

# **Отчёт по лабораторной работе №5**

**Дисциплина: Администрирование сетевых подсистем**

Мишина Анастасия Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Конфигурирование HTTP-сервера для работы через протокол HTTPS . . . . .	6
2.2	Конфигурирование HTTP-сервера для работы с РНР . . . .	10
2.3	Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	12
<b>3</b>	<b>Выводы</b>	<b>15</b>
<b>4</b>	<b>Ответы на контрольные вопросы</b>	<b>16</b>

# Список иллюстраций

2.1	Генерация ключа и заполнение сертификата . . . . .	7
2.2	Копирование сертификата в каталог /etc/ssl/certs . . . . .	7
2.3	Редактирование файла /etc/httpd/conf.d/www.aamishina.net	8
2.4	Внесение изменений в настройки межсетевого экрана, пе- резапуск веб-сервера . . . . .	9
2.5	Сообщение о незащищенности соединения . . . . .	9
2.6	Содержимое сертификата . . . . .	10
2.7	Замена файла /var/www/html/www.aamishina.net/index.html на index.php . . . . .	10
2.8	Редактирование index.php . . . . .	11
2.9	Корректирование прав доступа, восстановление контекста безопасности SELinux, перезагрузка HTTP-сервера . . . . .	11
2.10	Веб-страница с информацией об используемой версии PHP	12
2.11	Копируем в каталоги конфигурационные файлы . . . . .	13
2.12	Внесение изменений в скрипт http.sh . . . . .	14

# Список таблиц

# 1 Цель работы

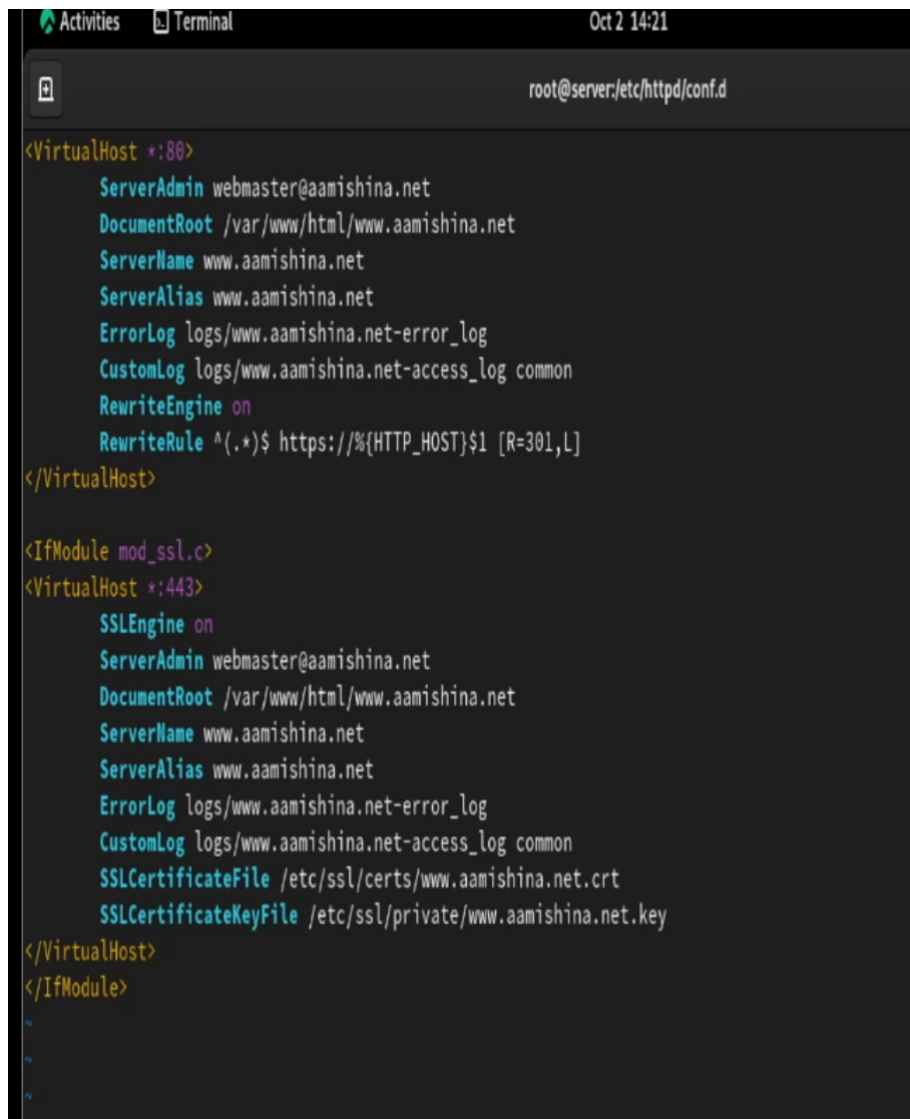
Приобретение практических навыков по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

