

# **Отчёт по лабораторной работе №5**

**Администрирование сетевых подсистем**

Ищенко Ирина НПИбд-02-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>14</b>
<b>4</b>	<b>Ответы на контрольные вопросы</b>	<b>15</b>

# Список иллюстраций

2.1	создание каталога . . . . .	6
2.2	Заполнение сертификата . . . . .	7
2.3	Содержимое каталогов /etc/ssl/private и /etc/ssl/certs . . . . .	7
2.4	Редактирование файла /etc/httpd/conf.d/www.ioithenko.net . . .	8
2.5	Внесение изменений в настройки межсетевого экрана, перезапуск веб-сервера . . . . .	8
2.6	Сообщение о незащищенности соединения . . . . .	9
2.7	Информация о веб-странице . . . . .	9
2.8	Содержимое сертификата . . . . .	10
2.9	PHP . . . . .	10
2.10	Редактирование index.php . . . . .	11
2.11	Права доступа, контекст безопасности . . . . .	11
2.12	Веб-страница с информацией об используемой версии PHP . . . .	11
2.13	Внутреннее окружение . . . . .	12
2.14	Внесение изменений в скрипт http.sh . . . . .	13

## **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

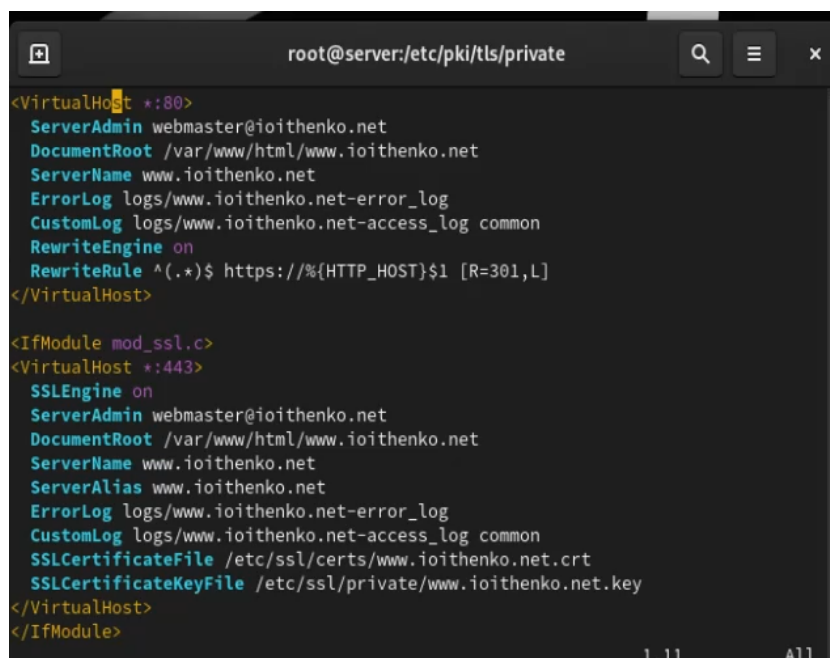
Запускаем ВМ через рабочий каталог. На ВМ server входим под собственным пользователем и переходим в режим суперпользователя. В каталоге /etc/ssl создаем каталог private (рис. 2.1):

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

Рис. 2.1: создание каталога

Генерируем ключ и сертификат (рис. 2.2), введя следующую команду:  
openssl req -x509 -nodes -newkey rsa:2048 -keyout www.ioithenko.net.key -out  
www.ioithenko.net.crt



