

Отчет по лабораторной работе 5

Генералов Даниил, НПИбд-01-21, 1032202280

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	14
5	Контрольные вопросы	15

Список иллюстраций

3.1	openssl	7
3.2	apache	8
3.3	apache	9
3.4	firefox	10
3.5	firefox	11
3.6	firefox	12
3.7	vagrant	13

Список таблиц

1 Цель работы

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

2 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).

3 Выполнение лабораторной работы

Первым делом после загрузки сервера я создал TLS-сертификат для домена dmgeneralov.net.

```
# Задание
> 1. Сгенерируйте криптографический ключ и сертификат для домена dmgeneralov.net.
> 2. Настройте веб-сервер для работы с HTTPS.
> Напишите (или скорректируйте) скрипт загрузки виртуальной машины server (см. раздел 2.1).

# Выполнение лабораторной работы

# Выводы
Я получил опыт настройки сервера Apache.

# Контрольные вопросы

server: Clearing any previously set network configurations...
server: Preparing network interfaces based on configuration:
server: Adapter 1: nat
server: Adapter 2: intnet
server: Forwarding ports...
server: 22 (guest) => 2222 (host) (adapter 1)
server: Running 'pre-boot' VM customizations
server: Booting VM...
server: Waiting for machine to boot. This may take a minute or two...
server: SSH address: 127.0.0.1:2222
server: SSH username: vagrant
server: SSH auth method: password
server: Machine booted and ready!
[server] GuestAdditions 7.0.12 running --- OK.
server: Checking for guest additions in VM...
server: Setting hostname...
server: Configuring and enabling network interface: eth0
server: Mounting shared folders...
server: /vagrant => /home/danya/Documents/vagrant
server: Machine already provisioned. Run 'vagrant provision' to update the list.
server: Flag to force provisioning: Provisioning not enabled so will not configure.
server: Running provisioner: common hostnames...
server: Running: /tmp/vagrant-shell20231122-122222
anyaa@archlinux vagrant$
```

Рис. 3.1: openssl

Этот сертификат и приватный ключ после этого я указал в настройках виртуального хоста. Мы также настраиваем, чтобы виртуальный хост на HTTP перенаправлял на HTTPS-сайт.

```
GNU nano 5.6.1 /etc/httpd/conf.d/www.dmgeneralov.net.conf
<VirtualHost *:80>
    ServerAdmin webmaster@dmgeneralov.net
    DocumentRoot /var/www/html/www.dmgeneralov.net
    ServerName www.dmgeneralov.net
    ServerAlias www.dmgeneralov.net
    ErrorLog logs/www.dmgeneralov.net-error_log
    CustomLog logs/www.dmgeneralov.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@dmgeneralov.net
    DocumentRoot /var/www/html/www.dmgeneralov.net
    ServerName www.dmgeneralov.net
    ServerAlias www.dmgeneralov.net
    ErrorLog logs/www.dmgeneralov.net-error_log
    CustomLog logs/www.dmgeneralov.net-access_log common
    SSLCertificateFile /etc/ssl/certs/www.dmgeneralov.net.crt
    SSLCertificateKeyFile /etc/ssl/private/www.dmgeneralov.net.key
</VirtualHost>
</IfModule>
```

Рис. 3.2: apache

После этого (после того, как я вспомнил скопировать сертификат в другую папку), я разрешил доступ к HTTPS-порту в firewalld и перезагрузил веб-сервер.


```
The unit httpd.service has entered the 'failed' state with result 'exit-code'.
Nov 22 19:38:01 server.user.net systemd[1]: Failed to start The Apache HTTP Server.
Subject: A start job for unit httpd.service has failed
Defined-By: systemd
Support: https://access.redhat.com/support

A start job for unit httpd.service has finished with a failure.

The job identifier is 2426 and the job result is failed.
[root@server ~]# cp /etc/pki/tls/
cert.pem          certs/          ct_log_list.cnf  fips_local.cnf  misc/          openssl.cnf     private/
[root@server ~]# cp /etc/pki/tls/private/www.dmgeneralov.net.crt /etc/pki/tls/certs/
[root@server ~]# firewall-cmd --add-service=https
Warning: ALREADY_ENABLED: 'https' already in 'public'
success
[root@server ~]# firewall-cmd --add-service=https --permanent
Warning: ALREADY_ENABLED: https
success
[root@server ~]# firewall-cmd --reload
success
[root@server ~]# systemctl restart httpd
[root@server ~]#
```

