

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
ИМЕНИ ПАТРИСА ЛУМУМБЫ

Факультет физико-математических и естественных наук
Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

Дисциплина «Администрирование сетевых подсистем»

Тема «Расширенная настройка HTTP-сервера Apache»

Студент: Щербак Маргарита Романовна

Ст. билет: 1032216537

Группа: НПИбд-02-21

МОСКВА

2023 г.

Цель работы

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

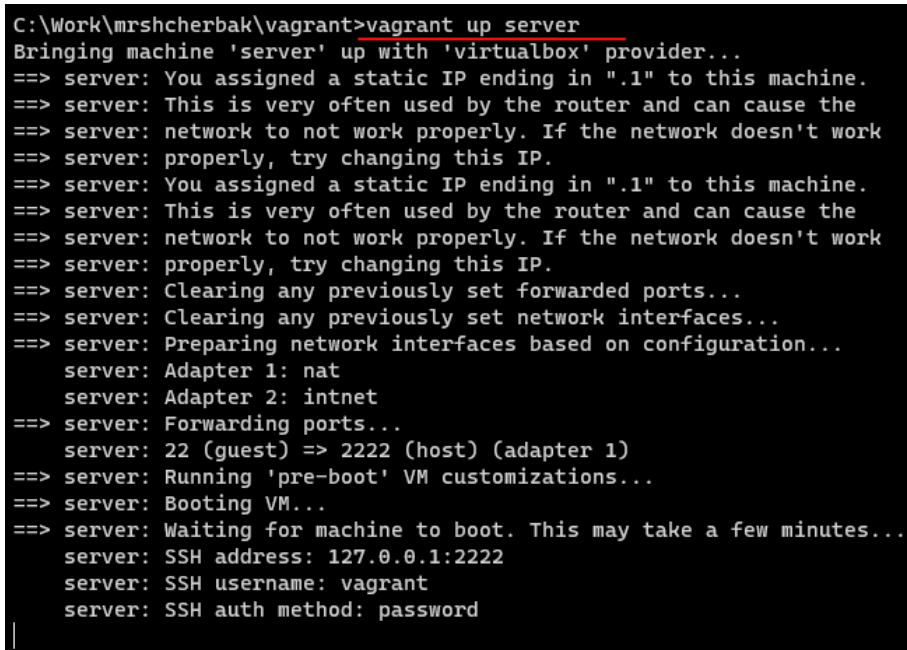
Задание

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.

Выполнение

1. Конфигурирование HTTP-сервера для работы через протокол HTTPS

1. Загрузила свою ОС и перешла в рабочий каталог с проектом, затем запустила виртуальную машину server с помощью команды `vagrant up server` (рис.1.1).



```
C:\Work\mrshcherbak\vagrant>vagrant up server
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Clearing any previously set forwarded ports...
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
server: Adapter 1: nat
server: Adapter 2: intnet
==> server: Forwarding ports...
server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Running 'pre-boot' VM customizations...
==> server: Booting VM...
==> server: Waiting for machine to boot. This may take a few minutes...
server: SSH address: 127.0.0.1:2222
server: SSH username: vagrant
server: SSH auth method: password
```

Рис.1.1. Запуск виртуальной машины Server

2. На виртуальной машине server вошла под своим пользователем и открыла терминал. Перешла в режим суперпользователя: `sudo -i`. В каталоге `/etc/ssl` создала

