II Corso RedHat per sistemisti INFN

RHEL/SL/CentOS 7

firewalld

firewalld

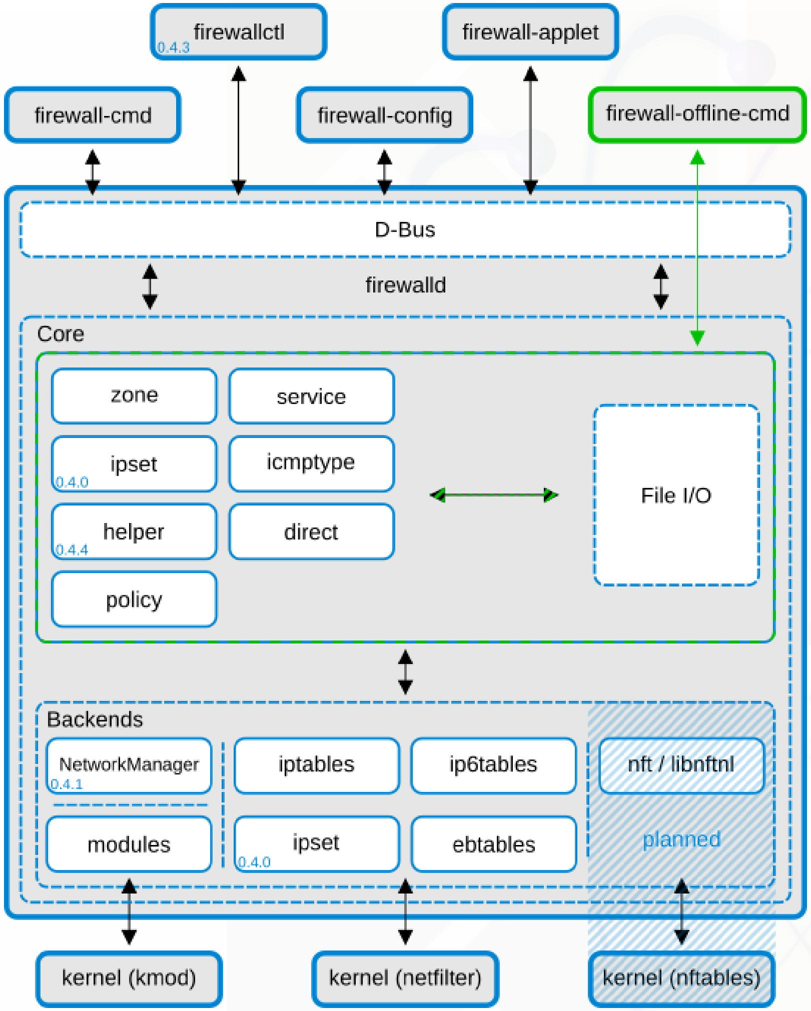
a service daemon with D-Bus interface

firewalld provides a dynamically managed firewall with support for network/firewall zones that define the trust level of network connections or interfaces. It also provides simplified interface for services or applications to add firewall rules directly.

Main features:

- firewall zones
- predefined list of zones, services and icmptypes – zones simplify configuration and segregation (separation of network traffic by zone and interface)
- Rich language for writing more flexible and comples rules; direct interface
- complete IPv4, IPv6 support (filtering and NAT)
- bridge and ipset support
- lockdown
- runtime and permanent configuration separation: changes can be done immediately in the runtime environment - no restart of the service or daemon is needed
- complete D-Bus API
- daemon runs in user-space; GUI works ☺
- still iptables/ip6tables/ebtables underneath

the big picture



Two layer design: the core layer and the D-Bus layer on top. The core layer is responsible for handling the configuration and the back ends like iptables, ip6tables, ebtables, ipset and the module loader. The firewalld D-Bus interface is the primary way to alter and create the firewall configuration. The interface is used by all firewalld provided online tools, like for example `firewall-cmd`, `firewallctl`, `firewall-config` and `firewall-applet`. firewalld does not depend on NetworkManager, but the use is recommended.

configuration

two configuration directories:

- **`/usr/lib/firewalld`**: default and fallback config for zones, services and icmptypes
- **`/etc/firewalld`**: system specific configuration

runtime vs ***permanent***: something like Cisco's *running-config* vs *startup-config* - the runtime configuration is the actual effective configuration applied to the firewall in the kernel, and commands usually operate on it unless the

```
--permanent
```

flag is provided; at firewalld service start the permanent configuration becomes the runtime configuration. Changes in the runtime configuration are not automatically saved to the permanent configuration, so it's often sensible to carry out changes in the runtime configuration and then – if everything's working fine – use the command

```
firewall-cmd --runtime-to-permanent
```

to persist them.

firewalld.conf

A few interesting parameters

```
# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=work

# Lockdown
# If set to enabled, firewall changes with the D-Bus interface will be limited
# to applications that are listed in the lockdown whitelist.
# The lockdown whitelist file is lockdown-whitelist.xml
# Default: no
Lockdown=no

# IPv6_rpfilter
# Performs a reverse path filter test on a packet for IPv6. If a reply to the
# packet would be sent via the same interface that the packet arrived on, the
# packet will match and be accepted, otherwise dropped.
# The rp_filter for IPv4 is controlled using sysctl.
# Default: yes
IPv6_rpfilter=yes
```

zones

A ***zone*** defines the trust level of the interface used for a connection, and manages a group of rules dictating what traffic should be permitted, thereby allowing segregation/separation of traffic.

firewalld filters incoming traffic into different zones depending on the particular rules applied to that zone. An incoming connection will use the following logic when determining which zone it will match (more on this later):

- *If the source IP address matches a source that has been defined for the zone, then the packet will be routed through that zone*.
- If the source IP address has not matched any zones, *next if the incoming interface for the packet matches a filter on that zone then this zone will be used*.
- Otherwise if the incoming traffic does not specifically match any of the defined zones, a *default zone* will be used.

Every pre-defined zone has a *default filtering policy* (a *default target* in the ip*tables language) in place which applies to traffic traversing it.

zones

Interfaces, or connections (in the *NetworkManager sense:* the firewall in the kernel is not able to handle network connections with the name shown by NetworkManager, it can only handle the network interfaces used by the connection - because of this NetworkManager tells firewalld to assign the network interface that is used for this connection to the zone defined in the configuration of that connection), and source addresses, are assigned (or bound) to *active* zones, and a zone can be associated with one or more connections, interfaces or sources.

Although the zone feature is specifically aimed at mobile systems (incoming traffic filtering), zones can be equally used on multi-homed systems, possibly associating each interface with its own appropriate zone – *IMHO firewalld is by no means a general purpose interface for building stand-alone firewalls* (yet it can be used for getting the job done, but you have to work with *ip*tables*…).

Zone definition files (in XML format) are located in:

- **`/usr/lib/firewalld/zones`**: default and fallback
- **`/etc/firewalld/zones`**: user created and custom

and zones can be created, modified, and deleted either using the standard firewalld configuration interfaces (**`firewall-cmd`**, **`firewall-config`**) or by editing configuration files.

zone management

```
# firewall-cmd --get-default-zone
trusted
# firewall-cmd --set-default-zone=work [--permanent]
success
# firewall-cmd –get-active-zones
work
  interfaces: bridge0 em2 em1
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
# firewall-cmd --get-zone-of-interface=bridge0
work
# firewall-cmd --path-zone=work --permanent
/etc/firewalld/zones/work.xml
```

An *active zone* is a zone that have bindings to an interface or a source – an active zone carries network traffic.

zone management

```
# firewall-cmd –permanent --new-zone=salca
success
# cat /etc/firewalld/zones/salca.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
</zone>
# firewall-cmd --zone=salca --set-target=%%REJECT%% \
  --permanent
success
# cat /etc/firewalld/zones/salca.xml
<?xml version="1.0" encoding="utf-8"?>
<zone target="%%REJECT%%">
</zone>
```

zone management

```
# firewall-cmd --zone=trusted --add-interface=trusted0

# firewall-cmd --zone=drop --add-interface=untrusted0

# firewall-cmd --zone=mgmt --add-source=192.168.27.0/24

# firewall-cmd --get-zone-of-source=192.168.27.0/24
mgmt

# firewall-cmd --zone=trusted --remove-interface=trusted0

# firewall-cmd --zone=mgmt --remove-source=192.168.27.0/24
```

default pre-defined zones -1

drop

drop all incoming traffic unless related to outgoing traffic (do not even respond with ICMP errors). Only outgoing network connections are allowed.