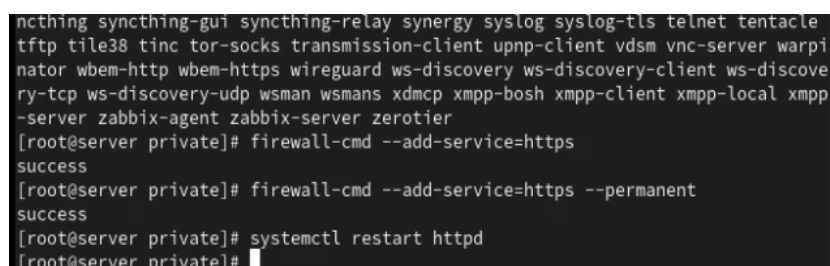


```
root@server:/etc/pki/tls/private
<VirtualHost *:80>
  ServerAdmin webmaster@ioithenko.net
  DocumentRoot /var/www/html/www.ioithenko.net
  ServerName www.ioithenko.net
  ErrorLog logs/www.ioithenko.net-error_log
  CustomLog logs/www.ioithenko.net-access_log common
  RewriteEngine on
  RewriteRule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

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

Рис. 2.4: Редактирование файла /etc/httpd/conf.d/www.ioithenko.net

Вносим изменения в настройки межсетевого экрана на сервере, перезапускаем веб-сервер (рис. 2.5)



```
ncthing syncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle
tftp tile38 tinc tor-socks transmission-client upnp-client vdsm vnc-server warpi
nator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discove
ry-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp
-server zabbix-agent zabbix-server zerotier
[root@server private]# firewall-cmd --add-service=https
success
[root@server private]# firewall-cmd --add-service=https --permanent
success
[root@server private]# systemctl restart httpd
[root@server private]#
```

Рис. 2.5: Внесение изменений в настройки межсетевого экрана, перезапуск веб-сервера

На VM client открываем в браузере страницу [www.ioithenko.net](http://www.ioithenko.net) с сообщением о незащищенности соединения (рис. 2.6). Добавив страницу в исключения, просматриваем информацию о сертификате (рис. 2.7), (рис. 2.8).



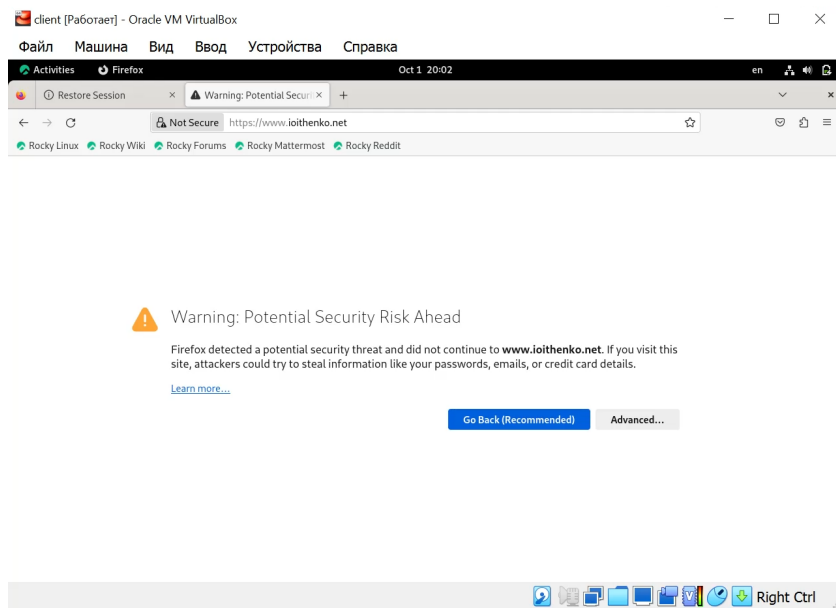


Рис. 2.6: Сообщение о незащищенности соединения

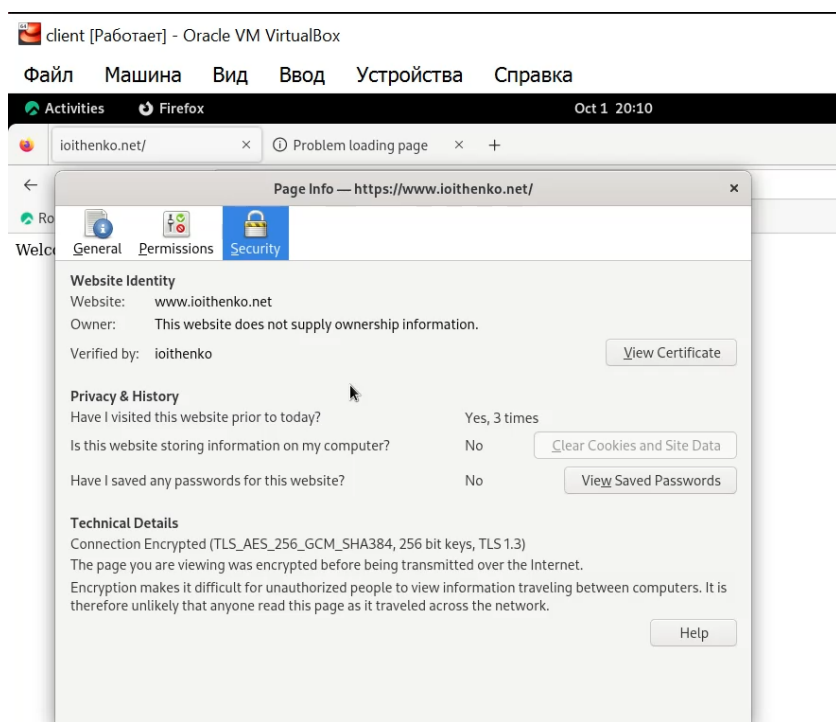


Рис. 2.7: Информация о веб-странице

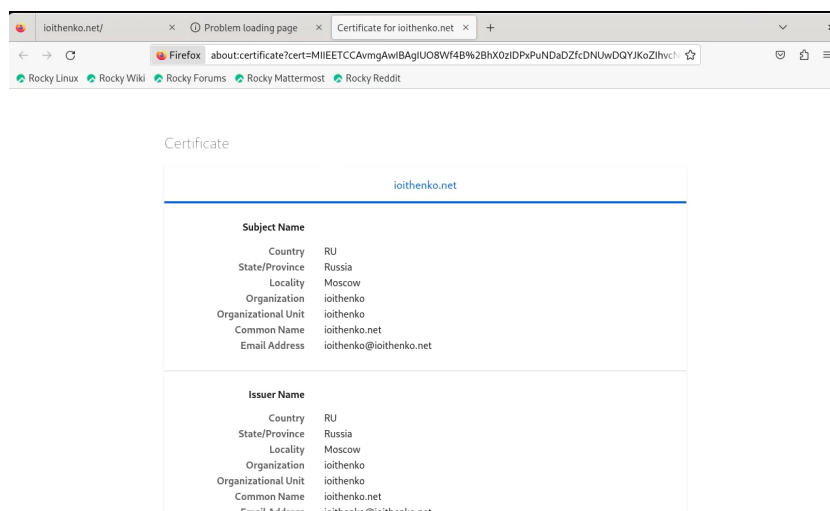


Рис. 2.8: Содержимое сертификата

Устанавливаю пакеты для работы с PHP: `dnf -y install php` (рис. 2.9).

