Отчёт по лабораторной работе 5

Расширенная настройка HTTP-сервера Apache

Сейдалиев Тагиетдин Ровшенович

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Выполнение

2.1 Конфигурирование HTTP-сервера для работы через протокол HTTPS

2.1.1 Генерация ключа и сертификата

В каталоге `/etc/pki/tls/private` был создан приватный ключ и самоподписанный сертификат для домена `www.trseidaliev.net`.
После перехода в каталог и создания символьной ссылки на `/etc/ssl/private` была выполнена генерация ключа и сертификата.
Заполненные поля сертификата включали страну, регион, город, логин пользователя, доменное имя и адрес электронной почты.

Итоговое содержимое каталога с ключом и сертификатом представлено ниже:

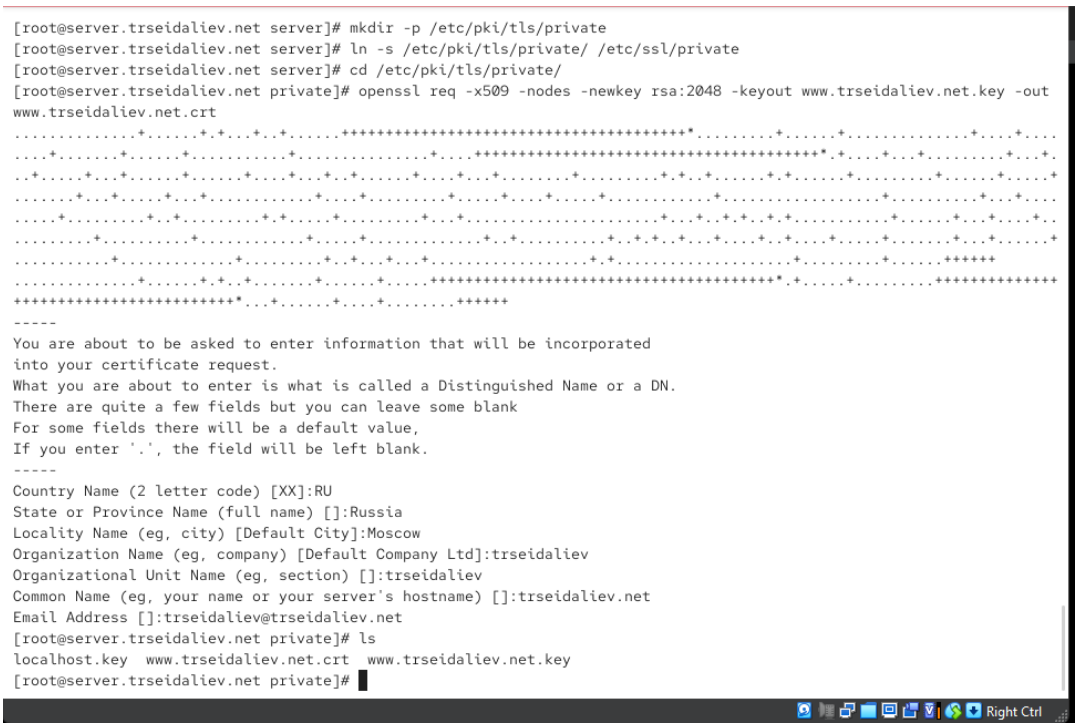


Рис. 2.1: Генерация ключа и сертификата

2.1.2 Настройка конфигурации виртуального хоста Apache

Конфигурационный файл `/etc/httpd/conf.d/www.trseidaliev.net.conf` был изменён в соответствии с требованиями для перехода сайта на HTTPS.

Содержимое файла приведено на скриншоте:

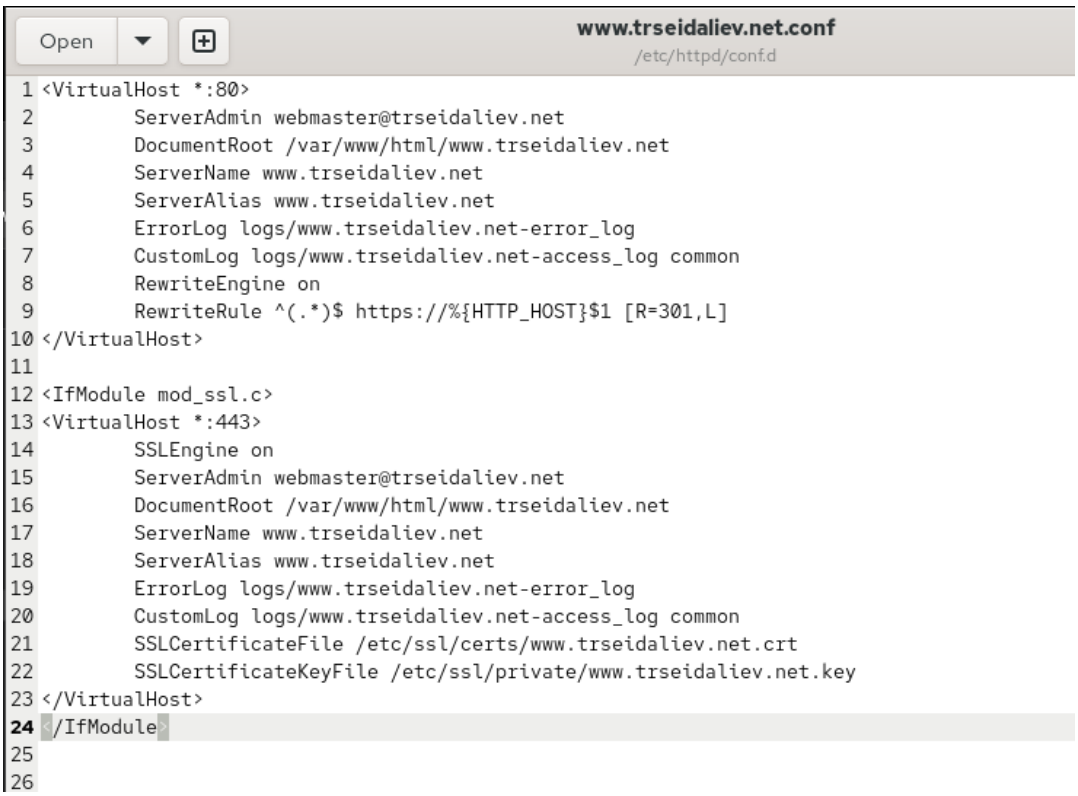


Рис. 2.2: Конфигурация виртуального хоста Apache

2.1.2.1 Пояснение конфигурации

Блок HTTP (порт 80)

Используется для перенаправления всего трафика на HTTPS.

Включён механизм `RewriteEngine`, выполняющий постоянное перенаправление (код 301) на соответствующий URL по протоколу HTTPS.

Блок HTTPS (порт 443)

Активируется при наличии модуля `mod_ssl`.

Включает SSL-движок, задаёт пути к сертификату и ключу, указанные при генерации.

Определяет DocumentRoot, основные параметры сервера и журналы доступа и ошибок.

2.1.3 Настройка межсетевого экрана

Для разрешения работы по протоколу HTTPS были внесены изменения в конфигурацию `firewalld`.

Сервис `https` был добавлен как временно, так и постоянно, после чего выполнена перезагрузка firewall.

Затем веб-сервер был перезапущен.

```
-trapper zabbix-web-service zero-k zerotier
[root@server.trseidaliev.net conf.d]# firewall-cmd --add-service=https
success
[root@server.trseidaliev.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@server.trseidaliev.net conf.d]# firewall-cmd --reload
success
[root@server.trseidaliev.net conf.d]# systemctl restart httpd
[root@server.trseidaliev.net conf.d]#
```

Рис. 2.3: Настройка firewall для HTTPS

2.1.4 Проверка работы HTTPS

При обращении к сайту `https://www.trseidaliev.net` с клиентской виртуальной машины произошло автоматическое перенаправление с HTTP на HTTPS.

Так как сертификат самоподписанный, браузер предложил добавить исключение безопасности.