## 2 Выполнение лабораторной работы

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

Запускаем ВМ через рабочий каталог. На ВМ server входим под собственным пользователем и переходим в режим суперпользователя. В каталоге `/etc/ssl` создаем каталог `private`: `mkdir -p /etc/pki/tls/private`, `ln -s /etc/pki/tls/private /etc/ssl/private`, `cd /etc/pki/tls/private`. Генерируем ключ и сертификат (рис. 2.1), введя следующую команду: `openssl req -x509 -nodes -newkey rsa:2048 -keyout www.aamishina.net.key -out www.aamishina.net.crt`





```
Oct 2 14:21
root@server:/etc/httpd/conf.d

<VirtualHost *:80>
    ServerAdmin webmaster@aamishina.net
    DocumentRoot /var/www/html/www.aamishina.net
    ServerName www.aamishina.net
    ServerAlias www.aamishina.net
    ErrorLog logs/www.aamishina.net-error_log
    CustomLog logs/www.aamishina.net-access_log common
    RewriteEngine on
    RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
</VirtualHost>

<IfModule mod_ssl.c>
<VirtualHost *:443>
    SSLEngine on
    ServerAdmin webmaster@aamishina.net
    DocumentRoot /var/www/html/www.aamishina.net
    ServerName www.aamishina.net
    ServerAlias www.aamishina.net
    ErrorLog logs/www.aamishina.net-error_log
    CustomLog logs/www.aamishina.net-access_log common
    SSLCertificateFile /etc/ssl/certs/www.aamishina.net.crt
    SSLCertificateKeyFile /etc/ssl/private/www.aamishina.net.key
</VirtualHost>
</IfModule>
~
~
~
```

Рис. 2.3: Редактирование файла /etc/httpd/conf.d/www.aamishina.net

Вносим изменения в настройки межсетевого экрана на сервере, перезапускаем веб-сервер (рис. 2.4)



```
[root@server.aamishina.net conf.d]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http ssh
[root@server.aamishina.net conf.d]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula bacula-cl
ient bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorren
t-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-
unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-clie
nt etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-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 ips
ec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver ku
be-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-n
etwork llmnr llmnr-client llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-w
bt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-stor
ageconsole ovirt-vmconsole plex pncd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-expor
ter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyn
cd rtsp salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptra
p spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy sysl
og syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vds vnc-server warpinator wbem-http
wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-clie
nt xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.aamishina.net conf.d]# firewall-cmd --add-service=https
success
[root@server.aamishina.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@server.aamishina.net conf.d]# firewall-cmd --reload
success
[root@server.aamishina.net conf.d]# systemctl restart httpd
```

Рис. 2.4: Внесение изменений в настройки межсетевого экрана, перезапуск веб-сервера

На VM client открываем в браузере страницу [www.aamishina.net](http://www.aamishina.net) с сообщением о незащищенности соединения (рис. 2.5). Добавив страницу в исключения, просматриваем информацию о сертификате (рис. 2.6).

