

Отчёт по лабораторной работе 5

Расширенная настройка HTTP-сервера Apache

Сейдалиев Тагьетдин Ровшенович

Содержание

1	Цель работы	5
2	Выполнение	6
2.1	Конфигурирование HTTP-сервера для работы через протокол HTTPS	6
2.1.1	Генерация ключа и сертификата	6
2.1.2	Настройка конфигурации виртуального хоста Apache	7
2.1.3	Настройка межсетевого экрана	9
2.1.4	Проверка работы HTTPS	9
2.2	Конфигурирование HTTP-сервера для работы с РНР	10
2.3	Внесение изменений в настройки внутреннего окружения виртуальной машины	12
3	Заключение	14
4	Контрольные вопросы	15
4.1	1. В чём отличие HTTP от HTTPS?	15
4.2	2. Каким образом обеспечивается безопасность контента веб-сервера при работе через HTTPS?	15
4.3	3. Что такое сертификационный центр? Приведите пример.	16

Список иллюстраций

2.1	Генерация ключа и сертификата	7
2.2	Конфигурация виртуального хоста Apache	8
2.3	Настройка firewall для HTTPS	9
2.4	Открытие сайта по HTTPS	9
2.5	Просмотр сертификата	10
2.6	index.php	11
2.7	phpinfo()	11
2.8	Копирование конфигурационных файлов в провизининг	12
2.9	Обновлённый скрипт http.sh	13

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

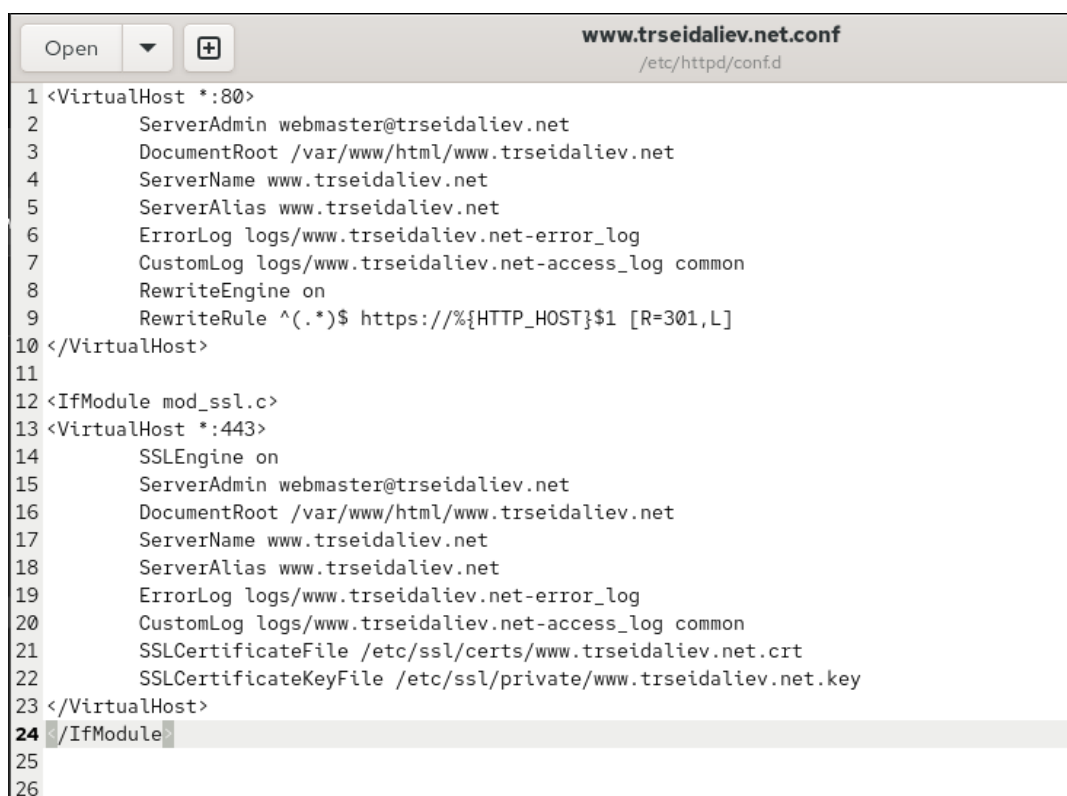
2.1.1 Генерация ключа и сертификата

В каталоге `/etc/pki/tls/private` был создан приватный ключ и самоподписанный сертификат для домена `www.trseidaliev.net`.

