

# Фильтр пакетов

## Часть 1

---

Славинский В.В.

29 ноября 2025

Российский университет дружбы народов, Москва, Россия Россия

## Информация

---

:::::::::: { .columns align=center } :::: { .column width="70%" }

- Славинский Владислав Вадимович
- Студент
- Российский университет дружбы народов
- [1132246169@pfur.ru]

:: :: { .column width="30%" }

## Вводная часть

---

## Определение текущей зоны по умолчанию

В терминале получим права администратора, определим текущую зону по умолчанию, введя:  
firewall-cmd –get-default-zone.

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

Рис. 1: sc1

## Определение доступных зон

Определим доступные зоны с помощью firewall-cmd –get-zones.

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

Рис. 2: sc2

## Доступные службы

Посмотрим службы, доступные на компьютере, используя firewall-cmd –get-services.

```
[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 dhcpcv6 dhcpcv6-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 gpgsql 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 managesieve matrix mdns memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-w
bt mssql murmur mysql nbd nebula-nebula-ns netdata-dashboard nfs nfs3 nmea-0183 nrpe ntp
nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmp
roxy 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 salt-master samba samba-client samba-dc sane
sip sips slp smtp smtp-submission smtps snmp snmpd snmpd-trap snmptrap spiderOak-lans
ync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthi
ng-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-
client upnp-client vdsm 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 ~]#
```

## Доступные службы в текущей зоне

Определим доступные службы в текущей зоне: firewall-cmd –list-services.

```
[root@slavinskiyv ~]# firewall-cmd --list-services  
cockpit dhcpcv6-client ssh  
[root@slavinskiyv ~]# █ █
```

Рис. 4: sc4

## Сравнение вывода информации

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

```
cockpit dhcpcv6-client ssh
[root@slavinskiyvv ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-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:
```

## Добавление VNC в брандмауэр

Добавим сервер VNC в конфигурацию брандмауэра: firewall-cmd –add-service=vnc-server

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

Рис. 6: sc6

## Проверка

Проверим, добавился ли vnc-server в конфигурацию: firewall-cmd –list-all.

```
[root@slavinskiyvv ~]# firewall-cmd --add-service=vnc-server  
success  
[root@slavinskiyvv ~]# firewall-cmd --list-all  
public (active)  I  
  target: default  
  icmp-block-inversion: no  
  interfaces: enp0s3  
  sources:  
  services: cockpit dhcpcv6-client ssh vnc-server  
  ports:  
  protocols:  
  forward: yes  
  masquerade: no  
  forward-ports:  
  source-ports:  
  icmp-blocks:  
  rich rules:
```

## Перезапуск службы

Перезапустим службу firewalld: systemctl restart firewalld.

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

Рис. 8: sc8

## Проверка

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

```
[root@slavinskiy ~]# systemctl restart firewalld
[root@slavinskiy ~]# firewall-cmd --list-all
public (active)
    target: default
    icmp-block-inversion: no
    interfaces: enp0s3
    sources:
    services: cockpit dhcpcv6-client ssh
    ports:
    protocols:
    forward: yes
    masquerade: no
    forward-ports:
    source-ports:
    icmp-blocks:
    rich rules:
```

## Добавление службы в постоянную

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

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

Рис. 10: sc10

## Проверка

Проверим наличие vnc-server в конфигурации: firewall-cmd –list-all.

```
success
[root@slavinskiyv ~]# firewall-cmd --list-all
public (active)
    target: default
    icmp-block-inversion: no
    interfaces: enp0s3
    sources:
    services: cockpit dhcpcv6-client ssh
    ports:
    protocols:
    forward: yes
    masquerade: no
    forward-ports:
    source-ports:
    icmp-blocks:
    rich rules:
[root@slavinskiyv ~]#
```

## Перезагрузка конфигурации

Перезагрузим конфигурацию firewalld и просмотрим конфигурацию времени выполнения: firewall-cmd –reload, firewall-cmd –list-all.

```
[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 dhcpcv6-client ssh vnc-server
    ports:
    protocols:
    forward: yes
    masquerade: no
    forward-ports:
    masquerade-ports:
```