Рис. 3.3: apache

Клиент замечает, что мы используем HTTPS, но пишет ошибку из-за того, что сертификат не подписан известным ему сертификационным центром. Мы игнорируем эту ошибку, используя принцип TOFU (trust on first use): мы ожидаем, что первое соединение не будет перехвачено, и поэтому мы сможем заметить, если оно будет перехвачено потом, потому что тогда сертификат будет другим (пусть и также неподписанным).

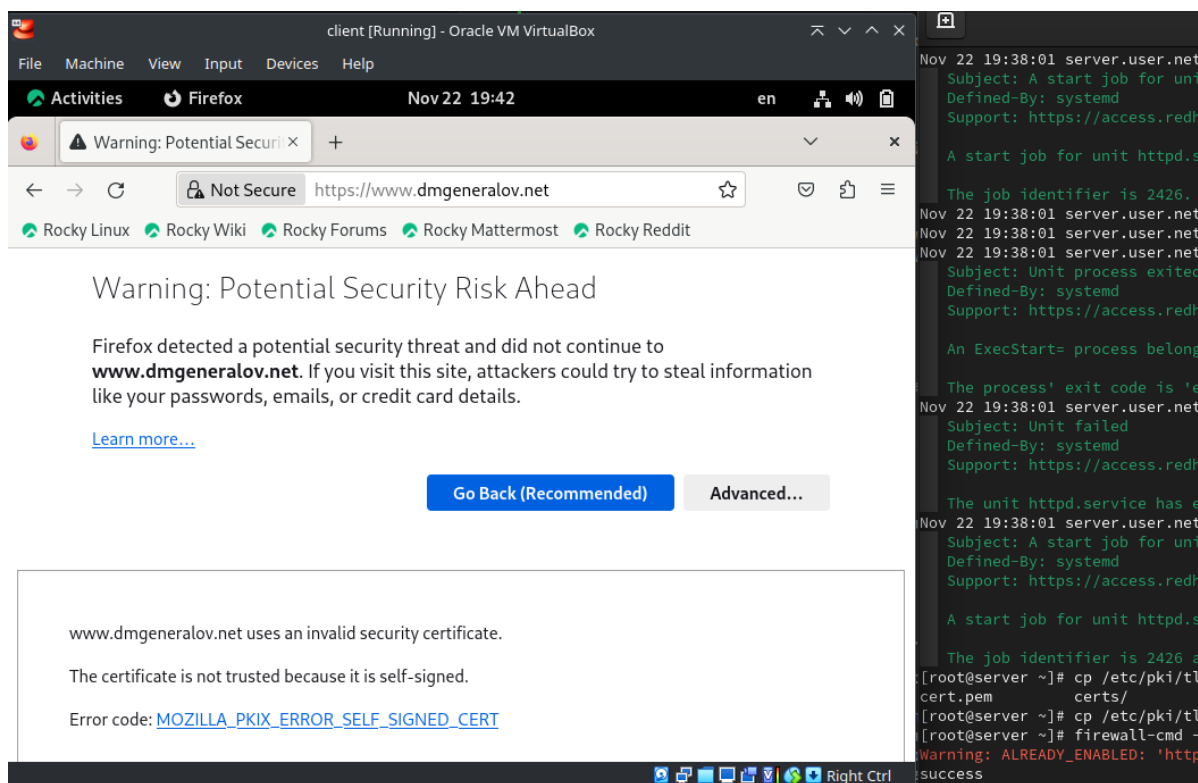


Рис. 3.4: firefox

Продолжив соединение, мы видим содержимое веб-страницы из лабораторной 4. Если посмотреть содержимое сертификата, то увидим, что он имеет все те параметры, как мы указали при создании.

