Отчет по лабораторной работе №5

Дисциплина: Администрирование сетевых подсистем

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

Целью этой работы является приобретение практических навыков по расширенному конфигурированию HTTPсервера Apache в части безопасности и возможности использования PHP.

# 2 Задание

1. Сгенерируйте криптографический ключ и самоподписанный сертификат безопасности для возможности перехода веб-сервера от работы через протокол HTTP к работе через протокол HTTPS (см. раздел 5.4.1).
2. Настройте веб-сервер для работы с PHP (см. раздел 5.4.2).
3. Напишите (или скорректируйте) скрипт для Vagrant, фиксирующий действия по расширенной настройке HTTP-сервера во внутреннем окружении виртуальной машины server (см. раздел 5.4.3).

# 3 Выполнение лабораторной работы

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

Загрузим операционную систему и перейдем в рабочий каталог с проектом: cd /var/tmp/user\_name/vagrant . Запустим виртуальную машину server: vagrant up server . (рис. 1).

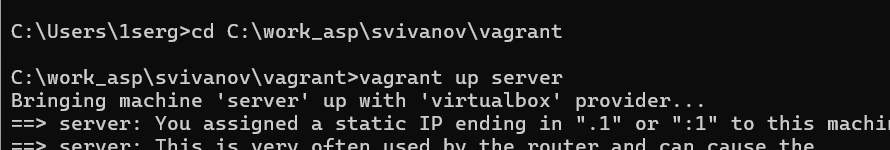


Рис. 1: Запуск server

На виртуальной машине server войдем под пользователем и откроем терминал. Перейдем в режим суперпользователя. В каталоге /etc/ssl создадим каталог private:

mkdir -p /etc/pki/tls/private

ln -s /etc/pki/tls/private /etc/ssl/private

cd /etc/pki/tls/private (рис. 2).

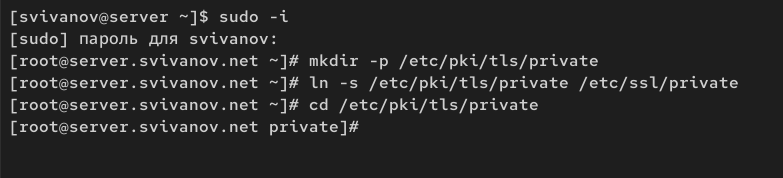


Рис. 2: Создание каталога private

Сгенерируем ключ и сертификат, используя следующую команду:

openssl req -x509 -nodes -newkey rsa:2048 -keyout www.svivanov.net.key -out www.svivanov.net.crt

mv www.svivanov.net.crt /etc/pki/tls/certs (рис. 3)

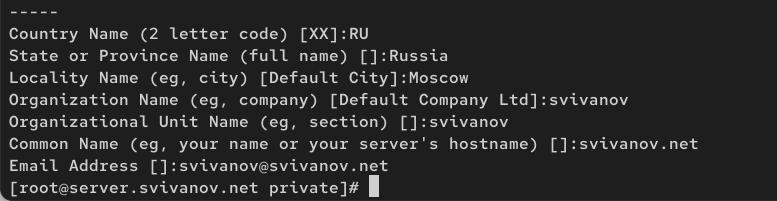


Рис. 3: Создание ключа и сертификата

Сгенерированные ключ и сертификат появятся в соответствующем каталоге /etc/ssl/private. Скопируем сертификат в каталог /etc/ssl/certs: cp /etc/ssl/private/www.svivanov.net.crt /etc/ssl/cert/ (рис. 4)

Копирование сертификата

Рис. 4: Копирование сертификата

Для перехода веб-сервера www.svivanov.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.svivanov.net.conf и заменим его содержимое на следующее: (рис. 5)

