

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

дисциплина: Администрирование сетевых подсистем

Студент: Махорин Иван Сергеевич

Студ. билет № 1032211221

Группа: НПИбд-02-21

МОСКВА

2023 г.

Цель работы:

Целью данной работы является приобретение практических навыков по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования РНР.

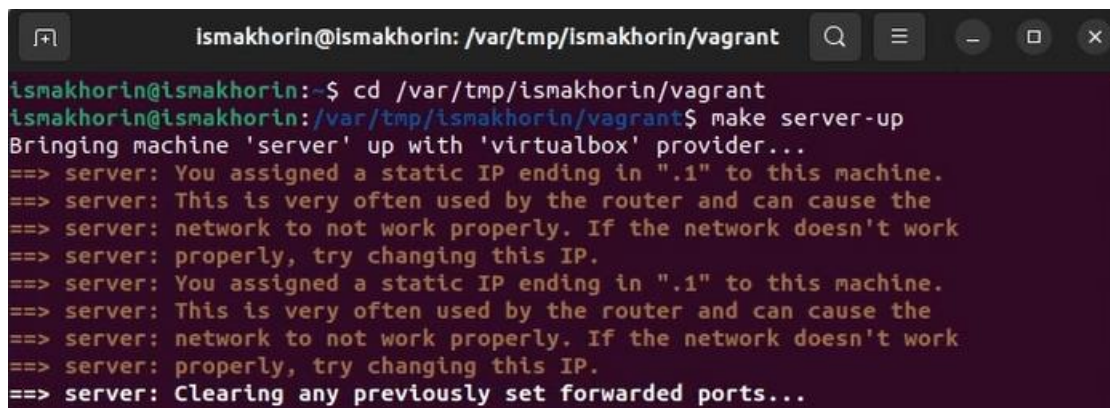
Выполнение работы:

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

```
cd /var/tmp/ismakhorin/vagrant
```

Далее запустим виртуальную машину server (Рис. 1.1):

```
make server-up
```



```
ismakhorin@ismakhorin: /var/tmp/ismakhorin/vagrant
ismakhorin@ismakhorin:~$ cd /var/tmp/ismakhorin/vagrant
ismakhorin@ismakhorin:/var/tmp/ismakhorin/vagrant$ make server-up
Bringing machine 'server' up with 'virtualbox' provider...
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: You assigned a static IP ending in ".1" to this machine.
==> server: This is very often used by the router and can cause the
==> server: network to not work properly. If the network doesn't work
==> server: properly, try changing this IP.
==> server: Clearing any previously set forwarded ports...
```

Рис. 1.1. Открытие рабочего каталога с проектом и запуск виртуальной машины server.

На виртуальной машине server войдём под нашим пользователем и откроем терминал. Далее перейдём в режим суперпользователя (Рис. 1.2):

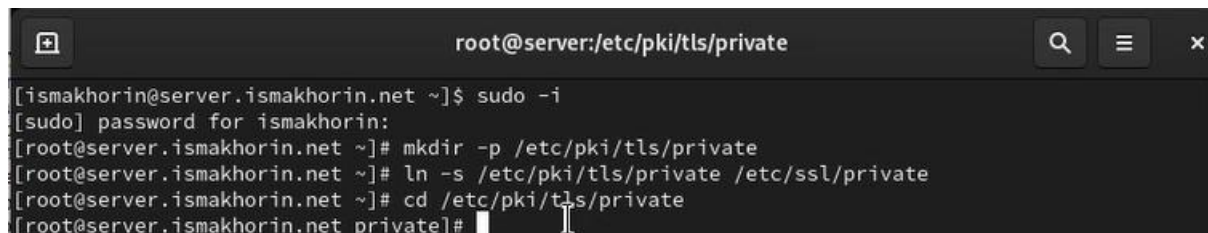
```
sudo -i
```

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
```



```
root@server:/etc/pki/tls/private

[ismakhorin@server.ismakhorin.net ~]$ sudo -i
[sudo] password for ismakhorin:
[root@server.ismakhorin.net ~]# mkdir -p /etc/pki/tls/private
[root@server.ismakhorin.net ~]# ln -s /etc/pki/tls/private /etc/ssl/private
[root@server.ismakhorin.net ~]# cd /etc/pki/tls/private
[root@server.ismakhorin.net private]#
```

Рис. 1.2. Переход в режим суперпользователя и создание в каталоге /etc/ssl каталога private.

Сгенерируем ключ (Рис. 1.3) и сертификат (Рис. 1.4), используя следующую команду:

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