block

any incoming network connections are rejected with an `icmp-host-prohibited` message for IPv4 and `icmp6-adm-prohibited` for IPv6. Only network connections initiated within this system are possible

dmz

for computers in your demilitarized zone that are publicly-accessible with limited access to your internal network. Only selected incoming connections are accepted.

public

for use in work areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.

default pre-defined zones -2

external

for use on external networks with masquerading enabled especially for routers. You do not trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.

work

for use in work areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.

home

internal

for use in home areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.

trusted

all network connections are accepted.

zone configuration

```
# firewall-cmd --zone=work --list-all
(or # firewall-cmd --info-zone=work)
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: bridge0 em2 em1
  sources:
  services: ssh dhcpv6-client http https iperf iperf3
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

/etc/firewalld/zones/work.xml

```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Work</short>
  <description>For use in work areas. You mostly
trust the other computers on networks to not harm
your computer. Only selected incoming connections
are accepted.</description>
  <service name="ssh"/>
  <service name="dhcpv6-client"/>
  <service name="http"/>
  <service name="https"/>
  <service name="iperf"/>
  <service name="iperf3"/>
</zone>
```

zone options

`target="ACCEPT|%%REJECT%%|DROP"`

Can be used to accept, reject or drop every packet that doesn't match any rule (port, service, etc.). The ACCEPT target is used in trusted zone to accept every packet not matching any rule. The %%REJECT%% target is used in block zone to reject (with default firewalld reject type) every packet not matching any rule. The DROP target is used in drop zone to drop every packet not matching any rule. If the target is not specified, every packet not matching any rule will be rejected (**default** target).

`interface (name="string")`

Is an optional empty-element tag and can be used several times. It can be used to bind an interface to a zone. You don't need this for NetworkManager-managed interfaces, because NetworkManager binds interfaces to zones automatically.

`source (address="address[/mask]"|mac="mac"|ipset="ipset")`

Is an optional empty-element tag and can be used several times. It can be used to bind a source address, address range, a MAC address or an ipset to a zone.

`rule`

Is an optional element tag and can be used several times to have more than one rich language rule entry.

zone options

`service (name=“string”)`

Is an optional empty-element tag and can be used several times to have more than one service entry enabled.

`port (port=“portid[-portid]” protocol=“tcp|udp”)`

Is an optional empty-element tag and can be used several times to have more than one port entry; the port can either be a single port number portid or a port range portid-portid.– all attributes are mandatory.

`protocol (name=“string”)`

Is an optional empty-element tag and can be used several times to have more than one protocol. The protocol can be any protocol supported by the system (see `/etc/protocols`).

`masquerade`

Is an optional empty-element tag. It can be used only once in a zone configuration and is not usable for IPv6. If it’s present masquerading is enabled for the zone. If you want to enable masquerading, you should enable it in the zone bound to the external interface.

standard rules: services

firewalld services are sets of firewall rules *to open ports* associated with a specific application or system service; a service can be a list of local ports, protocols and destinations (and additionally also a list of firewall helper modules) automatically loaded if a service is enabled.

firewalld daemon's default policy is to deny access, so any access needed has to be explicitly granted to a port/a set of ports associated with a specific service

Service definition files (in XML format) are located in:

- **`/usr/lib/firewalld/services`**: default and fallback
- **`/etc/firewalld/services`**: user created and custom

```
# cat /usr/lib/firewalld/services/https
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>Secure WWW (HTTPS)</short>
  <description>HTTPS is a modified HTTP used …</description>
  <port protocol="tcp" port="443"/>
</service>
```

service management

```
# firewall-cmd --list-services
ssh dhcpv6-client http https iperf iperf3
# firewall-cmd --zone=dmz --list-services
ssh
# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bitcoin
bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine
condor-collector ctdb dhcp dhcpv6 dhcpv6-client dns docker-registry
dropbox-lansync elasticsearch freeipa-ldap freeipa-ldaps freeipa-
replication freeipa-trust ftp ganglia-client ganglia-master high-
availability http https imap imaps iperf iperf3 ipp ipp-client ipsec iscsi-
target kadmin kerberos kibana klogin kpasswd kshell ldap ldaps libvirt
libvirt-tls managesieve mdns mosh mountd ms-wbt mssql mysql nfs nrpe ntp
openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole pmcd pmproxy
pmwebapi pmwebapis pop3 pop3s postgresql privoxy proxy-dhcp ptp pulseaudio
puppetmaster quassel radius rpc-bind rsh rsyncd samba samba-client sane sip
sips smtp smtp-submission smtps snmp snmptrap spideroak-lansync squid ssh
synergy syslog syslog-tls telnet tftp tftp-client tinc tor-socks
transmission-client vdsm vnc-server wbem-https xmpp-bosh xmpp-client xmpp-
local xmpp-server
```

service management

```
# firewall-cmd --permanent --new-service=iperf
success
# firewall-cmd --service=iperf \
--set-description="iperf - perform network throughput tests"
success
# firewall-cmd –permanent --service=iperf --set-short="iperf"
success
# firewall-cmd --permanent --service=iperf --add-port=5001/tcp
success
# firewall-cmd --permanent --service=iperf --add-port=5001/udp
success
# firewall-cmd --reload
success
# firewall-cmd --info-service=iperf
iperf
  ports: 5001/tcp 5001/udp
  ...
```

standard rules: service or port?

```
# firewall-cmd --zone=work --add-service=iperf
success
# firewall-cmd --info-zone=work
work (active)
  target: default
  interfaces: em1
  services: ssh dhcpv6-client http https iperf3 iperf
  ports:
# firewall-cmd --zone=work --add-port=6001/tcp
success
# firewall-cmd --info-zone=work
work (active)
  target: default
  interfaces: em1
  services: ssh dhcpv6-client http https iperf3 iperf
  ports: 6001/tcp
```

direct rules

It is possible to interact directly with the *ip*tables layer* using the –**direct** option – *direct interface* is aimed to help implementing rules not supported by firewalld directly, or addressing specific migration issues.

```
# firewall-cmd [--permanent] --direct --add-rule {ipv4|ipv6} \
    table chain priority args
```

```
# firewall-cmd --permanent --direct \
    --add-rule ipv4 filter OUTPUT 0 \
    -p tcp -m tcp --dport=80 -j ACCEPT
# firewall-cmd --permanent --direct \
    --add-rule ipv4 filter OUTPUT 1 -j DROP
# firewall-cmd --permanent --direct --get-all-rules
```

rich rules

firewalld rich language is an abstract representation of ip*tables rules, and rich rules are intended to provide a much greater level of control than standard rules, through more custom granular options, without having to deal with the ip*tables obscure syntax. Rich rules can also be used to configure logging, masquerading, port forwarding, and rate limiting, and use the option `--add-rich-rule`

General rule structure

```
rule
  [source]
  [destination]
  service|port|protocol|icmp-block|icmp-type|masquerade|\
      forward-port|source-port
  [log]
  [audit]
  [accept|reject|drop|mark]
```

rich rules options

```
rule [family="ipv4|ipv6"]
source [not] address="address[/mask]"|
             mac="mac-address"|
             ipset="ipset‟
destination [not] address="address[/mask]‟
service name="service name"
port port="port value" protocol="tcp|udp"
protocol value="protocol value"

log [prefix="prefix text"]
    [level="log level"]
    [limit value="rate/duration"]

accept|reject [type="reject type"]|drop
    [limit value="rate/duration"]
```

rich rules examples

Allow new IPv4 and IPv6 connections for service ftp and log 1 per minute using audit:

```
    rule service name="ftp" log limit value="1/m" audit accept
```

Allow new IPv4 connections from address 192.168.0.0/24 for service tftp and log 1 per minutes using syslog:

```
    rule family="ipv4" source address="192.168.0.0/24"
         service name="tftp" log prefix="tftp"
         level="info" limit value="1/m" accept
```

New IPv6 connections from 1:2:3:4:6:: to service radius are all rejected and logged at a rate of 3 per minute; new IPv6 connections from other sources are accepted.