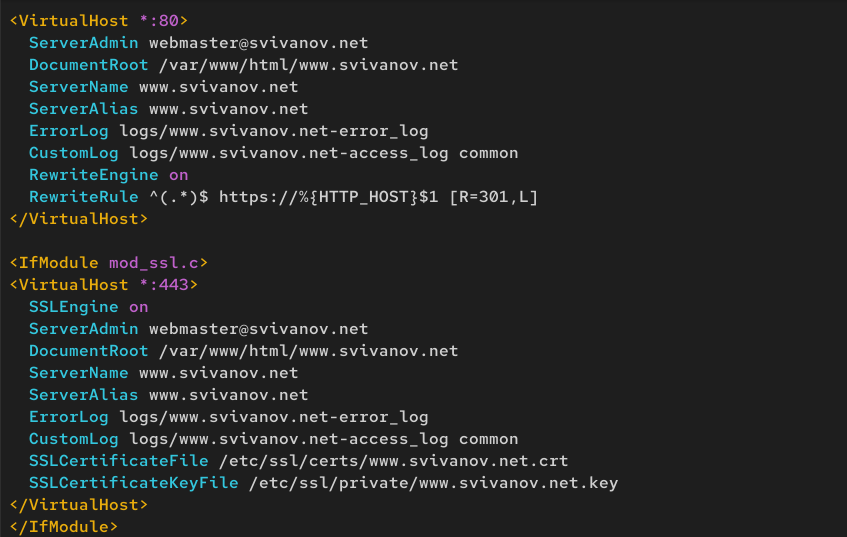


Рис. 5: Редактирование конф. файла

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

firewall-cmd –list-services

firewall-cmd –get-services

firewall-cmd –add-service=https

firewall-cmd –add-service=https –permanent

firewall-cmd –reload (рис. 6)

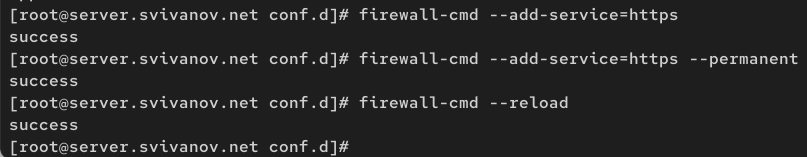


Рис. 6: Меняем настройки firewall

Перезапустим веб-сервер: systemctl restart httpd (рис. 7)

Перезапуск сервера

Рис. 7: Перезапуск сервера

На виртуальной машине client в строке браузера введем название веб-сервера www.svivanov.net и убедимся, что произойдёт автоматическое переключение на работу по протоколу HTTPS. (рис. 8)

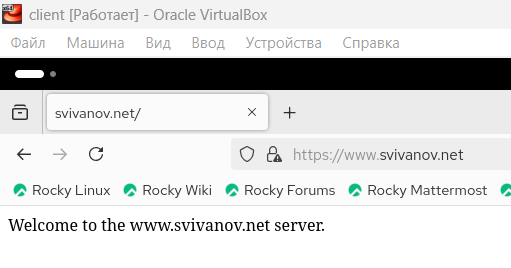


Рис. 8: Доступ к веб-серверу по HTTPS

На открывшейся странице с сообщением о незащищённости соединения нажмем кнопку «Дополнительно», затем добавим адрес вашего сервера в постоянные исключения. (рис. 9)

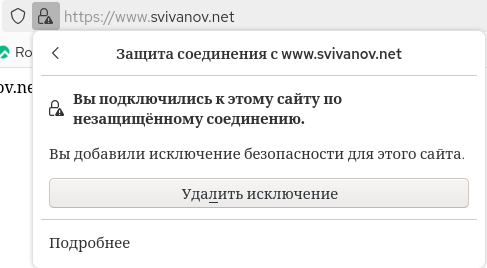


Рис. 9: Добавление в исключение

Затем просмотрим содержание сертификата. (рис. 10)

