

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ  
НАРОДОВ**

**Факультет физико-математических и естественных  
наук**

**Кафедра теории вероятностей и кибербезопасности**

**Отчет лабораторной работы 5**

**Дисциплина: Администрирование сетевых подсистем**

Студент: Астахова Марина

Группа: НПИбд-02-23

## *Тема: Расширенная настройка HTTP-сервера Apache*

### 5.1. Цель работы.

Приобретение практических навыков по расширенному конфигурированию HTTP сервера Apache в части безопасности и возможности использования PHP.

## 5.2. Выполнение работы

**1. Сгенерируйте криптографический ключи самоподписанный сертификат безопасности для возможности перехода веб-сервера от работы через протокол HTTP к работе через протокол HTTPS. Конфигурирование HTTP-сервера для работы через протокол HTTPS.**


В каталоге /etc/ssl создали каталог private и сгенерировали ключ:

[illegible]

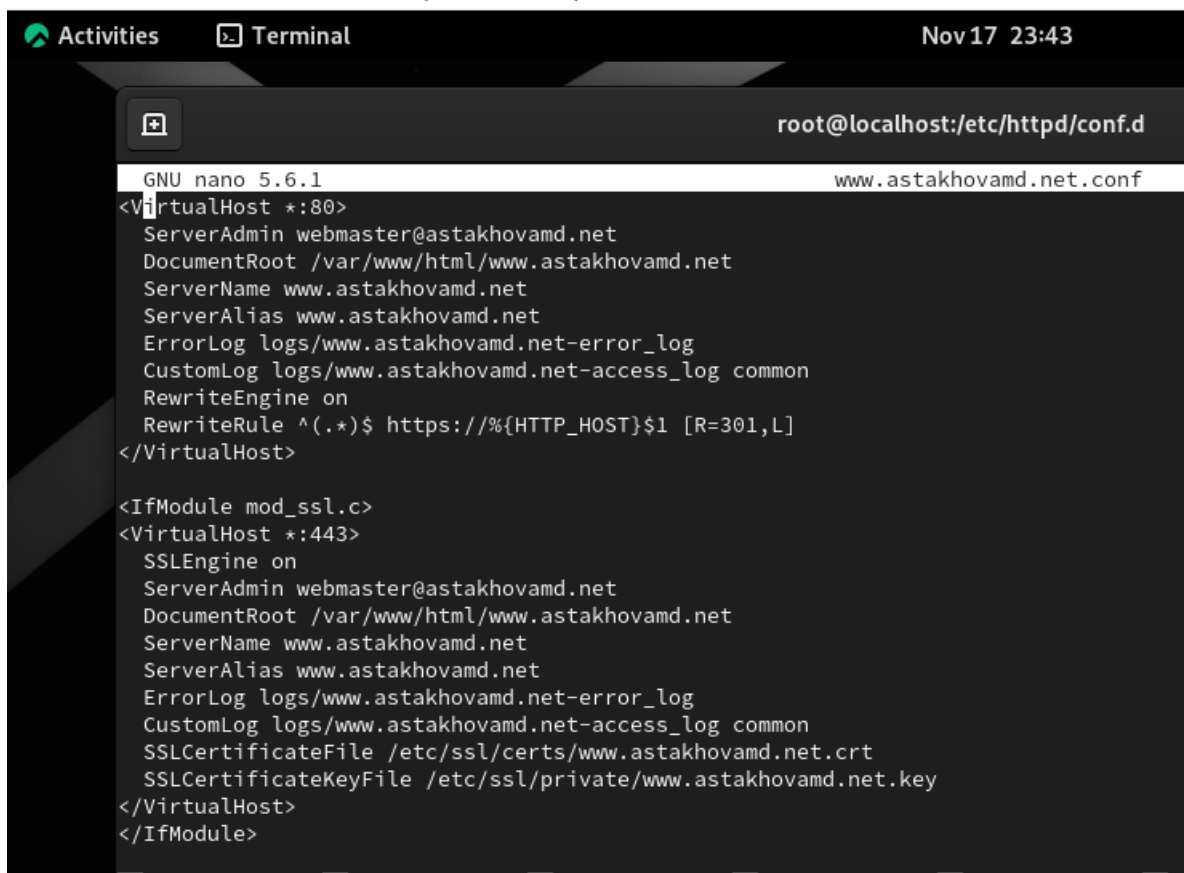
## Подготовка ключа

```
-----
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) [XX]:RU
State or Province Name (full name) []:Russia
Locality Name (eg, city) [Default City]:Moscow
Organization Name (eg, company) [Default Company Ltd]:astakhovamd
Organizational Unit Name (eg, section) []:astakhovamd
Common Name (eg, your name or your server's hostname) []:astakhovamd.net
Email Address []:astakhovamd@astakhovamd.net
[root@localhost.user.net private]# mv www.astakhovamd.net.crt /etc/pki/tls/certs
[root@localhost.user.net private]#
cp /etc/ssl/private/www.astakhovamd.net.crt /etc/ssl/cert/
cp: cannot stat '/etc/ssl/private/www.astakhovamd.net.crt': No such file or directory
[root@localhost.user.net private]#
```

