

# Лабораторная работа №5

## Расширенная настройка HTTP-сервера Apache

Сахно Никита НФИбд-02-23

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Выполнение лабораторной работы.....	1
4	Выводы.....	5
5	Контрольные вопросы.....	5

### 1 Цель работы

Приобрести практические навыки по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

### 2 Задание

1. Сгенерировать криптографический ключ и самоподписанный сертификат безопасности для возможности перехода веб-сервера от работы через протокол HTTP к работе через протокол HTTPS;
2. Настроить веб-сервер для работы с PHP;
3. Написать скрипт для Vagrant, фиксирующий действия по расширенной настройке HTTP-сервера во внутреннем окружении виртуальной машины server.

### 3 Выполнение лабораторной работы

#### Конфигурирование HTTP-сервера для работы через протокол HTTPS

Загрузим вашу операционную систему и перейдем в рабочий каталог с проектом: cd C:\Users\nikita\work\study\nvsakhno\vagrant

Запустим виртуальную машину server: make server-up

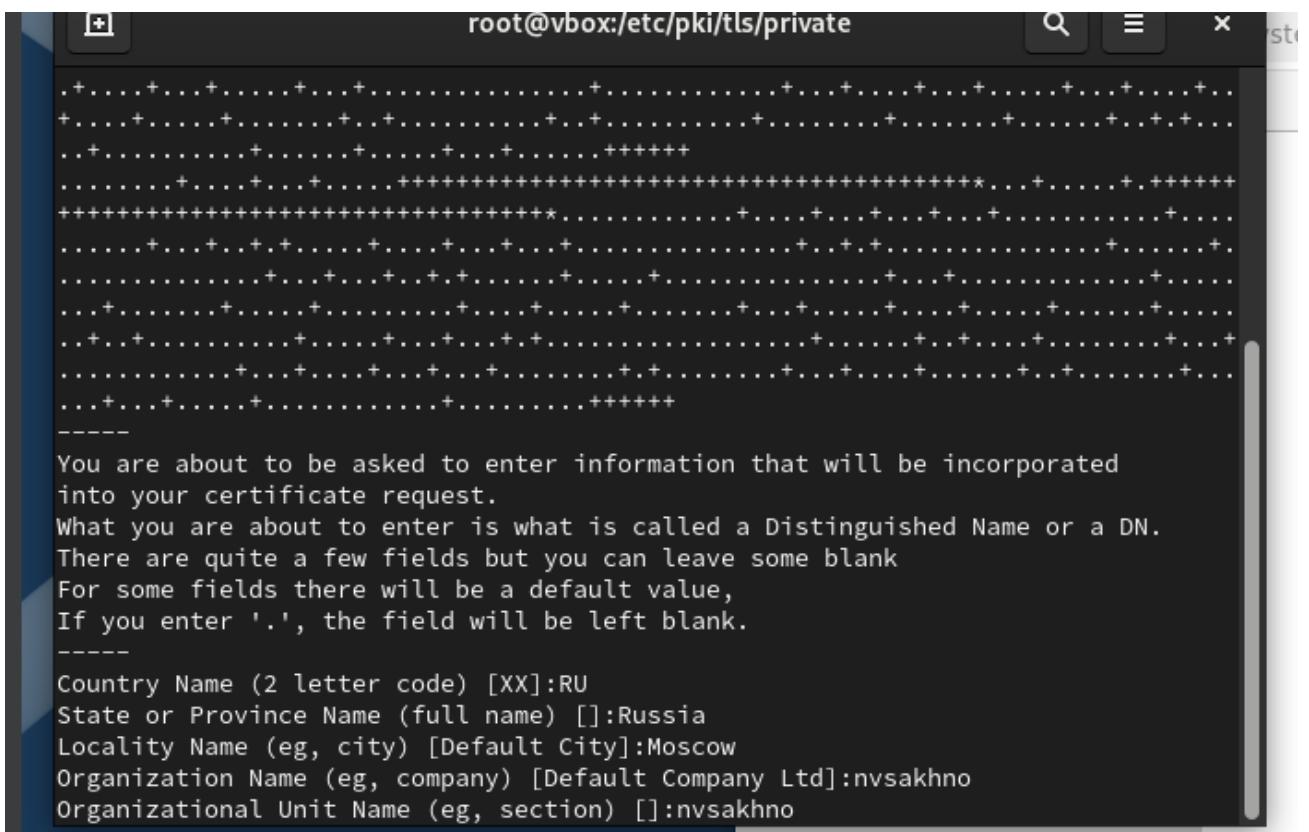
На виртуальной машине server войдем под своим пользователем и откроем терминал.  
Перейдем в режим суперпользователя: sudo -i