каталог private. Сгенерировала ключ и сертификат, после чего заполнила сертификат. Действия представлены на рис.1.2 – рис.1.3.

```
root@server:/etc/pki/tls/private

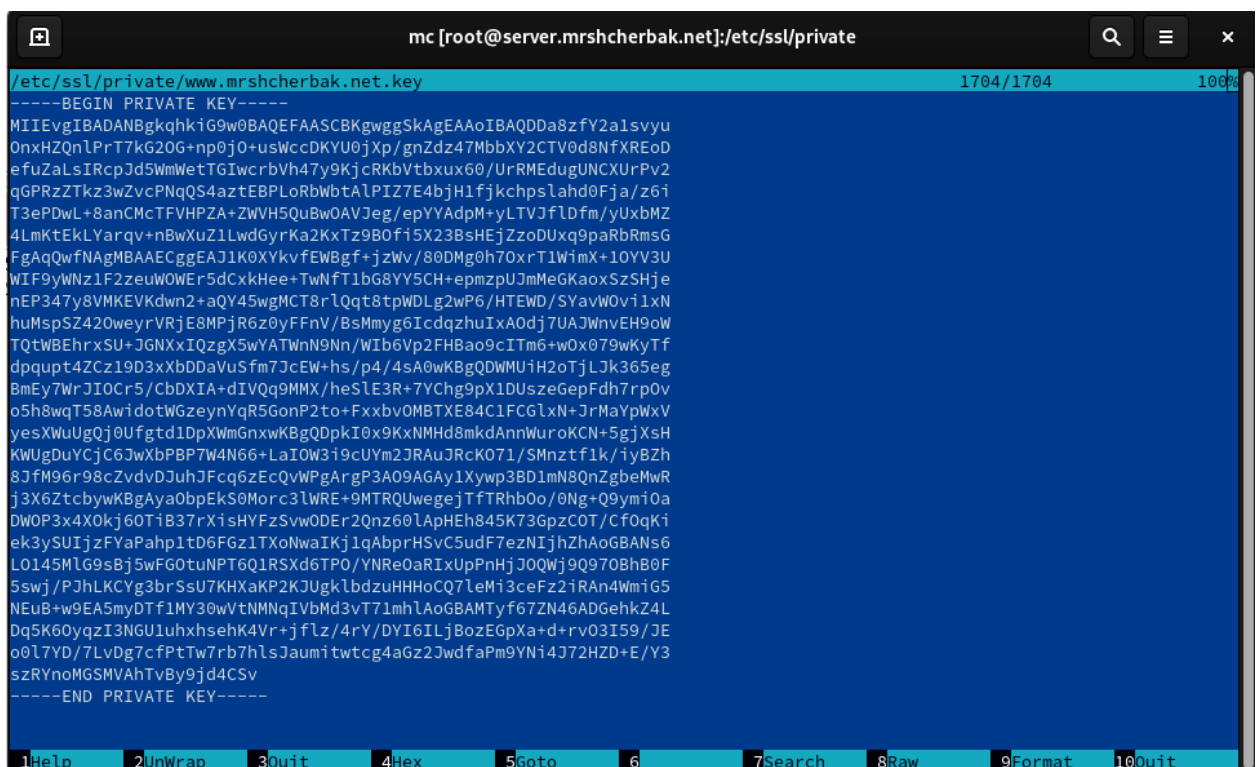
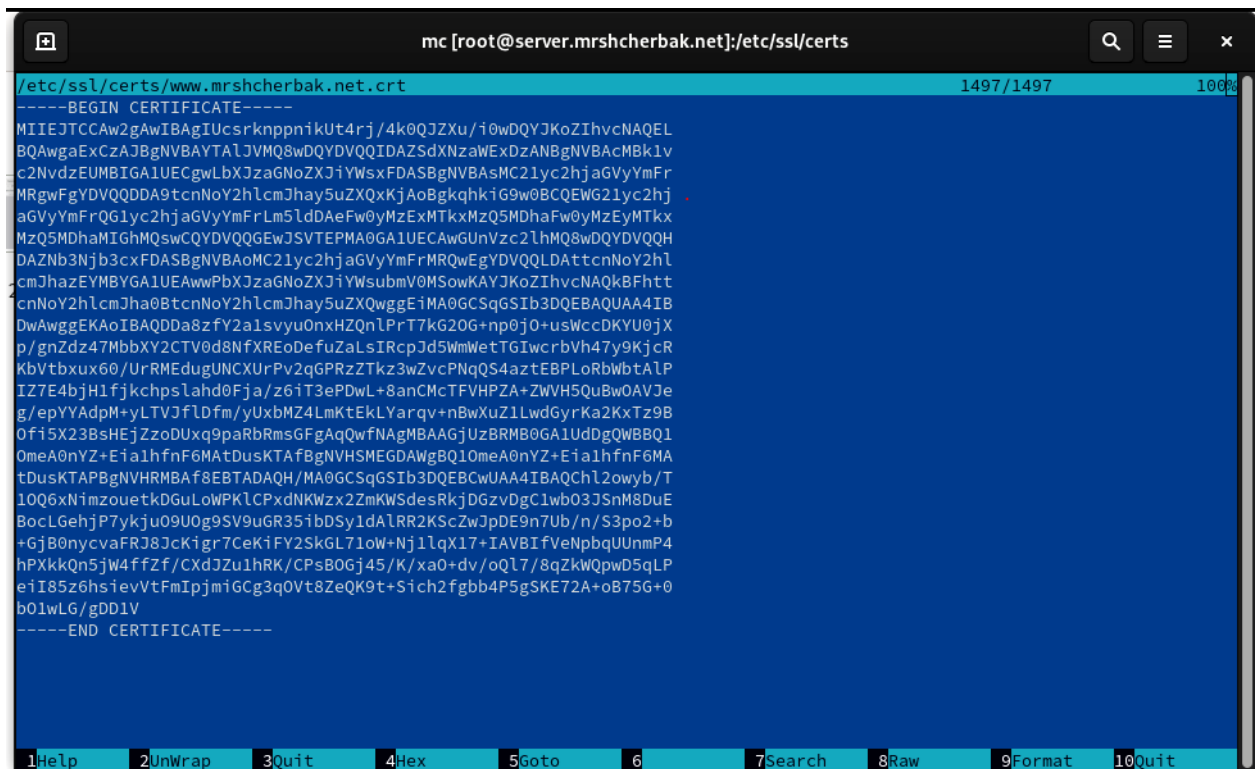
[mrshcherbak@server.mrshcherbak.net ~]$ sudo -i
[sudo] password for mrshcherbak:
[root@server.mrshcherbak.net ~]# mkdir -p /etc/pki/tls/private
[root@server.mrshcherbak.net ~]# ln -s /etc/pki/tls/private /etc/ssl/private
[root@server.mrshcherbak.net ~]# cd /etc/pki/tls/private
```

Рис.1.2. Выполнение команд

[illegible]

Рис.1.3. Выполнение действий

Сгенерированные ключ и сертификат появились в соответствующих каталогах /etc/ssl/private и /etc/ssl/certs (рис.1.4 – рис.1.5).



