

# **Лабораторная работа №5**

**Расширенная настройка HTTP-сервера Apache**

Демидова Екатерина Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Конфигурирование HTTP-сервера для работы через протокол HTTPS	6
3.2	Конфигурирование HTTP-сервера для работы с PHP . . . . .	11
3.3	Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	12
<b>4</b>	<b>Контрольные вопросы</b>	<b>14</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>

## Список иллюстраций

3.1	Создание каталога и генерация ключа и сертификата . . . . .	7
3.2	Заполнение сертификата . . . . .	7
3.3	Изменение конфигурационного файла <code>www.eademidova.net.conf</code> .	8
3.4	Внесение в настройки межсетевого экрана и перезапуск веб-сервера	9
3.5	Сообщение о незащищенности соединения . . . . .	10
3.6	Просмотр содержания сертификата . . . . .	10
3.7	Конфигурирование HTTP-сервера для работы с РНР . . . . .	11
3.8	Проверка работы сервера с РНР . . . . .	12
3.9	Создание окружения для внесения изменений в настройки окру- жающей среды . . . . .	12
3.10	Содержание <code>http.sh</code> . . . . .	13

# 1 Цель работы

Приобретение практических навыков по расширенному конфигурированию HTTP-сервера 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.
2. Настройте веб-сервер для работы с PHP.
3. Напишите (или скорректируйте) скрипт для Vagrant, фиксирующий действия по расширенной настройке HTTP-сервера во внутреннем окружении виртуальной машины `server`.

## 3 Выполнение лабораторной работы

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

Загрузим нашу операционную систему и перейдем в рабочий каталог с проектом:

```
cd /var/tmp/eademidova/vagran
```

Затем запустим виртуальную машину server:

```
make server-up
```

На виртуальной машине server войдем под созданным в предыдущей работе пользователем и откроем терминал. Перейдем в режим суперпользователя. В каталоге `/etc/ssl` создадим каталог `private` сгенерируем ключ и сертификат, используя следующую команду (рис. 3.1):



