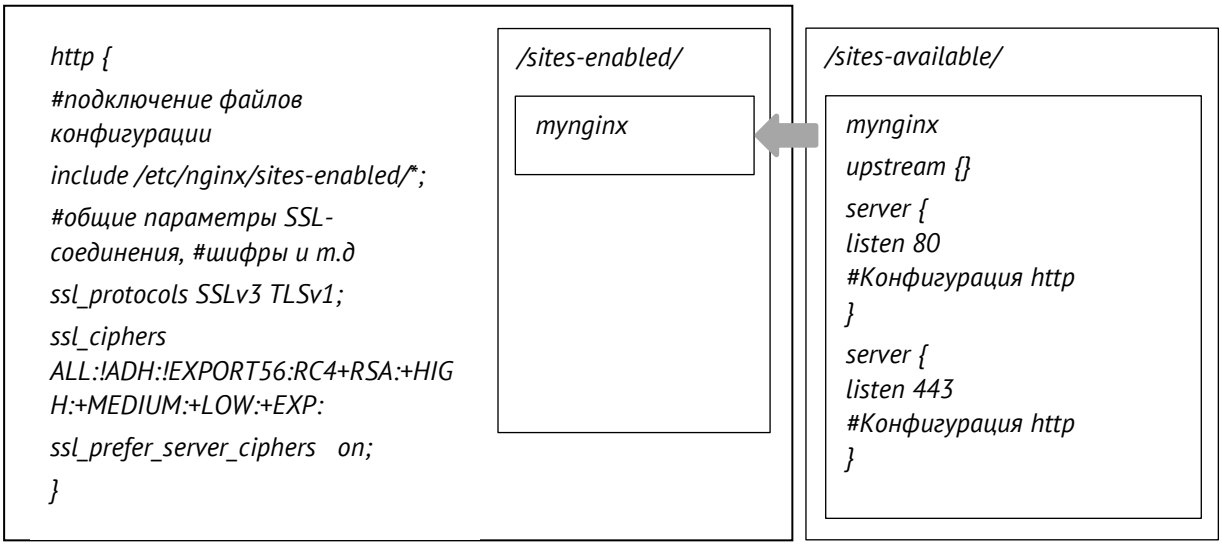
```
mkdir /var/log/nginx
mkdir /etc/nginx
./configure --with-http_ssl_module --conf-path=/etc/nginx/nginx.conf
--error-log-path=/var/log/nginx/error.log --http-lo
g-path=/var/log/nginx/access.log --pid-path=/var/log/nginx/nginx.pid
make
```

```
make install
ln -s /usr/local/nginx/sbin/nginx /usr/sbin/
```

9.2 Структура конфигурационных файлов

Конфигурационные файлы могут быть помещены в директорию /sites-enabled/ или директорию /vhosts/ в зависимости от того, какие папки подключены в файле nginx.conf

nginx.conf



Конфигурационный файл состоит из блоков со следующей иерархией

```
Main {
     http {
           server {
                  location {}
}}}
```

9.3 Переменные PHP

В HTTP-заголовка результаты аутентификации передаются в частности следующими переменными:

Переменная	*PHP*	*Описание*
HTTP_X_FORWARDED_FOR	*$_SERVER["HTTP_X_FORWARDED_FOR"]*	*Получение реального IP-адреса клиента, т.к. переменная REMOTE_ADDR будет содержать IP-адрес nGinx*
HTTP_SSL_PROTOCOL	*$_SERVER["HTTP_SSL_PROTOCOL"]*	*Показывает, действительно ли соединение установлено по HTTPS.*
HTTP_CLIENT_CERT_SERIAL	*$_SERVER["HTTP_CLIENT_CERT_SERIAL"]*	*Серийный номер сертификата клиента*

HTTP_CLIENT_CERT_DN	*$_SERVER["HTTP_CLIENT_CERT_DN"]*	*Имя DN субъекта клиентского сертификата*
HTTP_CLIENT_CERT_DN_ISSUER	*$_SERVER["HTTP_CLIENT_CERT_DN_ISSUER"]*	*Имя DN организации, выпустившей сертификат*
HTTP_CLIENT_CERT_VERIFY	*$_SERVER["HTTP_CLIENT_CERT_VERIFY"]*	*Результат проверки сертификата клиента*

Дополнительно, для возможности использовать любой другой веб-сервер в качестве сервера backend в файле конфигурации может настраиваться передача заголовков в следующие переменные:

Директива	*Описание*
proxy_set_header Client-Cert-Serial $ssl_client_serial;	*Серийный номер сертификата клиента*
proxy_set_header Client-Cert-DN $ssl_client_s_dn;	*Имя DN субъекта сертификата клиента*
proxy_set_header Client-Cert-DN-Issuer $ssl_client_i_dn;	*Имя DN центра сертификации, выпустившего сертификат клиента*
proxy_set_header Client-Cert-Verify $ssl_client_verify;	*Результат проверки сертификата клиента*
proxy_set_header SSL-Protocol $ssl_protocol;	*Параметры SSL-соединения*
proxy_set_header Client-Cert-Data $ssl_client_cert;	*Содержимое сертификата клиента в формате PEM (base64)*
proxy_set_header SSL-Session-ID $ssl_session_id;	*ID сессии*

