

Лабораторная работа 7

Генералов Даниил, НПИбд-01-21, 1032202280

2023

¹RUDN University, Moscow, Russian Federation

Задача

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).

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).

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 маскрадинг на виртуальной машине server для организации доступа клиента к сети Интернет (см. раздел 7.4.3).

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 маскерадинг 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).

Выполнение

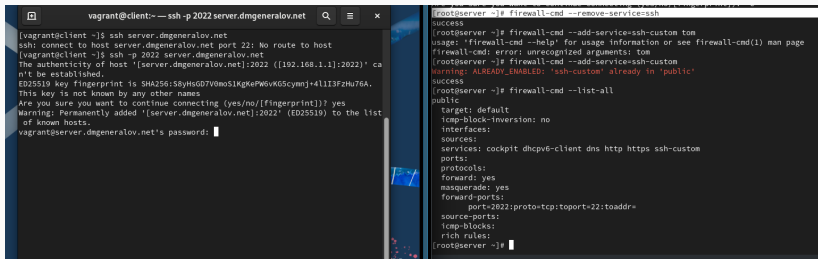
```
root@server:~  
GNU nano 5.6.1 /usr/lib/firewalld/services/ssh-custom.xml  
<?xml version="1.0" encoding="utf-8"?>  
<service>  
  <short>SSH</short>  
  <description>Secure Shell (SSH) customized.</description>  
  <port protocol="tcp" port="2022"/>  
</service>
```

Рис. 1: firewalld


```
[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
sql-mwbt 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 pmproxy pmwebapi pmwebapis pop3 pop3s postgresql proxyoxy prometheus prometheus-node-exporter proxy-d
hcp ps3netstrv 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 smtp smtp-submission smtps snmp snmpv1-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 vdsml 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. 2: firewalld

```
[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.forwarding = 1
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
net.ipv6.conf.all.forwarding = 0
```

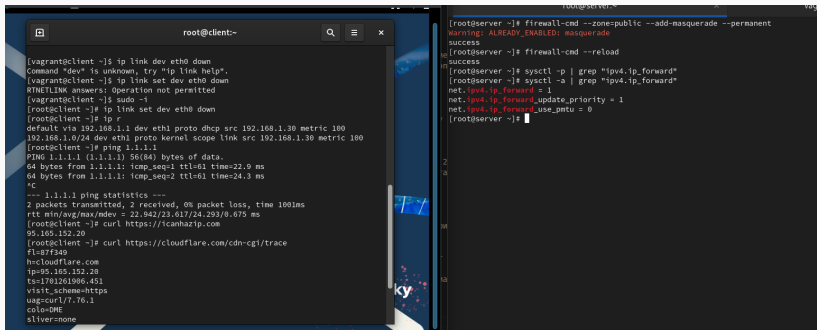


```
vagrant@client:~ -- ssh -p 2022 server.dmgeneralov.net

[vagrant@client ~]$ ssh server.dmgeneralov.net
ssh: connect to host server.dmgeneralov.net port 22: No route to host
[vagrant@client ~]$ ssh -p 2022 server.dmgeneralov.net
The authenticity of host '[server.dmgeneralov.net]:2022 ([192.168.1.1]:2022)' ca
n't be established.
EO25519 key fingerprint is SHA256:58yHsGD7V0moS1KgKePW6vK65cywnj+4lI3FzHu76A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[server.dmgeneralov.net]:2022' (EO25519) to the list
of known hosts.
vagrant@server.dmgeneralov.net's password:

[root@server ~]# firewall-cmd --remove-service=ssh
success
[root@server ~]# firewall-cmd --add-service=ssh-custom tom
usage: 'firewall-cmd --help' for usage information or see firewall-cmd(1) man page
firewall-cmd: error: unrecognized arguments: tom
[root@server ~]# firewall-cmd --add-service=ssh-custom
Warning: ALREADY_ENABLED: 'ssh-custom' already in 'public'
success
[root@server ~]# firewall-cmd --list-all
public
target: default
icmp-block-inversion: no
interfaces:
sources:
services: cockpit dhcpv6-client dns http https ssh-custom
ports:
protocols:
forward: yes
masquerade: yes
forward-ports:
  port=2022:proto=tcp:toport=22:toaddr=
source-ports:
icmp-blocks:
rich rules:
[root@server ~]#
```

Рис. 4: firewalld



The image shows two terminal windows side-by-side. The left window is titled 'root@client:~' and shows a series of commands and their outputs. The right window is titled 'root@server:~' and shows commands to configure the firewall on the server.

```
root@client:~  
[vagrant@client ~]$ ip link set dev eth0 down  
Command "dev" is unknown, try "ip link help".  
[vagrant@client ~]$ ip link set dev eth0 down  
RTNETLINK answers: Operation not permitted  
[vagrant@client ~]$ sudo -i  
[root@client ~]# ip link set dev eth0 down  
[root@client ~]# ip r  
default via 192.168.1.1 dev eth1 proto dhcp src 192.168.1.30 metric 100  
192.168.1.0/24 dev eth1 proto kernel scope link src 192.168.1.30 metric 100  
[root@client ~]# ping 1.1.1.1  
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data:  
64 bytes from 1.1.1.1: icmp_seq=1 ttl=61 time=22.9 ms  
64 bytes from 1.1.1.1: icmp_seq=2 ttl=61 time=24.3 ms  
^C  
--- 1.1.1.1 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 100ms  
rtt min/avg/max/mdev = 22.942/23.617/24.293/0.675 ms  
[root@client ~]# curl https://icanhazip.com  
95.165.152.20  
[root@client ~]# curl https://cloudflare.com/cdn-cgi/trace  
fl=87f349  
h=cloudflare.com  
ip=95.165.152.20  
ts=1701261906.451  
visit_scheme=https  
uag=curl/7.76.1  
colo=DME  
sliver=none
```

```
root@server:~  
[root@server ~]# firewall-cmd --zone=public --add-masquerade --permanent  
Warning: ALREADY_ENABLED: masquerade  
success  
[root@server ~]# firewall-cmd --reload  
success  
[root@server ~]# systemctl -p | grep "ipv4.ip_forward"  
[root@server ~]# systemctl -a | grep "ipv4.ip_forward"  
net.ipv4.ip_forward = 1  
net.ipv4.ip_forward_update_priority = 1  
net.ipv4.ip_forward_use_pmtu = 0  
[root@server ~]#
```

Рис. 5: firewalld

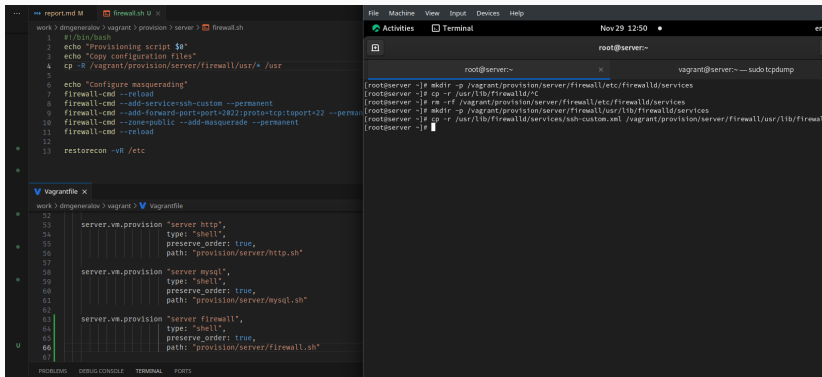


Рис. 6: vagrant

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