

Отчет по лабораторной работе 7

Генералов Даниил, НПИбд-01-21, 1032202280

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12
5	Контрольные вопросы	13

Список иллюстраций

3.1	firewalld	7
3.2	firewalld	8
3.3	firewalld	9
3.4	firewalld	10
3.5	firewalld	10
3.6	vagrant	11

Список таблиц

1 Цель работы

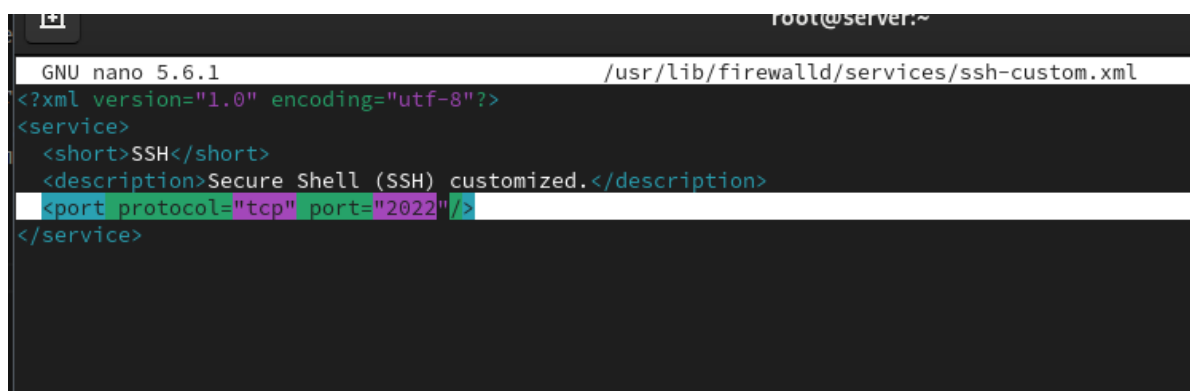
Получить навыки настройки межсетевого экрана в Linux в части переназначения портов и настройки Masquerading.

2 Задание

1. Настройте межсетевой экран виртуальной машины `server` для доступа к серверу по протоколу SSH не через 22-й порт, а через порт 2022 (см. разделы 7.4.1 и 7.4.2).
2. Настройте Port Forwarding на виртуальной машине `server` (см. разделы 7.4.3).
3. Настройте маскератинг на виртуальной машине `server` для организации доступа клиента к сети Интернет (см. раздел 7.4.3).
4. Напишите скрипт для Vagrant, фиксирующий действия по расширенной настройке межсетевого экрана. Соответствующим образом внести изменения в Vagrantfile (см. раздел 7.4.4).

3 Выполнение лабораторной работы

Сначала мы копируем файл с настройками службы SSH и изменяем его номер порта. В этом файле есть несколько полей, включая название и описание службы, а также список портов которые участвуют в службе.



```
root@server:~  
GNU nano 5.6.1 /usr/lib/firewalld/services/ssh-custom.xml  
<?xml version="1.0" encoding="utf-8"?>  
<service>  
  <short>SSH</short>  
  <description>Secure Shell (SSH) customized.</description>  
  <port protocol="tcp" port="2022"/>  
</service>
```

Рис. 3.1: firewalld

После этого можно перезагрузить firewalld и увидеть эту службу в списке, а затем добавить ее.

```

[vagrant@server ~]$ sudo -i
[root@server ~]# cp /usr/lib/firewalld/services/ssh.xml /usr/lib/firewalld/services/ssh-custom.xml
[root@server ~]# nano /usr/lib/firewalld/services/ssh-custom.xml
[root@server ~]# firewall-cmd --reload
success
[root@server ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula bacula-clien
t bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine checkmk-agent cockpit coll
ectd condor-collector cratedb ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-registry docker-swarm dropbox-lansync
elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication free
ipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availability http http3 https ident imap imaps ipfs
ipp ipp-client ipsec irc ircs iscsi-target isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell ku
be-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-no
deport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvi
rt-tls lightning-network llmnr llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms
smb mssql murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconso
le ovirt-vmconsole plex pmcd pmpoxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-d
hcp ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt-master sa
mba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spoti
fy-sync squid ssdp ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tft
p tile38 tinc tor-socks transmission-client upnp-client vdsms vnc-server wbem-http wbem-https wireguard ws-discovery ws-discovery-
client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-serv
er zerotier
[root@server ~]# firewall-cmd --add-service=ssh-custom
success
[root@server ~]# firewall-cmd --list-services
cockpit dhcpv6-client dns http https ssh ssh-custom
[root@server ~]#

```

Рис. 3.2: firewalld

Затем мы настраиваем переадресацию: соединения на порт 2022 перенаправляются на порт 22. Однако, по какой-то причине, эта переадресация не работает: порт 2022 остается закрытым.