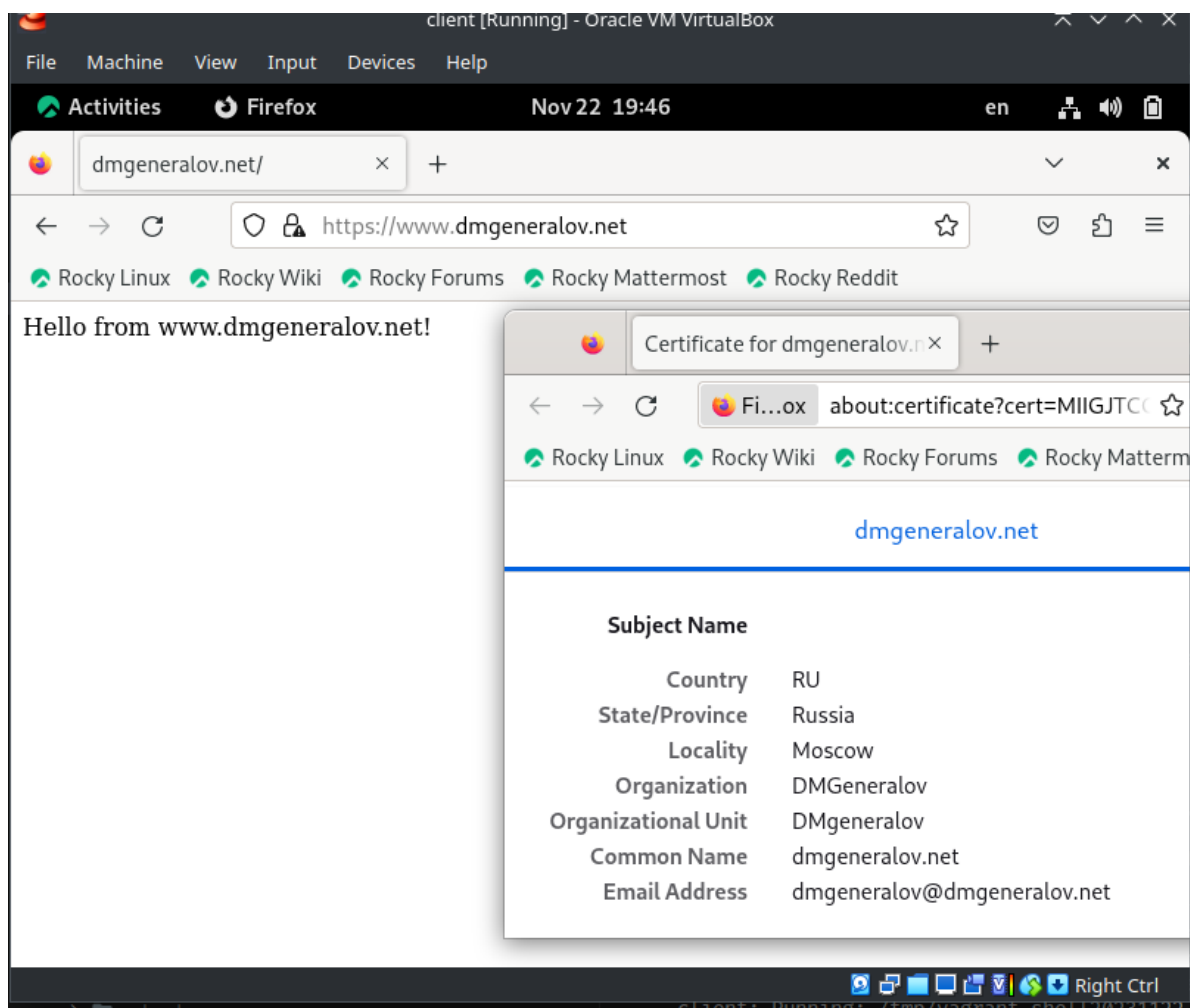


Рис. 3.5: firefox

Теперь мы настраиваем сервер, чтобы он использовал PHP. Для этого мы просто устанавливаем пакет php, а затем создаем файл index.php, и он автоматически будет использовать интерпретатор PHP, а не просто отправит содержимое страницы.