```
mv www.ismakhorin.net.crt /etc/pki/tls/certs
```


Откроем на редактирование файл `/etc/httpd/conf.d/www.ismakhorin.net.conf` и заменим его содержимое на то, которое дано нам в лабораторной работе (Рис. 1.6):



Рис. 1.6. Открытие файла `/etc/httpd/conf.d/www.ismakhorin.net.conf` на редактирование и замена содержимого.

Внесём изменения в настройки межсетевого экрана на сервере, разрешив работу с https (Рис. 1.7):

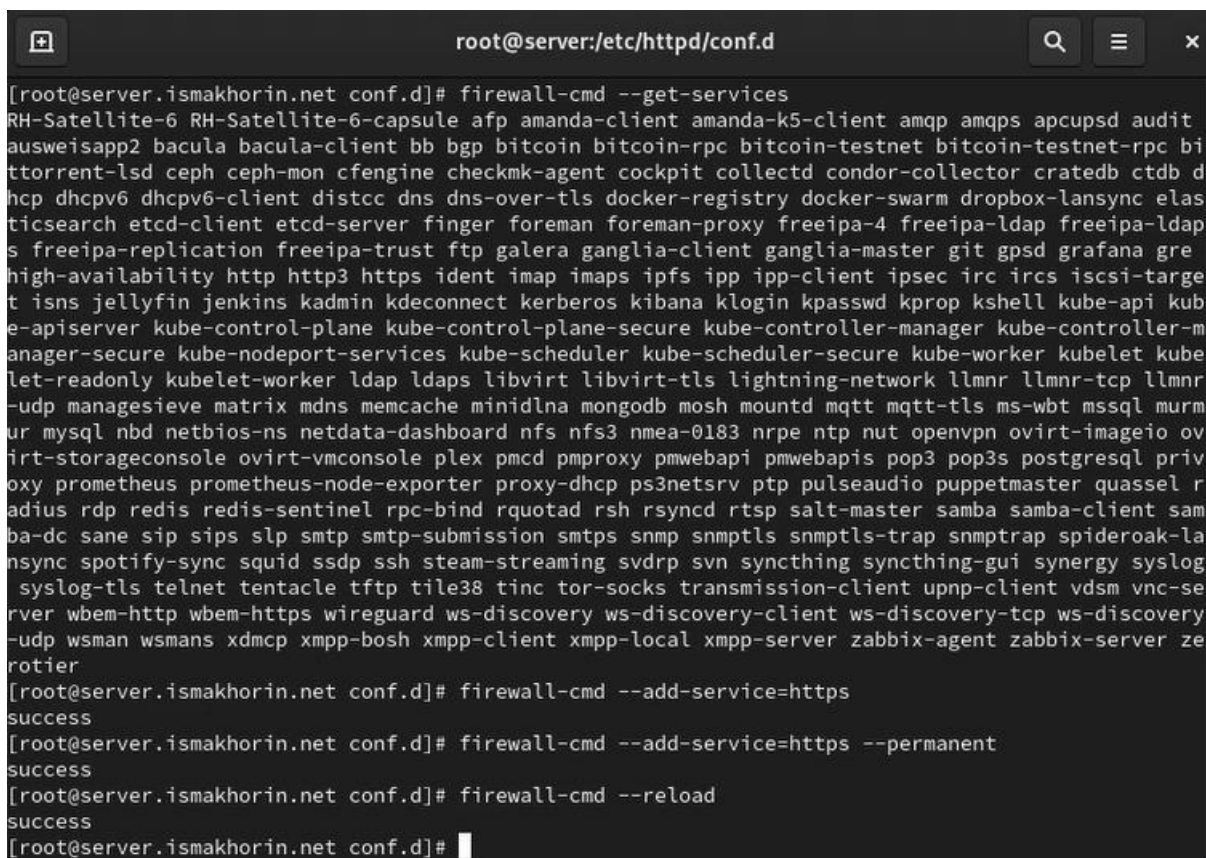
```
firewall-cmd --list-services
```