В каталоге /etc/ssl создадим каталог private.

```
# mkdir -p /etc/pki/tls/private
# ln -s /etc/pki/tls/private /etc/ssl/private
No such file or directory
# ln -s /etc/pki/tls/private /etc/ssl/private
# cd /etc/pki/tls/private
```

### *Создание каталога private*

Сгенерируем ключ и сертификат:



The screenshot shows a terminal window titled "root@vbox:/etc/pki/tls/private". It displays the OpenSSL command for generating a certificate request (CSR). The command is:

```
openssl req -x509 -nodes -newkey rsa:2048 -keyout private.key -out certificate.crt
```

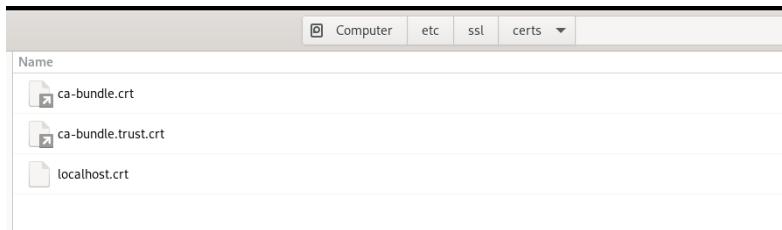
Below the command, the terminal shows the interactive process of entering certificate information. It asks for the country name (RU), state/province name (Russia), locality name (Moscow), organization name (nvsakhno), and organizational unit name (nvsakhno).

### *Генерация ключа и сертификата*

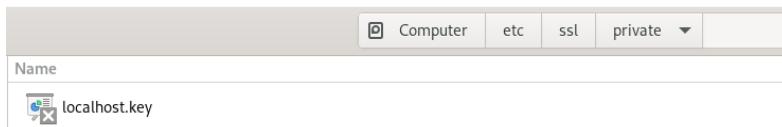
- req -x509 означает, что используется запрос подписи сертификата x509 (CSR);
- параметр -nodes указывает OpenSSL, что нужно пропустить шифрование сертификата SSL с использованием парольной фразы, т.е. позволить Apache читать файл без какого-либо вмешательства пользователя (без ввода пароля при попытке доступа к странице, в частности);
- параметр -newkey rsa: 2048 указывает, что одновременно создаются новый ключ и новый сертификат, причём используется 2048-битный ключ RSA;

- параметр `-keyout` указывает, где хранить сгенерированный файл закрытого ключа при создании;
- параметр `-out` указывает, где разместить созданный сертификат SSL. Далее требуется заполнить сертификат:

Сгенерированные ключ и сертификат появились в соответствующем каталогах `/etc/ssl/private` и `/etc/ssl/certs`.