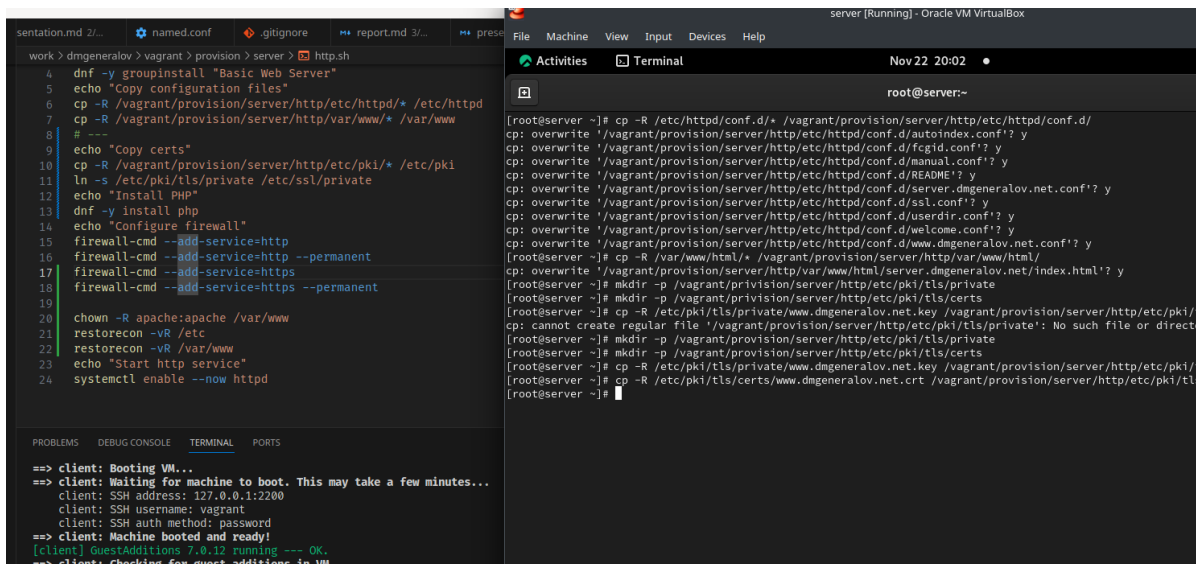


Рис. 3.7: vagrant

4 Выводы

Я получил опыт настройки сервера Apache, чтобы он использовал HTTPS и PHP.

5 Контрольные вопросы

1. В чём отличие HTTP от HTTPS?

HTTP – протокол общения с веб-сервером, а HTTPS – это тот же самый протокол поверх TLS-шифрования. Из-за этого, трафик HTTP можно перехватить, прочитать и изменить, в то время как HTTPS-трафик нельзя.

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

Прежде чем отправлять трафик по проводу, сервер шифрует его ключом, который он получил вместе с клиентом с помощью надежной схемы обмена ключами (например Diffie-Hellman); этот обмен ключами же может произойти только после того, как сервер докажет клиенту, что он обладает приватным ключом от сертификата, выданного доверенным сертификационным центром.

3. Что такое сертификационный центр? Приведите пример.

Сертификационный центр – это организация, которая занимается подписыванием сертификатов. Имея подписанный сертификат и соответствующий приватный ключ, веб-сервер может доказать клиенту, что он является тем, чье имя написано на сертификате, и если клиент доверяет сертификационному центру, то он транзитивно доверяет этому сертификату.

Например, Let's Encrypt – автоматизированный сертификационный центр. Чтобы получить у него сертификат на определенный домен, требуется разместить

на этом домене специальный файл с сгенерированным паролем. Успешное выполнение этого доказывает, что владелец домена – тот же самый сервер, который заказывает сертификат, и, когда Let's Encrypt успешно подтверждает существование и содержимое файла, то они подписывают ваш сертификат. После этого можно показывать этот сертификат клиентам, и, если они доверяют Let's Encrypt (или транзитивно, через Internet Security Research Group – корню этой цепочки сертификатов), то они могут быть уверены в безопасности соединения с этим сервером.