

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: Пакавира Арсениу Висенте Луиш

Студ. билет № 1032225105

Группа: НФИбд-02-23

МОСКВА

2025 г.

Цель работы:

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

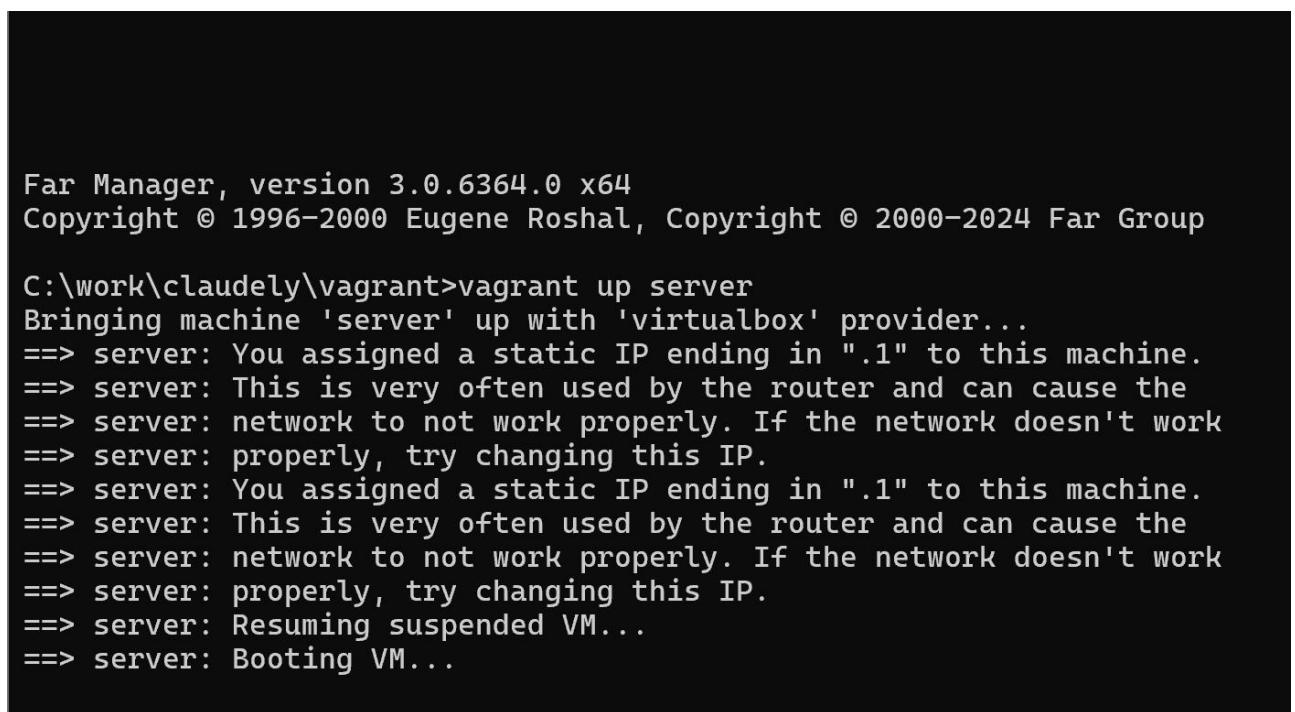
Выполнение работы:

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

```
cd /var/tmp/claudely/vagrant
```

Далее запустим виртуальную машину server (Рис. 1.1):

```
make server-up
```



```
Far Manager, version 3.0.6364.0 x64
Copyright © 1996-2000 Eugene Roshal, Copyright © 2000-2024 Far Group

C:\work\claudely\vagrant>vagrant up server
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: Resuming suspended VM...
==> server: Booting VM...
```

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):

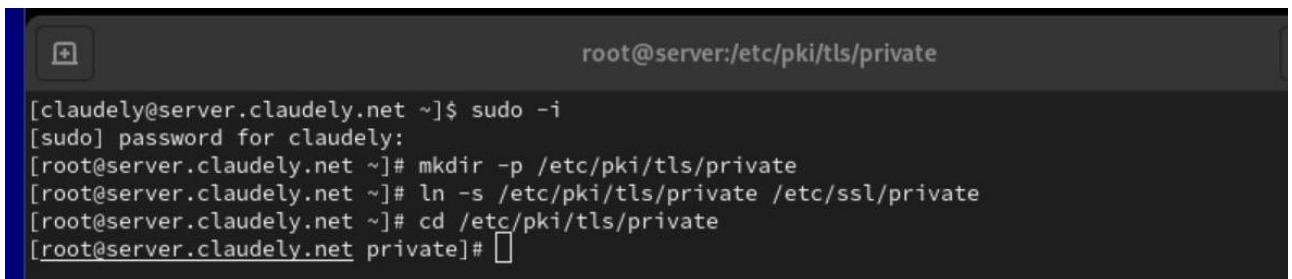
```
sudo -i
```

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
```



The screenshot shows a terminal window with a dark background and light-colored text. At the top right, it says "root@server:/etc/pki/tls/private". The terminal history shows the following commands:

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