```
firewall-cmd --get-services
```

```
firewall-cmd --add-service=https
```

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

```
firewall-cmd --reload
```

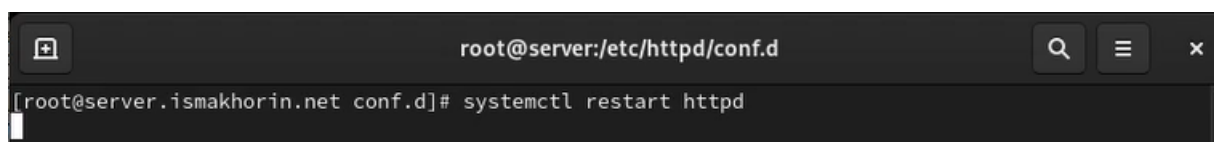
A terminal window titled 'root@server:/etc/httpd/conf.d' with search, menu, and close icons in the title bar. The terminal shows the output of 'firewall-cmd --get-services' listing numerous services. Then, 'firewall-cmd --add-service=https' is executed, returning 'success'. Next, 'firewall-cmd --add-service=https --permanent' is executed, also returning 'success'. Finally, 'firewall-cmd --reload' is executed, returning 'success'. The prompt returns to the root user.

```
root@server:/etc/httpd/conf.d
[root@server.ismakhorin.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-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bi
ttorrent-lsd ceph ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb d
hcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync elas
ticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldap
s 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-targe
t isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kub
e-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-m
anager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kube
let-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-tcp llmnr
-udp managesieve matrix mdns memcache minidlna mongodb mosh mounstd mqtt mqtt-tls ms-wbt mssql murm
ur mysql nbd netbios-ns netdata-dashboar nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ov
irt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql priv
oxy prometheus prometheus-node-exporter proxy-dhcp ps3netsrv ptp pulseaudio puppetmaster quassel r
adius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client sam
ba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-la
nsync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui synergy syslog
syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsd vnc-se
rver wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery
-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server ze
rotier
[root@server.ismakhorin.net conf.d]# firewall-cmd --add-service=https
success
[root@server.ismakhorin.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@server.ismakhorin.net conf.d]# firewall-cmd --reload
success
[root@server.ismakhorin.net conf.d]#
```

Рис. 1.7. Внесение изменений в настройки межсетевого экрана на сервере, разрешив работу с https.

Перезапустим веб-сервер (Рис. 1.8):

```
systemctl restart httpd
```

A terminal window titled 'root@server:/etc/httpd/conf.d' with search, menu, and close icons in the title bar. The terminal shows the command 'systemctl restart httpd' being executed. The prompt returns to the root user.