*Наличие ключа в каталоге*



*Наличие сертификата в каталоге*

Для перехода веб-сервера `www.dmbelicheva.net` на функционирование через протокол HTTPS требуется изменить его конфигурационный файл. Перейдем в каталог с конфигурационными файлами: `cd /etc/httpd/conf.d`

Откроем на редактирование файл `/etc/httpd/conf.d/www.nvsakhno.net.conf` и заменим его содержимое. Внесем изменения в настройки межсетевого экрана на сервере, разрешив работу с https. Перезапустим веб-сервер: `systemctl restart httpd`.

```
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps audit ausweisapp2 b
k-agent cockpit collectd condor-collector cratedb dhcp dhcpcv6 dhcpcv6-client distcc dns dns-over-tls docker-
p freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpgsql grafana gre
in kdeconnect kerberos kibana klogind kpasswd kprop kshell kube-api kube-apiserver kube-control-plane kube-contro
ure kube-worker kubefleet kubefleet-readyonly kubefleet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr l
etbios-ns netdata-dashboard nfs nfs3 nmea-0183 nppc ntp nntplib openvpn ovirt-imageio ovirt-storageconsole ovirt-vmc
tsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotar rsh rsyncd rtsp salt-m
nsync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui synergy syslog syslog-tls te
covery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local
```

*Настройка межсетевого экрана на сервере*

На виртуальной машине client в строке браузера введем название веб-сервера `www.user.net` и убедимся, что произойдёт автоматическое переключение на работу по протоколу HTTPS. На открывшейся странице с сообщением о незащищённости соединения нажмем кнопку «Дополнительно», затем добавим адрес сервера в постоянные исключения.

```
-rw----- . 1 root root 1704 фев 10 11:11 localhost.key
-rw----- . 1 root root 1704 фев 10 13:38 privkey.pem
-rw-r--r-- . 1 root root 1468 фев 10 13:44 www.nvsakhno.net.crt
-rw----- . 1 root root 1704 фев 10 13:44 www.nvsakhno.net.key
[root@vbox private]# mv www.nvsakhno.net.crt /etc/pki/tls/certs
[root@vbox private]# cp /etc/ssl/private/www.nvsakhno.net.crt /etc/ssl/cert
```

## Конфигурирование HTTP-сервера для работы с PHP

Установим пакеты для работы с PHP: `dnf -y install php`

```
Last metadata expiration check: 1:28:05 ago on Fri 24 Nov 2023 05:16:06 PM UTC.
Dependencies resolved.
=====
Package           Architecture   Version      Repository
=====
Installing:
php              x86_64        8.0.30-1.el9_2    appstream
Installing dependencies:
nginx-filesystem noarch       1:1.20.1-14.el9_2.1 appstream
php-common       x86_64        8.0.30-1.el9_2    appstream
Installing weak dependencies:
php-cti          x86_64        8.0.30-1.el9_2    appstream
php-fpm          x86_64        8.0.30-1.el9_2    appstream
php-mbstring     x86_64        8.0.30-1.el9_2    appstream
php-opcache      x86_64        8.0.30-1.el9_2    appstream
php-pdo          x86_64        8.0.30-1.el9_2    appstream
php-xml          x86_64        8.0.30-1.el9_2    appstream
=====
Transaction Summary
=====
Install 9 Packages
```

*Установка пакетов для работы с php*

В каталоге `/var/www/html/www.nvsakhno.net` заменим файл `index.html` на `index.php` следующего содержания:

```
<?php
phpinfo();
?>
```

```
GNU nano 5.6.1                               index.php
<?php
phpinfo();
?>
```

*Редактирование файла index.php*

Скорректируем права доступа в каталог с веб-контентом: `chown -R apache:apache /var/www`

Восстановим контекст безопасности в SELinux:

```
restorecon -vR /etc
restorecon -vR /var/www
```

