

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Администрирование сетевых подсистем

Студент: Боровикова Карина Владимировна

Группа: НПИбд-01-20

МОСКВА

2022 г.

Цель работы

Приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

Ход выполнения работы

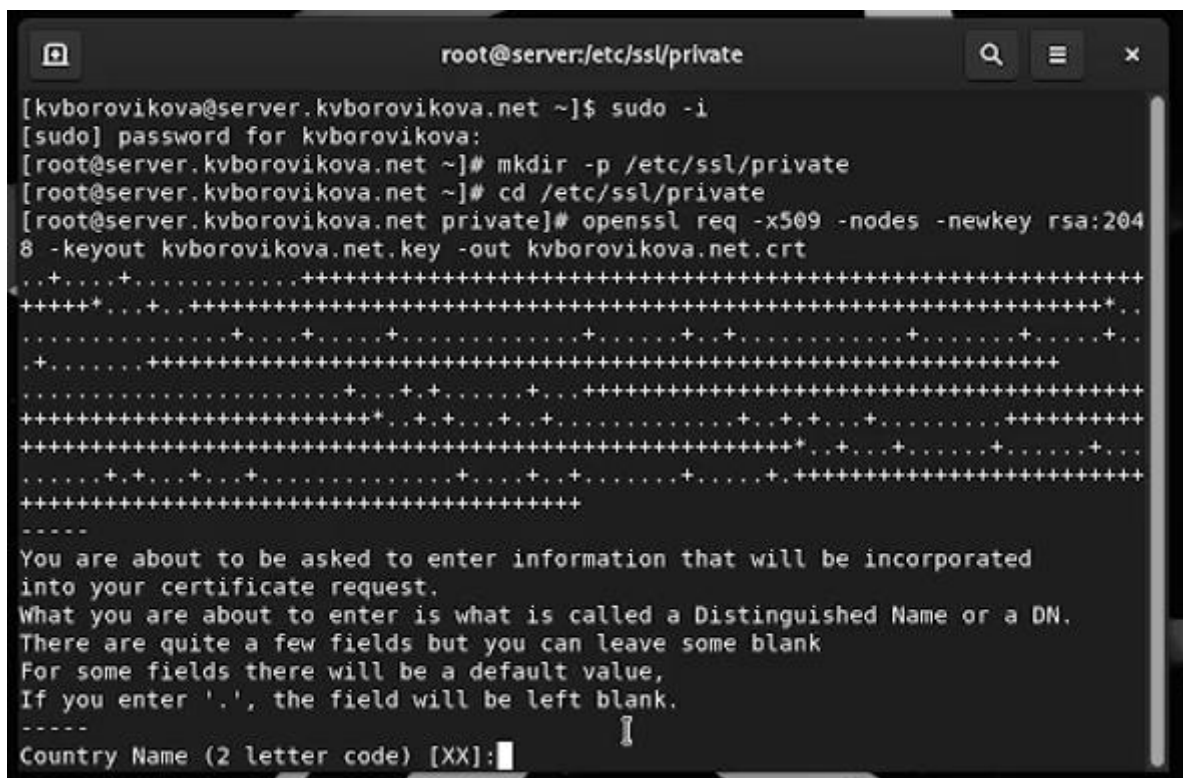
1. Конфигурирование HTTP-сервера для работы через протокол HTTPS

1. Загрузим нашу операционную систему и перейдем в рабочий каталог с проектом:
2. Запустим виртуальную машину server:
`vagrant up server`
3. На виртуальной машине server войдем под нашим пользователем и откроем терминал. Перейдем в режим суперпользователя:
`sudo -i`
4. В каталоге `/etc/ssl` создадим каталог `private`:
`mkdir -p /etc/ssl/private`
`cd /etc/ssl/private`
Сгенерируем ключ и сертификат, используя следующую команду:
`openssl req -x509 -nodes -newkey rsa:2048 -keyout user.net.key -out user.net.crt` В этой строке:
 - `req -x509` означает, что используется запрос подписи сертификата `x509` (CSR);
 - параметр `-nodes` указывает OpenSSL, что нужно пропустить шифрование сертификата SSL с использованием парольной фразы, т.е. позволить Apache читать файл без какого-либо вмешательства пользователя (без ввода пароля при попытке доступа к странице, в частности);
 - параметр `-newkey rsa: 2048` указывает, что одновременно создаются новый ключ и новый сертификат, причём используется 2048-битный ключ RSA;
 - параметр `-keyout` указывает, где хранить сгенерированный файл закрытого ключа при создании;
 - параметр `-out` указывает, где разместить созданный сертификат SSL.