```

root@server:~
[root@server ~]# dnf -y install php
Extra Packages for Enterprise Linux 9 - x86_64 68 kB/s | 36 kB 00:00
Extra Packages for Enterprise Linux 9 - x86_64 1.0 MB/s | 23 MB 00:21
Rocky Linux 9 - BaseOS 553 B/s | 4.1 kB 00:07
Rocky Linux 9 - BaseOS 193 kB/s | 2.3 MB 00:12
Rocky Linux 9 - AppStream 975 B/s | 4.5 kB 00:04
Rocky Linux 9 - AppStream 567 kB/s | 8.0 MB 00:14
Rocky Linux 9 - Extras 848 B/s | 2.9 kB 00:03
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
php x86_64 8.0.30-1.el9_2 appstream 7.7 k
Installing dependencies:
nginxfilesystem 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

```

Рис. 2.9: PHP

В каталоге `/var/www/html/www.ioithenko.net` заменяю `index.html` на `index.php`. Редактирую `index.php` (рис. 2.10).

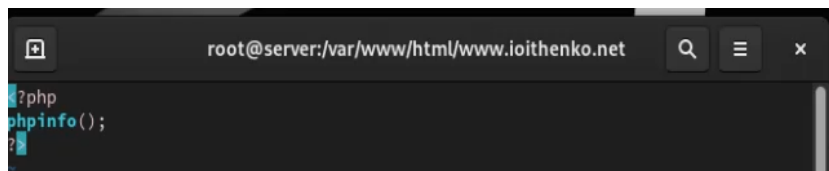


Рис. 2.10: Редактирование index.php

Корректирую права доступа в каталог с веб-контентом, восстанавливаю контекст безопасности в SELinux, перезагружаю HTTP-сервер (рис. 2.11).

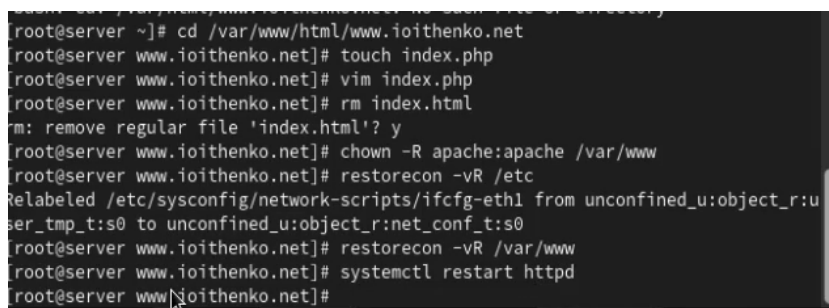


Рис. 2.11: Права доступа, контекст безопасности

На VM client ввожу в адресную строку браузера www.ioithenko.net и вижу веб-страницу с информацией об используемой версии PHP (рис. 2.12).

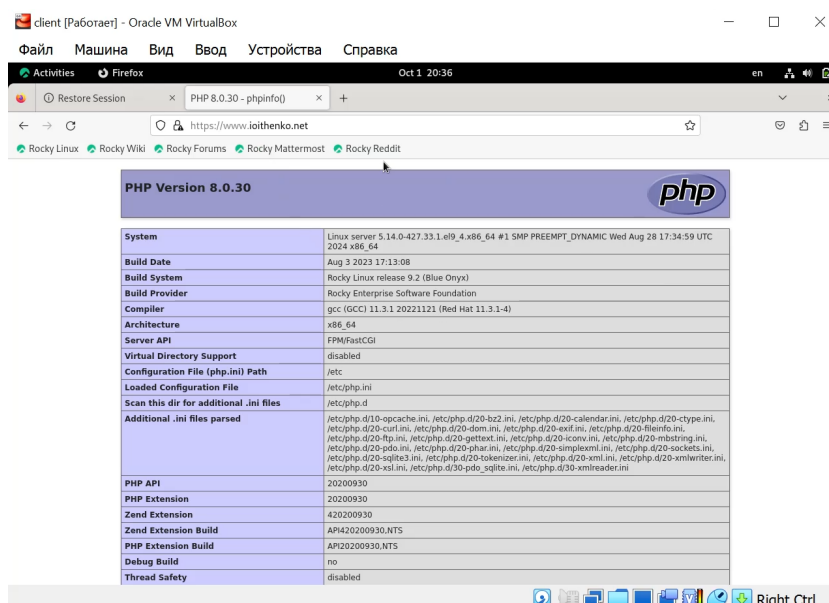
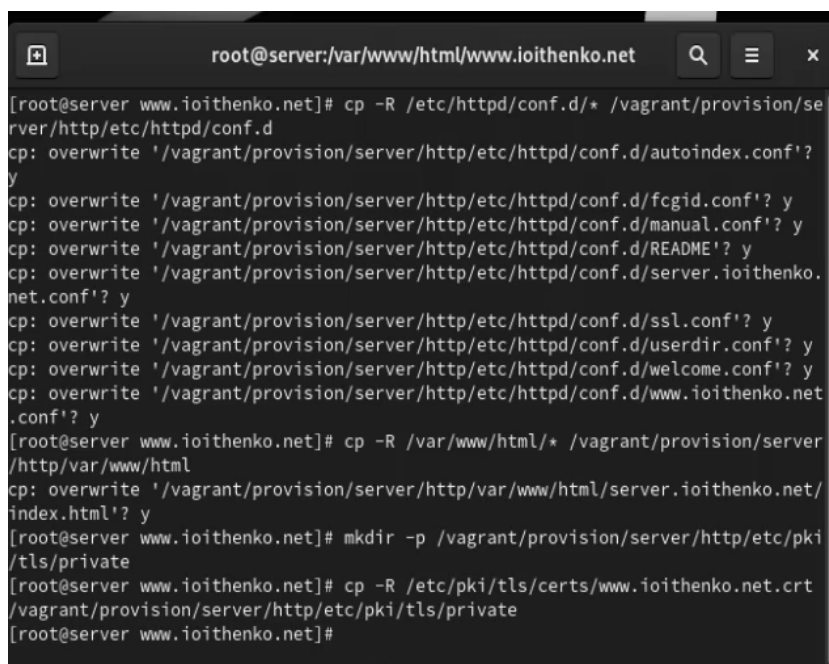


Рис. 2.12: Веб-страница с информацией об используемой версии PHP

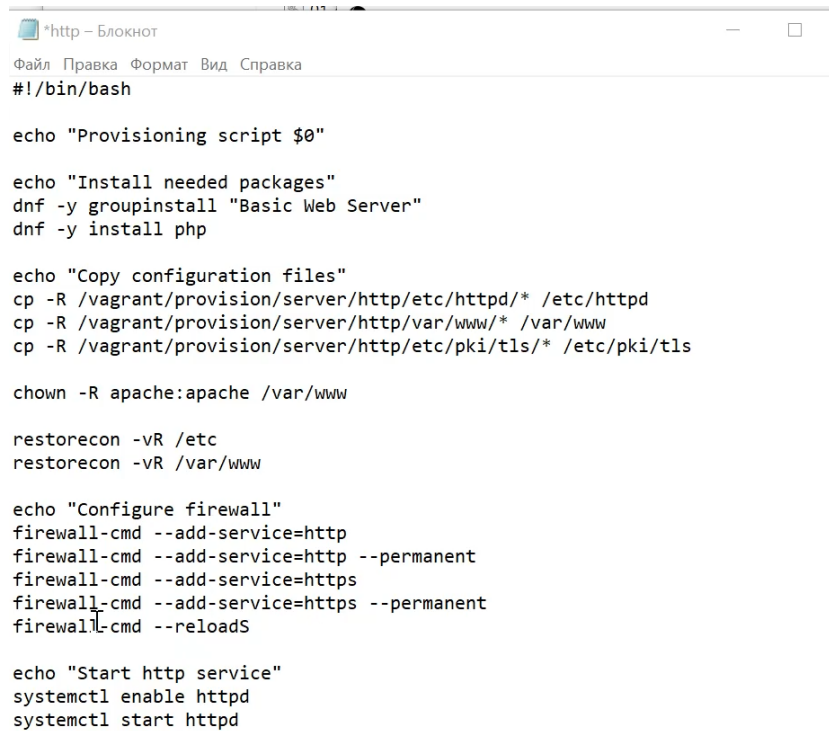
На VM server перехожу в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/` и копирую в соответствующие каталоги конфигурационные файлы (рис. 2.13).



