

Лабораторная работа №13

Отчет

Славинский Владислав Вадимович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15
4	Ответы на контрольные вопросы	16

Список иллюстраций

2.1	Определение текущей зоны по умолчанию	6
2.2	Определение доступных зон	6
2.3	Доступные службы	7
2.4	Доступные службы в текущей зоне	7
2.5	Сравнение вывода информации	8
2.6	Добавление VNC в брандмауэр	8
2.7	Проверка	9
2.8	Перезапуск службы	9
2.9	Проверка	10
2.10	Добавление службы в постоянную	10
2.11	Проверка	10
2.12	Перезагрузка конфигурации	11
2.13	Добавление порта	11
2.14	Запуск интерфейса GUI	12
2.15	Изменение параметров	12
2.16	Вывод информации	13
2.17	Перезагрузка	13
2.18	Создание конфигурации	13
2.19	Добавление сервисов в графическом интерфейсе	14
2.20	Перезагрузка конфигурации и проверка	14

Список таблиц

1 Цель работы

Получить навыки настройки пакетного фильтра в Linux.

2 Выполнение лабораторной работы

В терминале получим права администратора, определим текущую зону по умолчанию, введя: `firewall-cmd --get-default-zone` (рис. 2.1)

```
[slavinskiyv@slavinskiyv ~]$ su -  
Password:  
[root@slavinskiyv ~]# firewall-cmd --get-default-zone  
public  
[root@slavinskiyv ~]#
```

Рис. 2.1: Определение текущей зоны по умолчанию

Определим доступные зоны с помощью `firewall-cmd --get-zones`. (рис. 2.2)

```
public  
[root@slavinskiyv ~]# firewall-cmd --get-zones  
block dmz drop external home internal nm-shared public trusted work  
[root@slavinskiyv ~]#
```

Рис. 2.2: Определение доступных зон

Посмотрим службы, доступные на компьютере, используя `firewall-cmd --get-services`. (рис. 2.3)

```

[root@slavinskiyvv ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcups
d 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-ex
porter 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-registr
y docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman forema
n-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 https iden
t imap imaps ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnec
t kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver kube-control-plane k
ube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodep
ort-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kub
elet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp
llmnr-udp managiesicve matrix mdns memcache minidlna mongodb mosh mounid mqtt mqtt-tls ms-w
bt mssql murmur mysql nbd nebula netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp
nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pncd pmp
proxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter
proxy dhcp ps2link ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-se
ntinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane
sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lans
ync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthin
g-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-
client upnp-client vdsn vnc-server warpinator wbem-http wbem-https wireguard ws-discovery
ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-cl
ient xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@slavinskiyvv ~]#

```

Рис. 2.3: Доступные службы

Определим доступные службы в текущей зоне: `firewall-cmd --list-services`. (рис. 2.4)

```

[root@slavinskiyvv ~]# firewall-cmd --list-services
cockpit dhcpv6-client ssh
[root@slavinskiyvv ~]#

```

Рис. 2.4: Доступные службы в текущей зоне

Сравним результаты вывода информации при использовании команды `firewall-cmd --list-all` и команды `firewall-cmd --list-all --zone=public`. Вывод у нас одинаковый, так как первая команда показывает текущую зону по умолчанию, по умолчанию у нас зона `public`, а вторая команда показывает конкретно зону `public` (рис. 2.5)

```

cockpit dhcpv6-client ssh
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]# 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:
[root@slavinskiyvv ~]#

```

Рис. 2.5: Сравнение вывода информации

Добавим сервер VNC в конфигурацию брандмауэра: `firewall-cmd --add-service=vnc-server` (рис. 2.6)

```

rich rules:
[root@slavinskiyvv ~]# firewall-cmd --add-service=vnc-server
success
[root@slavinskiyvv ~]#

```

Рис. 2.6: Добавление VNC в брандмауэр

Проверим, добавился ли `vnc-server` в конфигурацию: `firewall-cmd --list-all`. (рис. 2.7)


```

[root@slavinskiyvv ~]# firewall-cmd --add-service=vnc-server
success
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#

```

Рис. 2.7: Проверка

Перезапустим службу firewalld: `systemctl restart firewalld`. (рис. 2.8)

```

rich rules:
root@slavinskiyvv ~]# systemctl restart firewalld
root@slavinskiyvv ~]#

```

Рис. 2.8: Перезапуск службы

Проверим, есть ли `vnc-server` в конфигурации: `firewall-cmd --list-all`. `vnc-server` пропал, потому что служба была добавлена только во временную конфигурацию. (рис. 2.9)

```
[root@slavinskiyvv ~]# systemctl restart firewalld
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#
```

Рис. 2.9: Проверка

Добавим службу vnc-server ещё раз, но на этот раз сделаем её постоянной, используя команду `firewall-cmd --add-service=vnc-server --permanent`. (рис. 2.10)

```
icmp-blocks:
rich rules:
[root@slavinskiyvv ~]# firewall-cmd --add-service=vnc-server --permanent
success
[root@slavinskiyvv ~]#
```