```
    rule family="ipv6" source address="1:2:3:4:6::" service name="radius‟
         log prefix="dns" level="info" limit value="3/m" reject
    rule family="ipv6" service name="radius" accept
```

standard, *rich* or *direct* rule?

The different methods to set a rule reflect the amount of fine grain control that can be delivered.

A *standard rule* applies to all traffic that matches the port/service.

A *rich rule* can deliver network based controls without needing a new zone or can configure logging of a traffic type.

The *direct rules* allow direct manipulation of the underlying iptables/ip6tables/ebtables rulesets for use cases that a rich rule cannot manage, and should really be the last resort if the goal can't be established with a standard or rich rule – as usual it's a matter of tradeoff between complexity (and management cost) and capabilities; although the most complicated iptables arrangements would only be possible with this type of rule the *cost* of doing so should be considered carefully. Moreover, *direct rules are not bound to any particular zone, and are applied with the highest priority*.

rules comparison

standard

```
# firewall-cmd --add-port 443/tcp
# firewall-cmd --add-service https
```

rich

```
# firewall-cmd --add-rich-rule \
  "rule port port="443" protocol="tcp" accept"
# firewall-cmd --add-rich-rule \
  "rule service name="https" accept‟
# firewall-cmd --add-rich-rule \
 "rule family="ipv4" port port="2222" protocol="tcp" drop"
```

direct

```
# firewall-cmd --direct --add-rule ipv4 filter INPUT 1 \
   -m tcp -p tcp --dport 443 -j ACCEPT
```

beware: direct rules are not saved in zones’ XML description files but are stored in **`/etc/firewalld/direct.xml`**

rules’ evaluation priority

- direct rules
- source address based zones
  - order: log deny allow
- interface based zone
  - order: log deny allow
- default zone
  - order: log deny allow

Within each log/deny/allow split of a zone the order is:

- rich rule
- port definition
- service definition

ie: ***the more abstract the rule, the lower the priority of evaluation***

IPset

IP sets are a framework inside the Linux kernel, (…). Depending on the type, an IP set may store IP addresses, networks, (TCP/UDP) port numbers, MAC addresses, interface names or combinations of them in a way, which ensures lightning speed when matching an entry against a set.

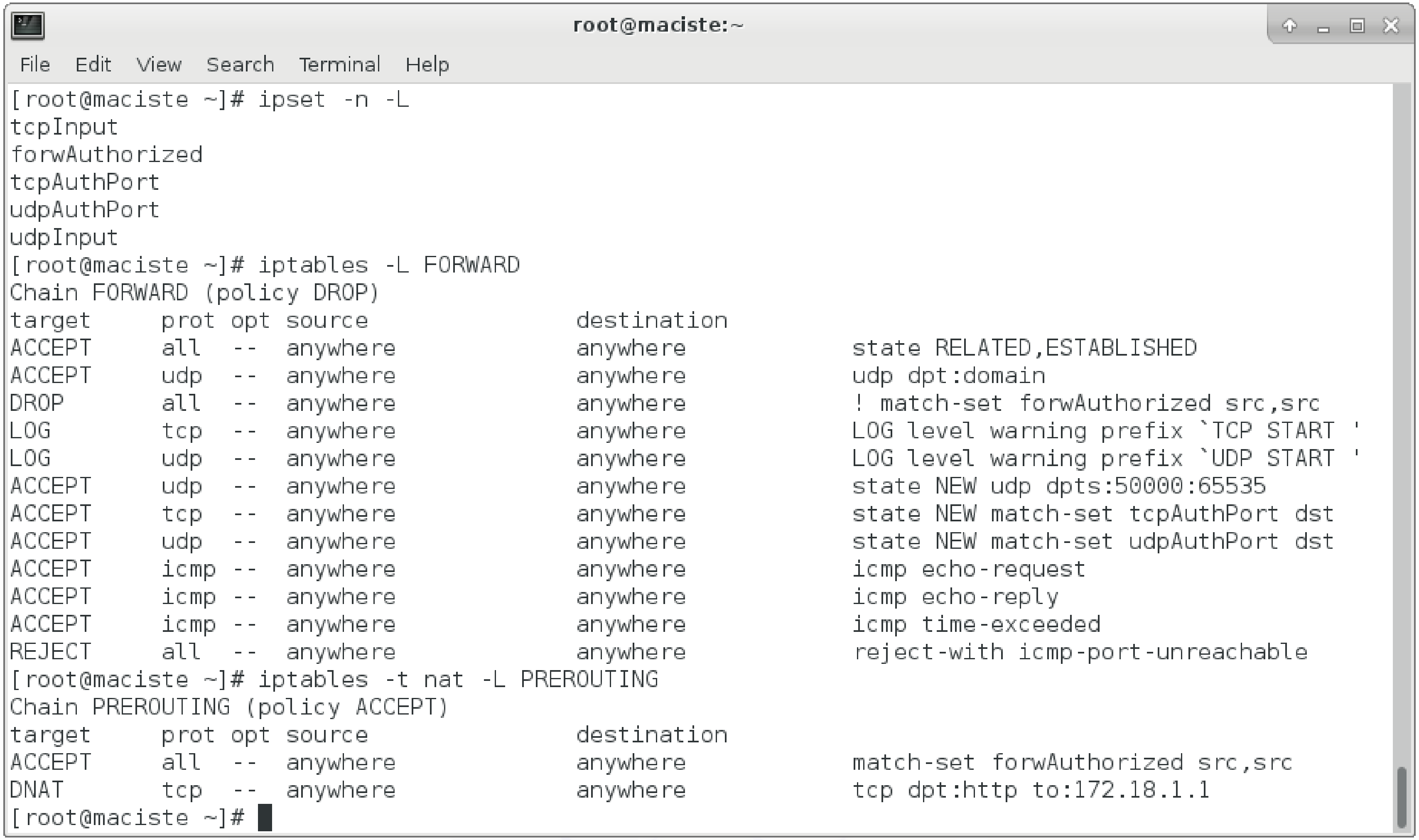
If you want to

- store multiple IPv4/IPv6 addresses or port numbers and match against the collection by iptables at one swoop;
- dynamically update iptables rules against IP addresses or ports without performance penalty;
- express complex IP address and ports based rulesets with one single iptables rule and benefit from the speed of IP sets

then ipset may be the proper tool for you.

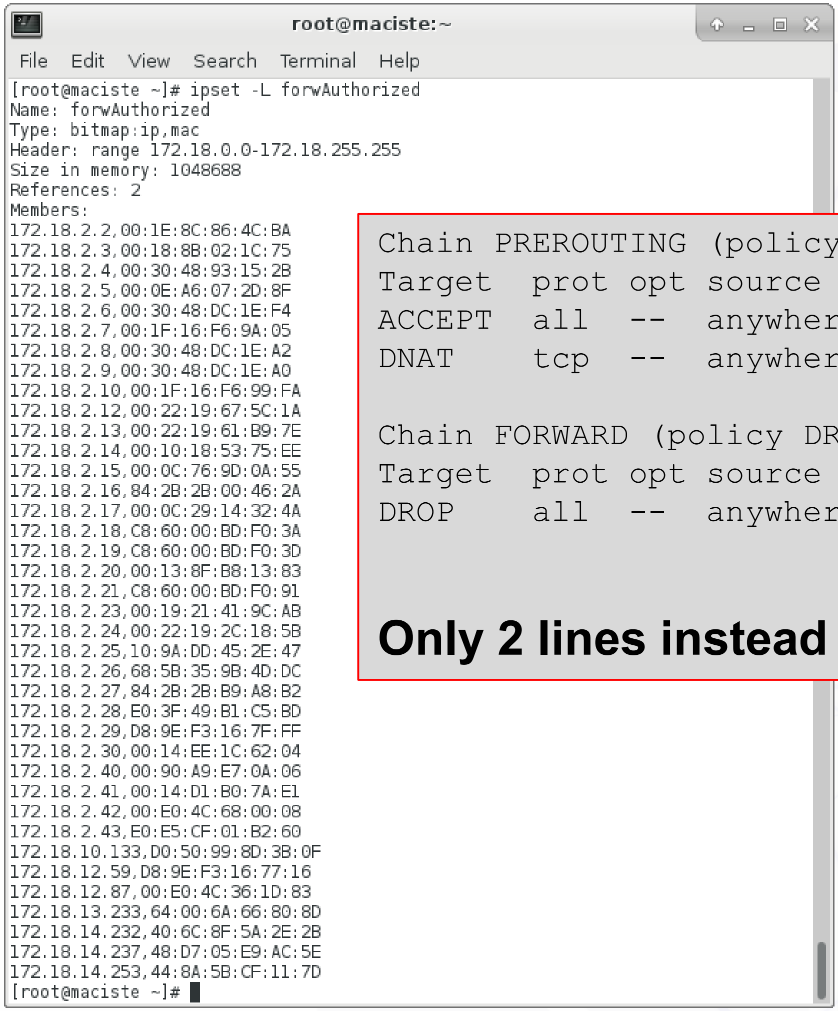
firewalld actually seems to be supporting the subset of IP set compatible with IPv6 addresses (only *hash* types, nor *bitmap* neither *list*).

IPset examples