9.4 Примеры конфигурационных файлов

nginx.conf.

```
# таймаут сессии, время когда клиент может использовать параметры сессии
ssl_session_timeout  1m;
      # доступные протоколы соединения
ssl_protocols SSLv3 TLSv1;
      # возможные шифры в формате OpenSSL
ssl_ciphers
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+EXP:+eNULL:!SSLv2:-EXP-EDH-
RSA-DES-CBC-SHA:-EXP-
RCALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+EXP:+eNULL:!SSLv2:-EXP-
EDH-RSA-DES-CBC-SHA:-EXP-RC;
      # устанавливает приоритет для шифров, поддерживаемых
      # сервером, а не клиентом
ssl_prefer_server_ciphers   on;
      # Задаёт тип и размеры кэшей для хранения параметров сессии.
      # Использование разделяемого кэша считается более эффективным
ssl_session_cache   shared:SSL:10m;
```

mynginx.conf

```
upstream  backend {
      # Указываем на какой сервер переадресовывать запросы,
      # прошедшие SSL аутентификацию.
server localhost:8080;
}

server {
      listen 80;
      # здесь могут быть другие параметры HTTP-соединения,
      # если разрешен доступ по протоколу HTTP
}
```

```
server {
      listen 443 default ssl;
      server_name mynginx.local;
      access_log  /var/log/nginx/example.com-access_log combined;
      proxy_read_timeout  600;

### Раздел пути к сертификатам ###
      # сертификат сервера без закрытого ключа
ssl_certificate     /etc/nginx/certificates/mynginx.cer;
      # закрытый ключ сертификата сервера
ssl_certificate_key /etc/nginx/certificates/mynginx.key;
      # корневой сертификат или цепочка сертификатов
ssl_client_certificate /etc/apache2/certificates/mycompany_root_ca.crt;
      # список отозванных сертификатов
ssl_crl /etc/apache2/certificates/mycompany.crl.pem;
      # Проверка сертификатов клиентов
ssl_verify_client on;

# Добавление собственных страниц ошибок
location /error-pages {
     charset windows-1251;
     root /etc/nginx/vhosts/mynginx.local/htdocs;
    }

error_page 495 /error-pages/495.html;# Ошибка при проверке сертификата
error_page 496 /error-pages/496.html;# Клиент не предоставил сертификат
error_page 497 /error-pages/497.html;# обычный запрос послан на порт HTTPS

      # небольшая оптимизации для iOS устройств,
      # помогающая избежать накопления ошибок в логах
location ~* apple-touch.+ {
        empty_gif;
}

location / {
proxy_pass      http://backend;
proxy_redirect  default;
proxy_set_header Host example.com;

      # Добавляем в заголовок поле X-Forwarded-For, по которому сервер,
      # куда переадресуются авторизованные запросы может получить
      # реальный IP-адрес клиента
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-Server $host;
proxy_set_header Range "";
proxy_set_header Request-Range "";

      # Добавляем дополнительные поля в заголовки,
      # по которым сервер, куда переадресуются авторизованные запросы
      # может получить дополнительную информацию
      # об авторизованном клиенте
proxy_set_header Client-Cert-Serial $ssl_client_serial;
proxy_set_header Client-Cert-DN $ssl_client_s_dn;
proxy_set_header Client-Cert-DN-Issuer $ssl_client_i_dn;
proxy_set_header Client-Cert-Verify $ssl_client_verify;
proxy_set_header SSL-Protocol $ssl_protocol;

      # передача содержимого сертификата
```

```
#proxy_set_header Client-Cert-Data $ssl_client_cert;

      # передача идентификатора ssl-сессии на backend
#proxy_set_header SSL-Session-ID $ssl_session_id;
    }
```

Возможные опции директивы ssl_verify_client:

`off`	*Сертификаты не используются*
`optional`	*Наличие сертификата не обязательно. При наличии сертификата пользователю будет предложено авторизоваться по сертификату*
`optional_no_ca`	*Пользователю будет предложено авторизоваться по сертификату, но сертификат не проверяется. Как правило, в настоящих проектах не используется.*
`on`	*Пользователь должен представить действительный сертификат, чтобы авторизоваться.*

После внесения изменений в конфигурационные файлы требуется перезагрузка nGinx.

9.5 Проверка пользователя в LDAP

Пользователь может быть заблокирован в Active Directory, но сертификат при этом автоматически не отзывается. Можно использовать дополнительную проверку учетной записи пользователя по протоколу LDAP.

Требуется модуль php_ldap. Для авторизации пользователя используется класс AD_USER.