3. Для перехода веб-сервера www.mrshcherbak.net на функционирование через протокол HTTPS требуется изменить его конфигурационный файл. Для этого перешла в каталог с конфигурационными файлами, открыла на редактирование

файл `/etc/httpd/conf.d/www.mrshcherbak.net.conf` и изменила его содержимое (рис.1.6).

Файл конфигурации Apache представляет собой виртуальный хост для веб-сайта `www.mrshcherbak.net`, настроенный для прослушивания на портах 80 (HTTP) и 443 (HTTPS).

`<VirtualHost *:80>`: начало блока конфигурации для виртуального хоста, прослушивающего порт 80.

`ServerAdmin webmaster@mrshcherbak.net`: адрес эл. почты администратора сервера.

`DocumentRoot /var/www/html/www.mrshcherbak.net`: путь к каталогу, где расположены файлы веб-сайта.

`ServerName www.mrshcherbak.net`: определяет основное имя хоста для этого виртуального хоста.

`ServerAlias www.mrshcherbak.net`: позволяет указать дополнительные имена хостов, которые также считаются этим виртуальным хостом.

`ErrorLog logs/www.mrshcherbak.net-error_log`: указывает файл журнала для записи ошибок.

`CustomLog logs/www.mrshcherbak.net-access_log common`: указывает файл журнала для записи доступа.

`RewriteEngine on`: включает модуль перенаправления URL.

`RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]`: определяет правило перезаписи, которое перенаправляет все запросы с HTTP на HTTPS (перенаправление с кодом ответа 301).

`</VirtualHost>`: закрывает блок конфигурации для виртуального хоста на порту 80.

`<IfModule mod_ssl.c>`: начинает блок конфигурации, который будет применен только в том случае, если модуль SSL загружен.

`<VirtualHost *:443>`: начало блока конфигурации для виртуального хоста, прослушивающего порт 443 (HTTPS).

`SSLEngine on`: включает поддержку SSL.

SSLCertificateFile /etc/ssl/certs/www.mrshcherbak.net.crt: путь к файлу сертификата SSL.

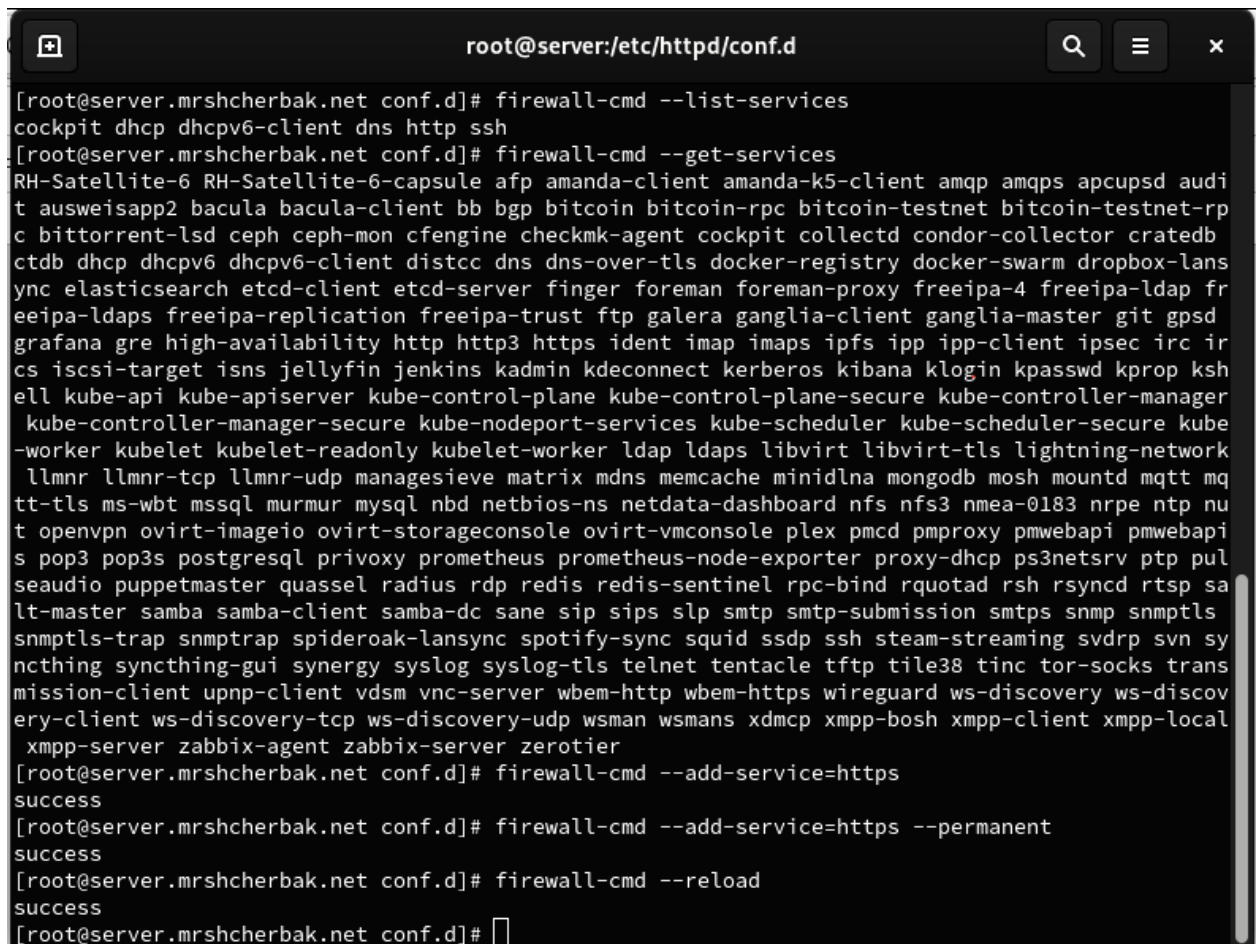
SSLCertificateKeyFile /etc/ssl/private/www.mrshcherbak.net.key: путь к файлу с закрытым ключом SSL.