```
root@server:/etc/httpd/conf.d
[root@server.ismakhorin.net conf.d]# systemctl restart httpd
```

Рис. 1.8. Перезапуск веб-сервера.

На виртуальной машине client в строке браузера введём название веб-сервера `www.ismakhorin.net` и убедимся, что произошло автоматическое переключение на работу по протоколу HTTPS (Рис. 1.9). На открывшейся странице с сообщением о незащищённости соединения нажмём кнопку

«Дополнительно», затем добавим адрес нашего сервера в постоянные исключения. Затем посмотрим содержание сертификата.

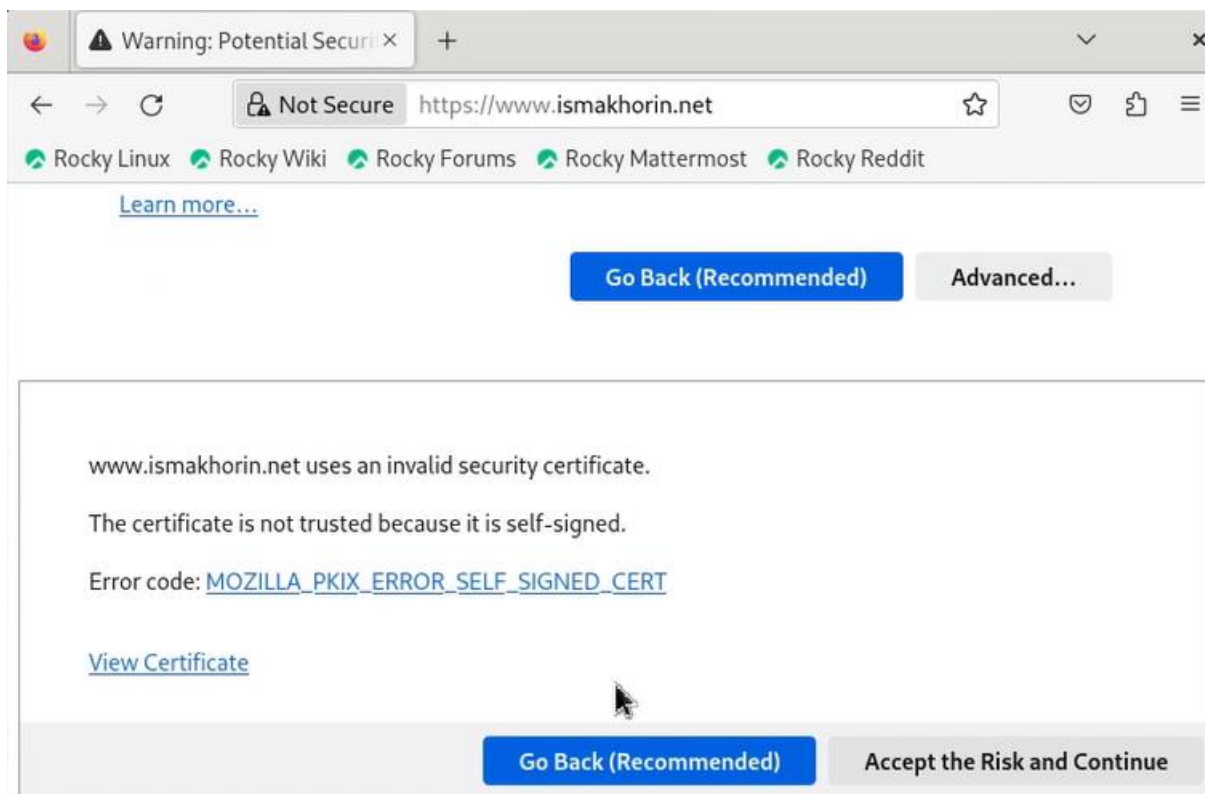


Рис. 1.9. Открытие веб-сервера www.ismakhorin.net и автоматическое переключение на работу по протоколу HTTPS.

Установим пакеты для работы с PHP (Рис. 2.1):

```
dnf -y install php
```

```
root@server:/etc/httpd/conf.d
[root@server.ismakhorin.net conf.d]# dnf -y install php
Rocky Linux 9 - BaseOS                    5.1 kB/s | 4.1 kB    00:00
Rocky Linux 9 - AppStream                 7.9 kB/s | 4.5 kB    00:00
Rocky Linux 9 - Extras                   5.5 kB/s | 2.9 kB    00:00
Dependencies resolved.
=====
Package                                Architecture Version                                Repository                                Size
=====
Installing:
php                                    x86_64      8.0.30-1.el9_2                        appstream                                7.7 k
Installing dependencies:
nginx-filesystem                      noarch      1:1.20.1-14.el9_2.1                  appstream                                8.5 k
php-common                            x86_64      8.0.30-1.el9_2                        appstream                                665 k
Installing weak dependencies:
php-cli                               x86_64      8.0.30-1.el9_2                        appstream                                3.1 M
php-fpm                               x86_64      8.0.30-1.el9_2                        appstream                                1.6 M
php-mbstring                         x86_64      8.0.30-1.el9_2                        appstream                                468 k
php-opcache                          x86_64      8.0.30-1.el9_2                        appstream                                509 k
php-pdo                               x86_64      8.0.30-1.el9_2                        appstream                                81 k
php-xml                               x86_64      8.0.30-1.el9_2                        appstream                                131 k
Transaction Summary
=====
Install 9 Packages