```
www.claudely.net.crt mv www.claudely.net.crt
```

```
/etc/pki/tls/certs
```

Рис. 1.3. Генерация ключа.

```
cert.pem certs ct_log_list.cnf openssl.cnf private  
[root@server.claudely.net ssl]# cp /etc/ssl/private/www.claudely.net.crt /etc/ssl/certs  
cp: overwrite '/etc/ssl/certs/www.claudely.net.crt'? yes  
[root@server.claudely.net ssl]#
```

Рис. 1.4. Генерация сертификата.

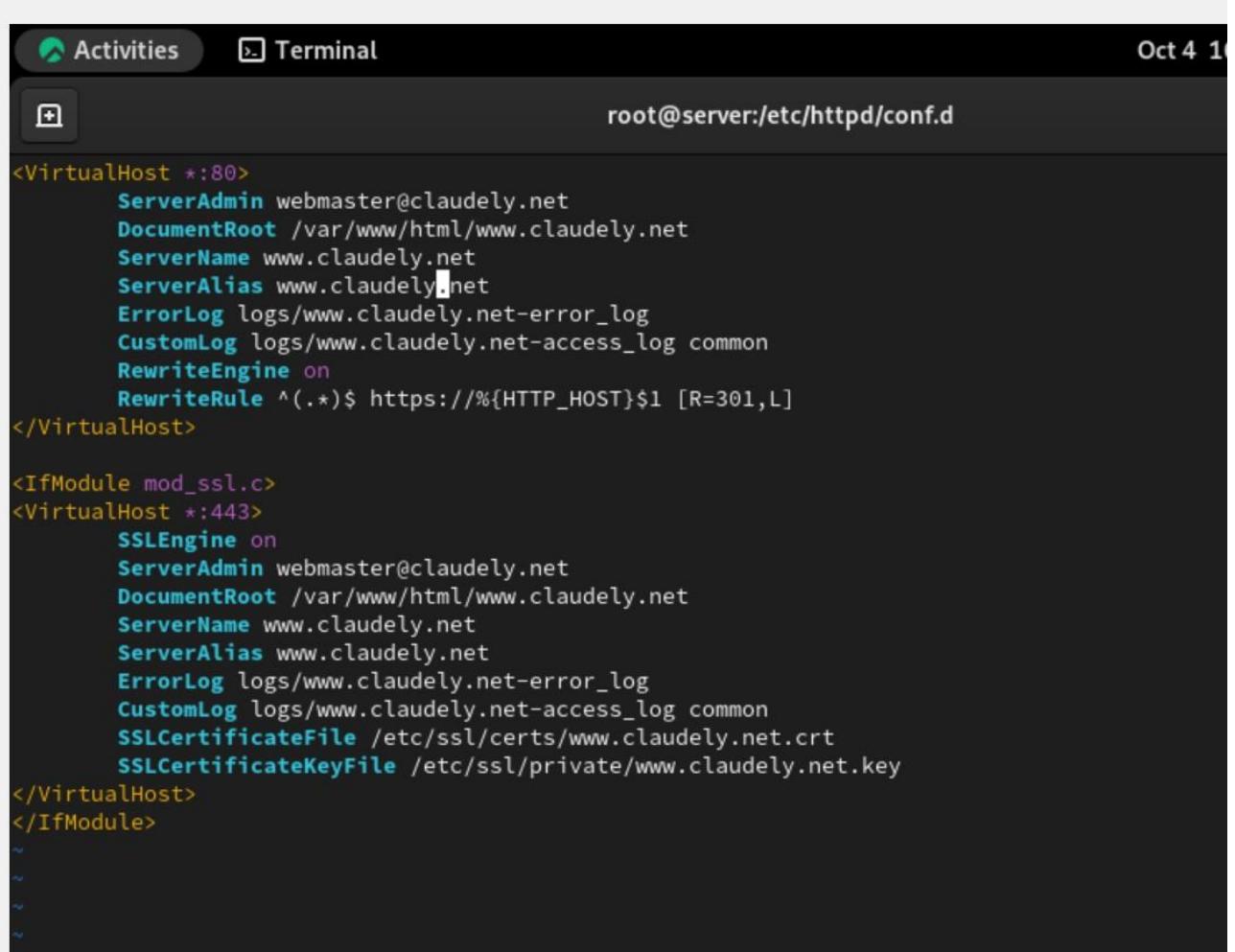
Для перехода веб-сервера www.claudely.net на функционирование через протокол HTTPS требуется изменить его конфигурационный файл. Для этого перейдём в каталог с конфигурационными файлами (Рис. 1.5):