```
root@server:var/www/html/www.ioithenko.net
[root@server www.ioithenko.net]# cp -R /etc/httpd/conf.d/* /vagrant/provision/se
rver/http/etc/httpd/conf.d
cp: overwrite '/vagrant/provision/server/http/etc/httpd/conf.d/autoindex.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.ioithenko.
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.ioithenko.net
.conf'? y
[root@server www.ioithenko.net]# cp -R /var/www/html/* /vagrant/provision/server
/http/var/www/html
cp: overwrite '/vagrant/provision/server/http/var/www/html/server.ioithenko.net/
index.html'? y
[root@server www.ioithenko.net]# mkdir -p /vagrant/provision/server/http/etc/pki
/tls/private
[root@server www.ioithenko.net]# cp -R /etc/pki/tls/certs/www.ioithenko.net.crt
/vagrant/provision/server/http/etc/pki/tls/private
[root@server www.ioithenko.net]#
```

Рис. 2.13: Внутреннее окружение

В скрипт `/vagrant/provision/server/http.sh` вношу изменения, добавив установку PHP и настройку межсетевого экрана для работы с https (рис. 2.14).



```
*http - Блокнот
Файл Правка Формат Вид Справка
#!/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. 2.14: Внесение изменений в скрипт http.sh

## **3 Выводы**

В ходе лабораторной работы я приобрела 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.