Рис. 2.10: Добавление службы в постоянную

Проверим наличие vnc-server в конфигурации: `firewall-cmd --list-all`. (рис. 2.11)

```
success
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#
```

Рис. 2.11: Проверка

Перезагрузим конфигурацию firewalld и посмотрим конфигурацию времени выполнения: `firewall-cmd --reload`, `firewall-cmd --list-all`. (рис. 2.12)

```
icmp-blocks:
rich rules:
[root@slavinskiyvv ~]# firewall-cmd --reload
success
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#
```

Рис. 2.12: Перезагрузка конфигурации

Добавим в конфигурацию межсетевого экрана порт 2022 протокола TCP: `firewall-cmd --add-port=2022/tcp --permanent`. Потом перезагрузим конфигурацию firewalld и проверим, что порт добавлен в конфигурацию. (рис. 2.13)

```
rich rules:
[root@slavinskiyvv ~]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@slavinskiyvv ~]# firewall-cmd --reload
success
[root@slavinskiyvv ~]# 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@slavinskiyvv ~]#
```

Рис. 2.13: Добавление порта

Откроем терминал и под учётной записью своего пользователя запустим ин-

терфейс GUI firewall-config: firewall-config(рис. 2.14)

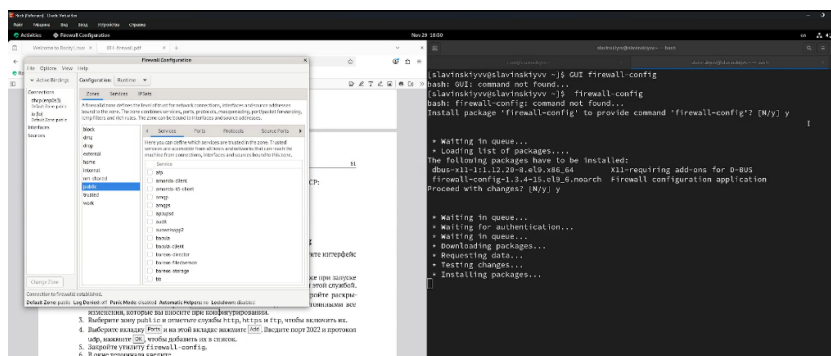


Рис. 2.14: Запуск интерфейса GUI

Далее в конфигурации выберем permanent. В зоне public выберем http ftp и https. Во вкладке ports введем 2022 и добавим протокол udp. (рис. 2.15)

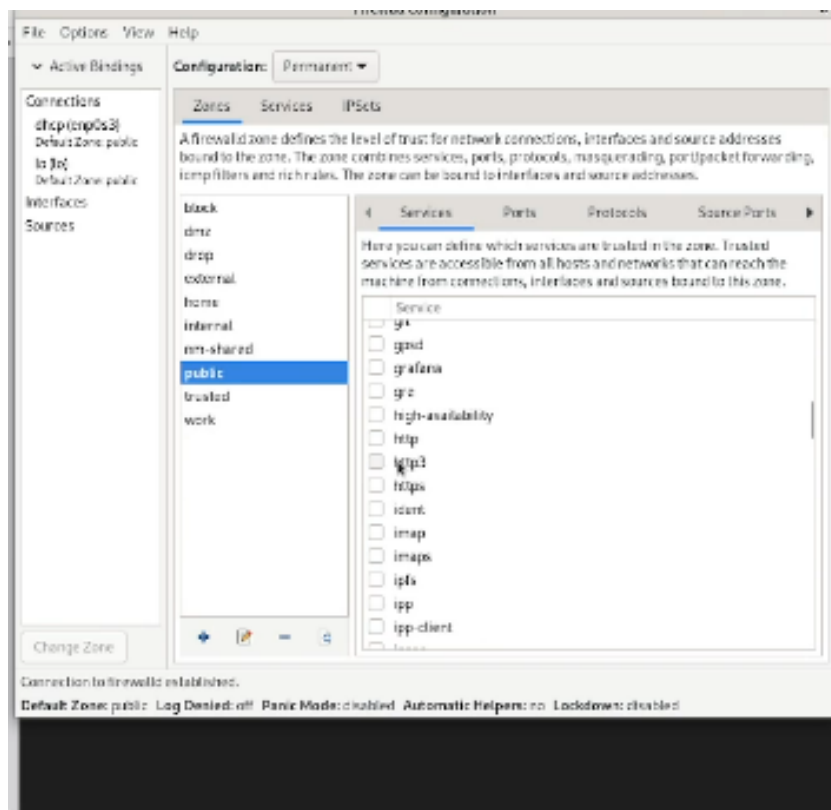


Рис. 2.15: Изменение параметров

В окне терминала введем `firewall-cmd --list-all` (рис. 2.16)

```

[slavinskiyvv@slavinskiyvv ~]$ firewall-config
[slavinskiyvv@slavinskiyvv ~]$ 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:
[slavinskiyvv@slavinskiyvv ~]$