```
cd /etc/httpd/conf
```

```
[root@server claudely.net certs]# cd /etc/httpd/conf.d  
[root@server claudely.net conf.d]#
```

Рис. 1.5. Открытие каталога с конфигурационными файлами.

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



```
<VirtualHost *:80>
    ServerAdmin webmaster@claudely.net
    DocumentRoot /var/www/html/www.claudely.net
    ServerName www.claudely.net
    ServerAlias www.claudely.net
    ErrorLog logs/www.claudely.net-error_log
    CustomLog logs/www.claudely.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@claudely.net
    DocumentRoot /var/www/html/www.claudely.net
    ServerName www.claudely.net
    ServerAlias www.claudely.net
    ErrorLog logs/www.claudely.net-error_log
    CustomLog logs/www.claudely.net-access_log common
    SSLCertificateFile /etc/ssl/certs/www.claudely.net.crt
    SSLCertificateKeyFile /etc/ssl/private/www.claudely.net.key
</VirtualHost>
</IfModule>
~
```

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

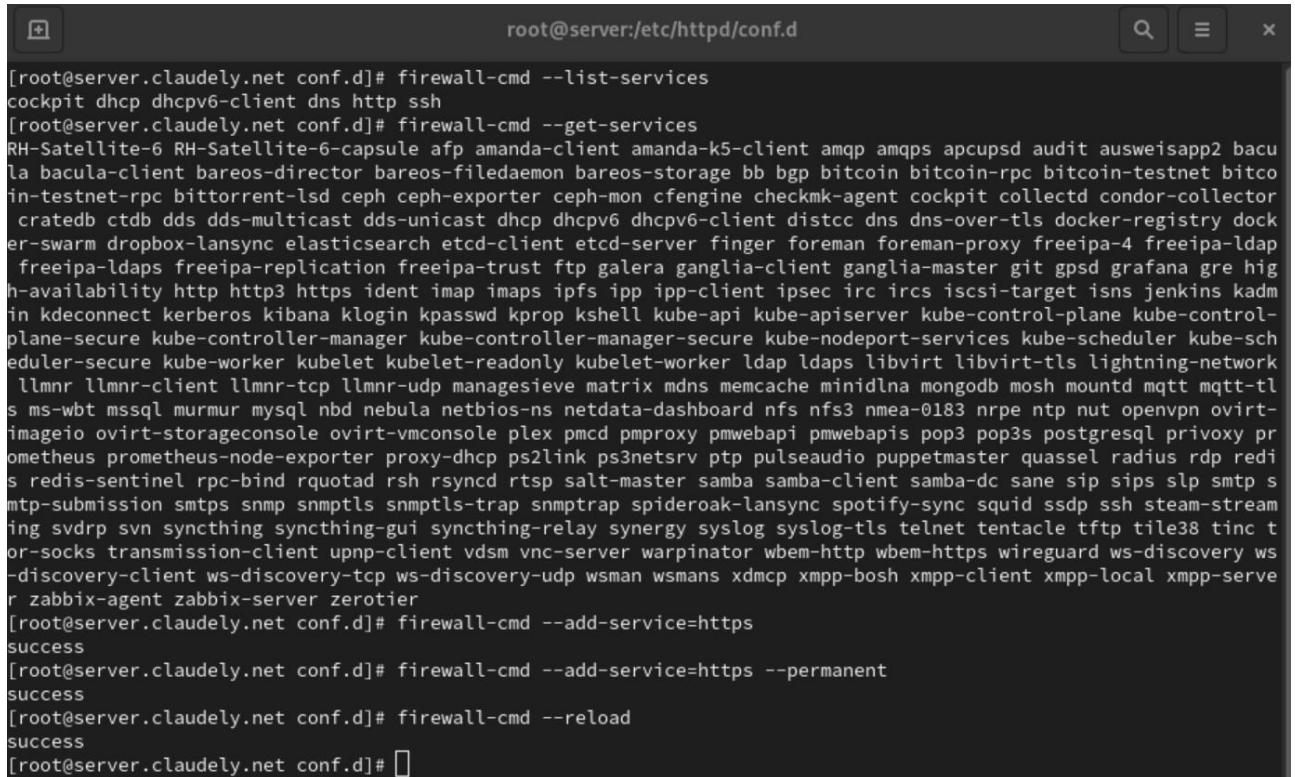
Внесём изменения в настройки межсетевого экрана на сервере, разрешив работу с https (Рис. 1.7):