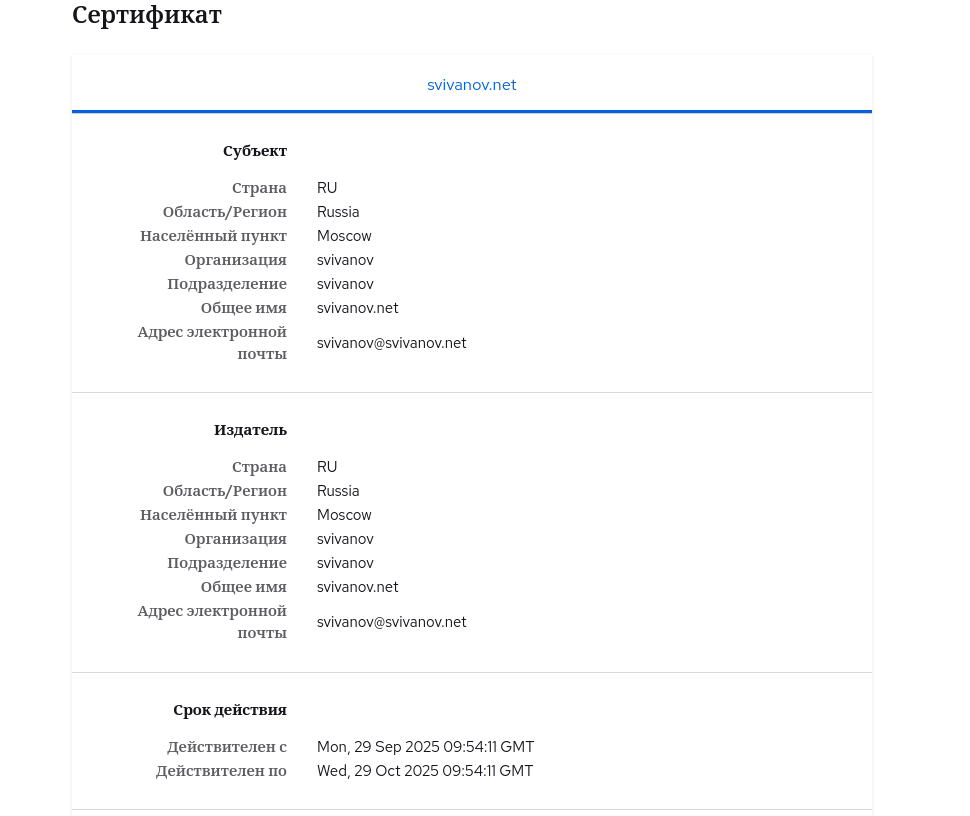


Рис. 10: Содержание сертификата

## 3.2 Конфигурирование HTTP-сервера для работы с PHP

Установим пакеты для работы с PHP: dnf -y install php. (рис. 11)

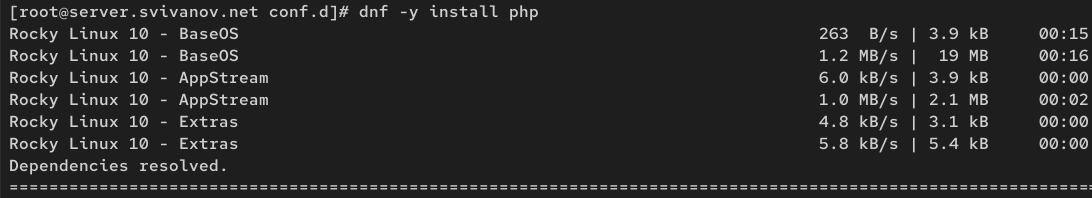


Рис. 11: Установка PHP

В каталоге /var/www/html/www.svivanov.net заменим файл index.html на index.php следующего содержания: (рис. 12)

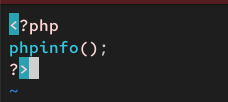


Рис. 12: Файл index.php

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 (рис. 13)

