**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).

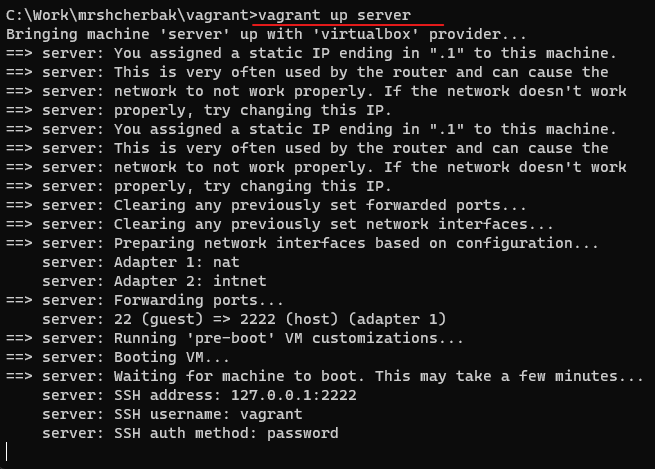


Рис.1.1. Запуск виртуальной машины Server

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.



Рис.1.2. Выполнение команд

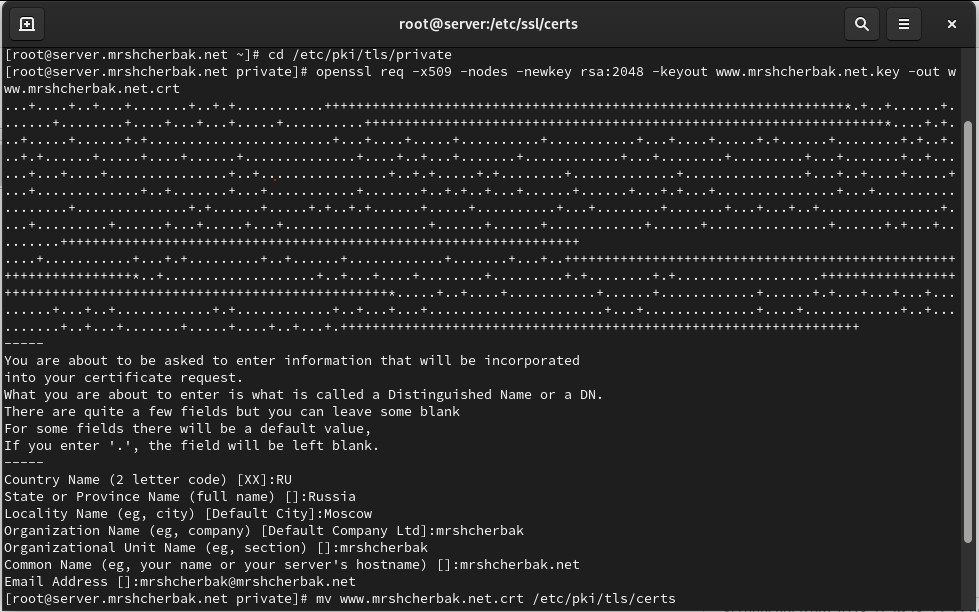


Рис.1.3. Выполнение действий

Сгенерированные ключ и сертификат появились в соответствующих каталогах

/etc/ssl/private и /etc/ssl/certs (рис.1.4 – рис.1.5).

