

Лабораторная работа №11

Администрирование сетевых подсистем

Машков Илья Евгеньевич

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
3.1 Запрет удалённого доступа по SSH для пользователя root	7
3.2 Ограничение списка пользователей для удалённого доступа по SSH	8
3.3 Настройка дополнительных портов для удалённого доступа по SSH	9
3.4 Настройка удалённого доступа по SSH по ключу	11
3.5 Организация туннелей SSH, перенаправление TCP-портов	12
3.6 Запуск графических приложений через SSH (X11Forwarding)	14
3.7 Внесение изменений в настройки внутреннего окружения виртуальной машины	15
4 Выводы	17
Список литературы	18

Список иллюстраций

3.1	Попытка обращения к серверу с помощью ssh	7
3.2	Запрет доступа для пользователя root	7
3.3	Перезапуск sshd	8
3.4	Попытка обращения к серверу	8
3.5	Редактирование sshd_config	8
3.6	Повторное редактирование sshd_config	9
3.7	Добавление организации через два порта	9
3.8	Перезапуск sshd и просмотр статуса	10
3.9	Логи sshd	10
3.10	Работа с метками безопасности, межсетевым экраном и пререзапуском службы	11
3.11	Повторное редактирование sshd_config	11
3.12	Генерация ssh-ключа	12
3.13	Обращение к серверу (и снова неудачное)	12
3.14	Просмотр процессов с протоколом TCP	13
3.15	Перенаправление портов	13
3.16	localhost:8080	14
3.17	Разрешение отображения граф. интерфейсов	14
3.18	sshd.sh	15
3.19	Редактирование Vagrantfile	16

Список таблиц

1 Цель работы

Приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

2 Задание

1. Настройте запрет удалённого доступа на сервер по SSH для пользователя `root`.
2. Настройте разрешение удалённого доступа к серверу по SSH только для пользователей группы `vagrant` и вашего пользователя.
3. Настройте удалённый доступ к серверу по SSH через порт 2022.
4. Настройте удалённый доступ к серверу по SSH по ключу.
5. Организуйте SSH-туннель с клиента на сервер, перенаправив локальное соединение с TCP-порта 80 на порт 8080.
6. Используя удалённое SSH-соединение, выполните с клиента несколько команд на сервере.
7. Используя удалённое SSH-соединение, запустите с клиента графическое приложение на сервере.
8. Напишите скрипт для Vagrant, фиксирующий действия по настройке SSH-сервера во внутреннем окружении виртуальной машины `server`. Соответствующим образом внесите изменения в `Vagrantfile`.

3 Выполнение лабораторной работы

3.1 Запрет удалённого доступа по SSH для пользователя root

Запускаю лог системных событий на сервере, а затем перехожу на клиент и пытаюсь получить доступ к серверу посредством ssh, но данная попытка терпит не удачу, т.к. client и server между собой никак не связаны (рис. [3.1]).

```
[user@client ~]$ ssh root@server.user.net
ssh: Could not resolve hostname server.user.net: Name or service not known
[user@client ~]$
```

Рис. 3.1: Попытка обращения к серверу с помощью ssh

В файле sshd_config запрещаю вход на сервер пользователю root (рис. [3.2]).

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Рис. 3.2: Запрет доступа для пользователя root

Перезапускаю службу sshd (рис. [3.3]).

```
[root@server ~]# systemctl restart sshd
[root@server ~]#
```

Рис. 3.3: Перезапуск sshd

Снова пытаюсь получить доступ к серверу, но, увы, неудачно (рис. [3.4]).

```
[user@client ~]$ ssh root@server
ssh: Could not resolve hostname server: Name or service not known
[user@client ~]$
```

Рис. 3.4: Попытка обращения к серверу

3.2 Ограничение списка пользователей для удалённого доступа по SSH

Снова открываю файл sshd_config и добавляю строку, показанную на скрине, а потом перезагружаю sshd (рис. [3.5]). Ожидаемо, что повторная отправка тоже закончилась неудачей.

```
#AllowAgentForwarding yes
#AllowTcpForwarding yes
AllowUsers vagrant
#GatewayPorts no
```

Рис. 3.5: Редактирование sshd_config

Снова редактирую sshd_config, добавляя к ранее внесённой строке имя пользователя (рис. [3.6]).

```
#AllowAgentForwarding yes  
#AllowTcpForwarding yes  
AllowUsers vagrant user  
#GatewayPorts no
```

Рис. 3.6: Повторное редактирование sshd_config

3.3 Настройка дополнительных портов для удалённого доступа по SSH

В файл sshd_config добавляю строки, позволяющие организовать весь процесс через два порта (рис. [3.7]).

```
#Port 22  
Port 22  
Port 2022  
#AddressFamily any  
#ListenAddress 0.0.0.0  
#ListenAddress ::
```