Total download size: 6.5 M
Installed size: 35 M
Downloading Packages:
(1/9): nginx-filesystem-1.20.1-14.el9_2.1.noarch.rpm    14 kB/s | 8.5 kB    00:00
(2-3/9): php-pdo-8.0.30-1.el9_1% [                    ] 255 kB/s | 72 kB    00:25 ETA
```

Рис. 2.1. Установка пакетов для работы с PHP.

В каталоге /var/www/html/www.ismakhorin.net заменим файл index.html на index.php следующего содержания (рис. 2.2):

```
*index.php
/var/www/html/www.ismakhorin.net
1 ?php
2 phpinfo();
3 ?
```

Рис. 2.2. Замена файла index.html на index.php с содержанием из лабораторной работы.

Скорректируем права доступа в каталог с веб-контентом:

```
chown -R apache:apache /var/www
```

После чего восстановим контекст безопасности в SELinux:

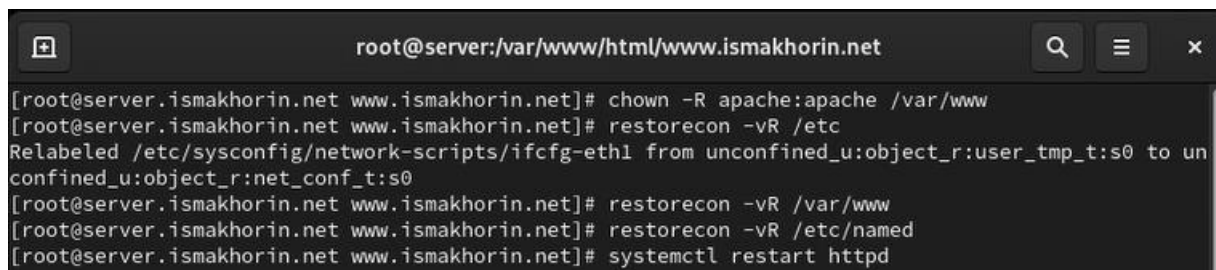
```
restorecon -vR /etc
```



```
restorecon -vR /var/www
```

И перезапустим HTTP-сервер (рис. 2.3):