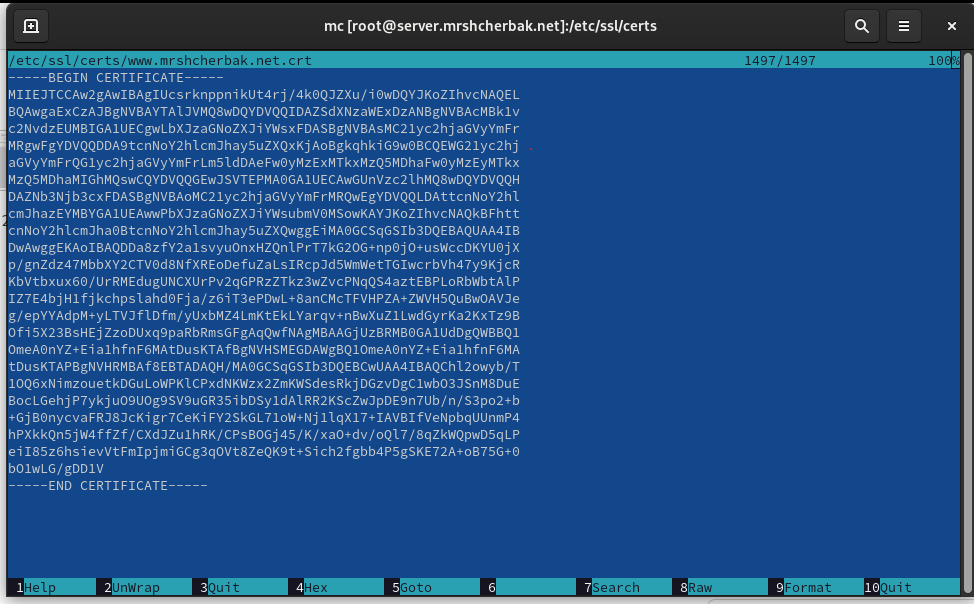


Рис.1.4. Сгенерированный сертификат

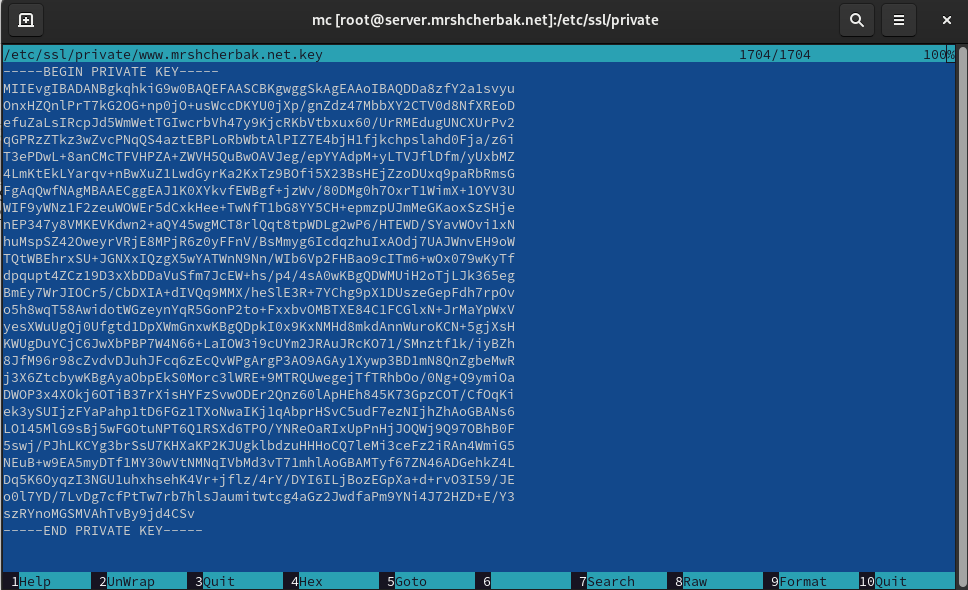


Рис.1.5. Сгенерированный ключ

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.