```
firewall-cmd --list-services
firewall-cmd --get-services
firewall-cmd --add-
```

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

```
service=https --permanent
```

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

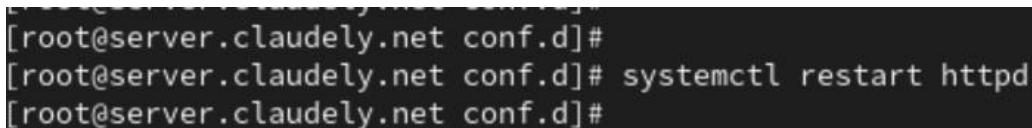


```
[root@server.claudely.net conf.d]# firewall-cmd --list-services
cockpit dhcp dhcpcv6-client dns http ssh
[root@server.claudely.net conf.d]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitco in-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfsengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcpc dhcpcv6 dhcpcv6-client distcc dns dns-over-tls docker-registry dock er-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpgsql grafana gre hig h-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadm in kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-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-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tl s ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane sip sipx slp smtp smtp-submission smtpts snmp snmpptls snmptrap snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vdsm vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsman xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.claudely.net conf.d]# firewall-cmd --add-service=https
success
[root@server.claudely.net conf.d]# firewall-cmd --add-service=https --permanent
success
[root@server.claudely.net conf.d]# firewall-cmd --reload
success
[root@server.claudely.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
```



```
[root@server.claudely.net conf.d]#
[root@server.claudely.net conf.d]# systemctl restart httpd
[root@server.claudely.net conf.d]#
```

Рис. 1.8. Перезапуск веб-сервера.

На виртуальной машине client в строке браузера введём название вебсервера www.claudely.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 просмотрим содержание сертификата.

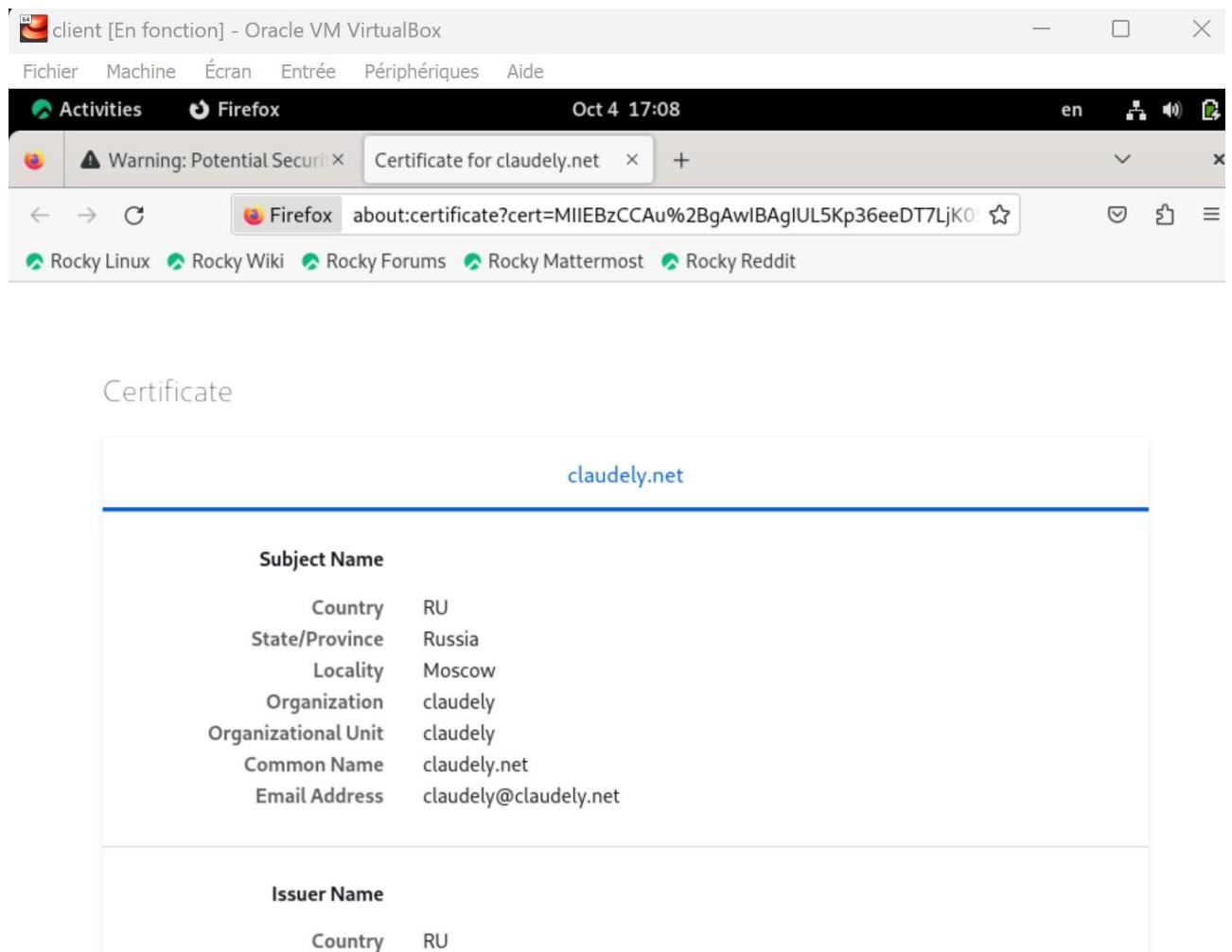


Рис. 1.9. Открытие веб-сервера www.claudely.net и автоматическое переключение на работу по протоколу HTTPS.

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