```

Рис. 2.16: Вывод информации

Перезагрузим конфигурацию firewalld: `firewall-cmd --reload`. И потом опять выведем список сервисов. Как видим, они добавились. (рис. 2.17)

```

rich rules:
[slavinskiyvv@slavinskiyvv ~]$ firewall-cmd --reload
success
[slavinskiyvv@slavinskiyvv ~]$ 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:
[slavinskiyvv@slavinskiyvv ~]$

```

Рис. 2.17: Перезагрузка

Создадим конфигурацию межсетевого экрана, которая позволяет получить доступ к службам telnet, imap, pop3, smtp. Через командную строку добавим telnet. (рис. 2.18)

```

rich rules:
[slavinskiyvv@slavinskiyvv ~]$ firewall-cmd --add-service=telnet --permanent

```

Рис. 2.18: Создание конфигурации

Далее делаем в графическом интерфейсе GUI. (рис. 2.19)

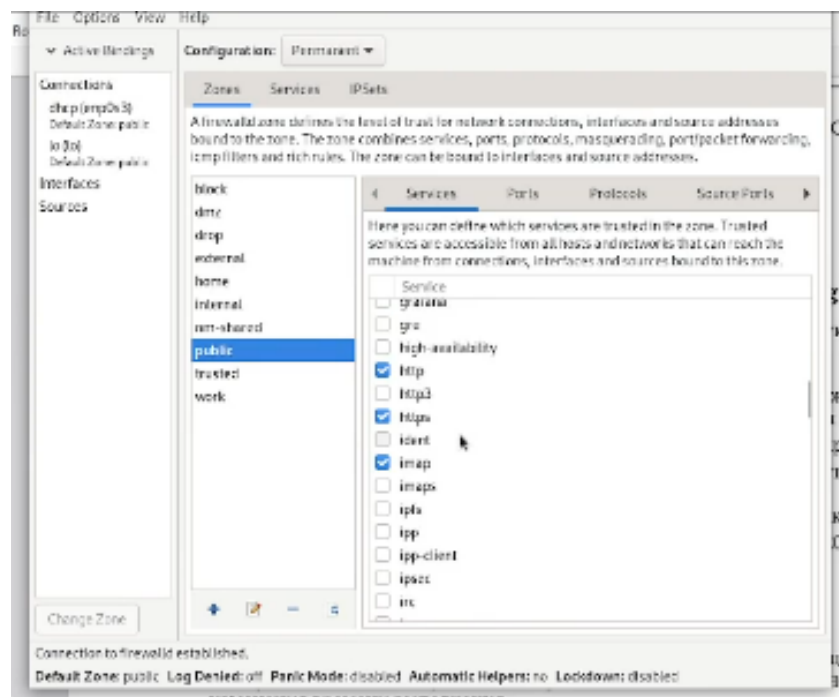


Рис. 2.19: Добавление сервисов в графическом интерфейсе

Перезагружаем конфигурацию firewalld и смотрим список доступных сервисов, как видим, все добавилось. (рис. 2.20)

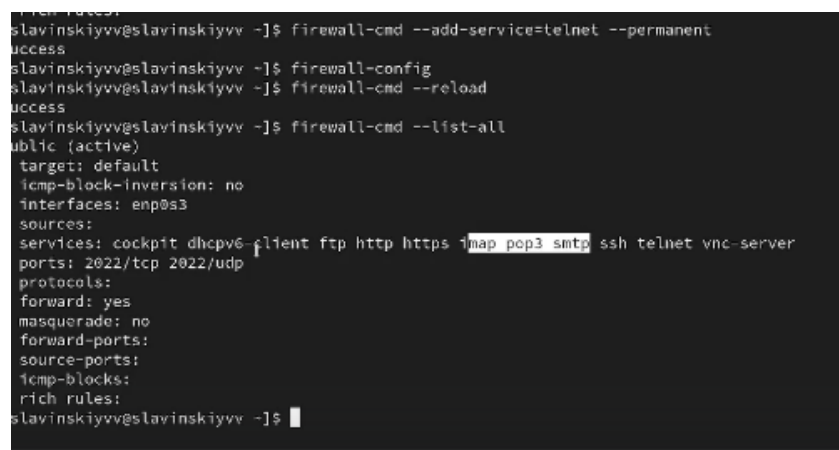


Рис. 2.20: Перезагрузка конфигурации и проверка

3 Выводы

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

4 Ответы на контрольные вопросы

1. `firewalld`
2. `firewall-cmd --add-port=2355/udp`
3. `firewall-cmd --list-all-zones`
4. `firewall-cmd --remove-service=vnc-server`
5. `firewall-cmd --reload`
6. `firewall-cmd --list-all`
7. `firewall-cmd --zone=public --add-interface=en01`
8. В зону по умолчанию - `public`