```
[root@maciste ~]# ipset -n -L
tcpInput
forwAuthorized
tcpAuthPort
udpAuthPort
udpInput
[root@maciste ~]# iptables -L FORWARD
Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            state RELATED,ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere            udp dpt:domain
DROP       all  --  anywhere             anywhere            ! match-set forwAuthorized src,src
LOG        tcp  --  anywhere             anywhere            LOG level warning prefix `TCP START '
LOG        udp  --  anywhere             anywhere            LOG level warning prefix `UDP START '
ACCEPT     udp  --  anywhere             anywhere            state NEW udp dpts:50000:65535
ACCEPT     tcp  --  anywhere             anywhere            state NEW match-set tcpAuthPort dst
ACCEPT     udp  --  anywhere             anywhere            state NEW match-set udpAuthPort dst
ACCEPT     icmp --  anywhere             anywhere            icmp echo-request
ACCEPT     icmp --  anywhere             anywhere            icmp echo-reply
ACCEPT     icmp --  anywhere             anywhere            icmp time-exceeded
REJECT     all  --  anywhere             anywhere            reject-with icmp-port-unreachable
[root@maciste ~]# iptables -t nat -L PREROUTING
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere            match-set forwAuthorized src,src
DNAT       tcp  --  anywhere             anywhere            tcp dpt:http to:172.18.1.1
[root@maciste ~]#
```

IPset examples



```
[root@maciste ~]# ipset -L forwAuthorized
Name: forwAuthorized
Type: bitmap:ip,mac
Header: range 172.18.0.0-172.18.255.255
Size in memory: 1048688
References: 2
Members:
172.18.2.2,00:1E:8C:86:4C:BA
172.18.2.3,00:18:8B:02:1C:75
172.18.2.4,00:30:48:93:15:2B
172.18.2.5,00:0E:A6:07:2D:8F
172.18.2.6,00:30:48:DC:1E:F4
172.18.2.7,00:1F:16:F6:9A:05
172.18.2.8,00:30:48:DC:1E:A2
172.18.2.9,00:30:48:DC:1E:A0
172.18.2.10,00:1F:16:F6:99:FA
172.18.2.12,00:22:19:67:5C:1A
172.18.2.13,00:22:19:61:B9:7E
172.18.2.14,00:10:18:53:75:EE
172.18.2.15,00:0C:76:9D:0A:55
172.18.2.16,84:2B:2B:00:46:2A
172.18.2.17,00:0C:29:14:32:4A
172.18.2.18,C8:60:00:BD:F0:3A
172.18.2.19,C8:60:00:BD:F0:3D
172.18.2.20,00:13:8F:B8:13:83
172.18.2.21,C8:60:00:BD:F0:91
172.18.2.23,00:19:21:41:9C:AB
172.18.2.24,00:22:19:2C:18:5B
172.18.2.25,10:9A:DD:45:2E:47
172.18.2.26,68:5B:35:9B:4D:DC
172.18.2.27,84:2B:2B:B9:A8:B2
172.18.2.28,E0:3F:49:B1:C5:BD
172.18.2.29,D8:9E:F3:16:7F:FF
172.18.2.30,00:14:EE:1C:62:04
172.18.2.40,00:90:A9:E7:0A:06
172.18.2.41,00:14:D1:B0:7A:E1
172.18.2.42,00:E0:4C:68:00:08
172.18.2.43,E0:E5:CF:01:B2:60
172.18.10.133,D0:50:99:8D:3B:0F
172.18.12.59,D8:9E:F3:16:77:16
172.18.12.87,00:E0:4C:36:1D:83
172.18.13.233,64:00:6A:66:80:8D
172.18.14.232,40:6C:8F:5A:2E:2B
172.18.14.237,48:D7:05:E9:AC:5E
172.18.14.253,44:8A:5B:CF:11:7D
[root@maciste ~]#
```

```
Chain PREROUTING (policy ACCEPT)
Target  prot opt source    destination
ACCEPT  all  --  anywhere  anywhere    match-set forwAuthorized src,src
DNAT    tcp  --  anywhere  anywhere    tcp dpt:http to:172.18.1.1

Chain FORWARD (policy DROP)
Target  prot opt source    destination
DROP    all  --  anywhere  anywhere    ! match-set forwAuthorized src,src
```

Only 2 lines instead of ~50-80

supported IPset types

```
# firewall-cmd --get-ipset-types
    hash:ip
    hash:ip,mark
    hash:ip,port
    hash:ip,port,ip
    hash:ip,port,net
    hash:mac
    hash:net
    hash:net,iface
    hash:net,net
    hash:net,port
    hash:net,port,net
```

IPset (black|white) list example

```
# firewall-cmd --permanent --new-ipset=bwlist --type=hash:ip \
                --family=inet
success