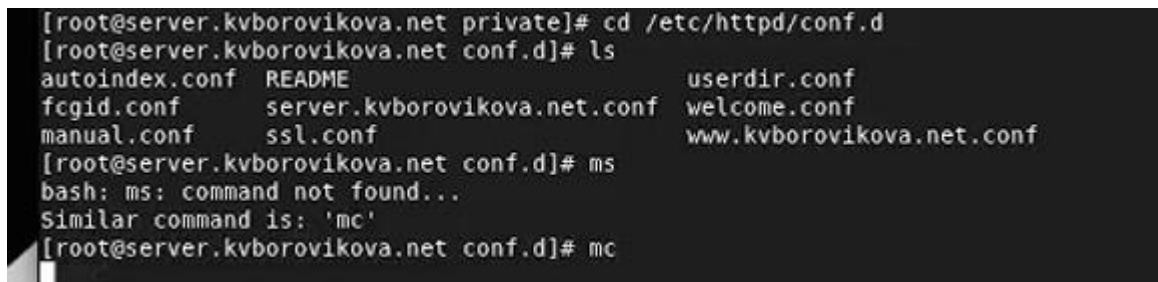
Далее требуется заполнить сертификат:

- в строке кода страны укажем `RU`;
- в строке названия страны укажем `Russia`;
- в строке названия города укажем `Moscow`;
- в строке названия организации укажем свой логин;

- в строке названия подразделения укажем свой логин;
- в строке названия хоста должно быть указано доменное имя нашего веб-сервера user.net (вместо user укажем свой логин);
- в строке email адреса должен быть указан user@user.net (вместо user укажем свой логин).



5. Для перехода веб-сервера `www.user.net` на функционирование через протокол HTTPS требуется изменить его конфигурационный файл. Перейдем в каталог с конфигурационными файлами:



Откроем на редактирование файл /etc/httpd/conf.d/www.user.net.conf и заменим его содержимое на следующее (вместо user укажем свой логин):

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

Виртуальный хост на порту 80

администратор сервера – webmaster@rmkipchakbaev.net

Директория, в которой находится контент - /var/www/html/www.rmkipchakbaev.net

Имя сервера – www.rmkipchakbaev.net

Псевдоним - www.rmkipchakbaev.net

Логи ошибок записываются в /logs/ www.rmkipchakbaev.net-error_log

Логи доступа записываются в /logs/ www.rmkipchakbaev.net-access_log

Включено манипулирование адресами URL

Правило манипулирования: ^(.*)\$ https://%{HTTP_HOST}\$1 [R=301,L]