```
systemctl restart httpd
```



```
root@server:/var/www/html/www.ismakhorin.net
[root@server.ismakhorin.net www.ismakhorin.net]# chown -R apache:apache /var/www
[root@server.ismakhorin.net www.ismakhorin.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.ismakhorin.net www.ismakhorin.net]# restorecon -vR /var/www
[root@server.ismakhorin.net www.ismakhorin.net]# restorecon -vR /etc/named
[root@server.ismakhorin.net www.ismakhorin.net]# systemctl restart httpd
```

Рис. 2.3. Корректировка прав доступа в каталог с веб-контентом, восстановление контекста безопасности в SELinux и перезапуск HTTP-сервера.

На виртуальной машине client в строке браузера введём название веб-сервера `www.ismakhorin.net` и убедимся, что будет выведена страница с информацией об используемой на веб-сервере версии PHP (рис. 2.4):

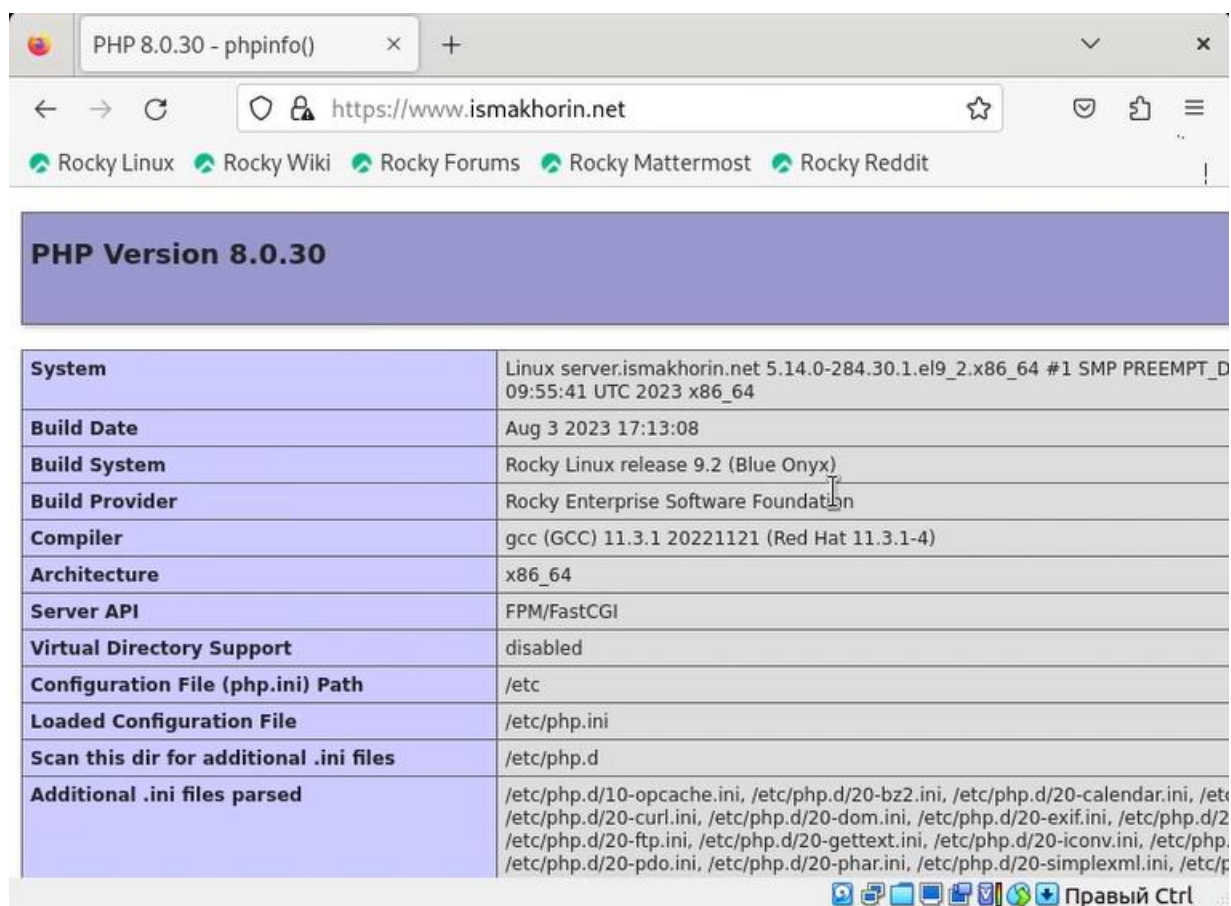
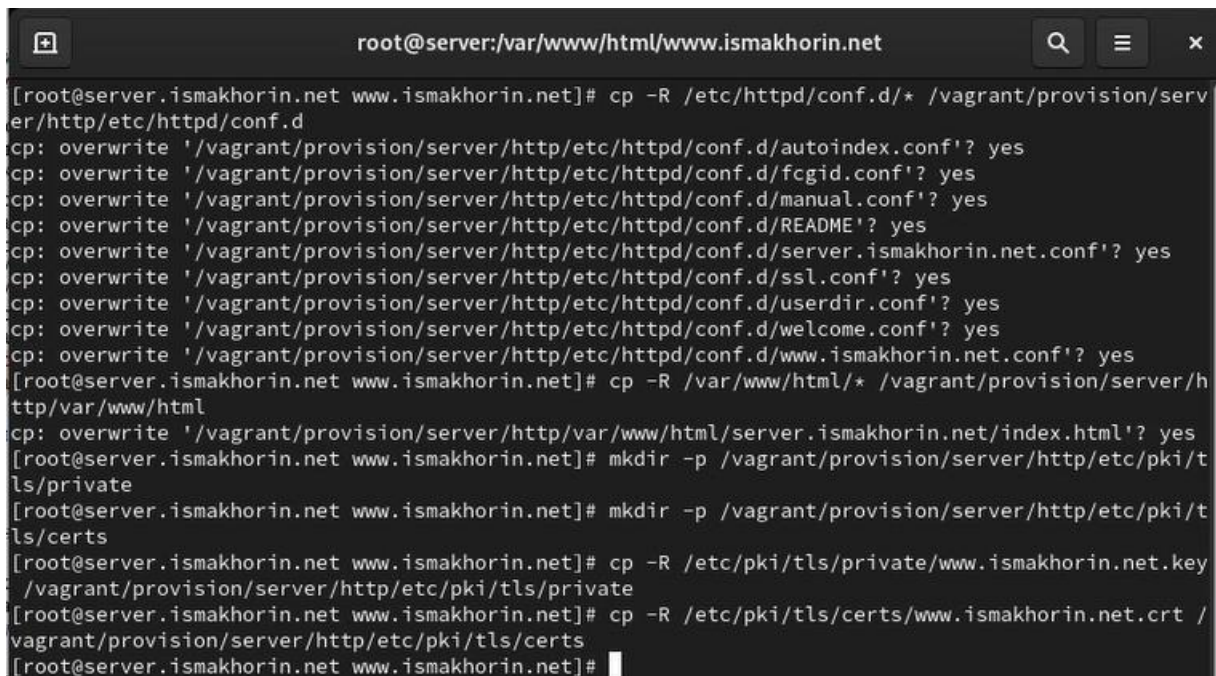


Рис. 2.4. Проверка вывода страницы с информацией об используемой на веб-сервере версии PHP.

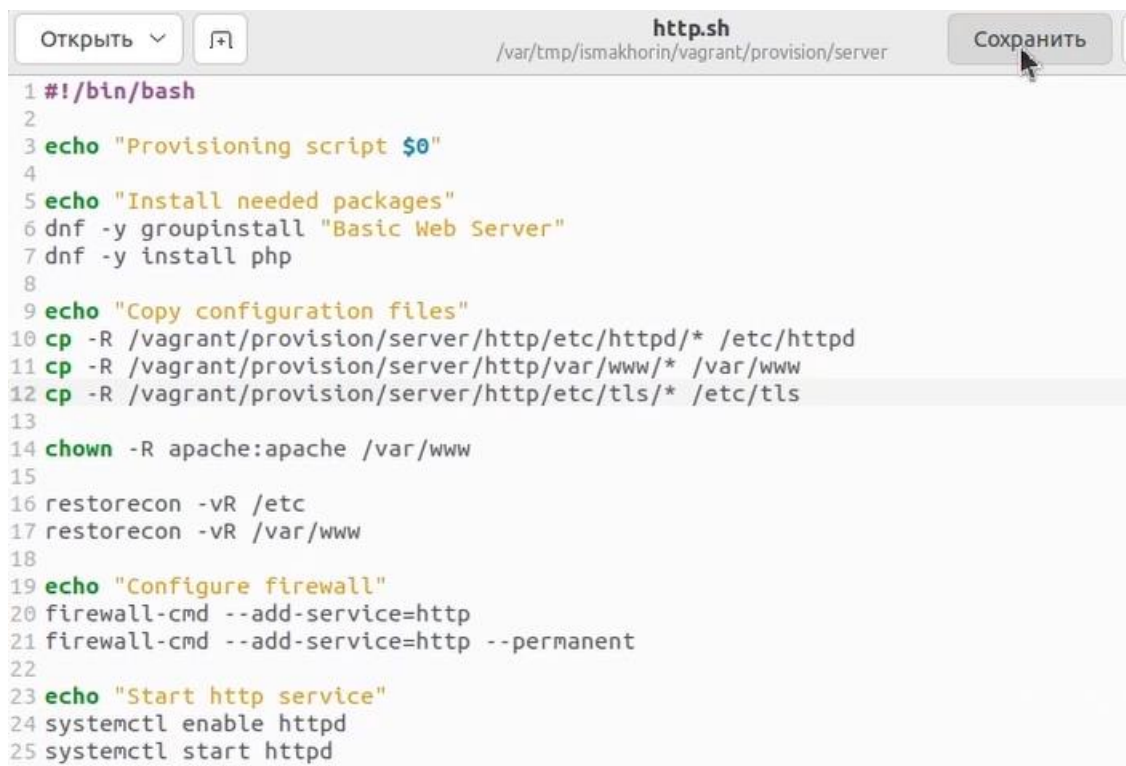
На виртуальной машине `server` перейдём в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/http` и в соответствующие каталоги скопируем конфигурационные файлы (рис. 3.1):



