

Отчёт о лабораторной работе

Лабораторная работа №13

Приходько Иван Иванович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	14
5	Ответы на контрольные вопросы	15

Список иллюстраций

3.1	Все доступные службы в текущей зоне	7
3.2	Все доступные службы в текущей зоне	8
3.3	Добавление VNC в конфигурацию брандмауэр	9
3.4	Проверка VNC	10
3.5	Проверка VNC на перманентное добавление	11
3.6	Запуск графического меню firewall	11
3.7	Добавление сервисов	12
3.8	Проверка сервисов	12
3.9	Проверка сервисов после самостоятельной работы	13

Список таблиц

1 Цель работы

Получить навыки настройки пакетного фильтра в Linux.

2 Задание

Поработать с пакетным фильтром в Linux.

3 Выполнение лабораторной работы

Для начала посмотрим все доступные службы в текущей зоне (рис. [3.1]).

```
[root@ivanprihodko ~]# firewall-cmd --get-default-zone
public
[root@ivanprihodko ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
[root@ivanprihodko ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcup
sd audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storag
e bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph
-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-collector cratedb ctdb
dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-re
gistry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman
foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp
galera ganglia-client ganglia-master git gpsd grafana gre high-availability http http3 ht
tps ident imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin
kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-contr
ol-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure
kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-
readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-clie
nt llmnr-tcp llmnr-udp managesieve matrix mdns memcache minidlna mongodb mosh mounstd mqtt
mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea
-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconso
le plex pmcd pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus promethe
us-node-exporter proxy-dhcp ps2link ps3netsh ptp pulseaudio puppetmaster quassel radius
rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-c
lient samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snm
ptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing s
yncthing-gui syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc t
or-socks transmission-client upnp-client vdsml vnc-server warpinator wbm-http wbm-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-server zerotier
[root@ivanprihodko ~]#
```

Рис. 3.1: Все доступные службы в текущей зоне

Выведем немного по-другому доступные службы (рис. [3.2]).

```

[root@ivanprihodko ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
[root@ivanprihodko ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@ivanprihodko ~]# firewall-cmd --list-all --zone=public
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
@ivanprihodko ~]#

```

Рис. 3.2: Все доступные службы в текущей зоне

Добавим VNC в конфигурацию брэндмауэр и проверим добавился ли он(рис. [3.3]).


```

[root@ivanprihodko ~]# firewall-cmd --add-service=vnc-server
success
[root@ivanprihodko ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh vnc-server
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@ivanprihodko ~]# systemctl restart firewalld
[root@ivanprihodko ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@ivanprihodko ~]# firewall-cmd --add-service=vnc-server --permanent
ss
[root@ivanprihodko ~]#

```

Рис. 3.3: Добавление VNC в конфигурацию брэндмауэр

Только после перезагрузки он появился (рис. [3.4]).

```
[root@ivanprihodko ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@ivanprihodko ~]# firewall-cmd --reload
success
[root@ivanprihodko ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh vnc-server
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@ivanprihodko ~]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@ivanprihodko ~]#
```

Рис. 3.4: Проверка VNC

Проверим добавился ли VNC после перманентного добавления (рис. [3.5]).

```
[root@ivanprihodko ~]# firewall-cmd --reload
success
[root@ivanprihodko ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh vnc-server
  ports: 2022/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@ivanprihodko ~]# su - ivanprihodko
ivanprihodko@ivanprihodko ~]$
```

Рис. 3.5: Проверка VNC на перманентное добавление

Теперь попробуем запустить графическое меню firewall (рис. [3.6]).

```
[root@ivanprihodko ~]# su - ivanprihodko
[ivanprihodko@ivanprihodko ~]$ firewall-config
bash: firewall-config: команда не найдена...
Установить пакет «firewall-config», предоставляющий команду «firewall-config»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов...
Следующие пакеты должны быть установлены:
dbus-x11-1:1.12.20-8.el9.x86_64      X11-requiring add-ons for D-BUS
firewall-config-1.3.4-15.el9_6.noarch Firewall configuration application
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

(firewall-config:4905): Gtk-WARNING **: 20:55:14.488: cannot open display:
[ivanprihodko@ivanprihodko ~]$ firewall-config
(firewall-config:5151): Gtk-WARNING **: 20:55:19.817: cannot open display:
[ivanprihodko@ivanprihodko ~]$ sudo firewall-config
(firewall-config:5160): Gtk-WARNING **: 20:55:29.798: cannot open display:
ivanprihodko@ivanprihodko ~]$
```

Рис. 3.6: Запуск графического меню firewall

Через графическое меню добавляем сервисы, что просят в задании (рис. [3.7]).

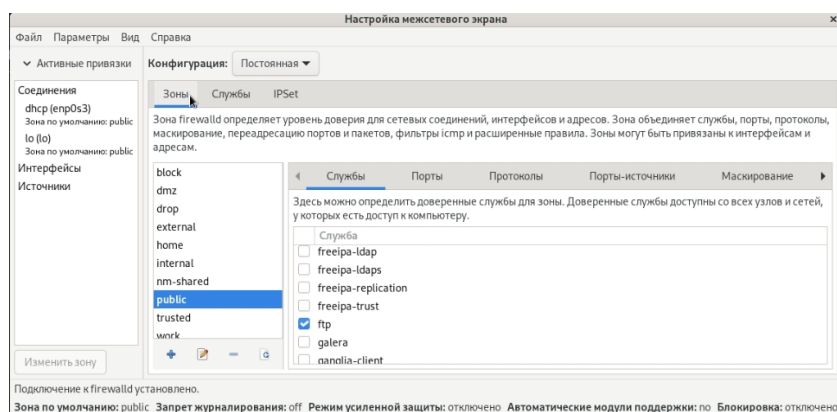


Рис. 3.7: Добавление сервисов

Проверим добавились ли они (рис. [3.8]).

```
[ivanprihodko@ivanprihodko ~]$ firewall-config
[ivanprihodko@ivanprihodko ~]$ firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh vnc-server
  ports: 2022/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[ivanprihodko@ivanprihodko ~]$ firewall-cmd --reload
success
[ivanprihodko@ivanprihodko ~]$ firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ftp http https ssh vnc-server
  ports: 2022/tcp 2022/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[ivanprihodko@ivanprihodko ~]$
```

Рис. 3.8: Проверка сервисов

После перезагрузки компьютера проверим, были добавлены изменения после

самостоятельной работы (рис. [3.9]).

```
[ivanprihodko@ivanprihodko ~]$ firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ftp http https ssh vnc-server
  ports: 2022/tcp 2022/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[ivanprihodko@ivanprihodko ~]$ firewall-cmd --reload
success
[ivanprihodko@ivanprihodko ~]$ firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ftp http https imap pop3 smtp ssh vnc-server
  ports: 2022/tcp 2022/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[ivanprihodko@ivanprihodko ~]$
```

Рис. 3.9: Проверка сервисов после самостоятельной работы

4 Выводы

В ходе данной лабораторной работы были получены навыки для настройки пакетного фильтра в Linux.

5 Ответы на контрольные вопросы

1. `firewalld`
2. `firewall-cmd --zone=public --add-port=2355/udp --permanent`
3. `firewall-cmd --list-all`
4. `firewall-cmd --remove-service=vnc-server --permanent`
5. `firewall-cmd --reload`
6. `firewall-cmd --zone=public --list-interfaces` или `firewall-cmd --list-all`
7. `firewall-cmd --zone=public --add-interface=en01 --permanent`
8. в зону по умолчанию (обычно `public`)