Подключение модуля ssl

Виртуальный хост на порту 443

Включить поддержку SSL

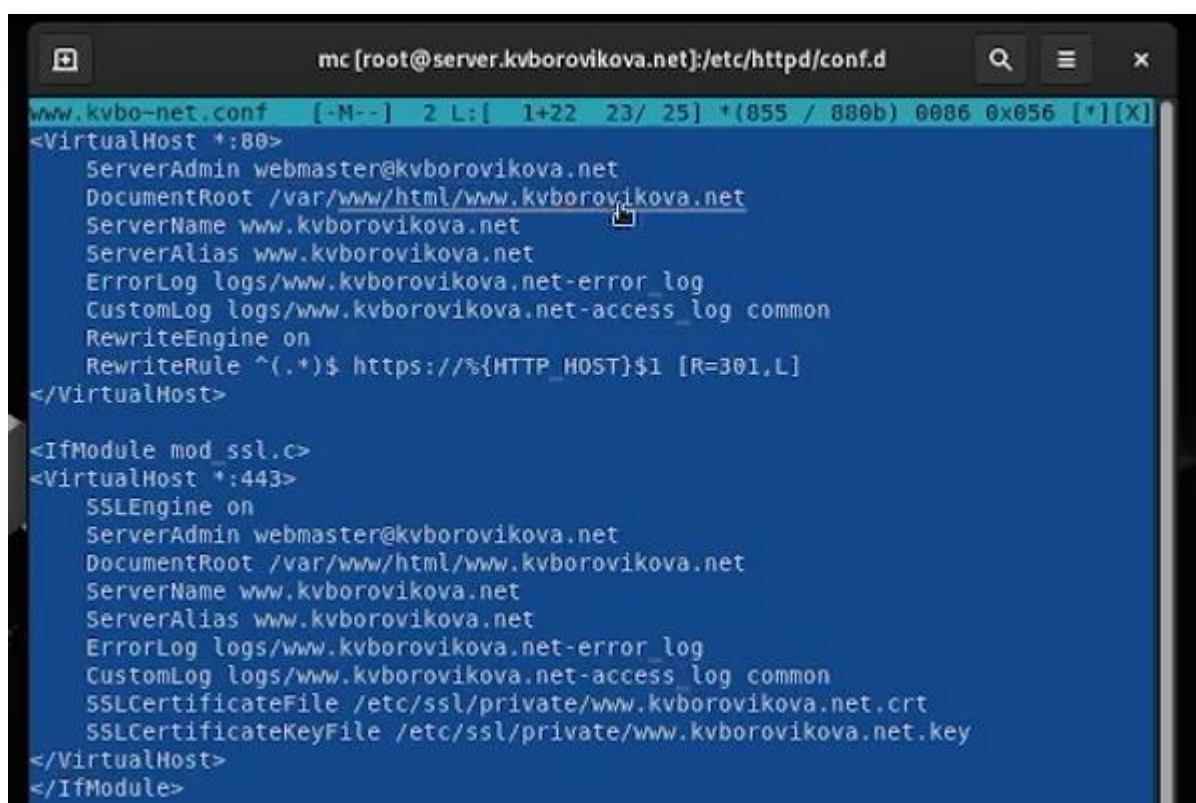
{*

Повторяющиеся параметры

*}

Местонахождение SSL сертификата: /etc/ssl/private/rmkipchakbaev.net.crt

Местонахождение SSL ключа: /etc/ssl/private/rmkipchakbaev.net.key



```
mc [root@server.kvborovikova.net]:/etc/httpd/conf.d
www.kvbo-net.conf [-M--] 2 L:[ 1+22 23/ 25] *(855 / 880b) 0086 0x056 [*][X]
<VirtualHost *:80>
    ServerAdmin webmaster@kvborovikova.net
    DocumentRoot /var/www/html/www.kvborovikova.net
    ServerName www.kvborovikova.net
    ServerAlias www.kvborovikova.net
    ErrorLog logs/www.kvborovikova.net-error_log
    CustomLog logs/www.kvborovikova.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@kvborovikova.net
    DocumentRoot /var/www/html/www.kvborovikova.net
    ServerName www.kvborovikova.net
    ServerAlias www.kvborovikova.net
    ErrorLog logs/www.kvborovikova.net-error_log
    CustomLog logs/www.kvborovikova.net-access_log common
    SSLCertificateFile /etc/ssl/private/www.kvborovikova.net.crt
    SSLCertificateKeyFile /etc/ssl/private/www.kvborovikova.net.key
</VirtualHost>
</IfModule>
```

Рисунок 3. Файл конфигурации, открытый на редактирование

- Внесем изменения в настройки межсетевого экрана на сервере, разрешив работу с https:

firewall-cmd --list-services

firewall-cmd --get-services

firewall-cmd --add-service=https

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

firewall-cmd --reload

```

firewall-cmd --list-services
[root@server.kvborovikova.net conf.d]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http ssh
[root@server.kvborovikova.net conf.d]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule amanda-client amanda-k5-client amqp amqps
apcupsd audit bacula bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bi
tcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine cockpit collectd condor-
collector ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry
docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger forem
an foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeip
a-trust ftp galera ganglia-client ganglia-master git grafana gre high-availabili
ty http https imap imaps ipp ipp-client ipsec irc ircs iscsi-target isns jenkins
kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-api
server kube-control-plane kube-controller-manager kube-scheduler kubelet-worker
ldap ldaps libvirt libvirt-tls lightning-network llmnr managesieve matrix mdns m
emcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd
netbios-ns nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storagec
onsole ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresq
l privoxy prometheus proxy-dhcp ptp pulseaudio puppetmaster quassel radius rdp r
edis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-cli
ent samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptrap spideroa
k-lansync spotify-sync squid sssd ssh steam-streaming svdrp svn syncthing syncth
ing-gui synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks tra
nsmission-client upnp-client vdsu vnc-server wbem-http wbem-https wireguard wsma
n wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-
server
[root@server.kvborovikova.net conf.d]# firewall-cmd --add-service=https
success
[root@server.kvborovikova.net conf.d]# firewall-cmd --add-service=https --perman
ent
success
[root@server.kvborovikova.net conf.d]# firewall-cmd --reload
success
[root@server.kvborovikova.net conf.d]#

```

Рисунок 4. Внесение изменений в настройки межсетевого экрана

7. Перезапустим веб-сервер:

systemctl restart httpd

```

[root@server.kvborovikova.net private]# systemctl restart httpd
[root@server.kvborovikova.net private]#

```

Рисунок 5. Перезапуск веб-сервера

8. На виртуальной машине client в строке браузера введем название веб-сервера `www.user.net` (вместо `user` укажем свой логин) и убедимся, что произойдёт автоматическое переключение на работу по протоколу HTTPS. На открывшейся странице с сообщением о незащищённости соединения нажмем кнопку «Дополнительно», затем добавим адрес нашего сервера в постоянные исключения. Затем просмотрим содержание сертификата (нажмем на значок с замком в адресной строке и кнопку «Подробнее»).