# firewall-cmd --ipset=bwlist --add-entry=192.168.100.70
success

# firewall-cmd --info-ipset=bwlist
blacklist
  type: hash:ip
  options: family=inet
  entries: 192.168.100.70
```

```
firewall-cmd --zone=drop --add-source=ipset:bwlist
DR:  () -m set --match-set bwlist src -j REJECT|ACCEPT
RR:  () source [not] address=ipset="ipset"
```

trying it all together

test setup:

virtone.mib.infn.it – multi homed CentOS7

- em1: 212.189.204.210/24, IPv6 auto
- em2: 192.168.100.65/24 (virtone.hmib.infn.it)

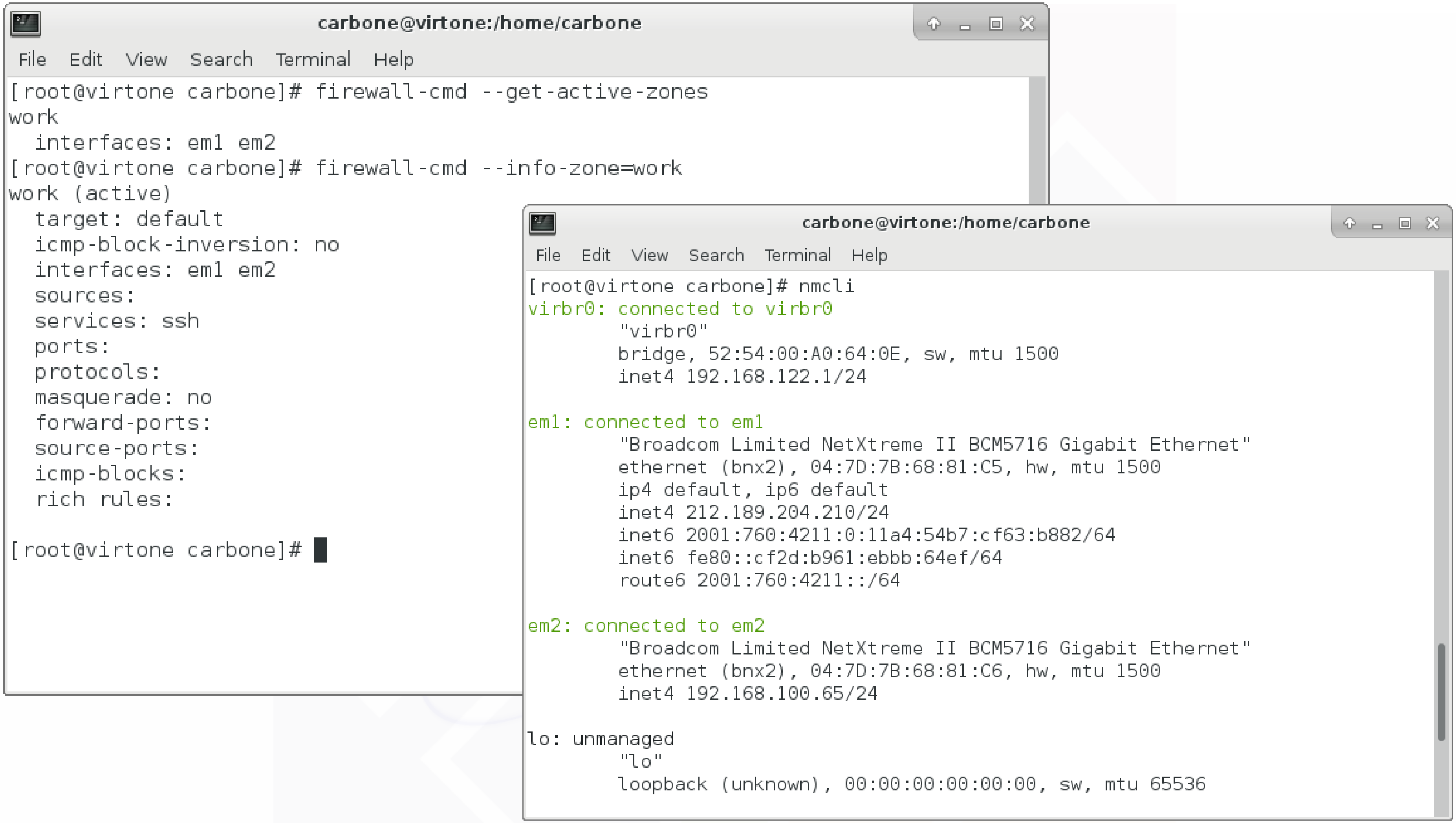
required configuration:

- ssh allowed from selected sources
- iperf/iperf3 allowed from selected sources
- http/https allowed from everywhere
- private network trusted
- 6201/tcp allowed from a single IP

a tentative solution

- em2 bound to *trusted* zone
- em1 bound to *work* zone
  - allow http/https
- create custom zone *netperf*
  - bind selected source(s)
  - allow iperf/iperf3
- create custom zone *mgmt*.
  - bind selected source(s)
  - allow ssh
- create *direct rule* for 6201/tcp access

starting point

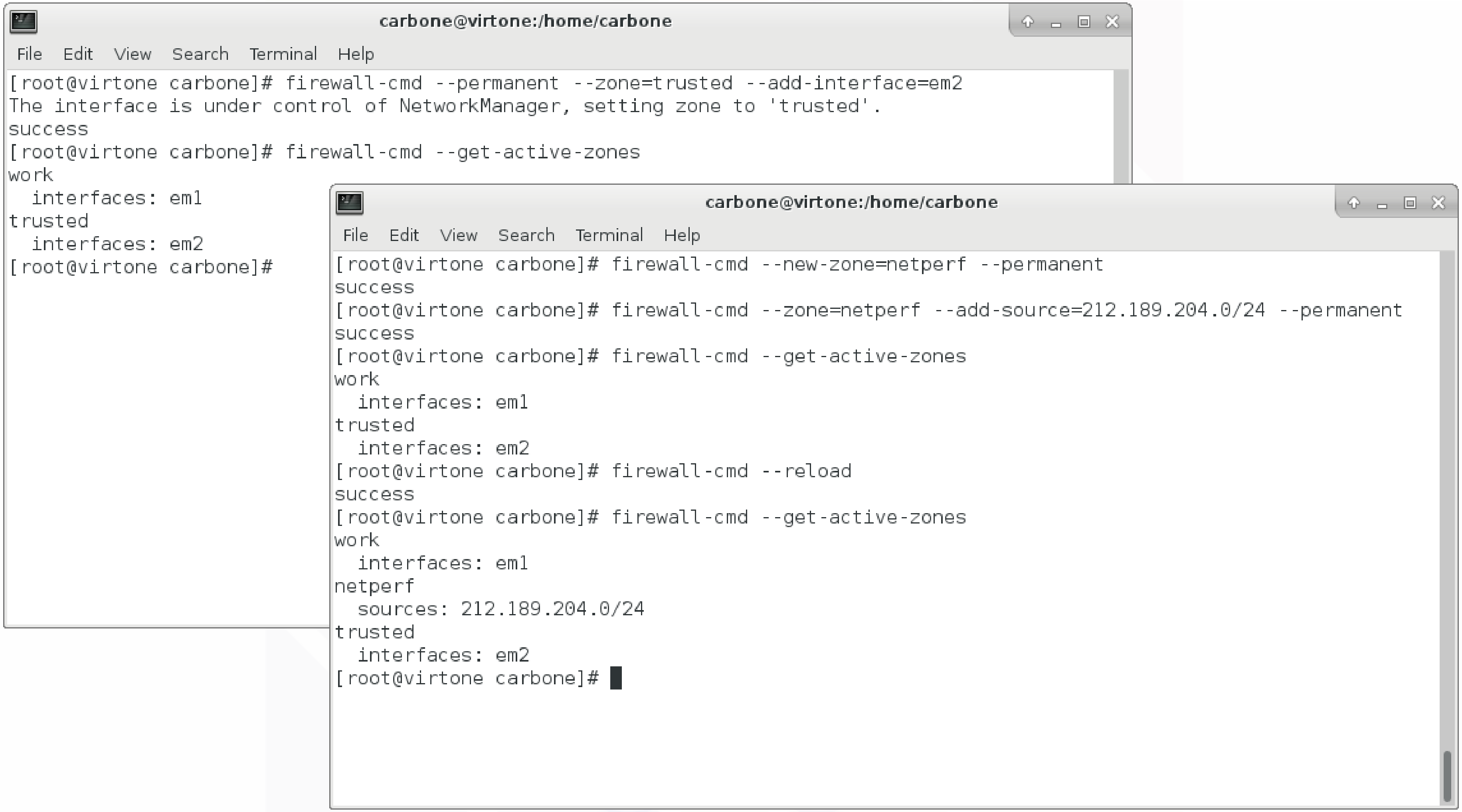

carbone@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --get-active-zones
work
  interfaces: em1 em2
[root@virtone carbone]# firewall-cmd --info-zone=work
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: em1 em2
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@virtone carbone]# 
```

carbone@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# nmcli
virbr0: connected to virbr0
        "virbr0"
        bridge, 52:54:00:A0:64:0E, sw, mtu 1500
        inet4 192.168.122.1/24

em1: connected to em1
        "Broadcom Limited NetXtreme II BCM5716 Gigabit Ethernet"
        ethernet (bnx2), 04:7D:7B:68:81:C5, hw, mtu 1500
        ip4 default, ip6 default
        inet4 212.189.204.210/24
        inet6 2001:760:4211:0:11a4:54b7:cf63:b882/64
        inet6 fe80::cf2d:b961:ebbb:64ef/64
        route6 2001:760:4211::/64

em2: connected to em2
        "Broadcom Limited NetXtreme II BCM5716 Gigabit Ethernet"
        ethernet (bnx2), 04:7D:7B:68:81:C6, hw, mtu 1500
        inet4 192.168.100.65/24

