

Лабораторная работа №7

Расширенные настройки межсетевого экрана

Шубина София Антоновна

2 октября 2025

Российский университет дружбы народов

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

1. Настроить межсетевой экран виртуальной машины server для доступа к серверу по протоколу SSH не через 22-й порт, а через порт 2022.
2. Настроить Port Forwarding на виртуальной машине server.
3. Настроить маскарading на виртуальной машине server для организации доступа клиента к сети Интернет.
4. Написать скрипт для Vagrant, фиксирующий действия по расширенной настройке межсетевого экрана. Соответствующим образом внести изменения в Vagrantfile.

Выполнение лабораторной работы

Перейдем в режим суперпользователя

```
[sashubina@server.sashubina.net ~]$ sudo -i  
[sudo] password for sashubina:  
[root@server.sashubina.net ~]#
```

Рис. 1: sudo -i

На основе существующего файла описания службы ssh создадим файл с собственным описанием и посмотрим содержимое файла службы.

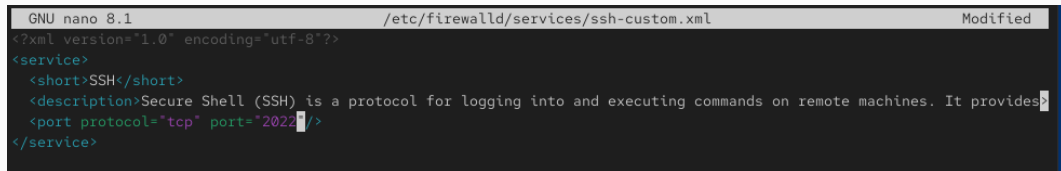
```
[root@server.sashubina.net ~]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml  
[root@server.sashubina.net ~]# cd /etc/firewalld/services/  
[root@server.sashubina.net services]#
```

Рис. 2: Создание файла с собственным описанием

```
[root@server.sashubina.net services]# cat /etc/firewalld/services/ssh-custom.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote machines. It provides
secure encrypted communications. If you plan on accessing your machine remotely via SSH over a firewalled interface, enable this option. You need the openssh-server package installed for this option to be useful.</description>
  <port protocol="tcp" port="22"/>
</service>
[root@server.sashubina.net services]#
```

Рис. 3: просмотр файла

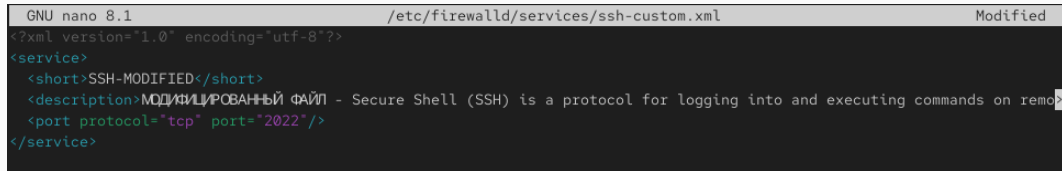
Откроем файл описания службы на редактирование и заменим порт 22 на новый порт (2022):



```
GNU nano 8.1 /etc/firewalld/services/ssh-custom.xml Modified
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote machines. It provides>
  <port protocol="tcp" port="2022"/>
</service>
```

Рис. 4: просмотр файла

В этом же файле скорректируем описание службы для демонстрации, что это модифицированный файл службы.



```
GNU nano 8.1 /etc/firewalld/services/ssh-custom.xml Modified
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH-MODIFIED</short>
  <description>МОДИФИЦИРОВАННЫЙ ФАЙЛ - Secure Shell (SSH) is a protocol for logging into and executing commands on remote systems</description>
  <port protocol="tcp" port="2022"/>
</service>
```

Рис. 5: Отредактированный файл описания службы

Просмотрим список доступных FirewallD служб:

```
[root@server.sashubina.net services]# firewall-cmd --get-services
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1602 anno-1800 apcups
d aseqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin
-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent civilization-
iv civilization-v cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client
distcc dns dns-over-https dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-serve
r factorio finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galer
a ganglia-client ganglia-master git gssd grafana gre high-availability http http3 https ident imap imaps iperf2 iperf3 i
pfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshel
l kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-s
ecure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ld
ap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache m
inecraft minidlna mndp mongodb mosh mountd mpd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula need-for-speed-most-wa
nted netbios-ns netdata-dashboard nfs nfs3 nmap-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconso
le ovirt-vmconsole plex pmcd pmpoxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporte
r proxy-dhcp ps2link ps3netrvr ptp pulseaudio puppetmaster quassel radius radsec rdp redis redis-sentinel rootd rpc-bind
rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane settlers-history-collection sip sips slimevr slp s
mtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid sssd ssh statshv steam
-lan-transfer steam-streaming stellaris stronghold-crusader stun stuns submission supertuxkart svdrp svn syncthing synct
hing-gui syncthing-relay synergy syscomlan syslog syslog-tls telnet tentacle terraria tftpd tile38 tinc tor-socks transmi
ssion-client turn turns upnp-client vdsd vnc-server vrrp warpinator wbem-http wbem-https wireguard ws-discovery ws-disco
very-client ws-discovery-host ws-discovery-tcp ws-discovery-udp wsdd wsdd-http wsman wsmans xdmcp xmpp-bosh xmpp-client
xmpp-local xmpp-server zabbix-agent zabbix-java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.sashubina.net services]#
```