- *Проверяются переменные окружения `HTTP_SSL_PROTOCOL` и `HTTP_CLIENT_CERT_VERIFY`, чтобы убедиться, что используется аутентификация по сертификату. Если значения переменных соответствуют требуемым, переходим к следующему этапу. Если `SSL_PROTOCOL` пуст, а SSL_CLIENT_VERIFY не равна SUCCESS, пользователь должен использовать стандартную авторизацию по логину/паролю.*
- *С помощью класса AD_USER имя DN из сертификата (`SSL_CLIENT_S_DN`) преобразуется в логин пользователя.*
- *Осуществляется проверка, не заблокирован ли пользователь.*
- *Если пользователь не заблокирован, устанавливается соединение.*

Замените в файле ad_user.class.php значения следующих переменных:

$ldaphost - IP домен-контроллера (AD)
$username – пользователь домена с административными привилегиями
$password - пароль пользователя
$base_dn - базовый DN

ad_user.class.php

```
<?php
class AD_USER {


      ###### Замените параметры на собственные #######
      var $allowed_ips = array("10.1.1.11");
      var $ldaphost = "10.1.1.5";
      var $ldapport = 389;
      var $username = "admin@mycompany.local";
      var $password = "password";
      var $base_dn = "DC=mycompany,DC=local";
```

```
	##############################################
		function __construct($invar=array()) {
		if(isset($invar['allowed_ips'])) $this-
>allowed_ips=$invar['allowed_ips'];
		if(isset($invar['ldaphost'])) $this->ldaphost=$invar['ldaphost'];
		if(isset($invar['ldapport'])) $this->ldapport=$invar['ldapport'];
		if(isset($invar['username'])) $this->username=$invar['username'];
		if(isset($invar['password'])) $this->password=$invar['password'];
		if(isset($invar['base_dn'])) $this->base_dn=$invar['base_dn'];
	}

		// Проверка IP адреса
	function checkRemoteIP() {
		if (!in_array($_SERVER["REMOTE_ADDR"], $this->allowed_ips)) {
			return array(false,$_SERVER["REMOTE_ADDR"]." not allowed -
ytf");
		}
		return array(true);
	}

	function checkUserAd() {
		$r=$this->checkRemoteIP();
		if(!$r[0]) return $r;

		$source_dn=trim($_SERVER["HTTP_CLIENT_CERT_DN"],"/");
		$dn_array = explode("/", $source_dn);
		$dn_array=preg_grep("/^emailAddress=/i", $dn_array, PREG_GREP_INVERT);
		$dn_array=array_reverse($dn_array);
		$ready_dn=implode(",",$dn_array);

		$filter = "(&(objectCategory=user)(distinguishedName=$ready_dn))";
		$attributes = array("samaccountname", "useraccountcontrol");
		//   LDAP
		if(($ldapconn = ldap_connect($this->ldaphost, $this-
>ldapport))===false)
			return array(false,"Can't connect to $this->ldaphost");

		ldap_set_option($ldapconn, LDAP_OPT_PROTOCOL_VERSION, 3);
		ldap_set_option($ldapconn, LDAP_OPT_REFERRALS, 0);
		$ldapbind = @ldap_bind($ldapconn, $this->username, $this->password);
		if (!$ldapbind) return array(false,"LDAP bind failed...");

		// поиск в Active Directory
		if (!($search = ldap_search($ldapconn, $this->base_dn, $filter,
$attributes)))
			return array(false,"Unable to search ldap server");

		$number_returned = ldap_count_entries($ldapconn,$search);
		if ($number_returned == 0)
			return array(false,"No user found ($ready_dn)");

		if ($number_returned > 1)
			return array(false,"too many users found for ($ready_dn)");

		$info = ldap_get_entries($ldapconn, $search);
		$login_enabled=1;
		if (($info[0]["useraccountcontrol"][0]&0x2) > 0) {
			$login_enabled=0;
		}
		$result=array(
			'dn' => $info[0]["dn"],
			'login' => $info[0]["samaccountname"][0],
			'enable' => $login_enabled,
		);
		return array(true,$result);
	}
	}
```

Пример вызова в файле, например login.php

```
// Создаем объект
$ad = new AD_USER;
// Проверяем пользователя
$reslogin=$ad->checkUserAd();
if(!$reslogin[0]) die($reslogin[1]);
if(!$reslogin[1]['enable']) die('Blocked User.');
```

9.6 Кастомизированные страницы ошибок

nGinx поддерживает несколько нестандартных кодов ошибок:

495 – при проверке клиентского сертификата произошла ошибка;

496 – клиент не предоставил требуемый сертификат;

497 – обычный запрос был послан на порт HTTPS.

Данные коды ошибок можно использовать в директиве error_page.

```
error_page 496      http://mynginx.local/missing_certificate.html;
```

Перенаправление делается после того, как запрос полностью разобран и доступны такие переменные, как `$request_uri`, `$uri`, `$args` и другие переменные.