</VirtualHost>: закрывает блок конфигурации для виртуального хоста на порту 443.

</IfModule>: закрывает блок конфигурации, применяемый только если модуль SSL загружен.

Рис.1.6. Содержимое файла /etc/httpd/conf.d/www.mrshcherbak.net.conf

4. Внесла изменения в настройки межсетевого экрана на сервере, разрешив работу с https (рис.1.7). После чего перезапустила веб-сервер с помощью команды `systemctl restart httpd`.

A terminal window titled 'root@server:/etc/httpd/conf.d' with search, menu, and close icons in the top right. The terminal shows the following commands and output:

```
[root@server.mrshcherbak.net conf.d]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http ssh
[root@server.mrshcherbak.net conf.d]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audi
t ausweisapp2 bacula bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rp
c bittorrent-lsd ceph ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb
ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lans
ync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap fr
eeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd
grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ir
cs iscsi-target isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop ksh
ell kube-api kube-apiserver kube-control-plane kube-control-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-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mq
tt-tls ms-wbt mssql murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nu
t openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapi
s pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps3netsrv ptp pul
seaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp sa
lt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls
snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn sy
ncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks trans
mission-client upnp-client vdsm vnc-server wbem-http wbem-https wireguard ws-discovery ws-discov
ery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local
xmpp-server zabbix-agent zabbix-server zerotier
[root@server.mrshcherbak.net conf.d]# firewall-cmd --add-service=https
success
[root@server.mrshcherbak.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@server.mrshcherbak.net conf.d]# firewall-cmd --reload
success
[root@server.mrshcherbak.net conf.d]#
```

Рис.1.7. Выполнение команд

5. На виртуальной машине client в строке браузера ввела название веб-сервера www.mrshcherbak.net и убедилась, что произошло автоматическое переключение на работу по протоколу HTTPS (рис.1.8). На открывшейся странице с сообщением о незащищённости соединения нажала кнопку «Дополнительно», затем добавила адрес своего сервера в постоянные исключения (рис.1.9). Затем просмотрела содержание сертификата (рис.1.10 – рис.1.11).