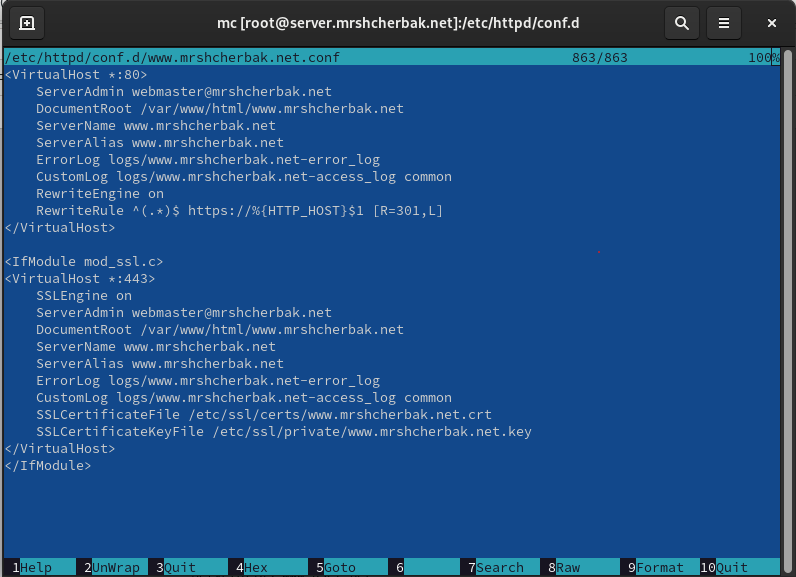


Рис.1.6. Содержимое файла /etc/httpd/conf.d/www.mrshcherbak.net.conf

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.

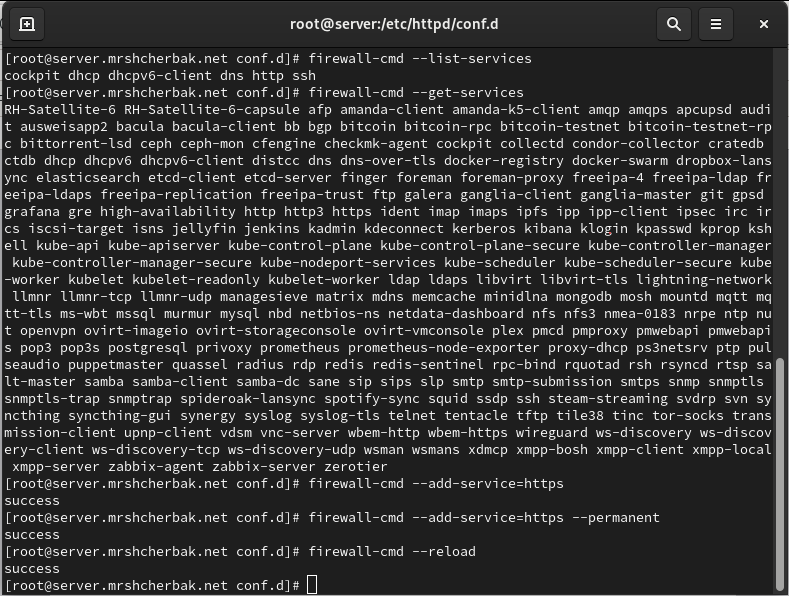


Рис.1.7. Выполнение команд

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).