Открыли на редактирование файл  
/etc/httpd/conf.d/www.astakhovamd.net.conf и заменили его содержимое на  
следующее

 vagrant\_server\_1762976634777\_2593 [Работает] - Oracle VirtualBox

Файл   Машина   Вид   Ввод   Устройства   Справка



```
Activities   Terminal   Nov 17 23:43

root@localhost:/etc/httpd/conf.d

GNU nano 5.6.1   www.astakhovamd.net.conf
<VirtualHost *:80>
  ServerAdmin webmaster@astakhovamd.net
  DocumentRoot /var/www/html/www.astakhovamd.net
  ServerName www.astakhovamd.net
  ServerAlias www.astakhovamd.net
  ErrorLog logs/www.astakhovamd.net-error_log
  CustomLog logs/www.astakhovamd.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@astakhovamd.net
  DocumentRoot /var/www/html/www.astakhovamd.net
  ServerName www.astakhovamd.net
  ServerAlias www.astakhovamd.net
  ErrorLog logs/www.astakhovamd.net-error_log
  CustomLog logs/www.astakhovamd.net-access_log common
  SSLCertificateFile /etc/ssl/certs/www.astakhovamd.net.crt
  SSLCertificateKeyFile /etc/ssl/private/www.astakhovamd.net.key
</VirtualHost>
</IfModule>
```

Внесли изменения в настройки межсетевого экрана на сервере, разрешив  
работу с https:

```
[root@localhost.user.net conf.d]# sudo nano www.astakhovamd.net.conf
[root@localhost.user.net conf.d]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http ssh
[root@localhost.user.net conf.d]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula bacula-client
bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ce
ph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp
dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server f
inger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client gangl
ia-master git gpsd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target
isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-plane kube-contro
l-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure
kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp
llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios
-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole ple
x pcmd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv p
tp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-
client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squ
id ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile3
8 tinc tor-socks transmission-client upnp-client vdsml vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discove
ry-client ws-discovery-tcp ws-discovery-udp wsmann wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-se
rver zerotier
[root@localhost.user.net conf.d]# firewall-cmd --add-service=https
success
[root@localhost.user.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@localhost.user.net conf.d]# firewall-cmd --reload
success
```

```
rver zerotier
[root@localhost.user.net conf.d]# firewall-cmd --add-service=https
success
[root@localhost.user.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@localhost.user.net conf.d]# firewall-cmd --reload
success
[root@localhost.user.net conf.d]# systemctl restart httpd
```

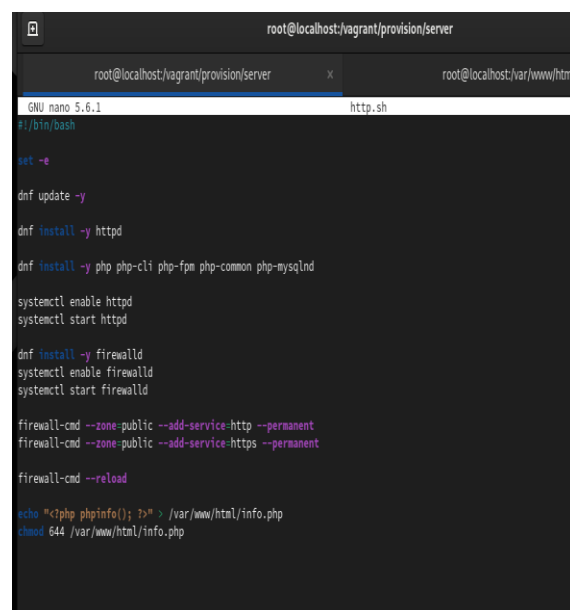
## ***2. Настроим веб-сервер для работы с PHP. Конфигурирование HTTP-сервера для работы с PHP.***