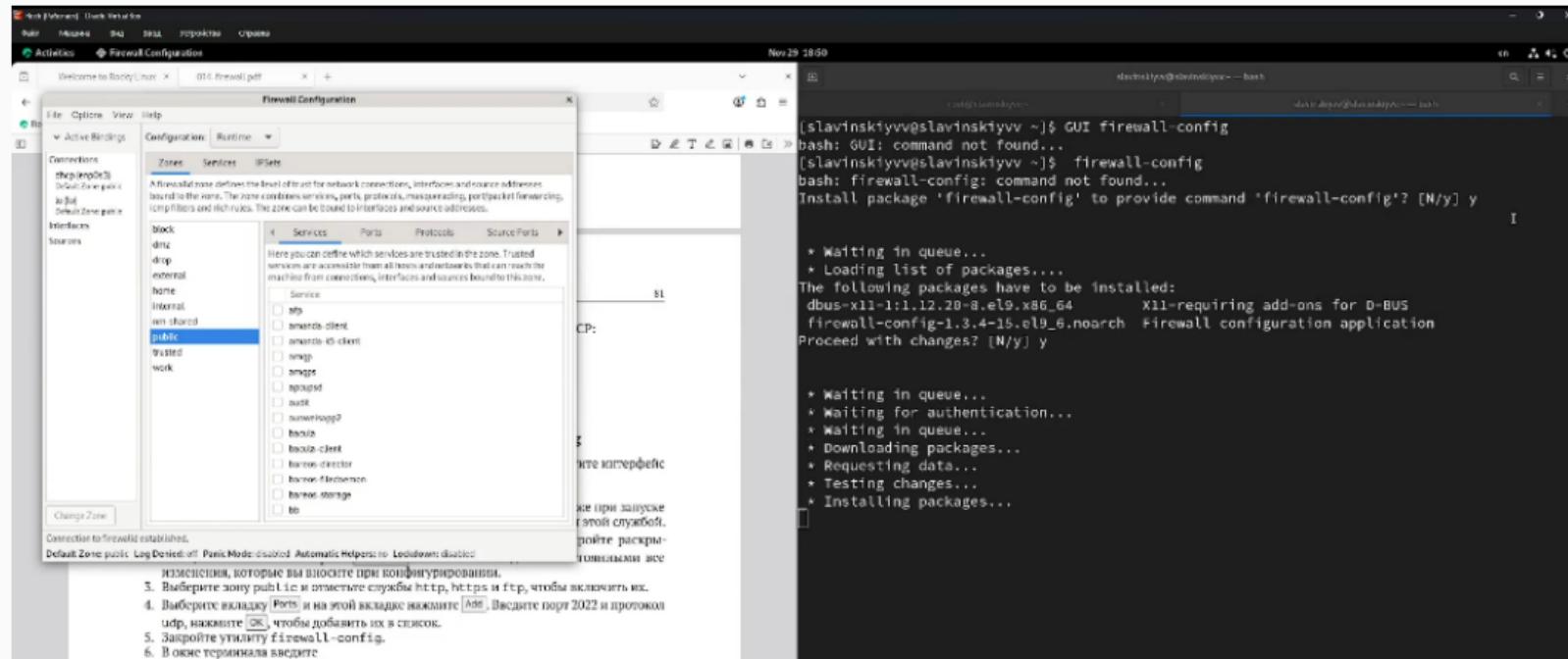
## Добавление порта

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.

```
[root@slavinskiyv ~]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@slavinskiyv ~]# firewall-cmd --reload
success
[root@slavinskiyv ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-client ssh vnc-server
  ports: 2022/tcp
  protocols:
  forward: yes
  masquerade: no
```

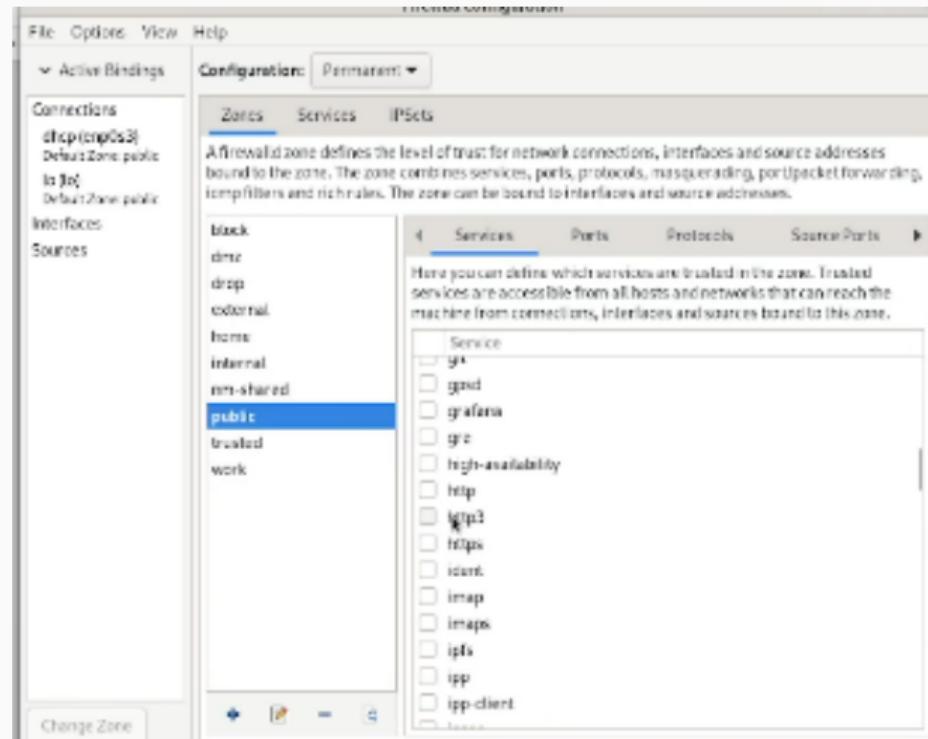
# Запуск интерфейса GUI

Откроем терминал и под учётной записью своего пользователя запустим интерфейс GUI  
firewall-config: firewall-config