Перезапустим HTTP-сервер: `systemctl restart httpd`

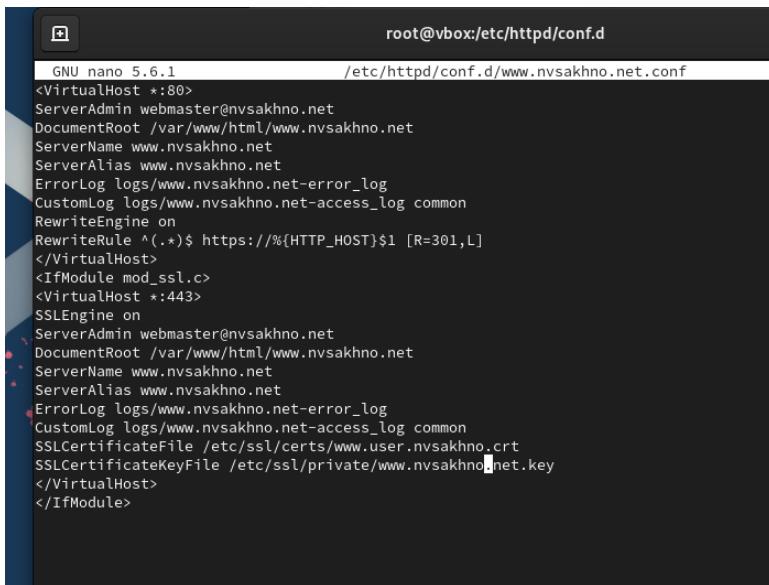
```
chown -R apache:apache /var/www  
restorecon -vR /etc  
restorecon -vR /var/www  
systemctl restart httpd
```

### *Права доступа и контекст безопасности в SELinux*

На виртуальной машине client в строке браузера введиим название веб-сервера www.nvsakhno.net и убедимся, что будет выведена страница с информацией об используемой на веб-сервере версии PHP.

### **Внесение изменений в настройки внутреннего окружения виртуальной машины**

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/http и в соответствующие каталоги скопируйте конфигурационные файлы. В имеющийся скрипт /vagrant/provision/server/http.sh внесем изменения, добавив установку PHP и настройку межсетевого экрана, разрешающую работать с https.



```
GNU nano 5.6.1          /etc/httpd/conf.d/www.nvsakhno.net.conf
<VirtualHost *:80>
ServerAdmin webmaster@nvsakhno.net
DocumentRoot /var/www/html/www.nvsakhno.net
ServerName www.nvsakhno.net
ServerAlias www.nvsakhno.net
ErrorLog logs/www.nvsakhno.net-error_log
CustomLog logs/www.nvsakhno.net-access_log common
RewriteEngine on
RewriteRule ^(.*)$ https:// %{HTTP_HOST}$1 [R=301,L]
</VirtualHost>
<IfModule mod_ssl.c>
<VirtualHost *:443>
SSLEngine on
ServerAdmin webmaster@nvsakhno.net
DocumentRoot /var/www/html/www.nvsakhno.net
ServerName www.nvsakhno.net
ServerAlias www.nvsakhno.net
ErrorLog logs/www.nvsakhno.net-error_log
CustomLog logs/www.nvsakhno.net-access_log common
SSLCertificateFile /etc/ssl/certs/www.user.nvsakhno.crt
SSLCertificateKeyFile /etc/ssl/private/www.nvsakhno.net.key
</VirtualHost>
</IfModule>
```

### *Редактирование скрипта*

## **4 Выводы**

в процессе выполнения данной лабораторной работы я приобрел практические навыки по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

## **5 Контрольные вопросы**

1. В чём отличие HTTP от HTTPS?

Отличие состоит в том, что HTTPS — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.

2. Каким образом обеспечивается безопасность контента веб-сервера при работе через HTTPS?

Улучшение безопасности при использовании HTTPS вместо HTTP достигается за счёт использования криптографических протоколов при организации HTTP-соединения и передачи по нему данных. Для шифрования может применяться протокол SSL (Secure Sockets Layer) или протокол TLC (Transport Layer Security). Оба протокола используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

3. Что такое сертификационный центр? Приведите пример.

Сертификационный центр (Certification authority, CA) представляет собой компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей.

Пример: IdenTrust, DigiCert.