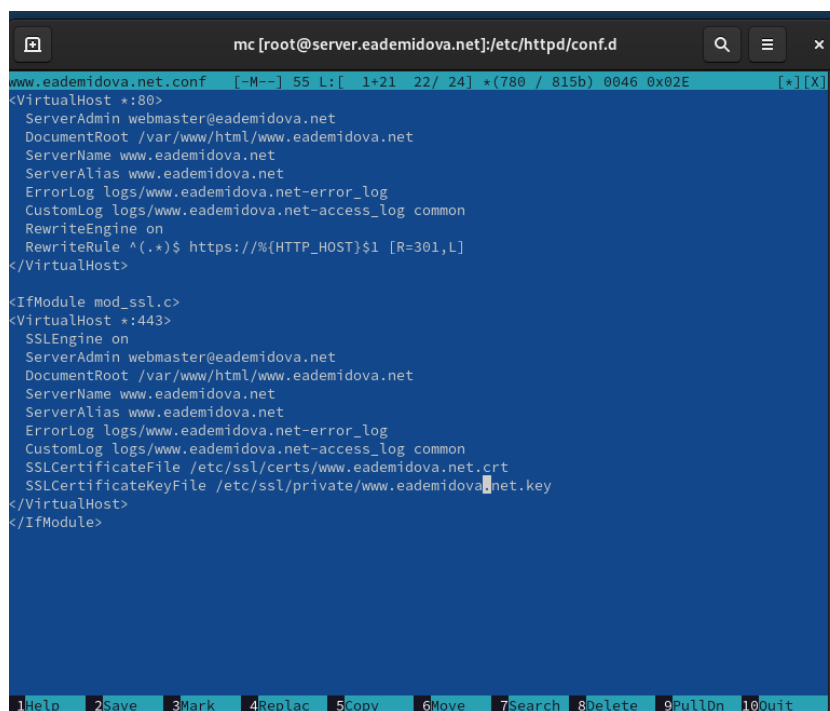


Рис. 3.3: Изменение конфигурационного файла www.eademidova.net.conf

Внесем изменения в настройки межсетевого экрана на сервере, разрешив работу с https и перезапустим веб-сервер(рис. 3.4):



```
root@server:/etc/httpd/conf.d

[root@server.eademidova.net conf.d]# mc

[root@server.eademidova.net conf.d]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http ssh
[root@server.eademidova.net conf.d]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd aud
it ausweisapp2 bacula bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-
rpc bittorrent-lsd ceph ceph-mon cfengine checkmk-agent cockpit collectd condor-collector crate
db ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-
lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ld
ap 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 ipsec
irc ircs iscsi-target isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogin kpasswd k
prop kshell kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controlle
r-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-s
ecure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightn
ing-network llmnr llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mo
untdd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmea-018
3 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmw
ebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps
3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquodad rs
h rsyncd rtsp salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission sm
tps snmp snmpv1 snmpv2 snmpv3 snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-stre
aming svdrp svn syncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tftp tile38 t
inc tor-socks transmission-client upnp-client vdsim vnc-server wbem-http wbem-https wireguard ws
-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsmann wsmans xdmcp xmpp-bosh x
mpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.eademidova.net conf.d]# firewall-cmd --add-service=https
success
[root@server.eademidova.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@server.eademidova.net conf.d]# firewall-cmd --reload
success
[root@server.eademidova.net conf.d]# systemctl restart httpd
[root@server.eademidova.net conf.d]#
```

Рис. 3.4: Всенесение в нстройки межсетевого экрана и перезапуск веб-сервера

Здесь в первом блоке указывается сервера администратора, указан файл, кото-  
рый используется для запроса, затем имя сервера, альтернативное имя хоста(в  
нашем случае такое же как основное), лог файл ошибок, лог файл расположения  
и журнала досутпа, включен механизм перезаписи, запрошенных URL-адресов и  
правило перезаписи. Во втором блоке настраивается ssl-доступ. Здесь порт 443.  
Кроме указанных в предыдущем блоке строк, также указан файл ssl-сертификата  
и его ключ.

На виртуальной машине client в строке браузера введем название веб-сервера  
www.eademidova.net и убедимся, что произойдёт автоматическое переключение  
на работу по протоколу HTTPS(рис. 3.5):

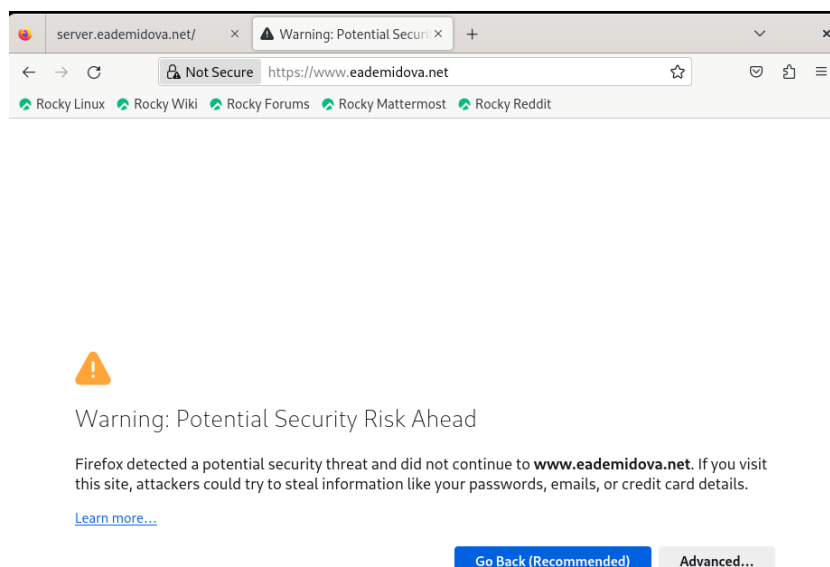


Рис. 3.5: Сообщение о незащищенности соединения

На открывшейся странице с сообщением о незащищённости соединения нажмем кнопку «Дополнительно», затем добавим адрес нашего сервера в постоянные исключения. Затем посмотрим содержание сертификата (нажав на значок с замком в адресной строке и кнопку «Подробнее»)(3.6):