lo: unmanaged
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
```

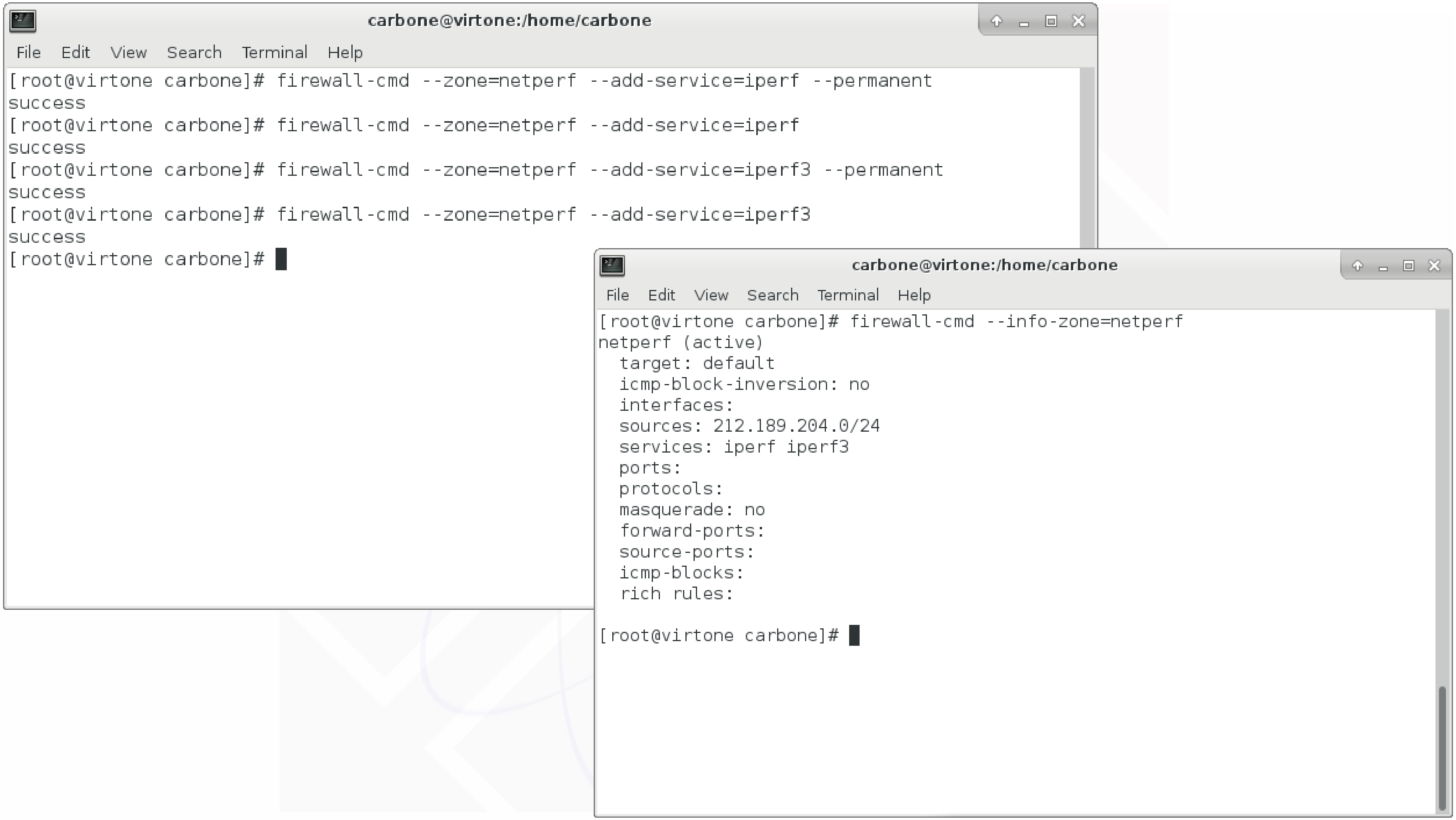
moving em2 & adding *netperf* zone


carbone@virtone:/home/carbone
File Edit View Search Terminal Help
carbone@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --permanent --zone=trusted --add-interface=em2
The interface is under control of NetworkManager, setting zone to 'trusted'.
success
[root@virtone carbone]# firewall-cmd --get-active-zones
work
  interfaces: em1
trusted
  interfaces: em2
[root@virtone carbone]#
```

```
[root@virtone carbone]# firewall-cmd --new-zone=netperf --permanent
success
[root@virtone carbone]# firewall-cmd --zone=netperf --add-source=212.189.204.0/24 --permanent
success
[root@virtone carbone]# firewall-cmd --get-active-zones
work
  interfaces: em1
trusted
  interfaces: em2
[root@virtone carbone]# firewall-cmd --reload
success
[root@virtone carbone]# firewall-cmd --get-active-zones
work
  interfaces: em1
netperf
  sources: 212.189.204.0/24
trusted
  interfaces: em2
[root@virtone carbone]# 
```

adding services

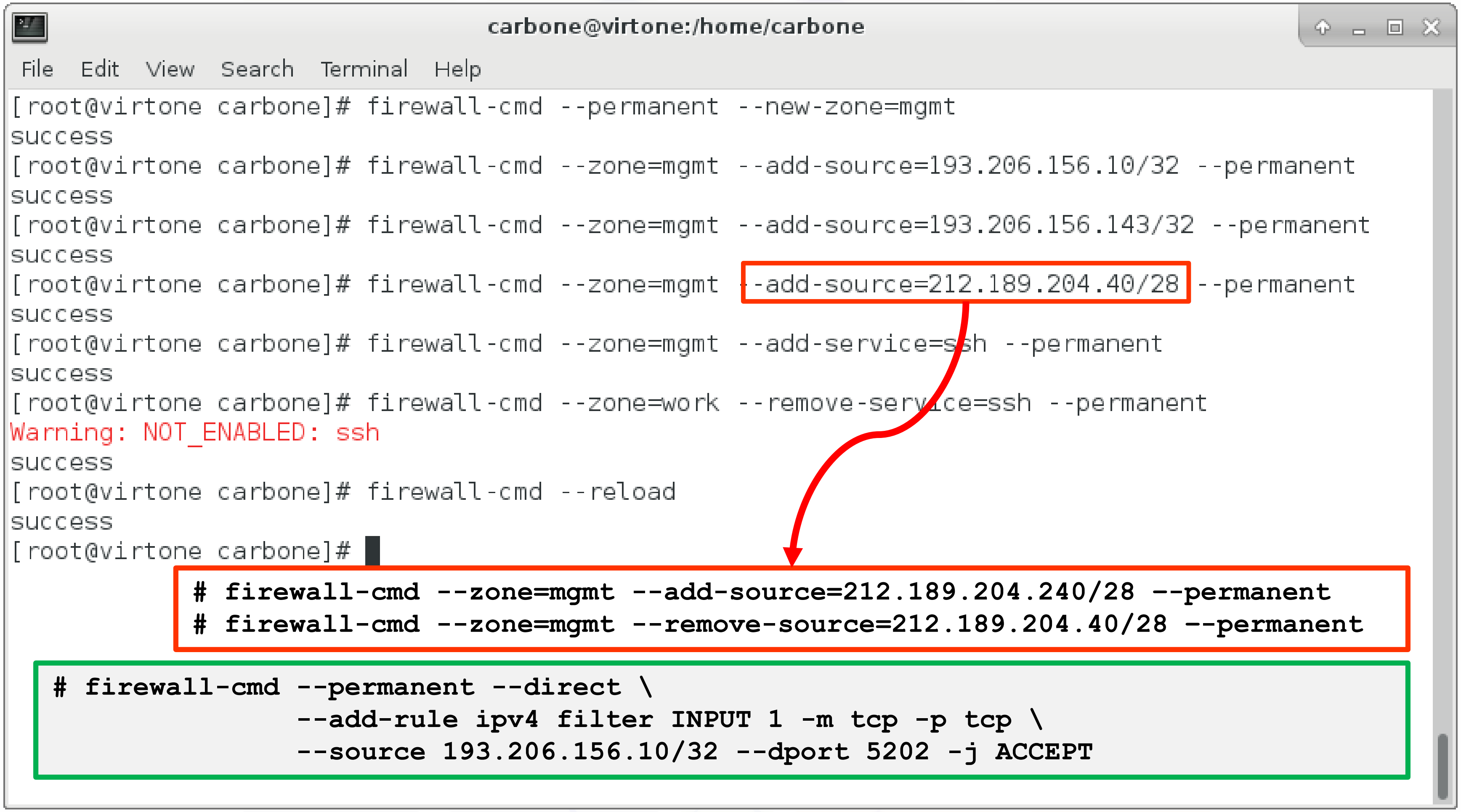

carbone@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --zone=netperf --add-service=iperf --permanent
success
[root@virtone carbone]# firewall-cmd --zone=netperf --add-service=iperf
success
[root@virtone carbone]# firewall-cmd --zone=netperf --add-service=iperf3 --permanent
success
[root@virtone carbone]# firewall-cmd --zone=netperf --add-service=iperf3
success
[root@virtone carbone]# 
```

carbone@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --info-zone=netperf
netperf (active)
  target: default
  icmp-block-inversion: no
  interfaces:
  sources: 212.189.204.0/24
  services: iperf iperf3
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@virtone carbone]# 
```