```
dnf -y install php
```

```

root@server.claudely.net conf.d]# dnf -y install php
Last metadata expiration check: 0:09:06 ago on Fri 04 Oct 2024 05:00:46 PM UTC.
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      1:1.20.1-16.el9_4.1  appstream    8.1 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-1.20.1-16.el9_4.1.noarch.rpm           4.2 kB/s | 8.1 kB   00:01
(2/9): php-pdo-8.0.30-1.el9_2.x86_64.rpm                      42 kB/s | 81 kB   00:01
(3/9): php-xml-8.0.30-1.el9_2.x86_64.rpm                     67 kB/s | 131 kB   00:01
(4/9): php-mbstring-8.0.30-1.el9_2.x86_64.rpm                2.9 MB/s | 468 kB   00:00
(5/9): php-opcache-8.0.30-1.el9_2.x86_64.rpm                 1.9 MB/s | 509 kB   00:00
(6/9): php-common-8.0.30-1.el9_2.x86_64.rpm                  2.9 MB/s | 665 kB   00:00
(7/9): php-cli-8.0.30-1.el9_2.x86_64.rpm                    2.9 MB/s | 131 kB   00:00
(8/9): php-fpm-8.0.30-1.el9_2.x86_64.rpm                   2.9 MB/s | 131 kB   00:00
(9/9): php-xml-8.0.30-1.el9_2.x86_64.rpm                  2.9 MB/s | 131 kB   00:00
=====

```

Рис. 2.1. Установка пакетов для работы с PHP.

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