Установили пакеты для работы с PHP:



В имеющийся скрипт `/vagrant/provision/server/http.sh` внесли изменения, добавив установку РНР и настройку межсетевого экрана, разрешающую работать с `https`.

```
[root@localhost.user.net conf.d]#
[root@localhost.user.net conf.d]#
[root@localhost.user.net conf.d]#
[root@localhost.user.net conf.d]#
[root@localhost.user.net conf.d]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/autoindex.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/fcgid.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/manual.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/README'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/server.astakhovamd.net.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/ssl.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/userdir.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/welcome.conf'? Y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/www.astakhovamd.net.conf'? Y
[root@localhost.user.net conf.d]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
cp: overwrite '/vagrant/provision/server/http/var/www/html/server.astakhovamd.net/index.html'? Y
cp: overwrite '/vagrant/provision/server/http/var/www/html/www.astakhovamd.net/index.html'? Y
[root@localhost.user.net conf.d]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/private
[root@localhost.user.net conf.d]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/certs
[root@localhost.user.net conf.d]# cp -R /etc/pki/tls/private/www.astakhovamd.net.key /vagrant/provision/server/http/etc/pki/tls/private
cp: missing destination file operand after '/etc/pki/tls/private/www.astakhovamd.net.key/vagrant/provision/server/http/etc/pki/tls/private'
Try 'cp --help' for more information
[root@localhost.user.net conf.d]# cp -R /etc/pki/tls/private/www.astakhovamd.net.key /vagrant/provision/server/http/etc/pki/tls/private
[root@localhost.user.net conf.d]# cp -R /etc/pki/tls/certs/www.astakhovamd.net.crt /vagrant/provision/server/http/etc/pki/tls/certs
[root@localhost.user.net conf.d]#
```



The screenshot shows a terminal window titled `root@localhost:/vagrant/provision/server`. It displays the execution of the `http.sh` script. The script performs several tasks: it updates the system, installs `httpd`, enables and starts the `httpd` service, installs `firewalld`, enables and starts the `firewalld` service, adds rules to the firewall to allow `http` and `https` traffic, and reloads the firewall. Finally, it checks the PHP version and the file permissions of `/var/www/html/info.php`.

```
root@localhost:/vagrant/provision/server
GNU nano 5.6.1 http.sh
#!/bin/bash

set -e

dnf update -y

dnf install -y httpd

dnf install -y php php-cli php-fpm php-common php-mysqlnd

systemctl enable httpd
systemctl start httpd

dnf install -y firewalld
systemctl enable firewalld
systemctl start firewalld

firewall-cmd --zone=public --add-service=http --permanent
firewall-cmd --zone=public --add-service=https --permanent

firewall-cmd --reload

echo "<?php phpinfo(); ?>" > /var/www/html/info.php
chmod 644 /var/www/html/info.php
```

### 5.3. Итог работы

Были приобретены практические навыки по расширенному конфигурированию HTTP сервера Apache в части безопасности и возможности использования РНР.

### 5.4. Контрольные вопросы

## ***1. В чём отличие HTTP от HTTPS?***

***HTTP (Hypertext Transfer Protocol)*** можно сравнить с передачей информации в открытом виде, аналогично отправке незапечатанной почтовой открытки. В такой ситуации данные не защищены от несанкционированного доступа, перехвата или прочтения третьими лицами в процессе их передачи по сети.