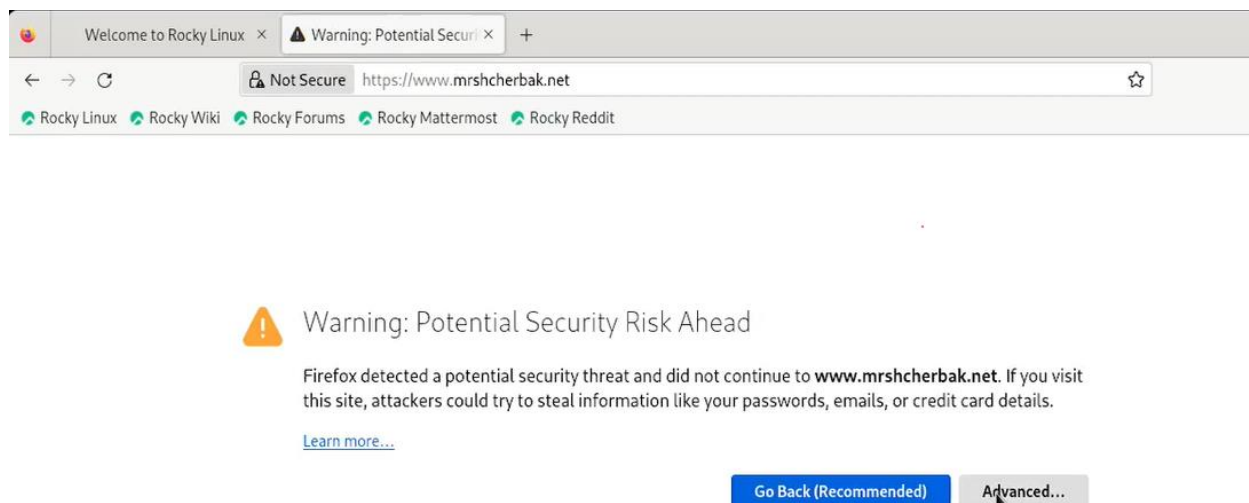


Рис.1.8. Автоматическое переключение на работу по протоколу HTTPS

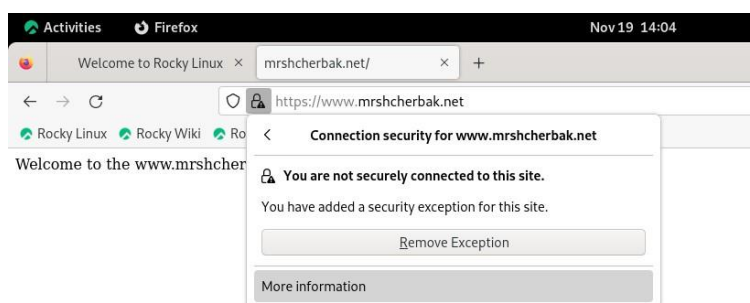


Рис.1.9. Добавление адреса сервера в постоянные исключения

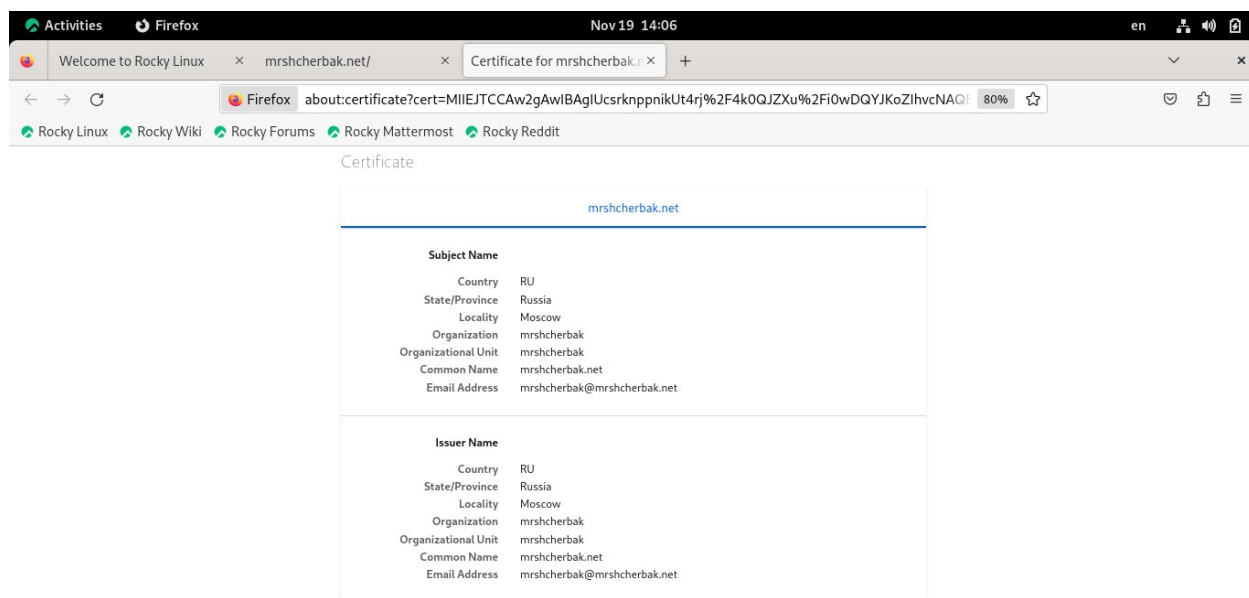


Рис.1.10. Просмотр информации о сертификате

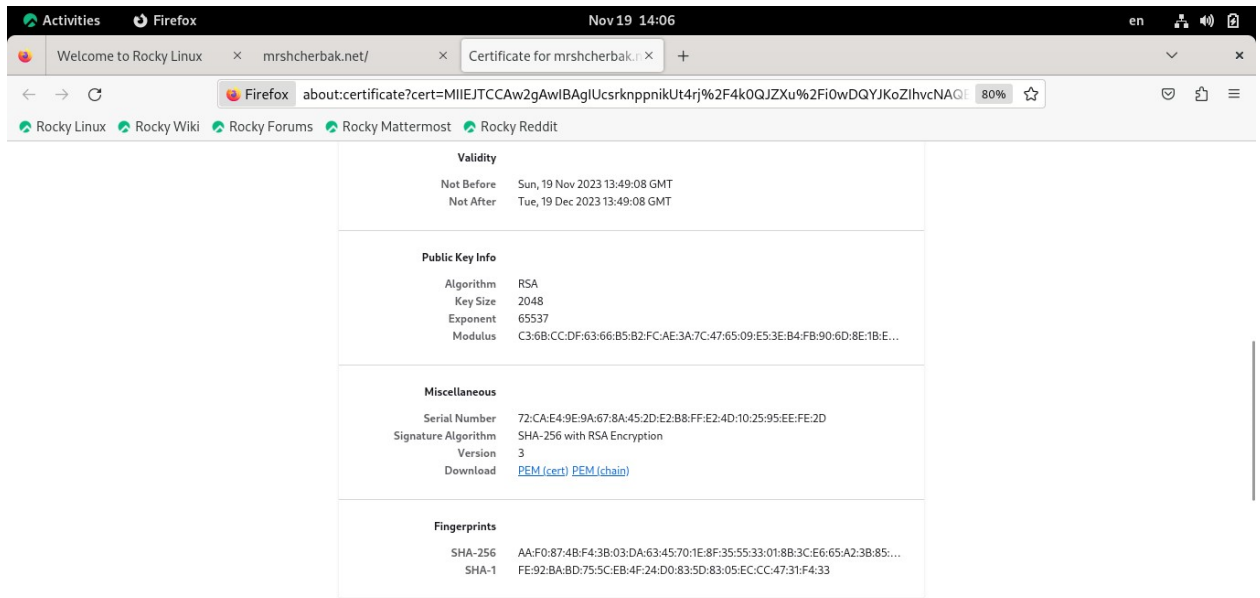