```
root@server:/var/www/html/www.ismakhorin.net
[root@server.ismakhorin.net www.ismakhorin.net]# cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/autoindex.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/fcgid.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/manual.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/README'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/server.ismakhorin.net.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/ssl.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/userdir.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/welcome.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/www.ismakhorin.net.conf'? yes
[root@server.ismakhorin.net www.ismakhorin.net]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
cp: overwrite '/vagrant/provision/server/http/var/www/html/server.ismakhorin.net/index.html'? yes
[root@server.ismakhorin.net www.ismakhorin.net]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/private
[root@server.ismakhorin.net www.ismakhorin.net]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.ismakhorin.net www.ismakhorin.net]# cp -R /etc/pki/tls/private/www.ismakhorin.net.key /vagrant/provision/server/http/etc/pki/tls/private
[root@server.ismakhorin.net www.ismakhorin.net]# cp -R /etc/pki/tls/certs/www.ismakhorin.net.crt /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.ismakhorin.net www.ismakhorin.net]#
```

Рис. 3.1. Внесение изменений в настройки внутреннего окружения /vagrant/provision/server/http и копирование конфигурационных файлов в каталоги.

В имеющийся скрипт /vagrant/provision/server/http.sh внесём изменения, добавив установку PHP и настройку межсетевого экрана, разрешающую работать с https (рис. 3.2):



```
1 #!/bin/bash
2
3 echo "Provisioning script $0"
4
5 echo "Install needed packages"
6 dnf -y groupinstall "Basic Web Server"
7 dnf -y install php
8
9 echo "Copy configuration files"
10 cp -R /vagrant/provision/server/http/etc/httpd/* /etc/httpd
11 cp -R /vagrant/provision/server/http/var/www/* /var/www
12 cp -R /vagrant/provision/server/http/etc/tls/* /etc/tls
13
14 chown -R apache:apache /var/www
15
16 restorecon -vR /etc
17 restorecon -vR /var/www
18
19 echo "Configure firewall"
20 firewall-cmd --add-service=http
21 firewall-cmd --add-service=http --permanent
22
23 echo "Start http service"
24 systemctl enable httpd
25 systemctl start httpd
```

Рис. 3.2. Внесение изменений в скрипт /vagrant/provision/server/http.sh, добавив установку PHP и настройку межсетевого экрана, позволяющую работать с https.

Вывод:

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

Ответы на контрольные вопросы:

1. В чём отличие HTTP от HTTPS? – **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? – **Обеспечение безопасности 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? Приведите пример. -
Сертификационный центр:

Определение: 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. Владелец веб-сайтов могут получить сертификат от Let's Encrypt, чтобы обеспечить шифрование и подтвердить свою легитимность в онлайн-среде.