adding mgmt zone

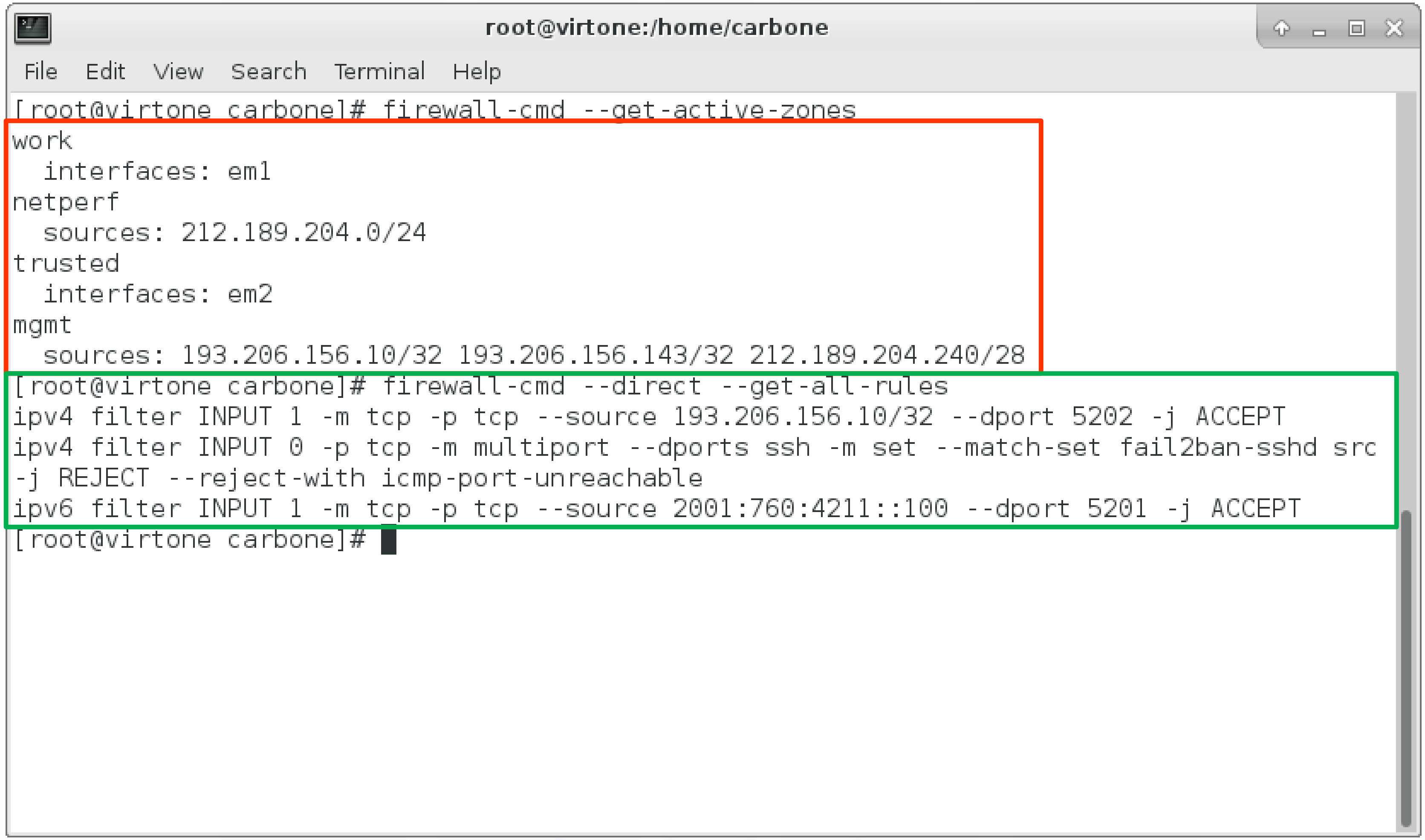

carbone@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --permanent --new-zone=mgmt
success
[root@virtone carbone]# firewall-cmd --zone=mgmt --add-source=193.206.156.10/32 --permanent
success
[root@virtone carbone]# firewall-cmd --zone=mgmt --add-source=193.206.156.143/32 --permanent
success
[root@virtone carbone]# firewall-cmd --zone=mgmt --add-source=212.189.204.40/28 --permanent
success
[root@virtone carbone]# firewall-cmd --zone=mgmt --add-service=ssh --permanent
success
[root@virtone carbone]# firewall-cmd --zone=work --remove-service=ssh --permanent
Warning: NOT_ENABLED: ssh
success
[root@virtone carbone]# firewall-cmd --reload
success
[root@virtone carbone]#
```

```
# firewall-cmd --zone=mgmt --add-source=212.189.204.240/28 --permanent
# firewall-cmd --zone=mgmt --remove-source=212.189.204.40/28 --permanent
```

```
# firewall-cmd --permanent --direct \
               --add-rule ipv4 filter INPUT 1 -m tcp -p tcp \
               --source 193.206.156.10/32 --dport 5202 -j ACCEPT
```

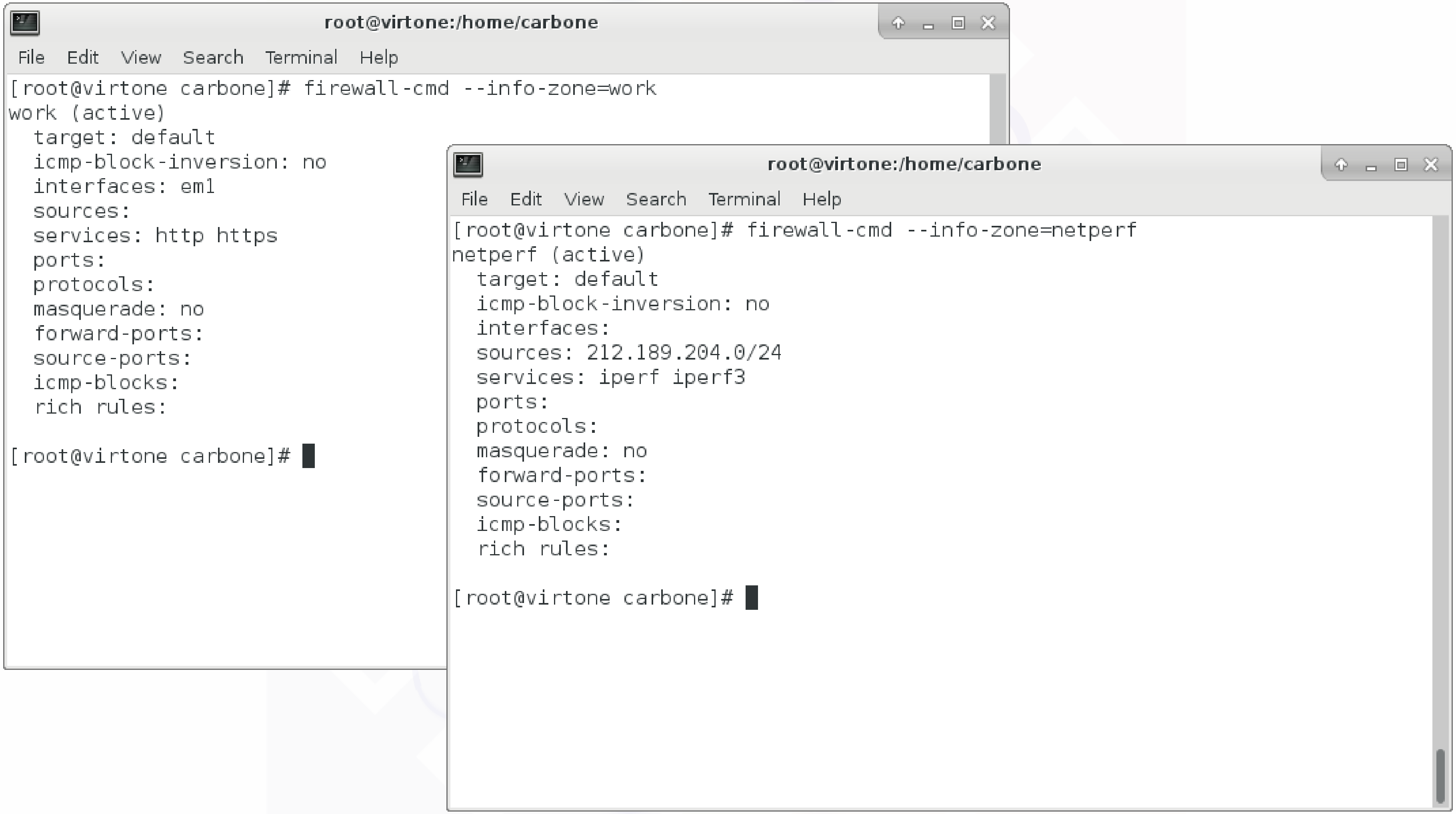
active zones & direct rules



root@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --get-active-zones
work
  interfaces: em1
netperf
  sources: 212.189.204.0/24
trusted
  interfaces: em2
mgmt
  sources: 193.206.156.10/32 193.206.156.143/32 212.189.204.240/28
[root@virtone carbone]# firewall-cmd --direct --get-all-rules
ipv4 filter INPUT 1 -m tcp -p tcp --source 193.206.156.10/32 --dport 5202 -j ACCEPT
ipv4 filter INPUT 0 -p tcp -m multiport --dports ssh -m set --match-set fail2ban-sshd src
-j REJECT --reject-with icmp-port-unreachable
ipv6 filter INPUT 1 -m tcp -p tcp --source 2001:760:4211::100 --dport 5201 -j ACCEPT
[root@virtone carbone]#
```