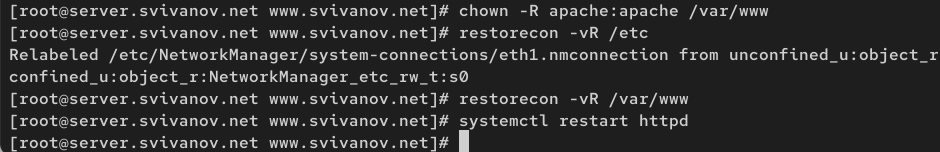


Рис. 13: Корректирование прав и перезапуск сервера

## 3.3 Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине client в строке браузера введем название веб-сервера www.svivanov.net и убедимся, что будет выведена страница с информацией об используемой на веб-сервере версии PHP. (рис. 14)

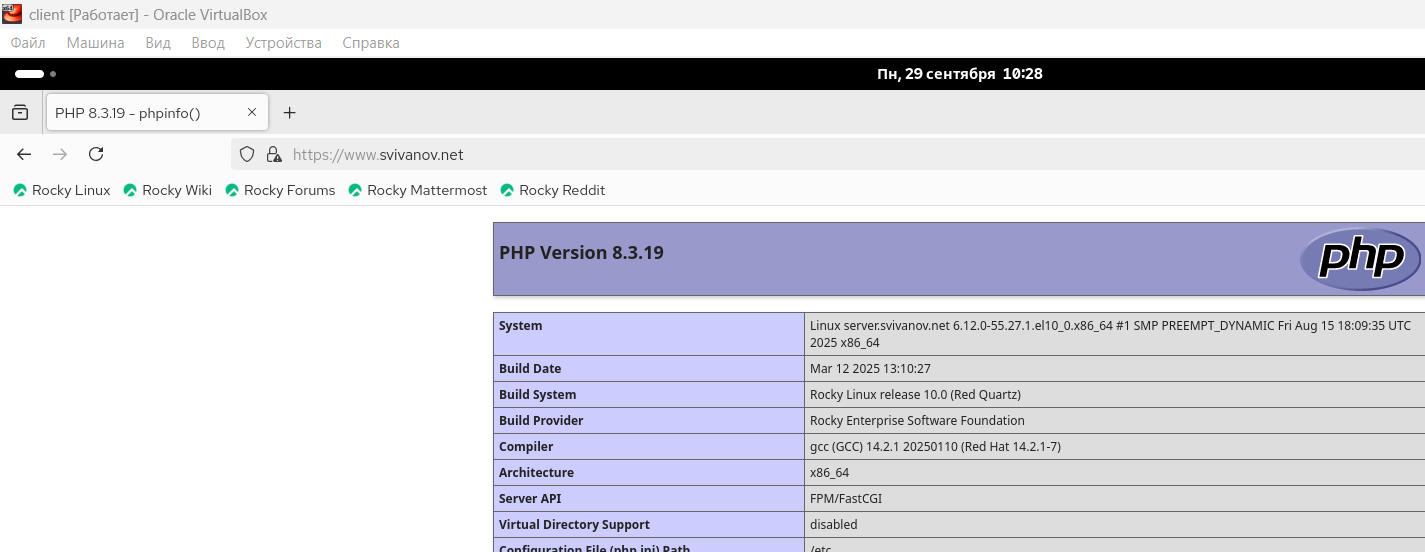


Рис. 14: Страница с PHP

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/http и в соответствующие каталоги скопируем конфигурационные файлы:

cp -R /etc/httpd/conf.d/\* /vagrant/provision/server/http/etc/httpd/conf.d

cp -R /var/www/html/\* /vagrant/provision/server/http/var/www/html

mkdir -p /vagrant/provision/server/http/etc/pki/tls/private

mkdir -p /vagrant/provision/server/http/etc/pki/tls/certs

cp -R /etc/pki/tls/private/www.svivanov.net.key /vagrant/provision/server/http/etc/pki/tls/private

cp -R /etc/pki/tls/certs/www.svivanov.net.crt /vagrant/provision/server/http/etc/pki/tls/certs (рис. 15)