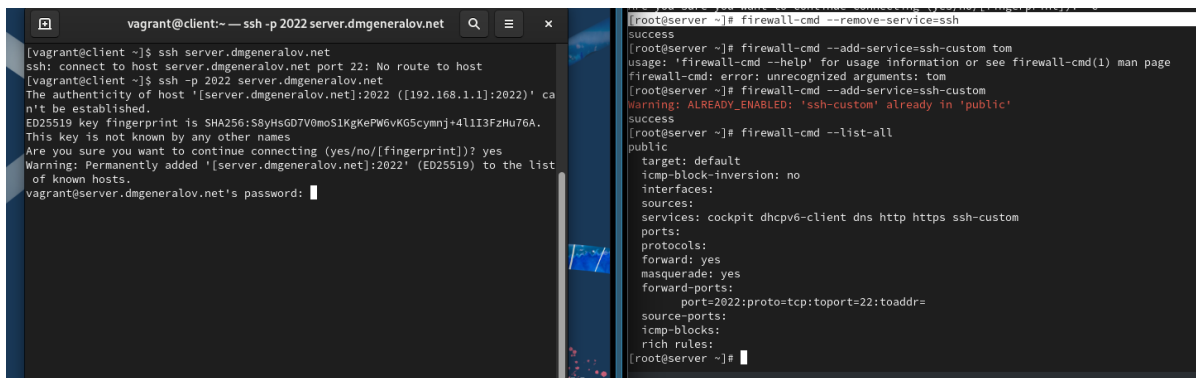

```

[root@server ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth1
  sources:
  services: cockpit dhcpv6-client dns http https ssh ssh-custom
  ports:
  protocols:
  forward: yes
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@server ~]# firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
success
[root@server ~]# ssh -p 2022 server.dmgeneralov.net
ssh: connect to host server.dmgeneralov.net port 2022: Connection refused
[root@server ~]# ssh -p 2022 localhost
ssh: connect to host localhost port 2022: Connection refused
[root@server ~]# sysctl -a | grep forward
net.ipv4.conf.all.bc_forwarding = 0
net.ipv4.conf.all.forwarding = 1
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.default.bc_forwarding = 0
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.eth0.bc_forwarding = 0
net.ipv4.conf.eth0.forwarding = 1
net.ipv4.conf.eth0.mc_forwarding = 0
net.ipv4.conf.eth1.bc_forwarding = 0
net.ipv4.conf.eth1.forwarding = 1
net.ipv4.conf.eth1.mc_forwarding = 0
net.ipv4.conf.lo.bc_forwarding = 0
net.ipv4.conf.lo.forwarding = 1
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.ip_forward = 1
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
net.ipv6.conf.all.forwarding = 0

```

Рис. 3.3: firewalld

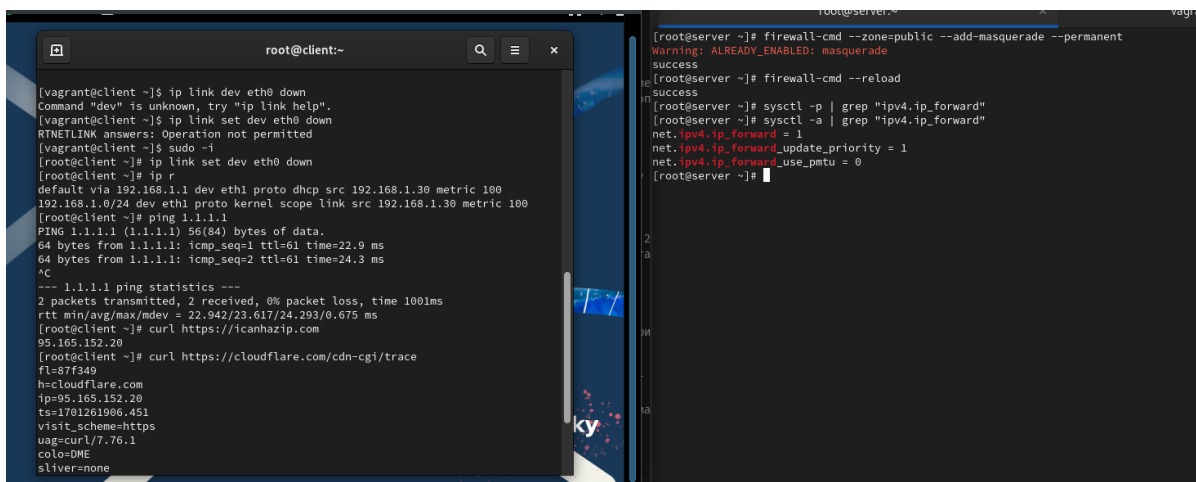
Но оказалось, что эти правила применяются только к трафику извне, например с клиента. Поэтому я удалил службу ssh, и теперь к серверу можно подключиться только через порт 2022. (После этого я восстановил это, потому что SSH используется для настройки виртуальной машины через Vagrant.)