***HTTPS (Hypertext Transfer Protocol Secure)***, напротив, обеспечивает безопасную передачу данных. Его функционирование эквивалентно отправке запечатанного и криптографически зашифрованного письма. Это гарантирует, что:

1. *Конфиденциальность*: Содержимое сообщения недоступно для прочтения посторонними лицами, поскольку оно зашифровано.
2. *Аутентификация*: Получатель может удостовериться в подлинности отправителя (веб-сервера) благодаря использованию цифровых сертификатов, что исключает возможность подмены отправителя.
3. *Целостность*: Получатель уверен, что данные не были изменены или повреждены в процессе передачи.

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

Безопасность контента при работе через HTTPS обеспечивается благодаря использованию протоколов SSL (Secure Sockets Layer) и его преемника TLS (Transport Layer Security). Эти протоколы работают на транспортном уровне и обеспечивают три ключевых аспекта безопасности:

### ***A. Шифрование (Confidentiality):***

Все данные, передаваемые между браузером и сервером (HTTP-запросы и ответы), шифруются. Это означает, что если злоумышленник перехватит трафик, он получит лишь бессмысленный набор символов, который не сможет прочитать.

Для шифрования используется комбинация асимметричного (для обмена ключами) и симметричного (для основной передачи данных) шифрования.

### ***B. Аутентификация (Authentication):***

HTTPS позволяет клиенту (браузеру) удостовериться в подлинности веб-сервера. Это предотвращает атаки типа "человек посередине" (Man-in-the-Middle), когда злоумышленник выдает себя за легитимный сервер.

Аутентификация осуществляется с помощью цифровых сертификатов. Сервер предоставляет свой сертификат браузеру. Браузер проверяет этот сертификат:

1. Действителен ли он (не истёк ли срок).
2. Выдан ли он доверенным Сертификационным Центром (CA).
3. Соответствует ли доменное имя в сертификате тому домену, к которому обращается пользователь.

### **С. Целостность данных (Integrity):**

HTTPS гарантирует, что данные не были изменены или повреждены во время передачи. Для этого используются криптографические хеш-функции и цифровые подписи. Сервер и клиент могут проверить, что полученные данные точно соответствуют отправленным.

### **3. Что такое сертифицикационный центр? Приведите пример.**

Сертификационный Центр (CA - Certificate Authority) — это доверенная сторонняя организация, которая выдает и управляет цифровыми сертификатами, используемыми для обеспечения безопасности в интернете (например, для HTTPS).

### **Роль Сертификационного Центра:**

#### **1. Проверка личности:**

CA проверяет личность владельца веб-сайта или организации, которая запрашивает сертификат. Уровень проверки может варьироваться от простой проверки владения доменом (Domain Validation - DV) до более строгой проверки организации (Organization Validation - OV) или расширенной проверки (Extended Validation - EV).

#### **2. Выдача сертификатов:**

После успешной проверки CA выдает цифровой сертификат, который содержит:

- A. Публичный ключ владельца.
- B. Имя домена (или доменов), для которого выдан сертификат.
- C. Срок действия сертификата.
- D. Название CA, который выдал сертификат.



- Е. Цифровую подпись самого СА, которая подтверждает подлинность сертификата.

### **3. Поддержание доверия:**

Браузеры и операционные системы имеют встроенный список "корневых" (root) сертификатов доверенных СА. Когда браузер получает сертификат от веб-сайта, он проверяет, подписан ли этот сертификат одним из доверенных корневых СА (или промежуточным СА, который, в свою очередь, подписан корневым СА). Если цепочка доверия действительна, браузер считает сертификат подлинным и устанавливает безопасное соединение.

#### **Пример Сертификационного Центра:**

Один из самых известных и широко используемых сертификационных центров, особенно для бесплатных сертификатов, это Let's Encrypt.

Let's Encrypt — это некоммерческий СА, который предоставляет бесплатные цифровые сертификаты X.509 для TLS-шифрования. Его цель — сделать использование HTTPS стандартным для всех веб-сайтов, упрощая процесс получения и установки сертификатов. Благодаря Let's Encrypt, многие мелкие и средние сайты могут использовать HTTPS без дополнительных затрат.

Другие известные коммерческие СА включают:

- a. DigiCert
- b. Sectigo (ранее Comodo)
- c. GlobalSign
- d. GoDaddy

