Лабораторная работа № 5. Расширенная настройка HTTP-сервера Apache

5.1. Цель работы

Приобретение практических навыков по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

5.2. Предварительные сведения

HTTPS (HyperText Transfer Protocol Secure) — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.

Улучшение безопасности при использовании HTTPS вместо HTTP достигается за счёт использования криптографических протоколов при организации HTTP-соединения и передачи по нему данных. Для шифрования может применяться протокол SSL (Secure Sockets Layer) или протокол TLC (Transport Layer Security). Оба протокола используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

Симметричное шифрование — способ шифрования, в котором для шифрования и дешифровывания данных применяется один и тот же криптографический ключ.

Асимметричное шифрование — способ шифрования, в котором для шифрования и дешифровывания данных применяется пара ключей — открытый и закрытый. Открытый ключ известен, передаётся по открытому каналу и используется для аутентификации пользователей и собственно для шифрования передаваемых данных. Закрытый ключ должен быть сохранён втайне и находиться на стороне получателя шифрованного сообщения. При помощи закрытого ключа сообщение дешифруется и таким образом подтверждается подлинность отправителя сообщения.

Криптографический ключ — секретная информация, используемая криптографическим алгоритмом при шифровании/дешифровании данных.

Основной характеристикой криптостойкости криптографического ключа является его длина, измеряемая, как правило, в битах. Для симметричных алгоритмов шифрования рекомендуемая минимальная длина ключа — 128 бит, для асимметричных — 1024 бит.

Сертификат открытого ключа — документ (электронный или бумажный), содержащий как сам открытый ключ, так и информацию о его владельце и области применения. Сертификат подписывается выдавшим его сертификационным центром, который подтверждает принадлежность открытого ключа владельцу.

По сути, *сертификационный центр (Certification authority, CA)* представляет собой компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей. Его открытый ключ широко известен общественности и не вызывает сомнений в подлинности.

5.3. Задание

1. Сгенерируйте криптографический ключ и самоподписанный сертификат безопасности для возможности перехода веб-сервера от работы через протокол HTTP к работе через протокол HTTPS (см. раздел 5.4.1).
2. Настройте веб-сервер для работы с PHP (см. раздел 5.4.2).

3. Напишите (или скорректируйте) скрипт для Vagrant, фиксирующий действия по расширенной настройке HTTP-сервера во внутреннем окружении виртуальной машины `server` (см. раздел 5.4.3).

5.4. Последовательность выполнения работы

5.4.1. Конфигурирование HTTP-сервера для работы через протокол HTTPS

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:
   ```
   cd /var/tmp/user_name/vagrant
   ```
   Здесь `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину `server`:
   ```
   make server
   ```
   (или, если вы работаете под ОС Windows, то `vagrant up server`).
3. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
4. В каталоге `/etc/ssl` создайте каталог `private`:
   ```
   mkdir -p /etc/ssl/private
   cd /etc/ssl/private
   ```
   Сгенерируйте 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 укажите RU;
   - в строке названия страны укажите Russia;
   - в строке названия города укажите Moscow;
   - в строке названия организации укажите свой логин;
   - в строке названия подразделения укажите свой логин;
   - в строке названия хоста должно быть указано доменное имя вашего веб-сервера user.net (вместо `user` укажите свой логин);
   - в строке email адреса должен быть указан user@user.net (вместо `user` укажите свой логин).

   Сгенерированные ключ и сертификат появятся в соответствующем каталогах `/etc/ssl/private`.
5. Для перехода веб-сервера `www.user.net` на функционирование через протокол HTTPS требуется изменить его конфигурационный файл. Перейдите в каталог с конфигурационными файлами:
   ```
   cd /etc/httpd/conf.d
   ```

Откройте на редактирование файл /etc/httpd/conf.d/www.user.net.conf и замените его содержимое на следующее (вместо user укажите 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>
```

В отчёте поясните построчно содержание этого файла.

6. Внесите 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
```

7. Перезапустите веб-сервер:

```
systemctl restart httpd
```

8. На виртуальной машине client в строке браузера введите название веб-сервера www.user.net (вместо user укажите свой логин) и убедитесь, что произойдёт автоматическое переключение на работу по протоколу HTTPS. На открывшейся странице с сообщением о незащищённости соединения нажмите кнопку «Дополнительно», затем добавьте адрес вашего сервера в постоянные исключения. Затем просмотрите содержание сертификата (нажмите на значок с замком в адресной строке и кнопку «Подробнее»).

5.4.2. Конфигурирование HTTP-сервера для работы с PHP

1. Установите пакеты для работы с PHP:

```
dnf -y install php
```

2. В каталоге /var/www/html/www.user.net (вместо user укажите свой логин) замените файл index.html на index.php следующего содержания:

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

3. Скорректируйте права доступа в каталог с веб-контентом:
```
chown -R apache:apache /var/www
```
4. Восстановите контекст безопасности в SELinux:
```
restorecon -vR /etc
restorecon -vR /var/www
```
5. Перезапустите HTTP-сервер:
```
systemctl restart httpd
```
6. На виртуальной машине `client` в строке браузера введите название веб-сервера `www.user.net` (вместо `user` укажите свой логин) и убедитесь, что будет выведена страница с информацией об используемой на веб-сервере версии PHP.

5.4.3. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите 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
```
2. В имеющийся скрипт `/vagrant/provision/server/http.sh` внесите изменения, добавив установку PHP и настройку межсетевого экрана, разрешающую работать с `https`.

5.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

5.6. Контрольные вопросы

1. В чём отличие HTTP от HTTPS?
2. Каким образом обеспечивается безопасность контента веб-сервера при работе через HTTPS?
3. Что такое сертификационный центр? Приведите пример.