Рис. 3.7: Добавление организации через два порта

Перезапускаю службу sshd и просматриваю её статус. Вижу, что организацию через порт 2022 не получилось сделать (рис. [3.8]).

```
[root@server ~]# systemctl restart sshd
[root@server ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
    Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: en>
      Active: active (running) since Thu 2026-02-12 14:25:48 MSK; 16s ago
        Docs: man:sshd(8)
               man:sshd_config(5)
    Main PID: 8489 (sshd)
       Tasks: 1 (limit: 48821)
      Memory: 1.4M
         CPU: 18ms
      CGroup: /system.slice/sshd.service
              └─8489 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 12 14:25:48 server systemd[1]: Starting OpenSSH server daemon...
Feb 12 14:25:48 server sshd[8489]: error: Bind to port 2022 on 0.0.0.0 failed>
Feb 12 14:25:48 server systemd[1]: Started OpenSSH server daemon.
Feb 12 14:25:48 server sshd[8489]: error: Bind to port 2022 on :: failed: Per>
Feb 12 14:25:48 server sshd[8489]: Server listening on 0.0.0.0 port 22.
Feb 12 14:25:48 server sshd[8489]: Server listening on :: port 22.
...skipping...
```

Рис. 3.8: Перезапуск sshd и просмотр статуса

Смотрю логи и замечаю, что в доступе было отказано из-за службы SELinux (рис. [3.9]).

```
Feb 12 14:25:51 server setroubleshoot[8490]: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket
port 2022. For complete SELinux messages run: sealert -l 8744b70a-c277-42bf-b66a-004a14b1835e
Feb 12 14:25:51 server setroubleshoot[8490]: SELinux is preventing /usr/sbin/sshd from name_bind access on the tcp_socket
port 2022.
```

Рис. 3.9: Логи sshd

Исправляю метки SELinux относительно порта 2022, перенастраиваю межсетевой экран на работу с TCP-портом 2022, перезапускаю службу и просматриваю статус. В этот раз всё исправно (рис. [3.10]).

```
[root@server ~]# semanage port -a -t ssh_port_t -p tcp 2022
[root@server ~]# firewall-cmd --add-port=2022/tcp
success
[root@server ~]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@server ~]# systemctl restart sshd
[root@server ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
    Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
    Active: active (running) since Thu 2026-02-12 14:30:42 MSK; 19s ago
      Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 8541 (sshd)
     Tasks: 1 (limit: 48821)
    Memory: 1.4M
       CPU: 13ms
      CGroup: /system.slice/sshd.service
              └─8541 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 12 14:30:42 server systemd[1]: Starting OpenSSH server daemon...
Feb 12 14:30:42 server sshd[8541]: Server listening on 0.0.0.0 port 2022.
Feb 12 14:30:42 server sshd[8541]: Server listening on :: port 2022.
Feb 12 14:30:42 server sshd[8541]: Server listening on 0.0.0.0 port 22.
Feb 12 14:30:42 server sshd[8541]: Server listening on :: port 22.
Feb 12 14:30:42 server systemd[1]: Started OpenSSH server daemon.
```

Рис. 3.10: Работа с метками безопасности, межсетевым экраном и пререзапуск службы

3.4 Настройка удалённого доступа по SSH по ключу

В файле `sshd_config` добавляю строку, показанную на скрине (рис. [3.11]).

```
PubkeyAuthentication yes
```

Рис. 3.11: Повторное редактирование `sshd_config`

Генерирую ssh-ключ (рис. [3.12]).

```
[user@client ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa
Your public key has been saved in /home/user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vR2dIrLaudVXGRUM21e9BA1mCxymrIBkI1T3kUW6Lis user@client
The key's randomart image is:
+---[RSA 3072]---+
| .o= . .+o,+***=|
| + + ..+ o.+ +++|
| . . o o ooo|
| . o. . o+|
| oS o o o..|
| . o = o .|
| . . o o .|
| E oo o .|
| ... +.|
+---[SHA256]---+
[user@client ~]$ ssh-copy-id user@server.user.net
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: ERROR: ssh: Could not resolve hostname server.user.net:
Name or service not known

[user@client ~]$
```

Рис. 3.12: Генерация ssh-ключа

Отправка ключа и обращение к серверу окончились ничем (рис. [3.13]).

```
[user@client ~]$ ssh user@server.user.net
ssh: Could not resolve hostname server.user.net: Name or service not known
[user@client ~]$
```

Рис. 3.13: Обращение к серверу (и снова неудачное)