## Изменение параметров

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



## Вывод информации

В окне терминала введем firewall-cmd –list-all

```
[slavinskiyvv@slavinskiyvv ~]$ firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpcv6-client ssh vnc-server
  ports: 2022/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[slavinskiyvv@slavinskiyvv ~]$ █
```

## Перезагрузка

Перезагрузим конфигурацию firewalld: firewall-cmd –reload. И потом опять выведем список сервисов. Как видим, они добавились.

```
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 dhcpcv6-client ftp http https ssh vnc-server  
    ports: 2022/tcp 2022/udp  
    protocols: I  
    forward: yes  
    masquerade: no  
    forward-ports:  
    source-ports:
```

## Создание конфигурации

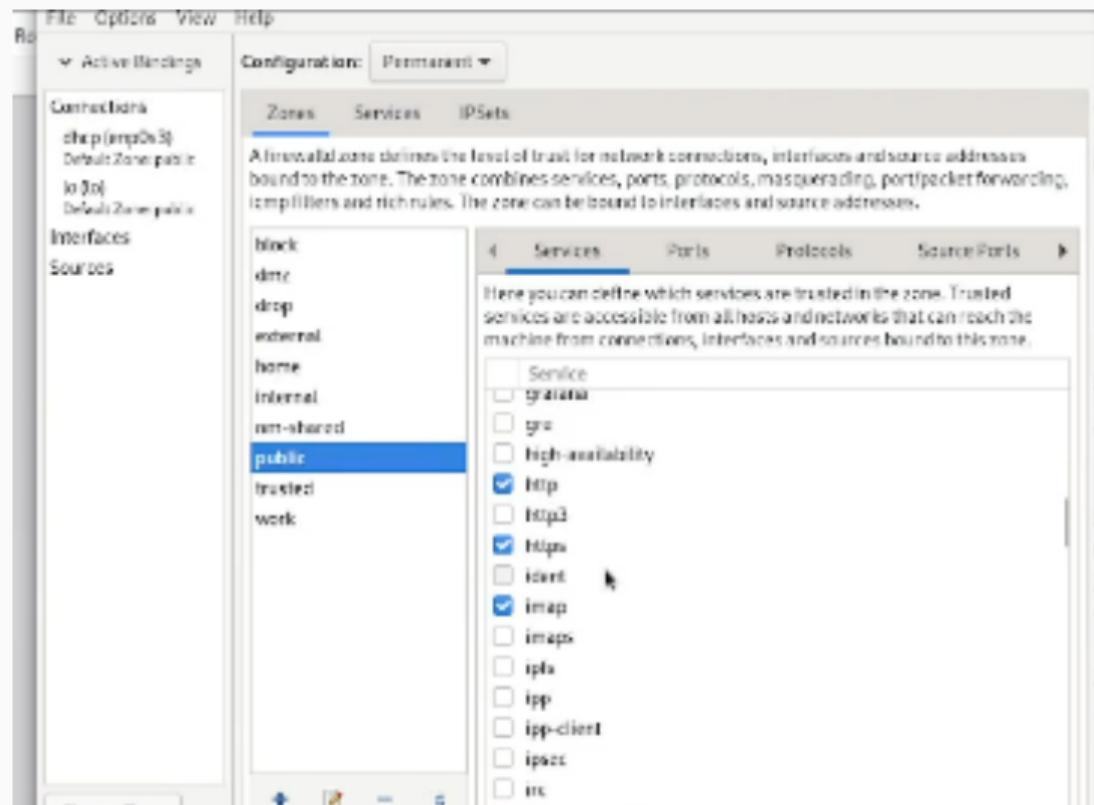
Создадим конфигурацию межсетевого экрана, которая позволяет получить доступ к службам telnet, imap, pop3, smtp. Через командную строку добавим telnet.

```
[root@slavinskiyvv ~]# firewall-cmd --add-service=telnet --permanent
```

Рис. 18: sc18

# Добавление сервисов в графическом интерфейсе

Далее делаем в графическом интерфейсе GUI.



## Перезагрузка конфигурации и проверка

Перезагружаем конфигурацию firewalld и смотрим список доступных сервисов, как видим, все добавилось.

```
[root@slavinskiyvv ~]# firewall-cmd --add-service=telnet --permanent
success
[root@slavinskiyvv ~]# firewall-config
[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 dhcpcv6-client ftp http https imap pop3 smtp ssh telnet vnc-server
  ports: 2022/tcp 2022/udp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
```