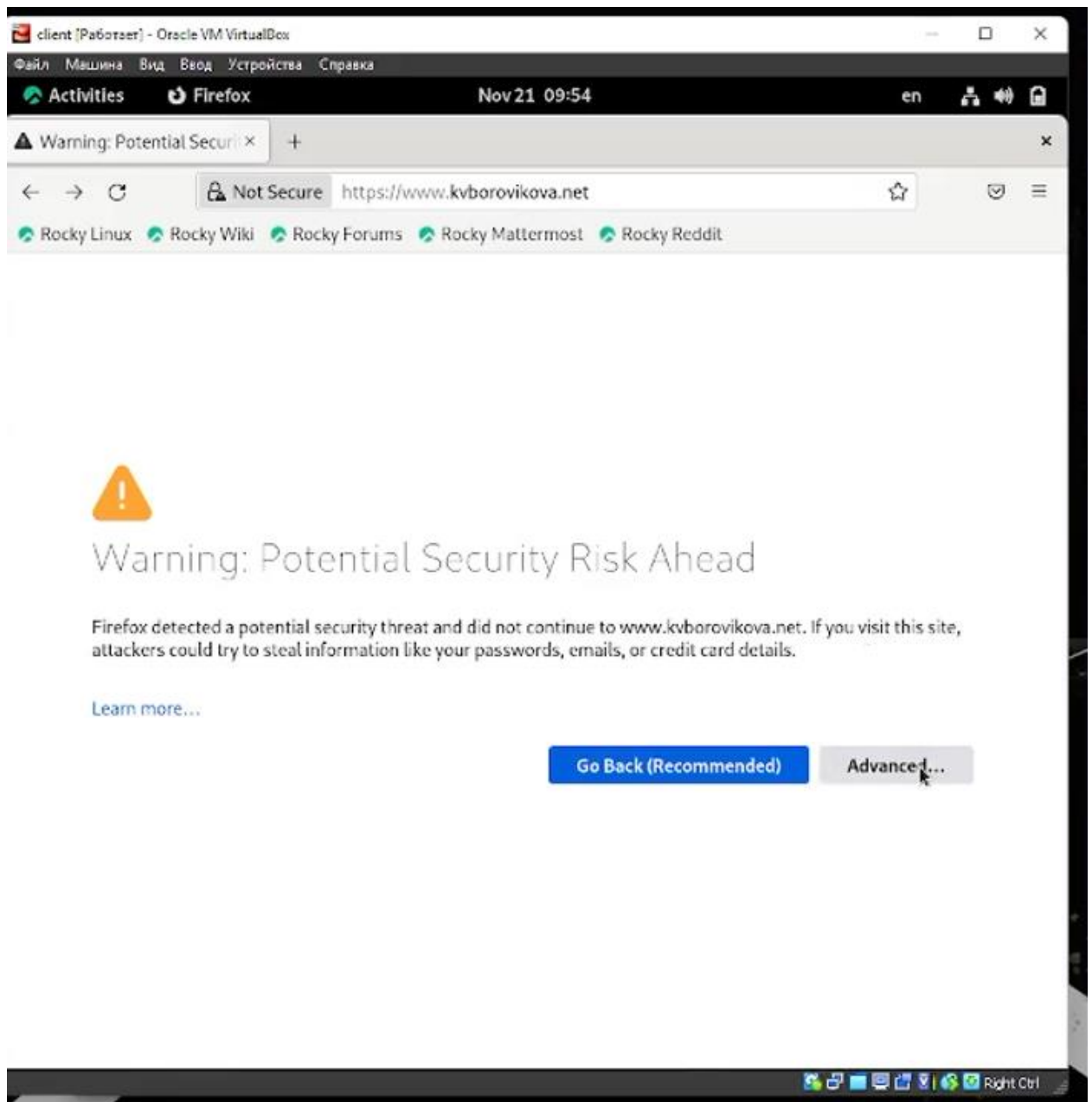


Рисунок 6. Ввели название веб-сервера, видим переключение на работу по протоколу https