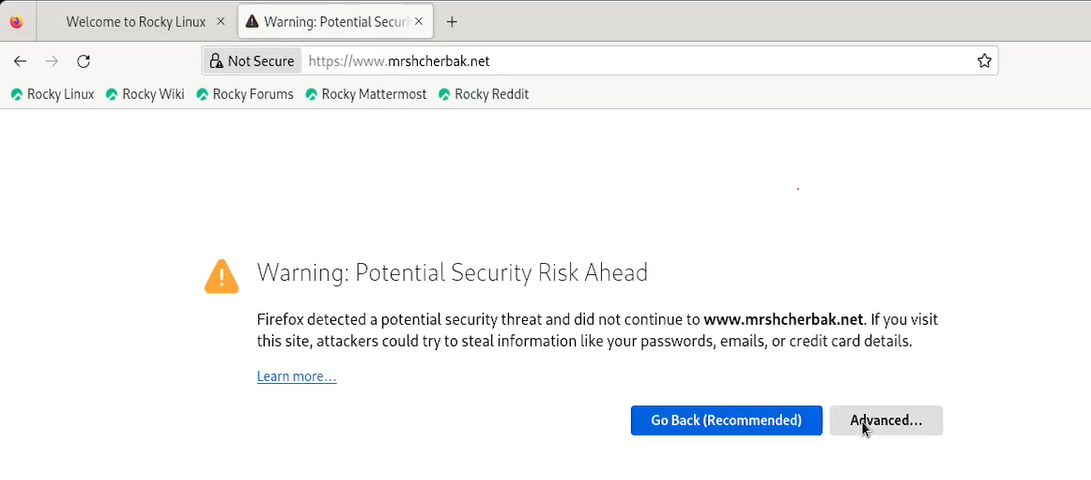


Рис.1.8. Автоматическое переключение на работу по протоколу HTTPS

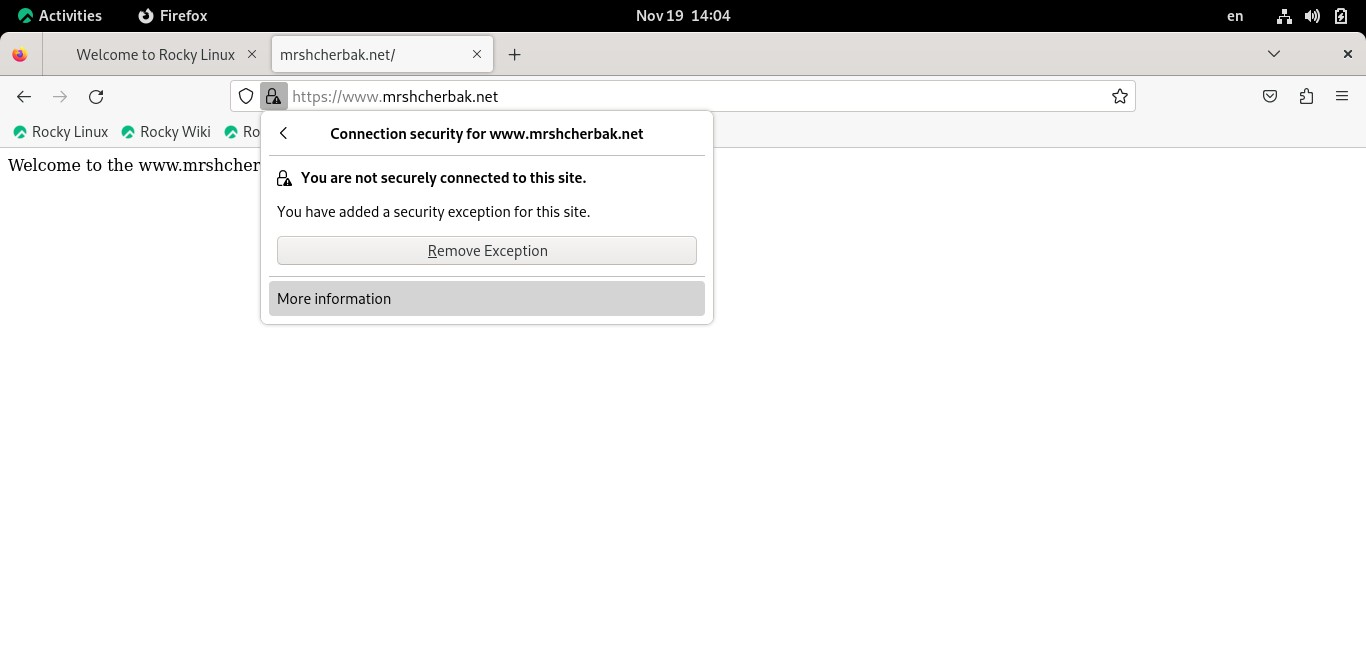


Рис.1.9. Добавление адреса сервера в постоянные исключения

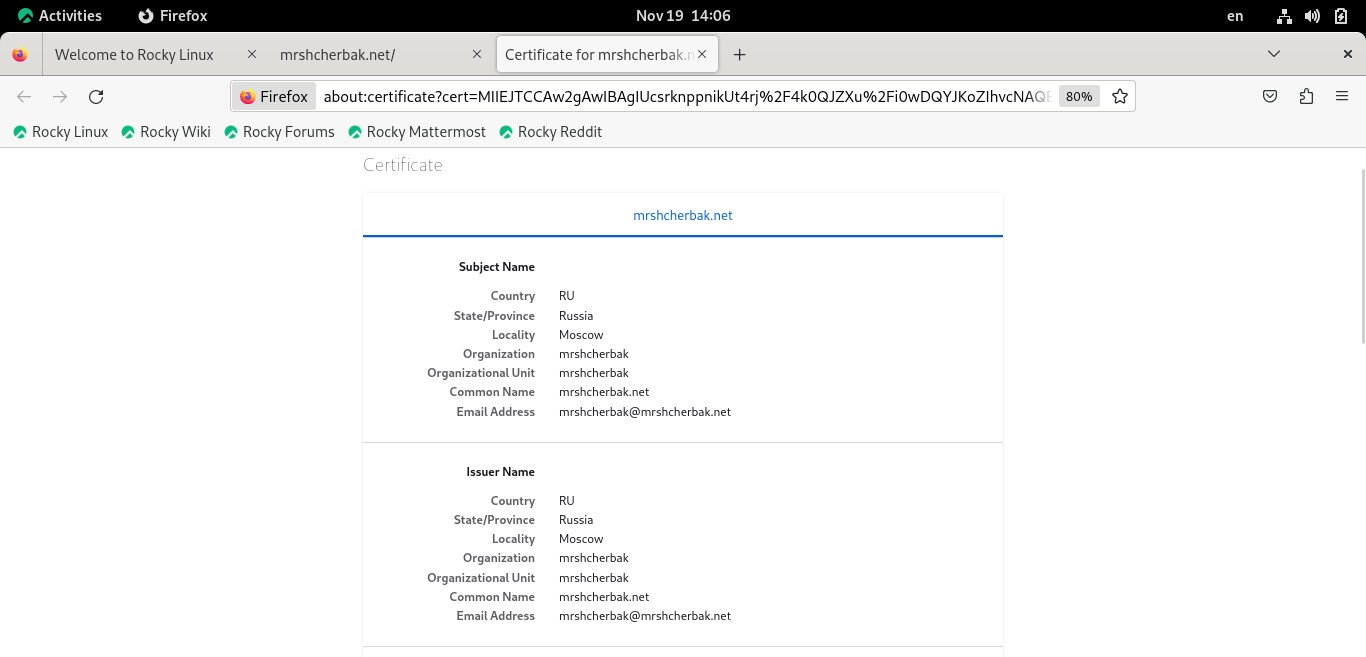