После перехода в каталог и создания символической 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:



```
1 <VirtualHost *:80>
2     ServerAdmin webmaster@trseidaliev.net
3     DocumentRoot /var/www/html/www.trseidaliev.net
4     ServerName www.trseidaliev.net
5     ServerAlias www.trseidaliev.net
6     ErrorLog logs/www.trseidaliev.net-error_log
7     CustomLog logs/www.trseidaliev.net-access_log common
8     RewriteEngine on
9     RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
10 </VirtualHost>
11
12 <IfModule mod_ssl.c>
13 <VirtualHost *:443>
14     SSLEngine on
15     ServerAdmin webmaster@trseidaliev.net
16     DocumentRoot /var/www/html/www.trseidaliev.net
17     ServerName www.trseidaliev.net
18     ServerAlias www.trseidaliev.net
19     ErrorLog logs/www.trseidaliev.net-error_log
20     CustomLog logs/www.trseidaliev.net-access_log common
21     SSLCertificateFile /etc/ssl/certs/www.trseidaliev.net.crt
22     SSLCertificateKeyFile /etc/ssl/private/www.trseidaliev.net.key
23 </VirtualHost>
24 </IfModule>
25
26
```

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.

```
- firewall-cmd --add-service=https --permanent
[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.

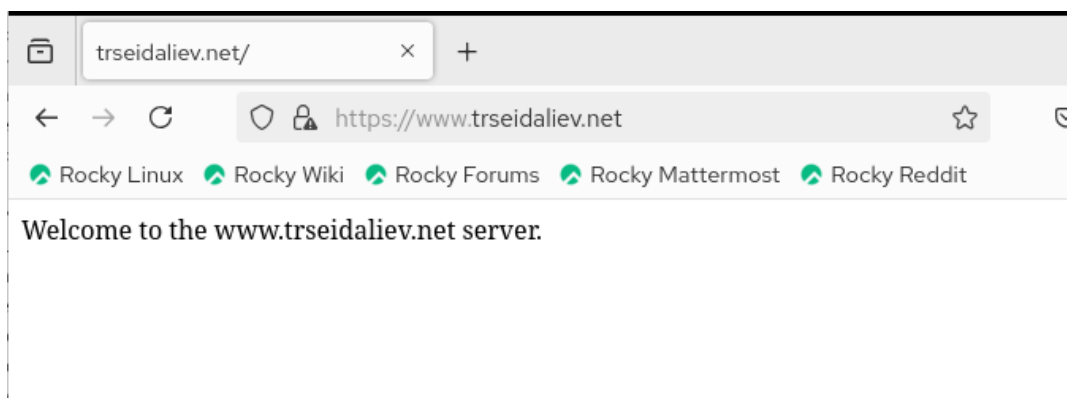


Рис. 2.4: Открытие сайта по HTTPS

Данные сертификата, отображаемые браузером:

		trseidaliev.net
	Subject Name	
	Country	RU
	State/Province	Russia
	Locality	Moscow
	Organization	trseidaliev
	Organizational Unit	trseidaliev
	Common Name	trseidaliev.net
	Email Address	trseidaliev@trseidaliev.net

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:

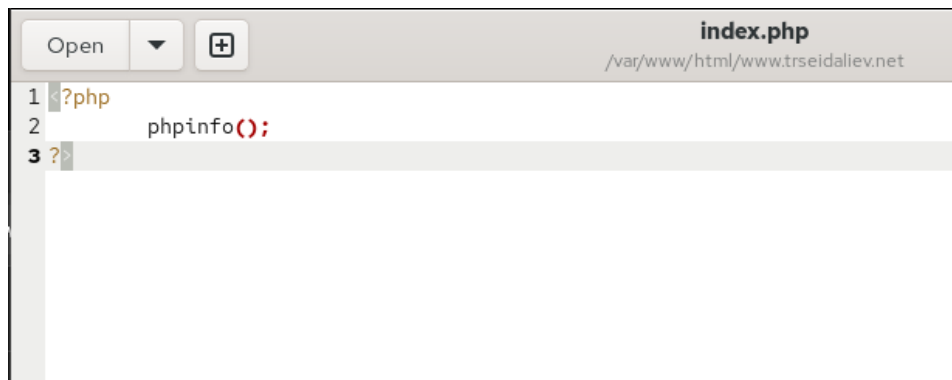


Рис. 2.6: index.php

В браузере отобразилась информация о конфигурации PHP:

PHP Version 8.3.19	
System	Linux server.trseidaliev.net 6.12.0-55.27.1.el10_0.x86_64 #1 SMP PREEMPT_DYNAMIC 2025 x86_64
Build Date	Mar 12 2025 13:10:27
Build System	Rocky Linux release 10.0 (Red Quartz)
Build Provider	Rocky Enterprise Software Foundation
Compiler	gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.

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:



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

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:

```
1  #!/bin/bash
2  echo "Provisioning script $0"
3  echo "Install needed packages"
4  dnf -y groupinstall "Basic Web Server"
5  dnf -y install php
6  echo "Copy configuration files"
7  cp -R /vagrant/provision/server/http/etc/httpd/* /etc/httpd
8  cp -R /vagrant/provision/server/http/var/www/* /var/www
9  chown -R apache:apache /var/www
10 restorecon -vR /etc
11 restorecon -vR /var/www
12 echo "Configure firewall"
13 firewall-cmd --add-service=http
14 firewall-cmd --add-service=http --permanent
15 firewall-cmd --add-service=https
16 firewall-cmd --add-service=https --permanent
17 echo "Start http service"
18 systemctl enable httpd
19 systemctl start httpd
20
```

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)