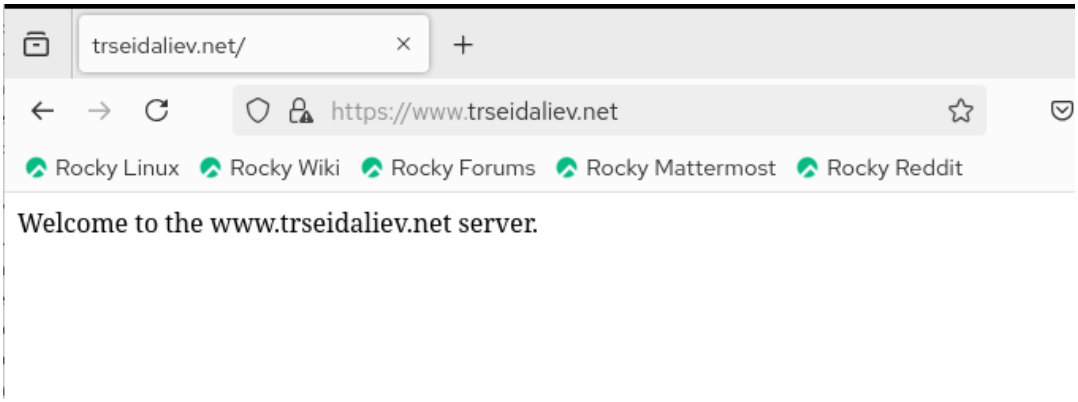


Рис. 2.4: Открытие сайта по HTTPS

Данные сертификата, отображаемые браузером:

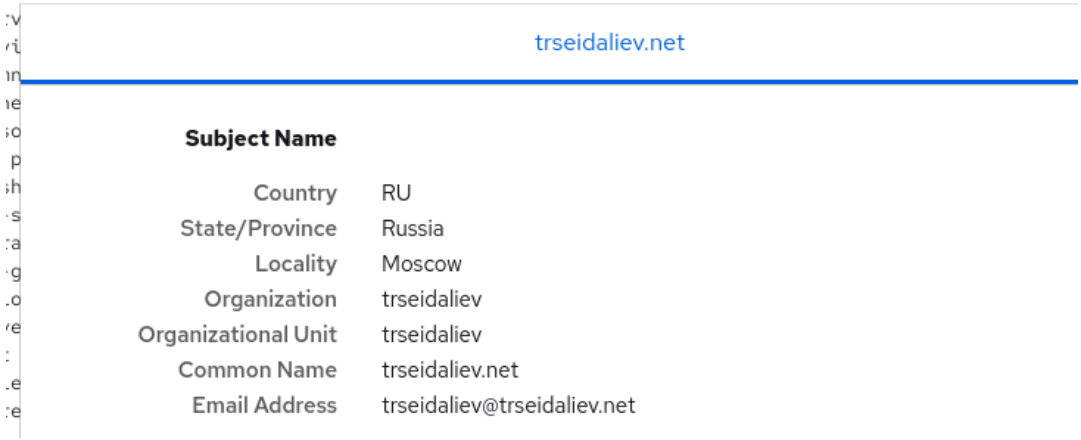


Рис. 2.5: Просмотр сертификата

2.2 Конфигурирование HTTP-сервера для работы с PHP

В каталоге `/var/www/html/www.trseidaliev.net` был заменён файл `index.html` на `index.php` со следующей конструкцией:

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

После установки PHP, корректировки владельцев каталога и восстановления SELinux контекстов был перезапущен HTTP-сервер.

Файл с PHP-кодом:

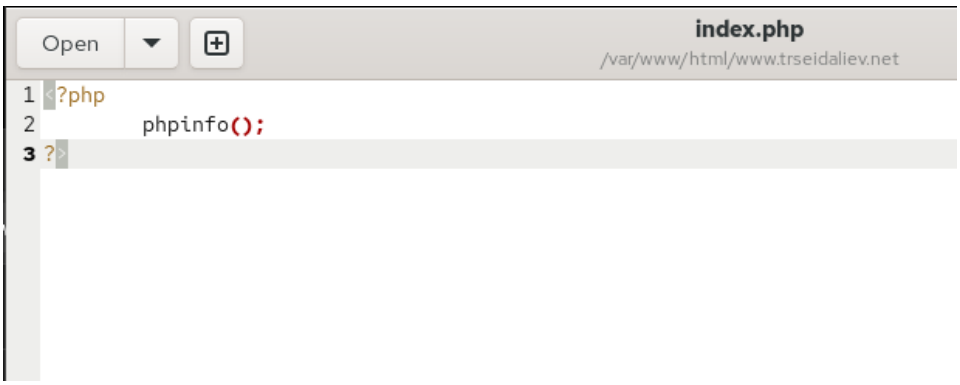


Рис. 2.6: index.php

В браузере отобразилась информация о конфигурации PHP:

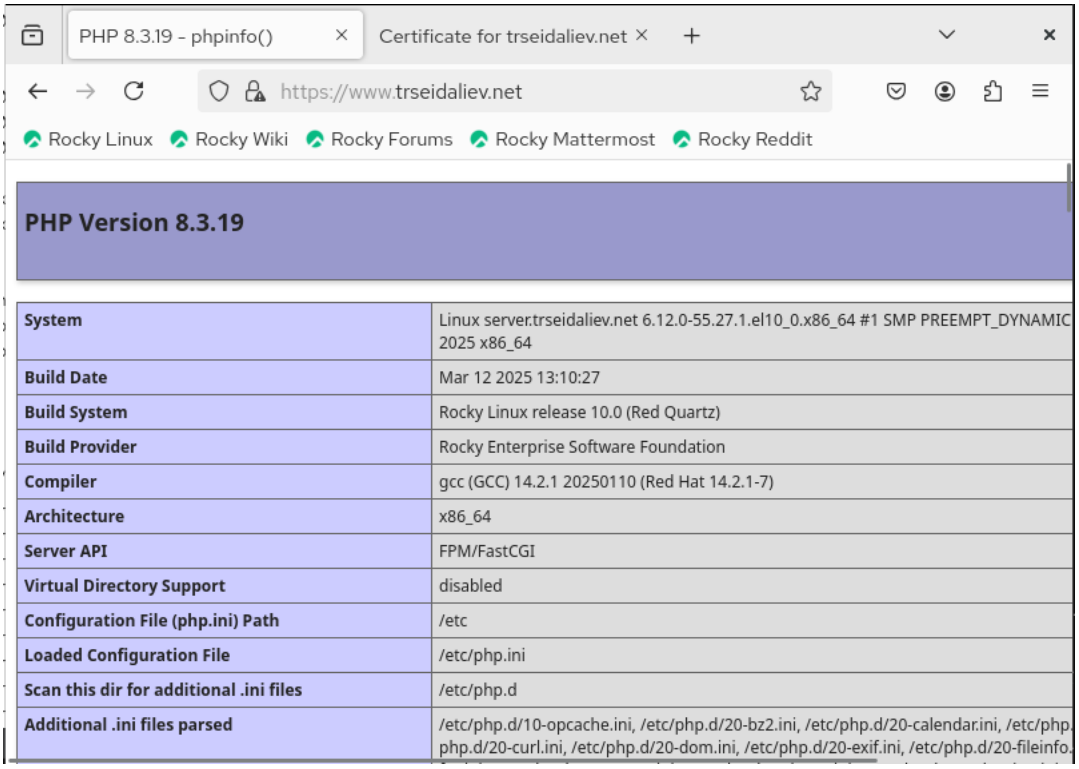


Рис. 2.7: phpinfo()

2.3 Внесение изменений в настройки внутреннего окружения виртуальной машины

Для сохранения текущего состояния веб-сервера и автоматизации повторного развёртывания были перенесены актуальные конфигурационные файлы в директорию провижининга `/vagrant/provision/server/http`.

Итоговое выполнение операций по копированию представлено на скриншоте:

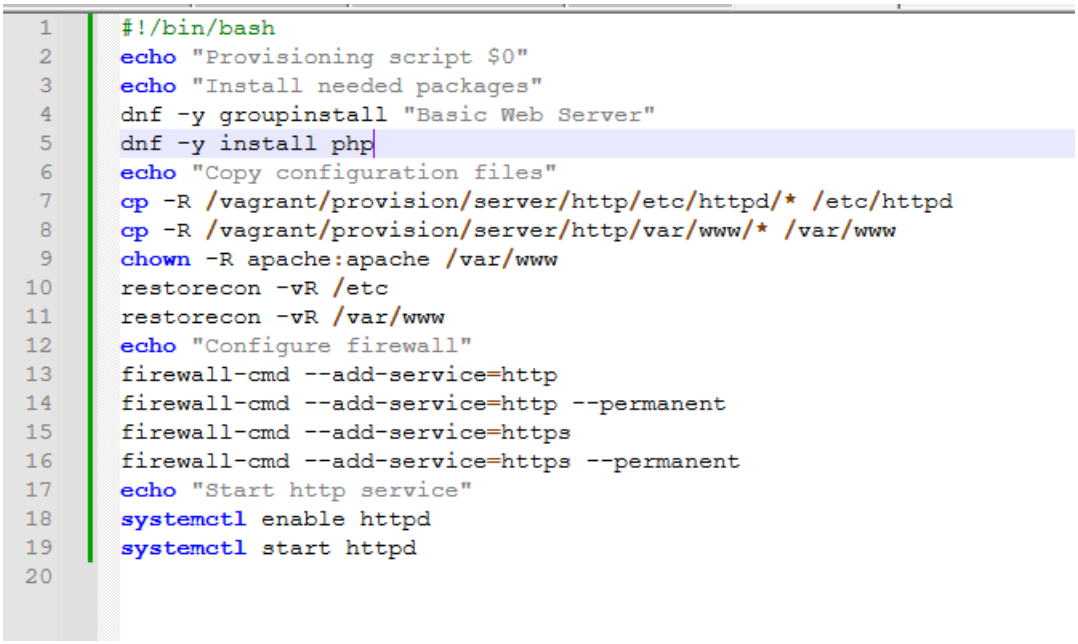
Рис. 2.8: Копирование конфигурационных файлов в провижининг

В файл `/vagrant/provision/server/http.sh` были добавлены команды для:

- установки PHP;
- настройки межсетевого экрана с разрешением сервиса HTTPS;
- перезапуска веб-сервера после применения настроек.

Это обеспечивает автоматическую установку всех необходимых компонентов при развёртывании виртуальной машины.

Фрагмент обновлённого скрипта показан ниже:



```
#!/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
```

Рис. 2.9: Обновлённый скрипт http.sh

3 Заключение

В ходе выполнения работы были последовательно выполнены все этапы настройки и проверки функционирования веб-сервера, его HTTPS-конфигурации и поддержки PHP. В результате:

- сгенерированы приватный ключ и самоподписанный сертификат для домена `www.trseidaliev.net`;
- настроен виртуальный хост Apache на работу по протоколу HTTPS с автоматическим перенаправлением с HTTP;
- внесены изменения в конфигурацию межсетевого экрана, что обеспечило доступность сервиса HTTPS;
- проведена проверка корректной работы SSL-сертификата и отображение сведений о нём в браузере;
- установлена поддержка PHP и успешно протестирован вывод информации с помощью `phpinfo()`;

- обновлён скрипт `http.sh`, содержащий установку PHP и разрешение HTTPS-трафика на уровне firewall.

4 Контрольные вопросы

4.1 1. В чём отличие HTTP от HTTPS?

HTTP — протокол передачи данных без шифрования.
Он не защищает трафик: данные передаются в открытом виде и могут быть перехвачены или изменены злоумышленником.

HTTPS — расширение HTTP, использующее криптографический протокол TLS/SSL.
Он обеспечивает шифрование канала, проверку подлинности сервера и целостности передаваемой информации.
Таким образом, HTTPS защищает данные от перехвата и подмены.

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

Безопасность обеспечивается комбинацией криптографических механизмов TLS:

- **Шифрование** — данные передаются в зашифрованном виде, что делает их недоступными для чтения третьими лицами.

- **Аутентификация сервера** — браузер проверяет сертификат, подтверждающий, что пользователь подключается именно к тому серверу, которому

доверяет.

- **Контроль целостности** — механизмы хеширования позволяют выявлять любые попытки изменения данных в процессе передачи.

Компания TLS гарантирует защищённый канал и минимизирует риски MITM-атак.

4.3 3. Что такое сертификационный центр? Приведите пример.

Сертификационный центр (Certificate Authority, CA) — организация, выдающая SSL/TLS-сертификаты.
Она подтверждает подлинность домена или организации, привязывая их к криптографическим ключам.
Браузеры и операционные системы доверяют сертификатам, подписанным такими центрами.

Примеры сертификационных центров:

- Let’s Encrypt
- DigiCert
- GlobalSign
- Comodo (Sectigo)