Рис.1.10. Просмотр информации о сертификате

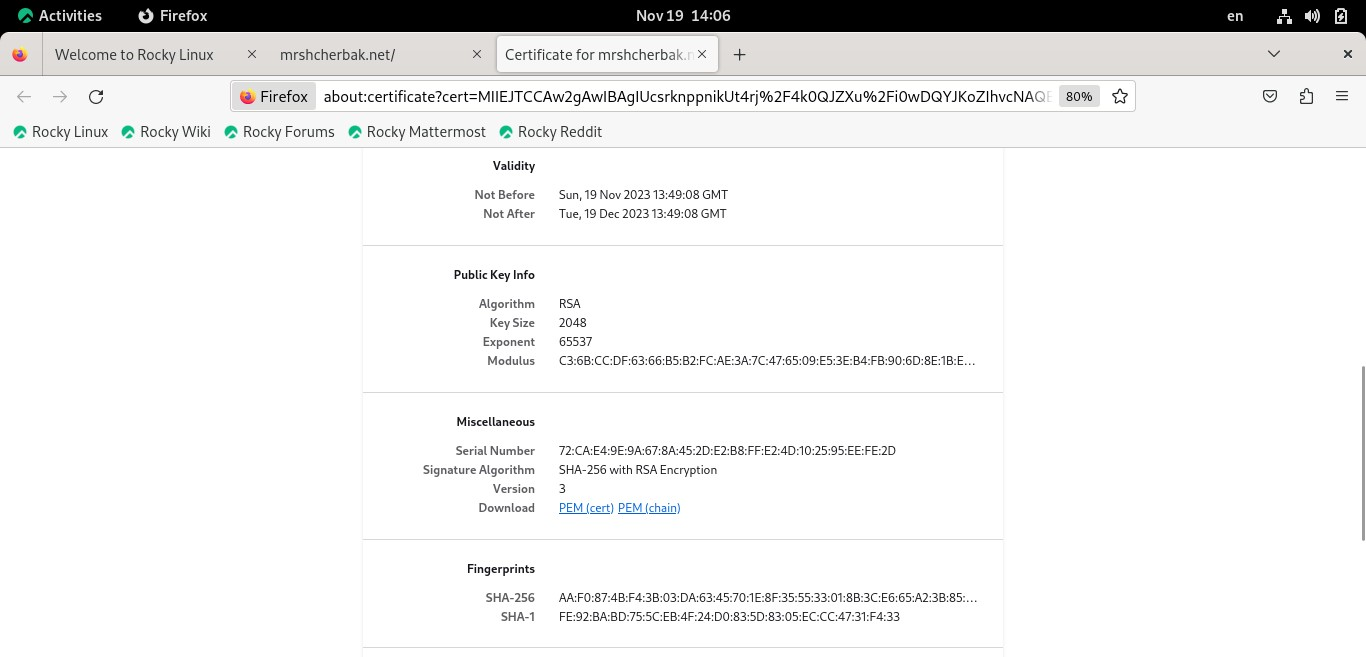


Рис.1.11. Просмотр информации о сертификате

**2. Конфигурирование HTTP-сервера для работы с PHP**

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