Рис.1.11. Просмотр информации о сертификате

2. Конфигурирование HTTP-сервера для работы с PHP

1. Установила пакеты для работы с PHP (рис.2.1).

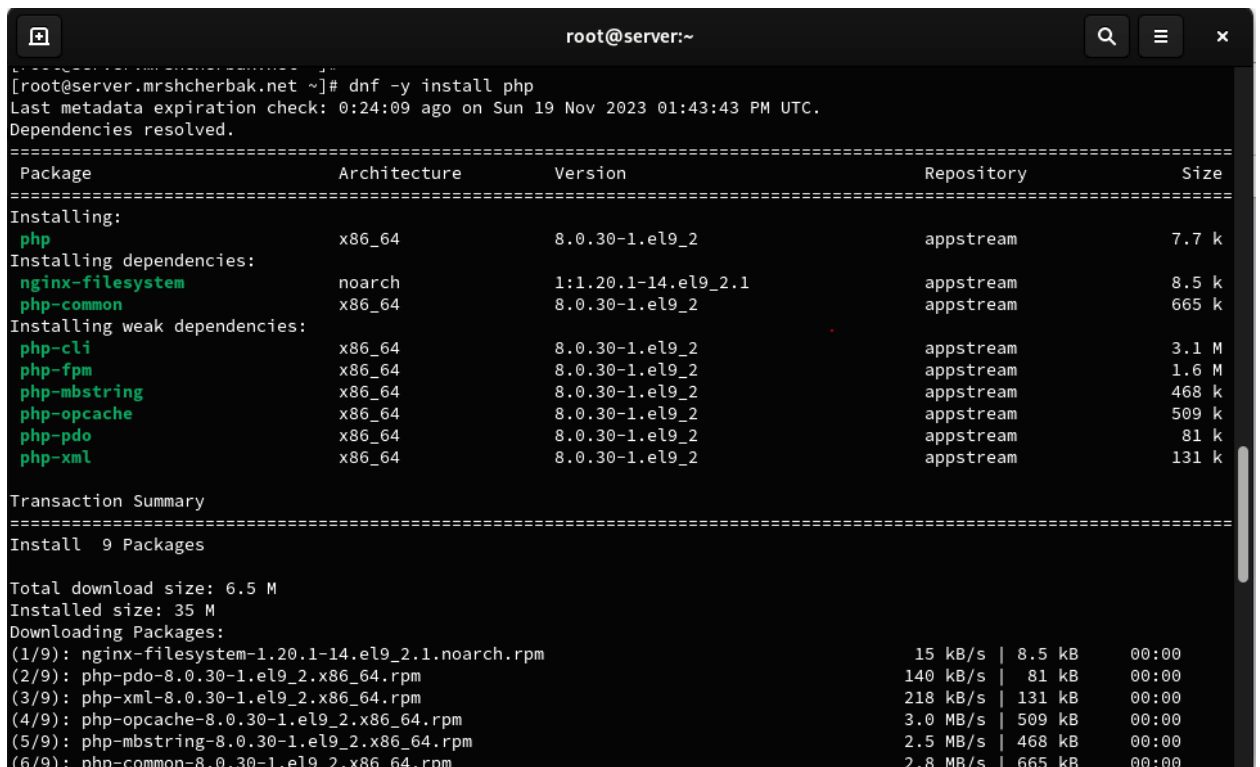


Рис.2.1. Установка пакетов для работы с PHP

2. В каталоге /var/www/html/www.mrshcherbak.net заменила файл index.html на index.php следующего содержания (рис.2.2).