Рисунок 7. Адрес сервера добавлен в исключения

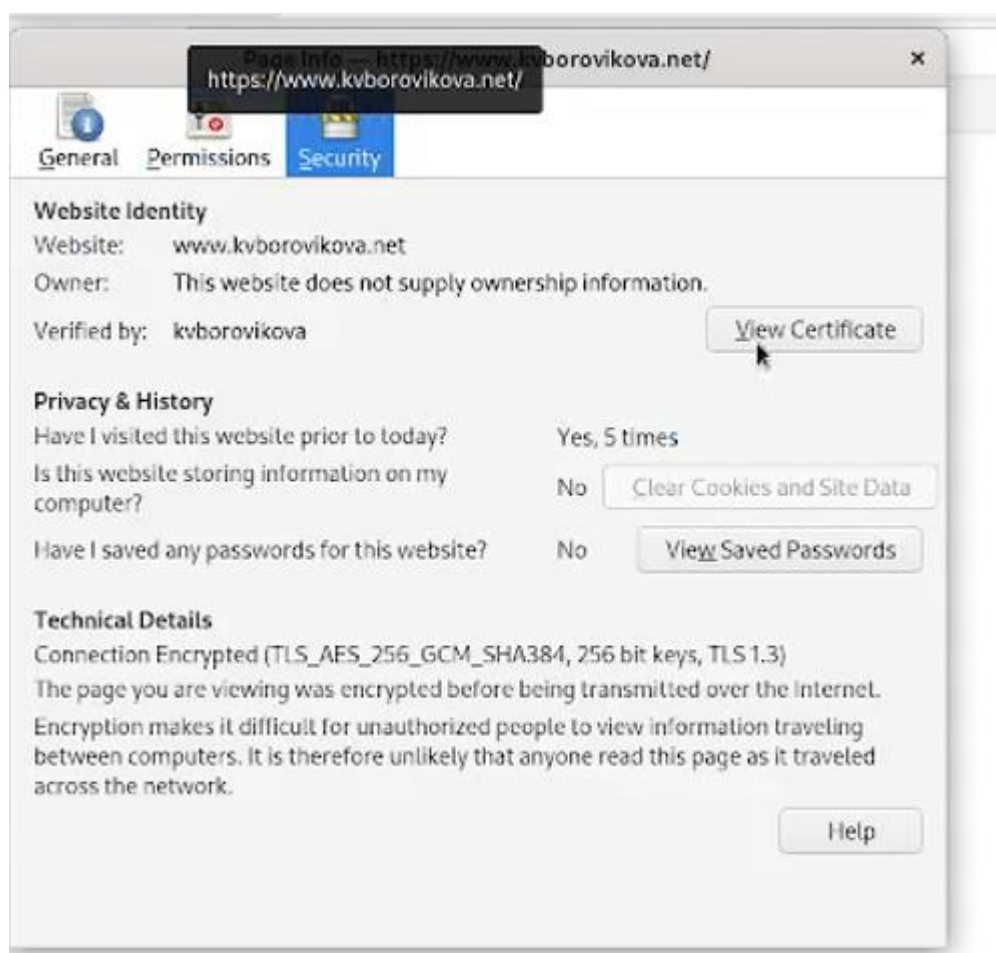


Рисунок 8. Открыли дополнительные сведения

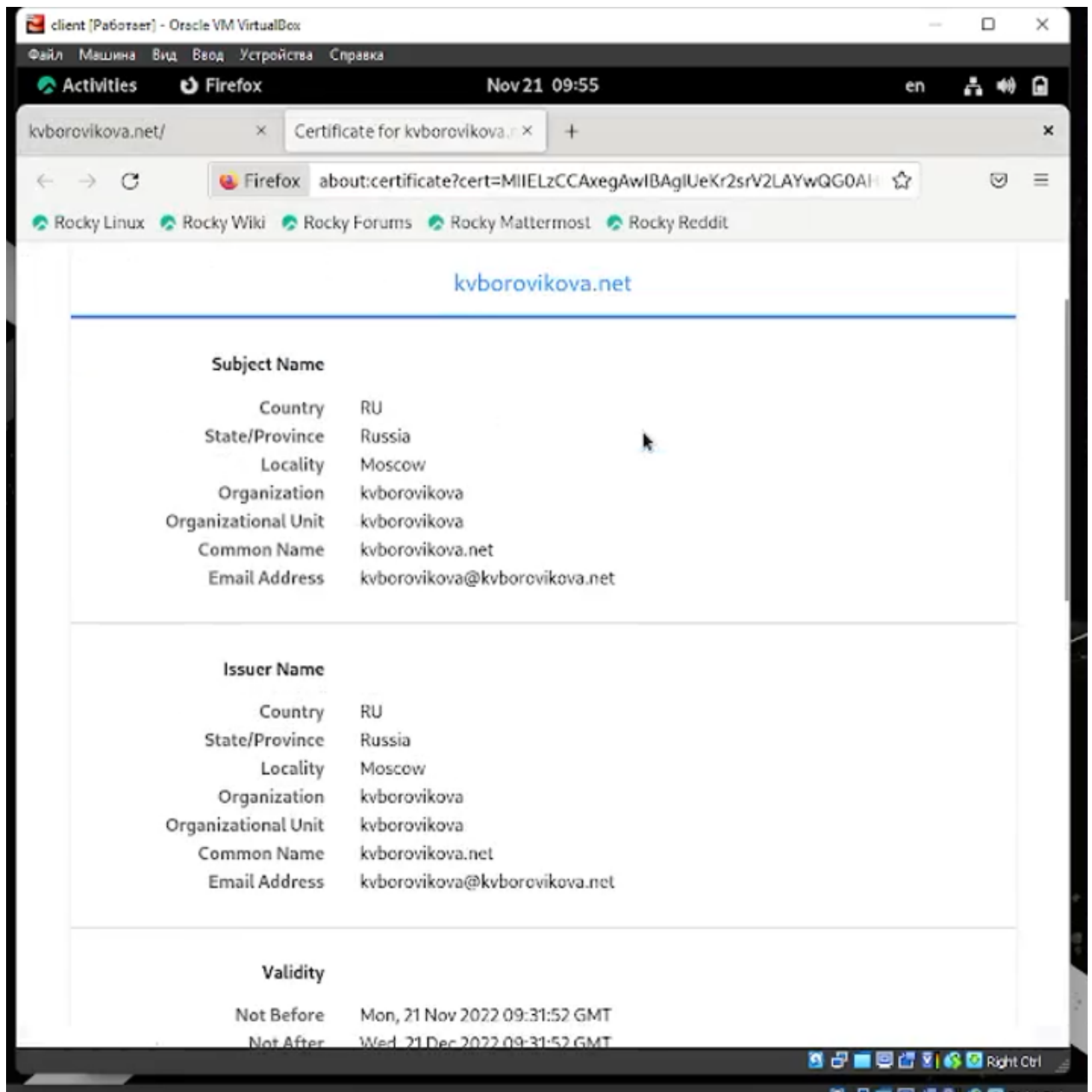


Рисунок 9. Данные в сертификате совпадают с теми, которые мы вводили в начале выполнения ЛР

2. Конфигурирование HTTP-сервера для работы с PHP

1. Установим пакеты для работы с PHP:

`dnf -y install php`

```
[root@server.kvborovikova.net rz]# dnf -y install php
Last metadata expiration check: 0:45:44 ago on Mon 21 Nov 2022 09:10:35 AM UTC.
Dependencies resolved.
=====
Package                Arch      Version      Repository    Size
=====
Installing:
php                    x86_64     8.0.13-2.el9_0    appstream     12 k
Installing dependencies:
nginx-filessystem      noarch     1:1.20.1-10.el9   appstream     11 k
php-common             x86_64     8.0.13-2.el9_0    appstream     665 k
Installing weak dependencies:
php-cli                x86_64     8.0.13-2.el9_0    appstream     3.1 M
php-fpm                x86_64     8.0.13-2.el9_0    appstream     1.6 M
php-mbstring           x86_64     8.0.13-2.el9_0    appstream     472 k
php-opcache            x86_64     8.0.13-2.el9_0    appstream     509 k
php-pdo                x86_64     8.0.13-2.el9_0    appstream     85 k
php-xml                x86_64     8.0.13-2.el9_0    appstream     134 k

Transaction Summary
=====
Install 9 Packages

Total download size: 6.5 M
Installed size: 35 M
Downloading Packages:
(1/9): php-pdo-8.0.13-2.el9_0.x86_64.rpm      15 kB/s | 85 kB    00:05
(2/9): nginx-filessystem-1.20.1-10.el9.noarch.rp 1.9 kB/s | 11 kB    00:05
(3/9): php-xml-8.0.13-2.el9_0.x86_64.rpm      24 kB/s | 134 kB   00:05
(4/9): php-opcache-8.0.13-2.el9_0.x86_64.rpm 652 kB/s | 509 kB   00:00
(5-7/9): php-mbstring-8.0.13-2.el9_0.x86_64.rpm 1.1 MB/s | 1.8 MB   00:04 ETA
```

Рисунок 10. Установка пакетов для работы с PHP

- В каталоге /var/www/html/www.user.net (вместо user укажем свой логин) заменим файл index.html на index.php следующего содержания:

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



Рисунок 11. Меняем содержимое файла index.php

- Скорректируем права доступа в каталог с веб-контентом:
chown -R apache:apache /var/www

```
[root@server.kvborovikova.net www.kvborovikova.net]# chown -R apache:apache /var
/var
[root@server.kvborovikova.net www.kvborovikova.net]# restorecon -vB /etc
```

Рисунок 12. Корректируем права доступа

4. Восстановим контекст безопасности в SELinux:

```
restorecon -vR /etc
```

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

5. Перезапустим HTTP-сервер:

```
systemctl restart httpd
```

```
[root@server.kvborovikova.net www.kvborovikova.net]# restorecon -vR /etc
[root@server.kvborovikova.net www.kvborovikova.net]# restorecon -vR /var/www
[root@server.kvborovikova.net www.kvborovikova.net]# systemctl restart httpd
[root@server.kvborovikova.net www.kvborovikova.net]#
```

Рисунок 13. Восстановление контекста безопасности и перезапуск http-сервера

6. На виртуальной машине client в строке браузера введем название веб-сервера `www.user.net` (вместо `user` укажем свой логин) и убедимся, что будет выведена страница с информацией об используемой на веб-сервере версии PHP.