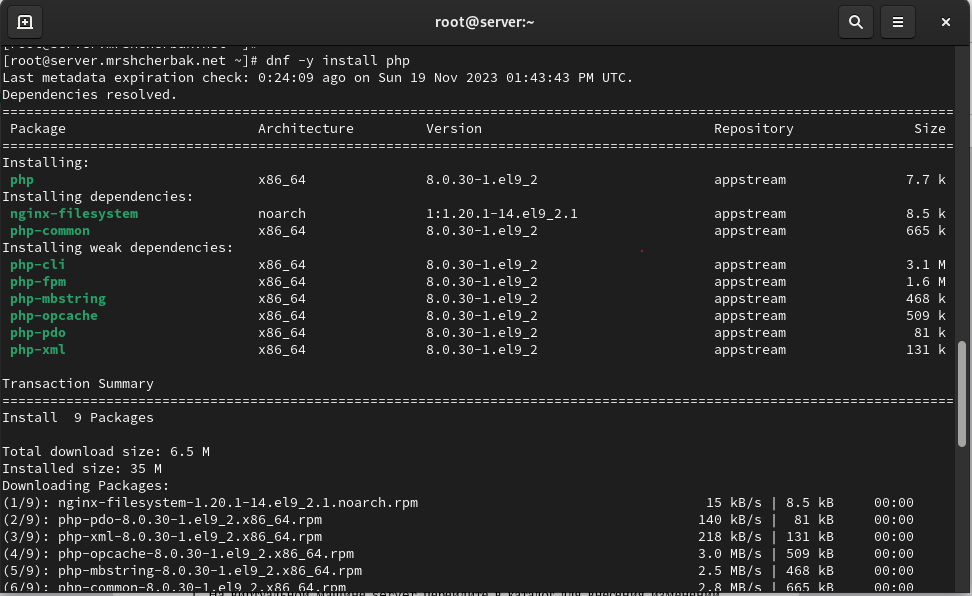


Рис.2.1. Установка пакетов для работы с PHP

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).

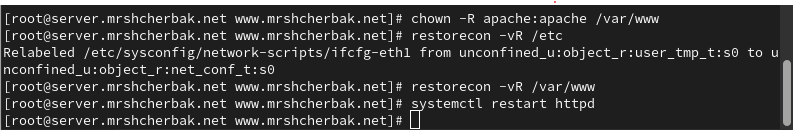


Рис.2.3. выполнение команд

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).