Рис. 6: Список доступных FirewallD служб

Добавим новую службу в FirewallD и выведем на экран список активных служб:

```
[root@server.sashubina.net services]# firewall-cmd --reload
success
[root@server.sashubina.net services]# firewall-cmd --get-services
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1602 anno-1800 apcups
d aseqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin
-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent civilization-
iv civilization-v cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client
distcc dns dns-over-quick dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-serve
r factorio finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galer
a ganglia-client ganglia-master git gssd grafana gre high-availability http http3 https ident imap imaps iperf2 iperf3 i
pfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshel
l kube-api kube-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-s
ecure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ld
ap ldaps libvirt libvirt-tls lightning-network llmn llmn-client llmn-tcp llmn-udp managesieve matrix ndns nmapcache n
inecraft minidlna mndp mongodb mosh mountd mpd mqtt mqtt-tls ms-wbt mssql muzmur mysql nbd nebula need-for-speed-most-wa
nted netbios-ns netdata-dashboard nfs nfs3 nmap-0183 nripe ntp nut opentelemetry openvpn ovirt-imagedio ovirt-storageconso
le ovirt-vmconsole plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporte
r proxy-dhcp ps2link ps3netshr ptp pulseaudio puppetmaster quassel radius radsec rdp redis redis-sentinel rootd rpc-bind
rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane settlers-history-collection sip sips slimevr slp s
ntp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh ssh-custom st
atsrv steam-lan-transfer steam-streaming stellaris stronghold-crusader stun stuns submission supertuxkart svdrp svn sync
thing syncthing-gui syncthing-relay synergy sysconlan syslog syslog-tls telnet tentacle terraria tftp tile38 tinc tor-so
cks transmission-client turn turns upnp-client vdsim vnc-server vrtp warpinator wben-http wben-https wireguard ws-discover
y ws-discovery-client ws-discovery-host ws-discovery-tcp ws-discovery-udp wsd wsd-http wsmn wsmans xdmcp xmpp-bosh x
mpp-client xmpp-local xmpp-server zabbix-agent zabbix-java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-
k zerotier
[root@server.sashubina.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh
[root@server.sashubina.net services]#
```

Рис. 7: Список FirewallD служб и добавление новой службы в FirewallD

Организуем на сервере переадресацию с порта 2022 на порт 22:

```
[root@server.sashubina.net services]# firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22  
success  
[root@server.sashubina.net services]#
```

Рис. 8: с порта 2022 на порт 22

На клиенте попробуем получить доступ по SSH к серверу через порт 2022:

```
[sashubina@client.sashubina.net ~]$ ssh -p 2022 sashubina@server.sashubina.net
The authenticity of host '[server.sashubina.net]:2022 ([192.168.1.100]:2022)' can't be established.
ED25519 key fingerprint is SHA256:LJpSqM044iVZ+0Xnopuu/dN//5izIhYfJcVSZXbg00k.
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.sashubina.net]:2022' (ED25519) to the list of known hosts.
Connection closed by 192.168.1.100 port 2022
[sashubina@client.sashubina.net ~]$
```

Рис. 9: Переадресация и получение доступа по SSH

Настройка Port Forwarding и Masquerading

На сервере посмотрим, активирована ли в ядре системы возможность перенаправления IPv4-пакетов пакетов:

```
[root@server.sashubina.net services]# sysctl -a | grep forward
net.ipv4.conf.all.bc_forwarding = 0
net.ipv4.conf.all.forwarding = 0
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.default.bc_forwarding = 0
net.ipv4.conf.default.forwarding = 0
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.eth0.bc_forwarding = 0
net.ipv4.conf.eth0.forwarding = 0
net.ipv4.conf.eth0.mc_forwarding = 0
net.ipv4.conf.eth1.bc_forwarding = 0
net.ipv4.conf.eth1.forwarding = 0
net.ipv4.conf.eth1.mc_forwarding = 0
net.ipv4.conf.lo.bc_forwarding = 0
net.ipv4.conf.lo.forwarding = 0
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.ip_forward = 0
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
net.ipv6.conf.all.forwarding = 0
net.ipv6.conf.all.mc_forwarding = 0
net.ipv6.conf.default.forwarding = 0
net.ipv6.conf.default.mc_forwarding = 0
net.ipv6.conf.eth0.forwarding = 0
net.ipv6.conf.eth0.mc_forwarding = 0
net.ipv6.conf.eth1.forwarding = 0
net.ipv6.conf.eth1.mc_forwarding = 0
net.ipv6.conf.lo.forwarding = 0
net.ipv6.conf.lo.mc_forwarding = 0
[root@server.sashubina.net services]#
```