```

root@server:/var/www/html/www.claudely.net
root@server:/var/www/html/www... x
?php
phpinfo();
?b
~
~
```

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

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 вебсервера `www.claudely.net` и убедимся, что будет выведена страница с информацией об используемой на веб-сервере версии PHP (рис. 2.4):

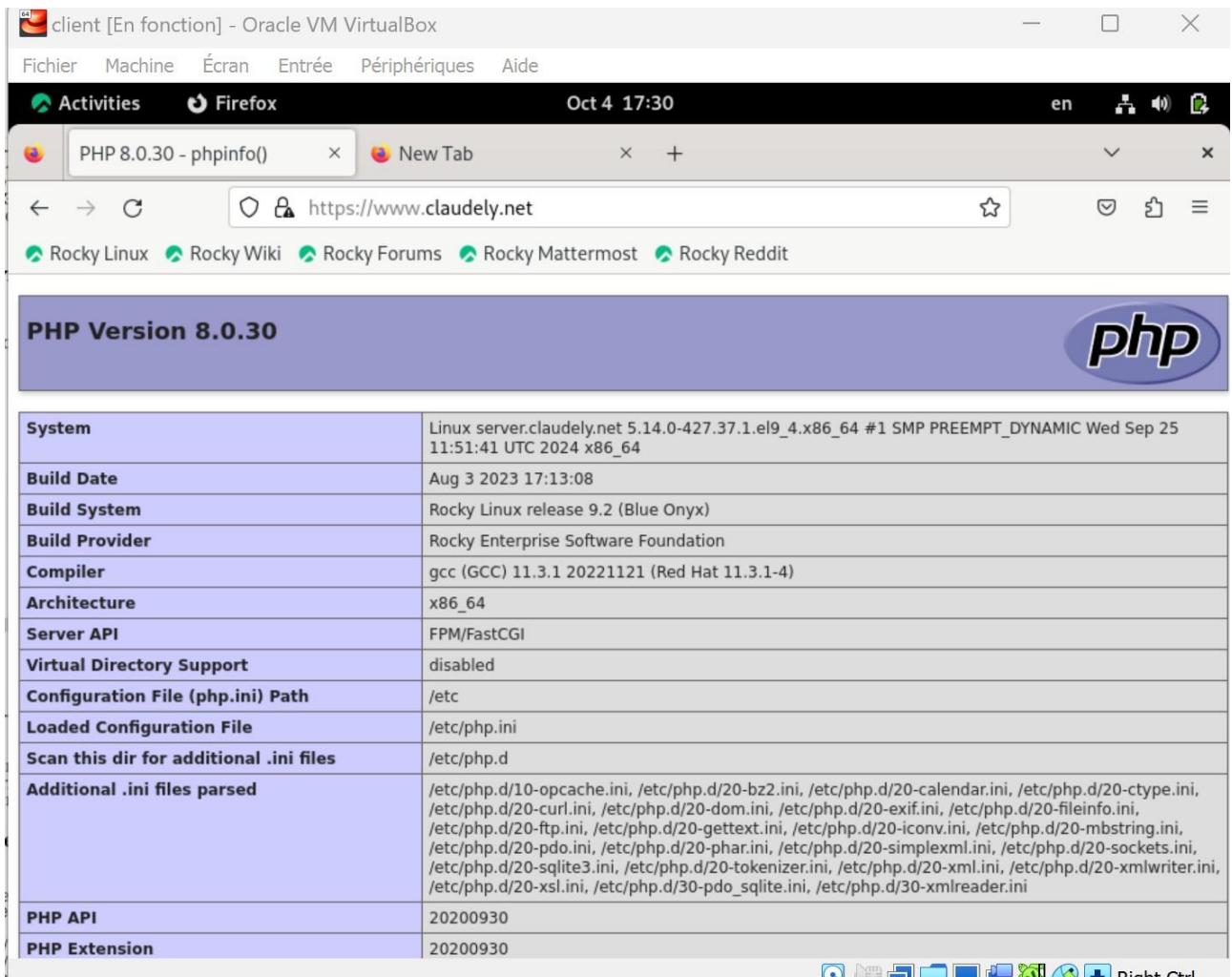


Рис. 2.4. Проверка вывода страницы с информацией об используемой на веб-сервере версии PHP.

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.claudely.net www.claudely.net]# cd /vagrant/provision/server/http
[root@server.claudely.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/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.claudely.net.conf'? yes
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/server.user.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.user.net.conf'? yes
[root@server.claudely.net http]# cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
cp: overwrite '/vagrant/provision/server/http/var/www/html/server.claudely.net/index.html'? yes
[root@server.claudely.net http]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/private
[root@server.claudely.net http]# mkdir -p /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.claudely.net http]# cp -R /etc/pki/tls/private/www.claudely.net.key /vagrant/provision/server/http/etc/pki/tls/private
[root@server.claudely.net http]# cp -R /etc/pki/tls/certs/www.claudely.net.crt /vagrant/provision/server/http/etc/pki/tls/certs
[root@server.claudely.net http]#
```

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):

```
root@server:/vagrant/provision/server/http x clau
#!/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/tls/* /etc/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=https --permanent

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

Рис. 3.2. Внесение изменений в скрипт /vagrant/provision/server/http.sh, добавив установку PHP и настройку межсетевого экрана, разрешающую 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, веббраузером) 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 вебсайтов. Владельцы веб-сайтов могут получить сертификат от Let's Encrypt, чтобы обеспечить шифрование и подтвердить свою легитимность в онлайнсреде.