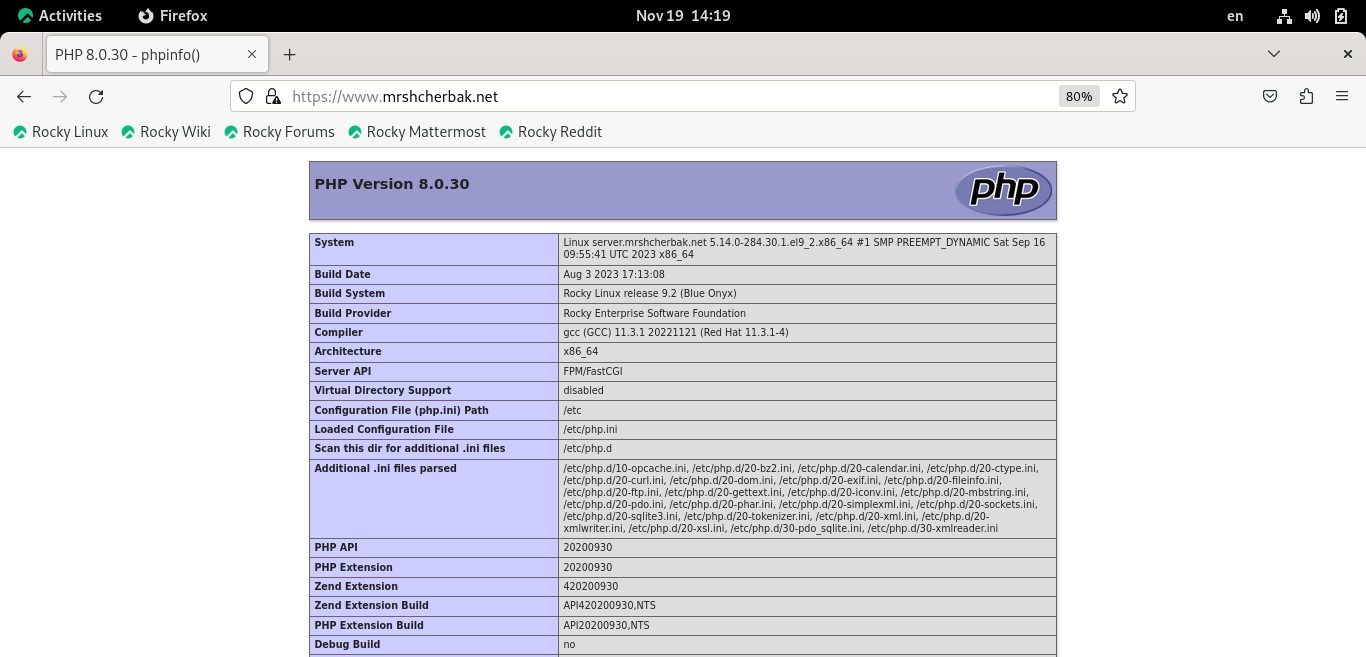


Рис.2.4. Страница с информацией об используемой на веб-сервере версии PHP

**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).



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).

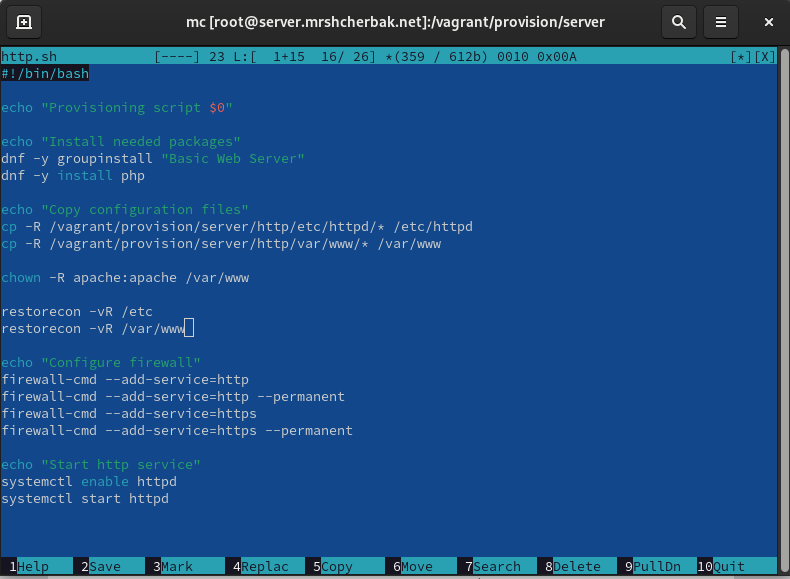


Рис.3.2. Содержимое файла /vagrant/provision/server/http.sh

# 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.

1. 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.

1. 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.