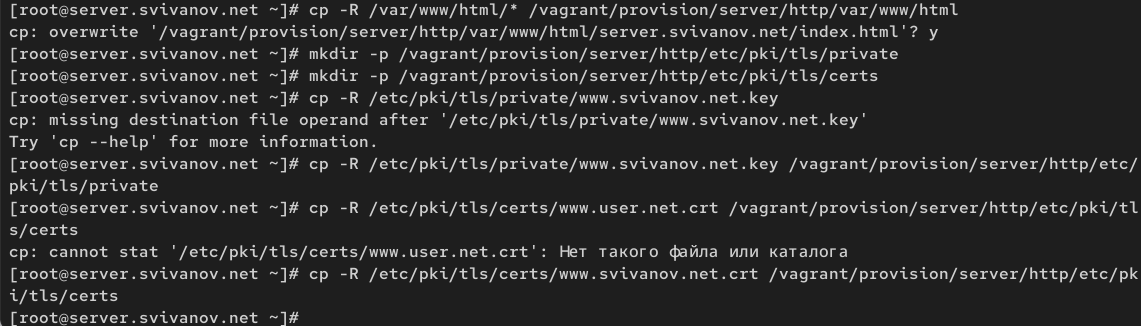


Рис. 15: Создание каталогов и копирование файлов

В имеющийся скрипт /vagrant/provision/server/http.sh внесем изменения, добавив установку PHP и настройку межсетевого экрана, разрешающую работать с https. (рис. 16)

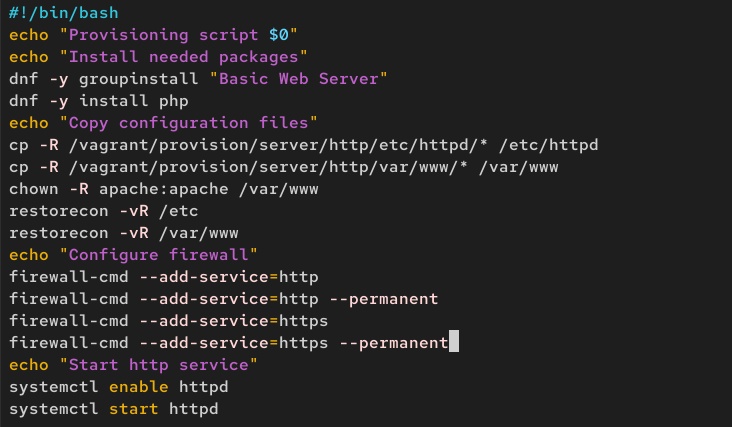


Рис. 16: Изменение скрипта http.sh

# 4 Ответы на контрольные вопросы

**1. В чём отличие HTTP от HTTPS?**

HTTP - это стандартный протокол для передачи данных, который не использует шифрование. Все данные (логины, пароли) передаются в открытом виде.

HTTPS - это безопасная версия HTTP, которая шифрует весь трафик между браузером и сервером с помощью протокола SSL/TLS.

Отличие в том, что HTTPS обеспечивает конфиденциальность и целостность данных, а HTTP - нет.

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

Безопасность обеспечивается протоколом SSL/TLS по трем направлениям:

1. Шифрование: Все данные передаются в зашифрованном виде, что защищает их от перехвата.
2. Аутентификация: Сервер предъявляет браузеру цифровой сертификат, подтверждающий его подлинность.
3. Целостность: Специальные механизмы гарантируют, что данные не были изменены при передаче.

**3. Что такое сертификационный центр? Приведите пример.**

Сертификационный центр (Certificate Authority, CA) - это доверенная организация, которая выпускает цифровые сертификаты для веб-сайтов.

Он проверяет владельца домена и digitally подписывает сертификат. Браузеры доверяют сертификатам от известных ЦС, чьи корневые сертификаты в них предустановлены.

Примеры: Let’s Encrypt (бесплатный), DigiCert, Sectigo.

# 5 Выводы

В ходе выполнения лабораторной работы мы приобрели практические навыков по расширенному конфигурированию HTTPсервера Apache в части безопасности и возможности использования PHP.