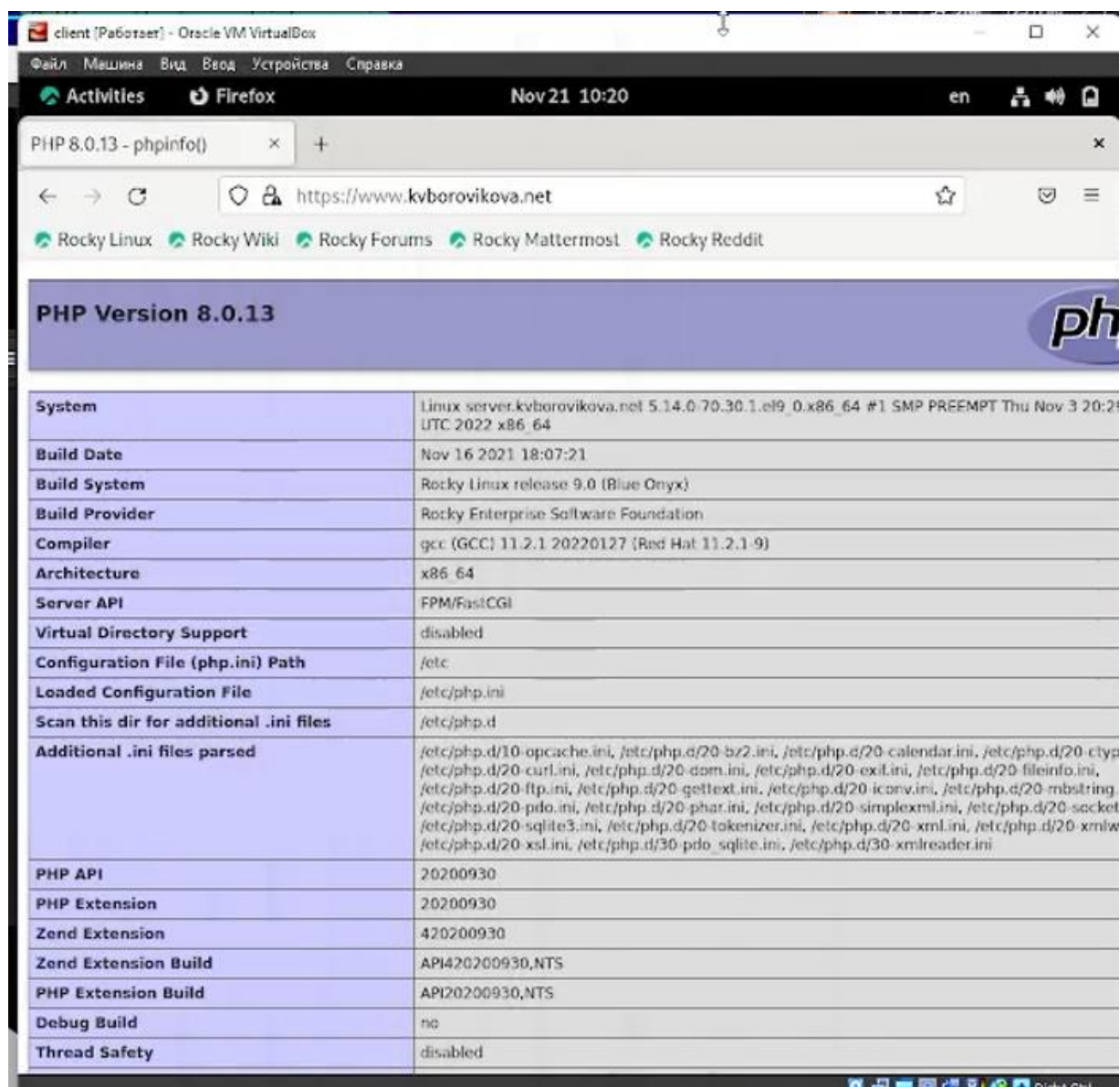


Рисунок 14. Страница с информацией о версии PHP

3. Внесение изменений в настройки внутреннего окружения виртуальной машины

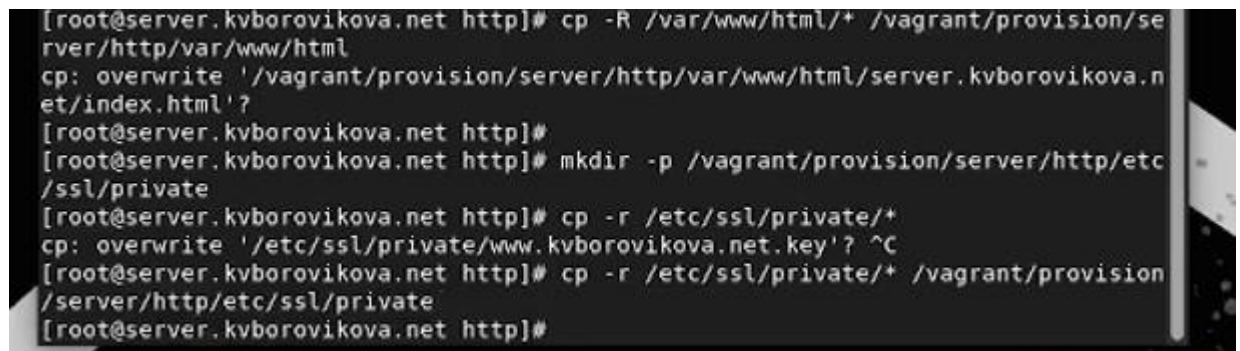
1. На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/http` и в соответствующие каталоги скопируйте конфигурационные файлы:

```
cp -R /etc/httpd/conf.d/* /vagrant/provision/server/http/etc/httpd/conf.d
```

```
cp -R /var/www/html/* /vagrant/provision/server/http/var/www/html
```

```
mkdir -p /vagrant/provision/server/http/etc/ssl/private
```