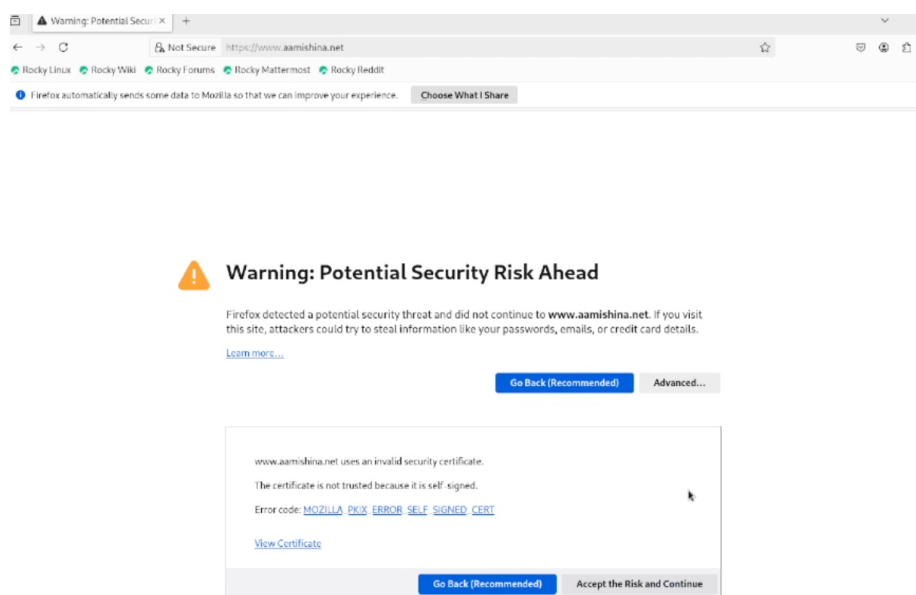


Рис. 2.5: Сообщение о незащищенности соединения

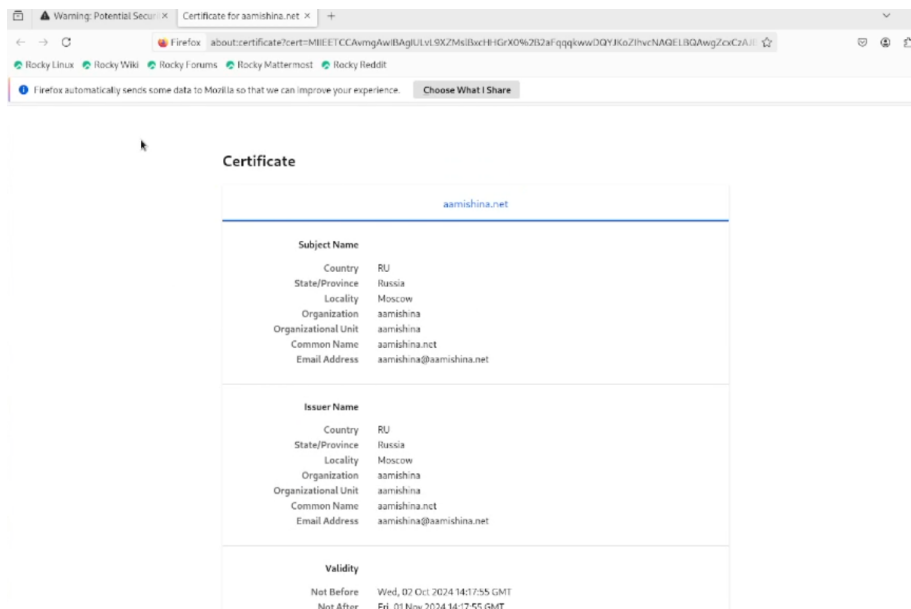


Рис. 2.6: Содержимое сертификата

## 2.2 Конфигурирование HTTP-сервера для работы с PHP

Устанавливаем пакеты для работы с PHP: `dnf -y install php`.

В каталоге `/var/www/html/www.aamishina.net` заменяем `index.html` на `index.php` (рис. 2.7).

```
Installed:
nginxfilesystem-1:1.20.1-16.el9_4.1.noarch      php-8.0.30-1.el9_2.x86_64      php-cli-8.0.30-1.el9_2.x86_64
php-common-8.0.30-1.el9_2.x86_64              php-fpm-8.0.30-1.el9_2.x86_64  php-mbstring-8.0.30-1.el9_2.x86_64
php-opcache-8.0.30-1.el9_2.x86_64             php-pdo-8.0.30-1.el9_2.x86_64  php-xml-8.0.30-1.el9_2.x86_64

Complete!
[root@server.aamishina.net conf.d]# cd /var/www/html/www.aamishina.net
[root@server.aamishina.net www.aamishina.net]# ls
index.html
[root@server.aamishina.net www.aamishina.net]# rm index.html
rm: remove regular file 'index.html'? y
[root@server.aamishina.net www.aamishina.net]# touch index.php
[root@server.aamishina.net www.aamishina.net]# vim index.php
```

Рис. 2.7: Замена файла `/var/www/html/www.aamishina.net/index.html` на `index.php`

Редактируем `index.php` (рис. 2.8).