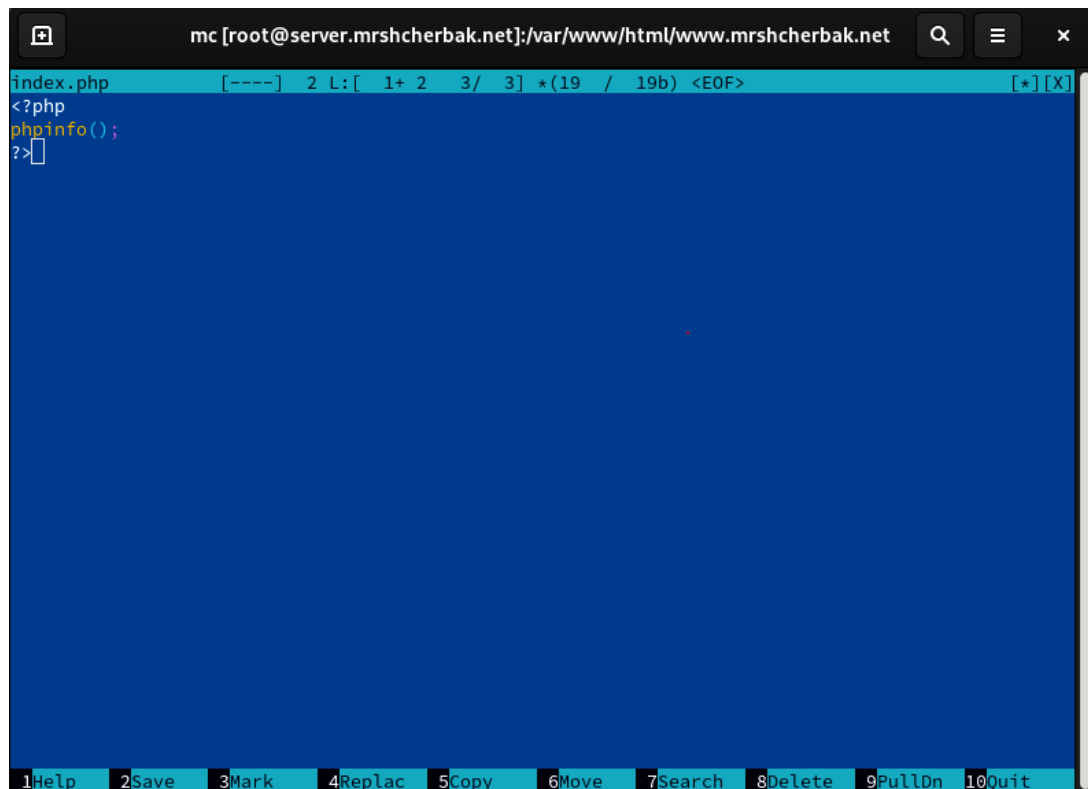


Рис.2.2. Содержимое файла index.php

3. Скорректировала права доступа в каталог с веб-контентом и восстановила контекст безопасности в SELinux, затем перезапустила HTTP-сервер (рис.2.3).

```
[root@server.mrshcherbak.net www.mrshcherbak.net]# chown -R apache:apache /var/www
[root@server.mrshcherbak.net www.mrshcherbak.net]# restorecon -vR /etc
Relabeled /etc/sysconfig/network-scripts/ifcfg-eth1 from unconfined_u:object_r:user_tmp_t:s0 to u
nconfined_u:object_r:net_conf_t:s0
[root@server.mrshcherbak.net www.mrshcherbak.net]# restorecon -vR /var/www
[root@server.mrshcherbak.net www.mrshcherbak.net]# systemctl restart httpd
[root@server.mrshcherbak.net www.mrshcherbak.net]#
```

Рис.2.3. выполнение команд

4. На виртуальной машине client в строке браузера ввела название веб-сервера www.mrshcherbak.net и убедилась, что выводится страница с информацией об используемой на веб-сервере версии PHP (рис.2.4).