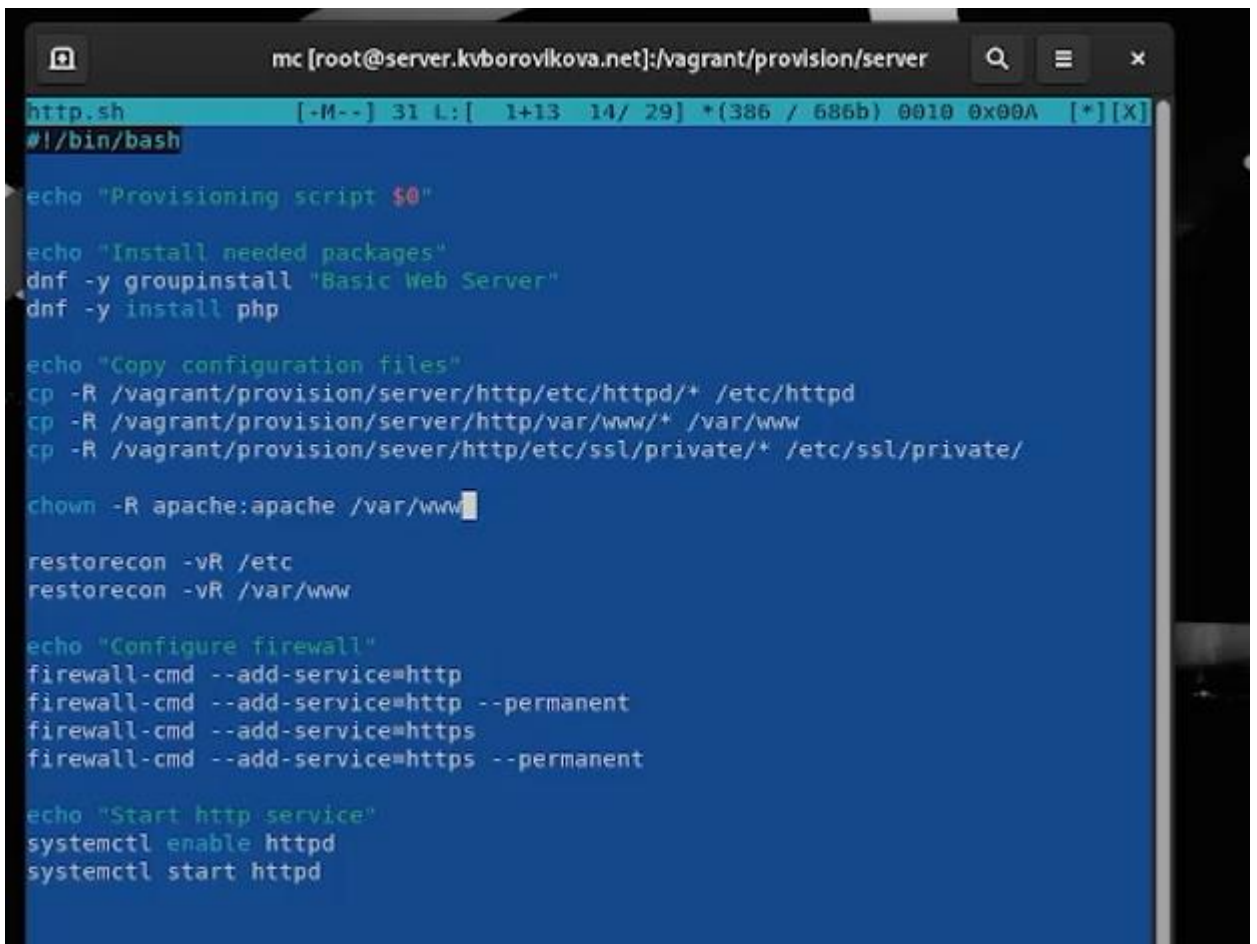
```
cp -R /etc/ssl/private/* /vagrant/provision/server/http/etc/ssl/private
```



```
[root@server.kvborovikova.net http]# cp -R /var/www/html/* /vagrant/provision/se  
rver/http/var/www/html  
cp: overwrite '/vagrant/provision/server/http/var/www/html/server.kvborovikova.n  
et/index.html'?  
[root@server.kvborovikova.net http]#  
[root@server.kvborovikova.net http]# mkdir -p /vagrant/provision/server/http/etc  
/ssl/private  
[root@server.kvborovikova.net http]# cp -r /etc/ssl/private/*  
cp: overwrite '/etc/ssl/private/www.kvborovikova.net.key'? ^C  
[root@server.kvborovikova.net http]# cp -r /etc/ssl/private/* /vagrant/provision  
/server/http/etc/ssl/private  
[root@server.kvborovikova.net http]#
```

Рисунок 15. Копирование конфигурационных файлов

2. В имеющийся скрипт `/vagrant/provision/server/http.sh` внесем изменения, добавив установку РНР и настройку межсетевого экрана, разрешающую работать с https.

A screenshot of a terminal window titled 'mc [root@server.kvborovikova.net]:/vagrant/provision/server'. The terminal shows the execution of a script named 'http.sh'. The script's content is as follows:

```
#!/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/sever/http/etc/ssl/private/* /etc/ssl/private/

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

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

Рисунок 16. Содержимое файла http.sh

Вывод

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

Контрольные вопросы

1. В чём отличие HTTP от HTTPS?

HTTPS и HTTP – два протокола, с помощью которых передается информация в Интернете. Они предназначены для передачи текстовых данных между клиентом и сервером, а главное различие между ними – в наличии и отсутствии шифрования передаваемых данных соответственно

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

HTTPS использует SSL/TLS для шифрования данных

3. Что такое сертификационный центр? Приведите пример.

Центр сертификации или удостоверяющий центр (англ. Certification authority,

CA) — сторона (отдел, организация), чья честность неоспорима, а открытый ключ широко известен. Задача центра сертификации — подтверждать подлинность ключей шифрования с помощью сертификатов электронной подписи. Центрами сертификации можно назвать Comodo, Geotrust, Thawte и Symantec (ранее VeriSign). Например, если нам необходим сертификат открытого ключа (не самоподписанный), мы можем обратиться в сертификационный центр или к партнерам сертификационного центра для того, чтобы купить сертификат.