The image shows two terminal windows. The left window is a client terminal (vagrant@client) with an SSH session to a server (server.dmgeneralov.net). It shows the SSH connection process, including fingerprint verification and password entry. The right window is a server terminal (root@server) showing the execution of firewall commands: `firewall-cmd --remove-service=ssh`, `firewall-cmd --add-service=ssh-custom`, and `firewall-cmd --list-all`. The output of `--list-all` shows the default target, blocked ICMP inversion, interfaces, sources, services (cockpit, dhcpv6-client, dns, http, https, ssh-custom), ports, protocols, forward settings, masquerade settings, and source ports.

Рис. 3.4: firewalld

После этого, когда masquerading аключен, а на клиенте выключен интерфейс eth0, мы все равно можем подключиться к интернету, но через сервер.



The image shows two terminal windows. The left window is a client terminal (root@client) showing the process of bringing down the `eth0` interface using `ip link set dev eth0 down`. It then shows the output of `ip netns exec` and `ping` commands, indicating successful connectivity to the internet. The right window is a server terminal (root@server) showing the execution of `firewall-cmd --zone=public --add-masquerade --permanent`, `firewall-cmd --reload`, and `sysctl` commands to enable IPv4 forwarding.

Рис. 3.5: firewalld

Наконец, можно экспортировать настройки в Vagrant.

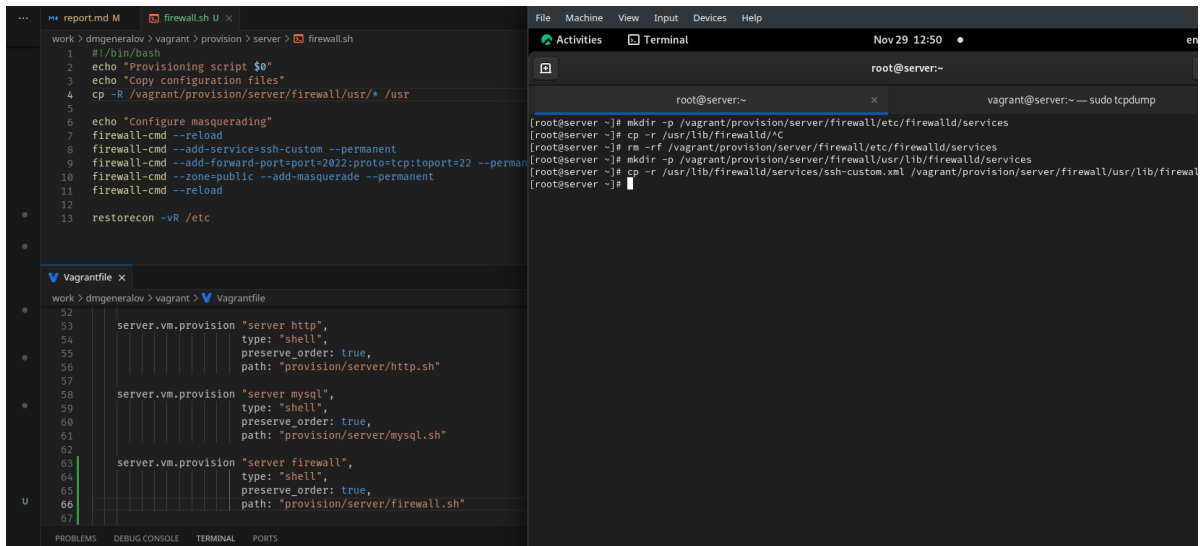


Рис. 3.6: vagrant

4 Выводы

Я получил опыт настройки port-forwarding и masquerading с помощью firewall.

5 Контрольные вопросы

1. Где хранятся пользовательские файлы firewalld?

Файлы служб можно найти в `/usr/lib/firewalld/services`

2. Какую строку надо включить в пользовательский файл службы, чтобы указать порт TCP 2022?

```
<port protocol="tcp" port="2022" />
```

3. Какая команда позволяет вам перечислить все службы, доступные в настоящее время на вашем сервере?

`firewall-cmd --get-services` показывает все службы, которые известны `firewalld`; `firewall-cmd --list-services` показывает те службы, которые сейчас активны.

4. В чем разница между трансляцией сетевых адресов (NAT) и маскарadingом (masquerading)?

NAT – это процесс изменения пакетов, проходящих через роутер, чтобы IP-адрес пакета был другим. Source NAT – это выполнение этой операции для пакетов, которые идут из внутренней сети во внешнюю, а Destination NAT – это аналогичный процесс для пакетов из внешней сети во внутреннюю. Masquerading – это особая форма Source NAT, при которой роутер сам определяет, какой IP-адрес прописать в исходящем пакете; это позволяет динамически использовать IP-адрес внешнего интерфейса роутера.

5. Какая команда разрешает входящий трафик на порт 4404 и перенаправляет его в службу ssh по IP-адресу 10.0.0.10?

```
firewall-cmd --add-forward-port=port=4404:proto=tcp:toaddr=10.0.0.10:toport=22
```

6. Какая команда используется для включения маскарадинга IP-пакетов для всех пакетов, выходящих в зону public?

```
firewall-cmd --add-masquerade
```