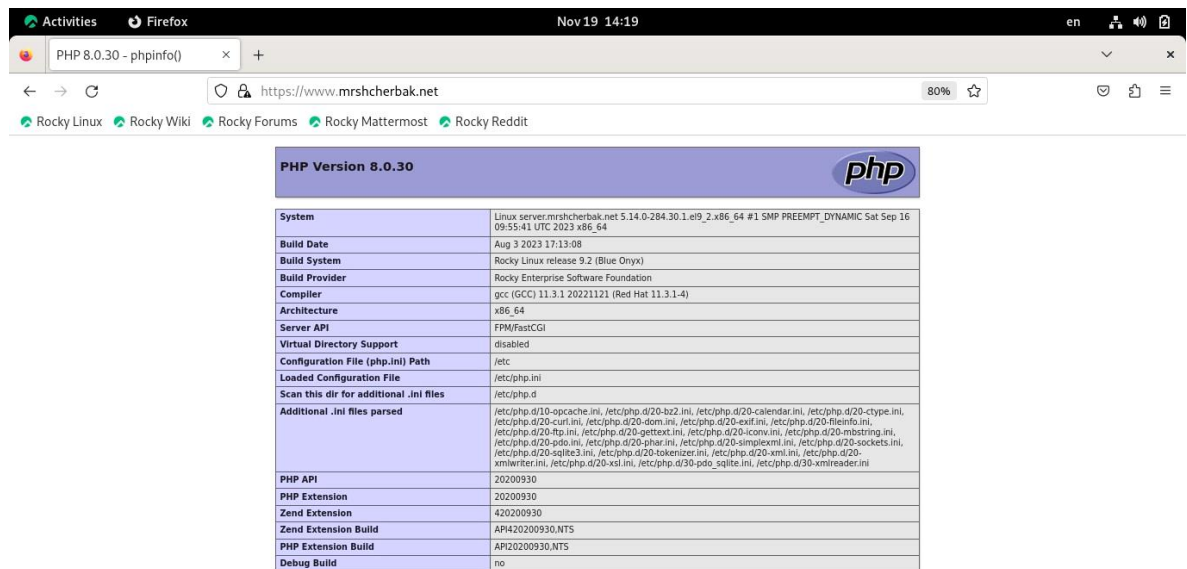


Рис.2.4. Страница с информацией об используемой на веб-сервере версии PHP

3. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине server перешла в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/http и в соответствующие каталоги скопировала конфигурационные файлы (рис.3.1).

```
[root@server.mrshcherbak.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/php.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.mrshcherbak.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.mrshcherbak.net.conf'? y
[root@server.mrshcherbak.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.mrshcherbak.net/index.html'? y
cp: overwrite '/vagrant/provision/server/http/var/www/html/www.mrshcherbak.net/index.php'? y
[root@server.mrshcherbak.net conf.d]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/private
[root@server.mrshcherbak.net conf.d]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.mrshcherbak.net conf.d]# cp -R /etc/pki/tls/private/www.mrshcherbak.net.key /vagrant/provision/server/http/etc/pki/tls/private
cp: overwrite '/vagrant/provision/server/http/etc/pki/tls/private/www.mrshcherbak.net.key'? y
[root@server.mrshcherbak.net conf.d]# cp -R /etc/pki/tls/certs/www.mrshcherbak.net.crt /vagrant/provision/server/http/etc/pki/tls/certs
cp: overwrite '/vagrant/provision/server/http/etc/pki/tls/certs/www.mrshcherbak.net.crt'? y
[root@server.mrshcherbak.net conf.d]#
```

Рис.3.1. Выполнение команд

2. В имеющийся скрипт /vagrant/provision/server/http.sh внесла изменения, добавив установку PHP и настройку межсетевого экрана, разрешающую работать с https (рис.3.2).

```
mc [root@server.mrshcherbak.net]:/vagrant/provision/server
http.sh [----] 23 L: [ 1+15 16/ 26] *(359 / 612b) 0010 0x00A [*][X]
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y groupinstall "Basic Web Server"
dnf -y install php

echo "Copy configuration files"
cp -R /vagrant/provision/server/http/etc/httpd/* /etc/httpd
cp -R /vagrant/provision/server/http/var/www/* /var/www

chown -R apache:apache /var/www

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

echo "Configure firewall"
firewall-cmd --add-service=http
firewall-cmd --add-service=http --permanent
firewall-cmd --add-service=https
firewall-cmd --add-service=https --permanent

echo "Start http service"
systemctl enable httpd
systemctl start httpd

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис.3.2. Содержимое файла /vagrant/provision/server/http.sh

Вывод: таким образом, в ходе выполнения л/р №5, я приобрела практические навыки по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

Контрольные вопросы

1. В чём отличие HTTP от HTTPS?

Основное отличие между HTTP (HyperText Transfer Protocol) и HTTPS (HyperText Transfer Protocol Secure) заключается в безопасности передачи данных. HTTP передает информацию в открытом виде, что делает её уязвимой для перехвата и прослушивания. В то время как HTTPS представляет собой расширение HTTP с добавлением протокола шифрования TLS/SSL, обеспечивая защищенную передачу данных между клиентом и сервером. Это шифрование защищает информацию от несанкционированного доступа, что особенно важно при передаче чувствительных данных, таких как логины, пароли и банковские данные.

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

- Шифрование данных: HTTPS использует протокол TLS/SSL для шифрования данных, передаваемых между клиентом и сервером. Это обеспечивает конфиденциальность и целостность информации.
- Идентификация сервера: при использовании HTTPS, сервер предоставляет цифровой сертификат, который подтверждает его подлинность. Это помогает предотвращать атаки типа "Man-in-the-Middle", где злоумышленник пытается подменить реальный сервер.
- Цифровая подпись: цифровые сертификаты используют цифровые подписи для подтверждения подлинности сервера. Это также обеспечивает доверие к серверу и поддерживает безопасную передачу данных.
- Защита от атак: HTTPS помогает защититься от различных видов атак, таких как перехват данных, внедрение кода и подделка идентификации.

3. Что такое сертификационный центр? Приведите пример

Сертификационный центр (Центр сертификации) - это доверенная организация, ответственная за выдачу цифровых сертификатов, подтверждающих подлинность электронных сущностей, таких как веб-сайты, электронные почтовые ящики и программы.

Пример сертификационного центра - "Let's Encrypt". Это бесплатный, автоматизированный и общедоступный центр сертификации, который предоставляет цифровые сертификаты для обеспечения безопасной передачи данных по протоколу HTTPS. Let's Encrypt сделал процесс получения и установки сертификатов более доступным для владельцев веб-сайтов, способствуя широкому распространению защищенного соединения в интернете.