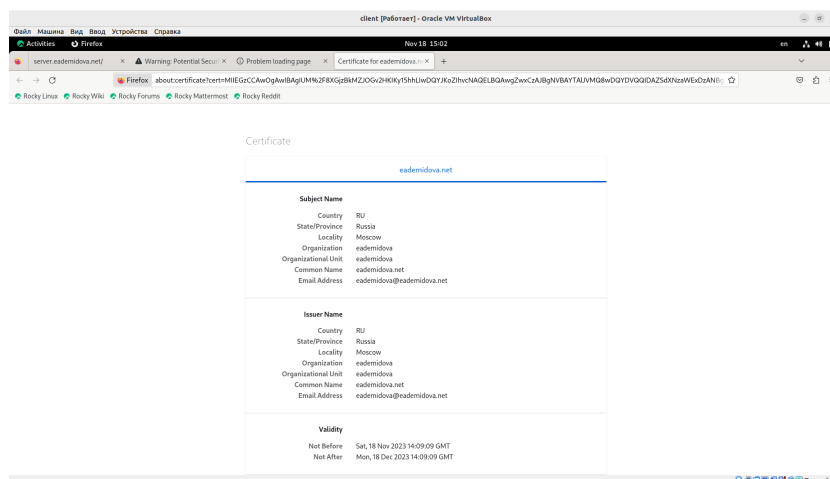


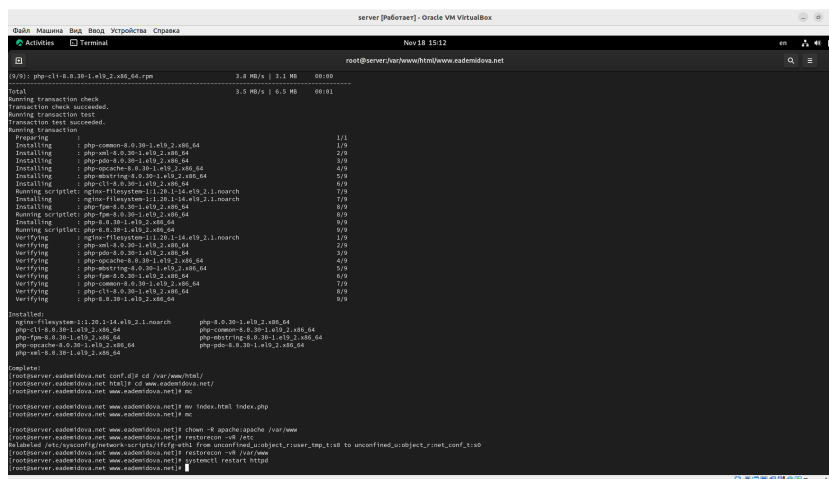
Рис. 3.6: Просмотр содержания сертификата

## 3.2 Конфигурирование HTTP-сервера для работы с PHP

Установим пакеты для работы с PHP, затем в каталоге `/var/www/html/www.eademidova.net` заменим файл `index.html` на `index.php` следующего содержания:

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

Скорректируем права доступа в каталог с веб-контентом, восстановим контекст безопасности в SELinux и перезапустим HTTP-сервер(3.7):