3.5 Организация туннелей SSH, перенаправление TCP-портов

На клиенте просматриваю активные службы с протоколом TCP (рис. [3.14]).

```
[user@client ~]$ lsof | grep TCP
firefox 6246 user 51u IPv4 43301
      0t0      TCP client:43264->93.243.107.34.bc.googleusercontent.com:htt
ps (ESTABLISHED)
firefox 6246 user 53u IPv4 43699
      0t0      TCP client:49708->146.75.117.91:https (ESTABLISHED)
firefox 6246 6268 gmain user 51u IPv4 43301
      0t0      TCP client:43264->93.243.107.34.bc.googleusercontent.com:htt
ps (ESTABLISHED)
firefox 6246 6268 gmain user 53u IPv4 43699
      0t0      TCP client:49708->146.75.117.91:https (ESTABLISHED)
firefox 6246 6269 gdbus user 53u IPv4 43301
      0t0      TCP client:43264->93.243.107.34.bc.googleusercontent.com:htt
ps (ESTABLISHED)
firefox 6246 6269 gdbus user 53u IPv4 43699
      0t0      TCP client:49708->146.75.117.91:https (ESTABLISHED)
firefox 6246 6273 glean.dis user 51u IPv4 43301
      0t0      TCP client:43264->93.243.107.34.bc.googleusercontent.com:htt
ps (ESTABLISHED)
```

Рис. 3.14: Просмотр процессов с портом TCP

Перенаправляю 80-ый порт на сервер server.user.net на порт 8080 на локальной машине (рис. [3.15]).

```
[user@client ~]$ ssh -fNL 8080:localhost:80 user@server.user.net
ssh: Could not resolve hostname server.user.net: Name or service not known
[user@client ~]$
```

Рис. 3.15: Перенаправление портов

Затем просматриваю список служб с тем же протоколом, но, т.к. нет связи клиента с сервером, ничего не поменялось. Потом перехожу на адрес localhost:8080 и вижу, что к такому адресу я обратиться не могу по вышеобозначенной причине (рис. [3.16]).

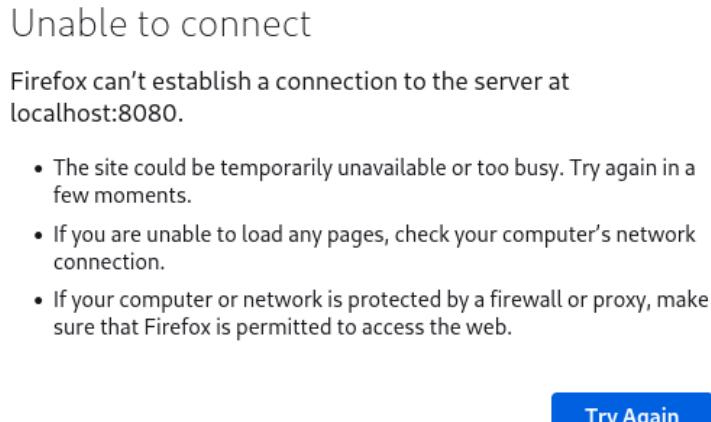


Рис. 3.16: localhost:8080

3.6 Запуск графических приложений через SSH (X11Forwarding)

В файле sshd_config разрешаю отображать на локальном клиентском компьютере графические интерфейсы X11 (рис. [3.17]).

```
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
```

Рис. 3.17: Разрешение отображения граф. интерфейсов

3.7 Внесение изменений в настройки внутреннего окружения виртуальной машины

Копирую все задействованные в этой лабе конфигурационные файлы в файлы внутреннего окружения машины server. Создаю файл sshd.sh, который будет повторять все ключевые моменты из этой лабы при запуске системы (рис. [3.18]).

```
#!/bin/bash
echo "Provisioning script $0"
echo "Copy configuration files"
cp -R /vagrant/provision/server/ssh/etc/* /etc
restorecon -vR /etc
echo "Configure firewall"
firewall-cmd --add-port=2022/tcp
firewall-cmd --add-port=2022/tcp --permanent
echo "Tuning SELinux"
semanage port -a -t ssh_port_t -p tcp 2022
echo "Restart sshd service"
systemctl restart sshd
```

Рис. 3.18: sshd.sh

Для отработки скрипта добавляю соответствующую запись в Vagrantfile (рис. [3.19]).

```
server.vm.provision "server ssh",
  type: "shell",
  preserve_order: true,
  path: "provision/server/ssh.sh"
```

Рис. 3.19: Редактирование Vagrantfile

4 Выводы

В процессе выполнения данной лабораторной работы я приобрёл практические навыки по настройке удалённого доступа к серверу с помощью SSH.

Список литературы

Администрирование сетевых подсистем