Рис. 10: Проверка активации перенаправления IPv4-пакетов

Включим перенаправление IPv4-пакетов на сервере. Включим маскарадинг на сервере и перезапустим систему:

```
[root@server.sashubina.net services]# echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
[root@server.sashubina.net services]# sysctl -p /etc/sysctl.d/90-forward.conf
net.ipv4.ip_forward = 1
[root@server.sashubina.net services]#
```

Рис. 11: Включение перенаправление IPv4-пакетов и маскарадинга на сервере

Включим маскарадинг на сервере:

```
[root@server.sashubina.net services]# firewall-cmd --zone=public --add-masquerade --permanent  
success  
[root@server.sashubina.net services]# firewall-cmd --reload  
success  
[root@server.sashubina.net services]#
```

Рис. 12: firewall-cmd--zone=public--add-masquerade--permanent и firewall-cmd--reload

На клиенте проверим доступность выхода в Интернет.

```
[sashubina@client.sashubina.net ~]$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=255 time=29.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=255 time=19.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=255 time=23.6 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 19.917/24.275/29.341/3.879 ms
[sashubina@client.sashubina.net ~]$
```

Рис. 13: Проверка доступности выхода в Интернет

На клиенте проверим доступность выхода в Интернет.

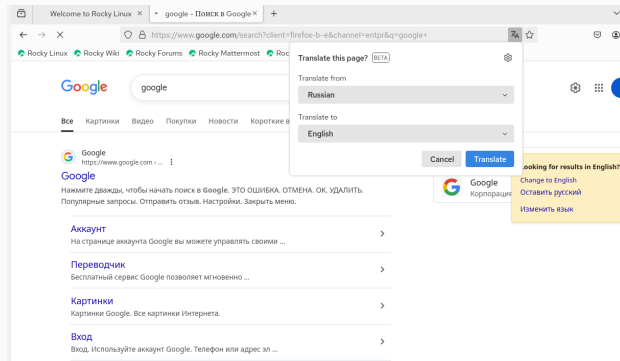


Рис. 14: Проверка доступности выхода в Интернет

Внесение изменений в настройки внутреннего окружения виртуальной машины

Внесения изменений в настройки внутреннего окружения

```
[root@server.sashubina.net services]# cd /vagrant/provision/server
[root@server.sashubina.net server]# mkdir -p /vagrant/provision/server/firewall/etc/firewalld/services
[root@server.sashubina.net server]# mkdir -p /vagrant/provision/server/firewall/etc/sysctl.d
[root@server.sashubina.net server]# cp -r /etc/firewalld/services/ssh-custom.xml /vagrant/provision/server/firewall/etc/
firewalld/services/
[root@server.sashubina.net server]# cp -r /etc/sysctl.d/90-forward.conf /vagrant/provision/server/firewall/etc/sysctl.d/
[root@server.sashubina.net server]#
```

Рис. 15: Внесения изменений в настройки внутреннего окружения

создадим файл `firewall.sh`.

```
[root@server.sashubina.net server]# cd /vagrant/provision/server  
[root@server.sashubina.net server]# touch firewall.sh  
[root@server.sashubina.net server]# chmod +x firewall.sh
```

Рис. 16: Внесения изменений в настройки внутреннего окружения

Открыв его на редактирование, пропишите в нём следующий скрипт:

```
GNU nano 8.1                               firewall.sh
#!/bin/bash
echo "Provisioning script $0"
echo "Copy configuration files"
cp -R /vagrant/provision/server/firewall/etc/* /etc
echo "Configure masquerading"
firewall-cmd --add-service=ssh-custom --permanent
firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22 --permanent
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload
restorecon -vR /etc
```

Рис. 17: Редактирование файла

в конфигурационном файле Vagrantfile необходимо добавить в разделе конфигурации для сервера:

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

Рис. 18: Редактирование файла

В процессе выполнения данной лабораторной работы я получила навыки настройки межсетевого экрана в Linux в части переадресации портов и настройки Masquerading.

Контрольные вопросы

1. Где хранятся пользовательские файлы firewalld?

`/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
```

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

При маскарadingе вместо адреса отправителя(как делается это в NAT) динамически подставляется адрес назначенного интерфейса (сетевой адрес + порт).

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

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

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

```
firewall-cmd --zone=public --add-masquerade --permanent
```