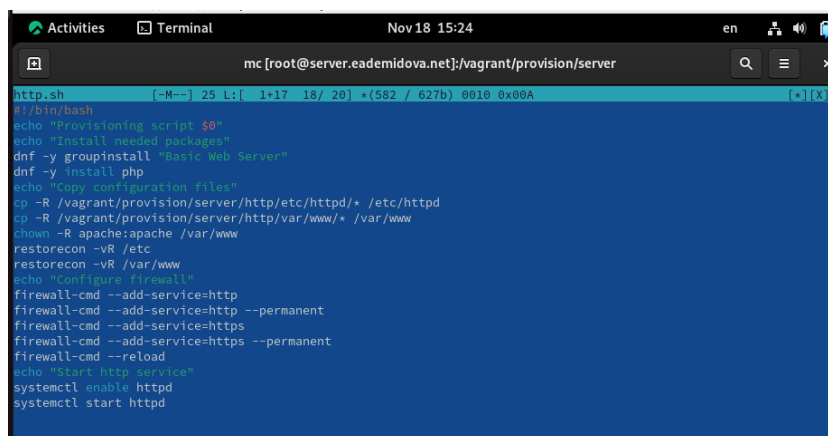
```
server [php@server ~]$ sudo yum install php php-common php-cli php-fpm php-mysqlnd php-xml php-xmlrpc php-ldap php-gd php-mbstring php-intl php-openssl
Total
Installing transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Installing : php-common-8.0.30-1.el8_2.x86_64 1/1
Installing : php-xml-8.0.30-1.el8_2.x86_64 2/9
Installing : php-pdo-8.0.30-1.el8_2.x86_64 3/9
Installing : php-openssl-8.0.30-1.el8_2.x86_64 4/9
Installing : php-mbstring-8.0.30-1.el8_2.x86_64 5/9
Installing : php-cli-8.0.30-1.el8_2.x86_64 6/9
Installing : php-fpm-8.0.30-1.el8_2.x86_64 7/9
Running scriptlet: php-fpm-8.0.30-1.el8_2.x86_64 7/9
Installing : php-8.0.30-1.el8_2.x86_64 8/9
Running scriptlet: php-8.0.30-1.el8_2.x86_64 8/9
Installing : php-8.0.30-1.el8_2.x86_64 9/9
Verifying : php-8.0.30-1.el8_2.x86_64 1/9
Verifying : php-xml-8.0.30-1.el8_2.x86_64 2/9
Verifying : php-pdo-8.0.30-1.el8_2.x86_64 3/9
Verifying : php-openssl-8.0.30-1.el8_2.x86_64 4/9
Verifying : php-mbstring-8.0.30-1.el8_2.x86_64 5/9
Verifying : php-fpm-8.0.30-1.el8_2.x86_64 6/9
Verifying : php-cli-8.0.30-1.el8_2.x86_64 7/9
Verifying : php-8.0.30-1.el8_2.x86_64 8/9
Verifying : php-8.0.30-1.el8_2.x86_64 9/9
Installed:
  php-8.0.30-1.el8_2.x86_64
  php-cli-8.0.30-1.el8_2.x86_64
  php-fpm-8.0.30-1.el8_2.x86_64
  php-openssl-8.0.30-1.el8_2.x86_64
  php-pdo-8.0.30-1.el8_2.x86_64
  php-xml-8.0.30-1.el8_2.x86_64
  php-xmlrpc-8.0.30-1.el8_2.x86_64
Complete!
[root@server ~]# cd /var/www/html/
[root@server ~]# ls
index.html
[root@server ~]# mv index.html index.php
[root@server ~]# ls
index.php
[root@server ~]# chown -R apache:apache /var/www
[root@server ~]# systemctl restart httpd
[root@server ~]# systemctl restart httpd
```

Рис. 3.7: Конфигурирование HTTP-сервера для работы с PHP

На виртуальной машине client в строке браузера введем название веб-сервера `www.eademidova.net` и убедимся, что будет выведена страница с информацией об используемой на веб-сервере версии PHP(3.8):



Открыв `http.sh` на редактирование, добавим в него следующие строки(3.10):

A screenshot of a terminal window titled "Terminal" with the date "Nov 18 15:24". The terminal shows the content of a file named "http.sh" located at "mc [root@server.eademidova.net]:/vagrant/provision/server". The script content is as follows:

```
http.sh
#!/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
firewall-cmd --reload
echo "Start http service"
systemctl enable httpd
systemctl start httpd
```

Рис. 3.10: Содержание `http.sh`

## 4 Контрольные вопросы

### 1. В чём отличие HTTP от HTTPS?

Основное отличие между HTTP и HTTPS заключается в том, что HTTPS использует шифрование для обеспечения безопасной передачи данных, в то время как HTTP передает информацию в открытом виде.

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

Безопасность контента веб-сервера при работе через HTTPS обеспечивается с помощью шифрования данных, используя SSL/TLS протоколы, что позволяет защитить информацию от несанкционированного доступа.

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

Сертификационный центр (Certification authority, CA) представляет собой компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей. Примеры: Let's Encrypt, Comodo, Symantec.

## 5 Выводы

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