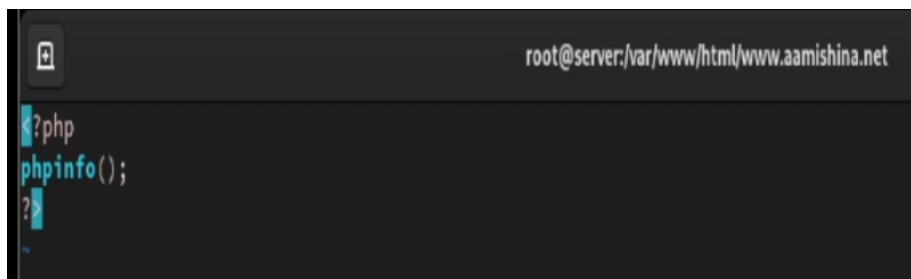


Рис. 2.8: Редактирование index.php

Корректируем права доступа в каталог с веб-контентом, восстанавливаем контекст безопасности в SELinux, перезагружаем HTTP-сервер: `chown -R apache:apache /var/www`, `restorecon -vR /etc`, `restorecon -vR /var/www` и `systemctl restart httpd` (рис. 2.9).

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

Рис. 2.9: Корректирование прав доступа, восстановление контекста безопасности SELinux, перезагрузка HTTP-сервера

На VM client вводим в адресную строку браузера `www.aamishina.net` и видим веб-страницу с информацией об используемой версии PHP (рис. 2.10).



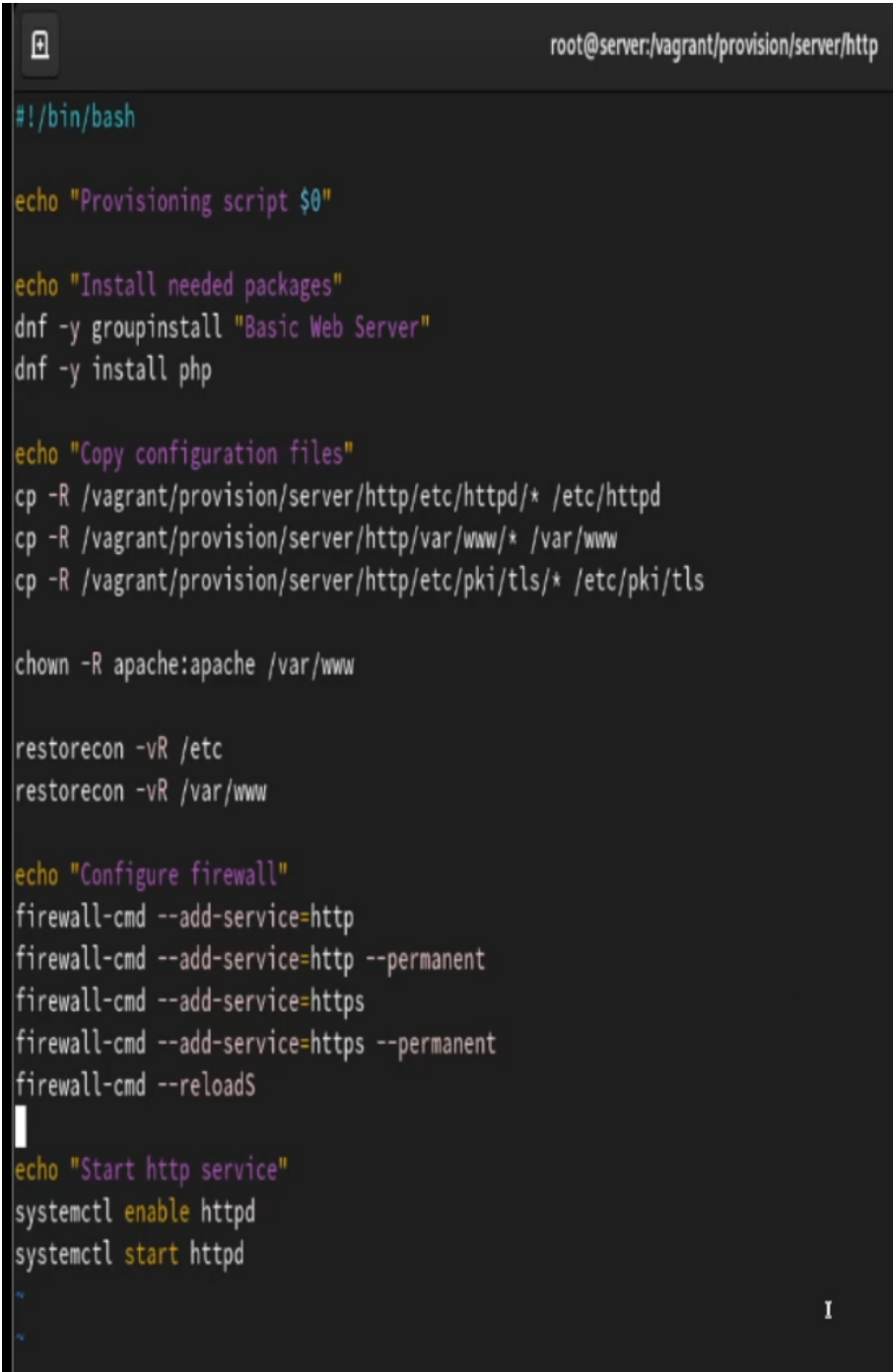
```

[root@server.aamishina.net www.aamishina.net]# cd /vagrant/provision/server/http
[root@server.aamishina.net http]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/autodisk.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/README'? y
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/server.aamishina.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.aamishina.net.conf'? y
[root@server.aamishina.net http]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
cp: overwrite '/vagrant/provision/server/http/var/www/html/server.aamishina.net/index.html'? y
[root@server.aamishina.net http]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/private
[root@server.aamishina.net http]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.aamishina.net http]# cp -R /etc/pki/tls/private/www.aamishina.net.key /vagrant/provision/server/http/etc/pki/tls/private
[root@server.aamishina.net http]# cp -R /etc/pki/tls/certs/www.aamishina.net.crt /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.aamishina.net http]# vim /vagrant/provision/server/http.sh

```

Рис. 2.11: Копируем в каталоги конфигурационные файлы

В скрипт `/vagrant/provision/server/http.sh` вносим изменения, добавив установку PHP и настройку межсетевого экрана для работы с https (рис. 2.12).

A terminal window with a dark background. The title bar at the top right shows the path 'root@server:/vagrant/provision/server/http'. The terminal content is a shell script for provisioning a web server. It includes comments for each section, package installation using 'dnf', file copying for configuration and content, setting permissions for the web directory, restoring permissions, configuring the firewall to allow HTTP and HTTPS, and finally enabling and starting the httpd service.

```
#!/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
cp -R /vagrant/provision/server/http/etc/pki/tls/* /etc/pki/tls

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
firewall-cmd --reload

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

Рис. 2.12: Внесение изменений в скрипт http.sh

## 3 Выводы

В результате выполнения работы были приобретены практические навыки по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

## 4 Ответы на контрольные вопросы

### 1. В чём отличие HTTP от HTTPS?

- **HTTP** (HyperText Transfer Protocol) – это протокол передачи данных, который используется для передачи информации между клиентом (например, веб-браузером) и сервером. Однако он не обеспечивает шифрование данных, что делает их уязвимыми к перехвату злоумышленниками.
- **HTTPS** (HyperText Transfer Protocol Secure) – это расширение протокола HTTP с добавлением шифрования, обеспечивающее безопасную передачу данных между клиентом и сервером. Протокол HTTPS использует SSL (Secure Sockets Layer) или более современный TLS (Transport Layer Security) для шифрования данных.

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

- Шифрование данных: при использовании HTTPS данные, передаваемые между клиентом и сервером, шифруются, что делает их невозможными для прочтения злоумышленниками, перехватывающими трафик.
- Идентификация сервера: сервер предоставляет цифровой сертификат, подтверждающий его легитимность. Этот сертификат выдается



сертификационным центром и содержит информацию о владельце сертификата, публичный ключ для шифрования и подпись, подтверждающую подлинность сертификата.

### 3. Что такое сертификационный центр?

- Сертификационный центр (Центр сертификации) - это доверенная сторона, которая выдает цифровые сертификаты, подтверждающие подлинность владельца сертификата. Пример: Одним из известных сертификационных центров является "Let's Encrypt". Он предоставляет бесплатные SSL-сертификаты, которые используются для обеспечения безопасного соединения на множестве веб-сайтов. Владельцы веб-сайтов могут получить сертификат от Let's Encrypt, чтобы обеспечить шифрование и подтвердить свою легитимность в онлайн-среде.