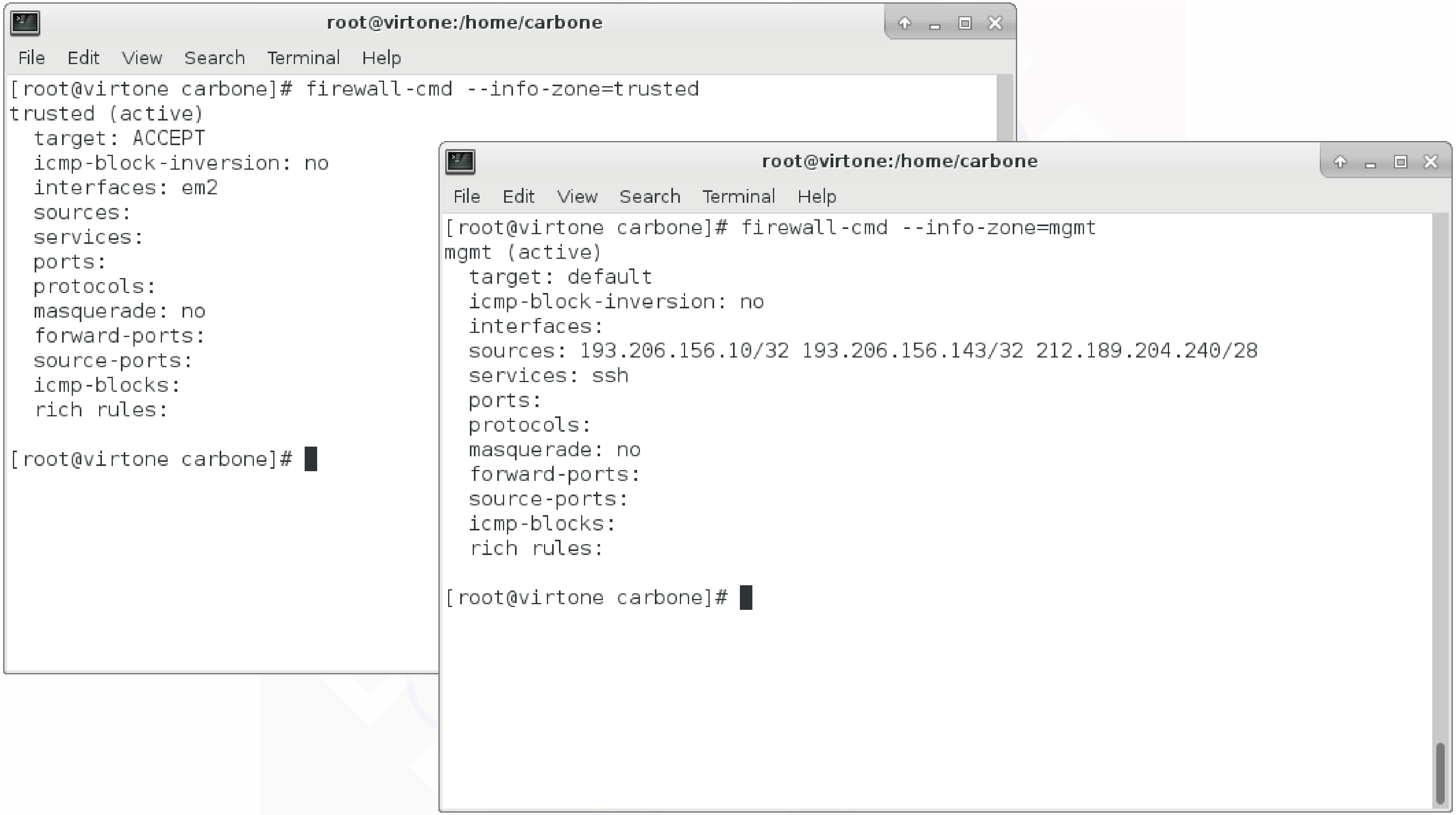
active zones detail


root@virtone:/home/carbone
File Edit View Search Terminal Help
root@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --info-zone=work
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: em1
  sources:
  services: http https
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@virtone carbone]#
```

```
[root@virtone carbone]# firewall-cmd --info-zone=netperf
netperf (active)
  target: default
  icmp-block-inversion: no
  interfaces:
  sources: 212.189.204.0/24
  services: iperf iperf3
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@virtone carbone]#
```

active zones detail


root@virtone:/home/carbone
File Edit View Search Terminal Help

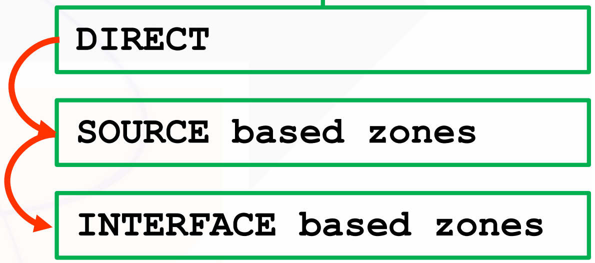
```
[root@virtone carbone]# firewall-cmd --info-zone=trusted
trusted (active)
  target: ACCEPT
  icmp-block-inversion: no
  interfaces: em2
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@virtone carbone]# 
```

root@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --info-zone=mgmt
mgmt (active)
  target: default
  icmp-block-inversion: no
  interfaces:
  sources: 193.206.156.10/32 193.206.156.143/32 212.189.204.240/28
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@virtone carbone]# 
```

underlying iptables layer

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
target          prot opt in    out  source     destination
ACCEPT                all  --  any  any  anywhere   anywhere   ctstate RELATED,ESTABLISHED
ACCEPT                all  --  lo   any  anywhere   anywhere
INPUT_direct          all  --  any  any  anywhere   anywhere
INPUT_ZONES_SOURCE    all  --  any  any  anywhere   anywhere
INPUT_ZONES           all  --  any  any  anywhere   anywhere
DROP                  all  --  any  any  anywhere   anywhere   ctstate INVALID
REJECT                all  --  any  any  anywhere   anywhere   reject-with icmp-host-prohibited

Chain INPUT_ZONES (1 references)
target      prot opt in     out  source     destination
IN_work     all  --  em1    any  anywhere   anywhere    [goto]
IN_trusted  all  --  em2    any  anywhere   anywhere
IN_work     all  --  +      any  anywhere   anywhere    [goto]

Chain INPUT_ZONES_SOURCE (1 references)
target      prot opt in   out  source                destination
IN_mgmt     all  --  any  any  ssire.mib.infn.it     anywhere   [goto]
IN_mgmt     all  --  any  any  groppone.mib.infn.it  anywhere   [goto]
IN_mgmt     all  --  any  any  212.189.204.240/28    anywhere   [goto]
IN_netperf  all  --  any  any  212.189.204.0/24      anywhere   [goto]
```



DIRECT

SOURCE based zones

INTERFACE based zones

underlying iptables layer

```
Chain INPUT_direct (1 references)
target     prot opt in   out  source               destination
REJECT     tcp  --  any  any  anywhere             anywhere
                              multiport dports ssh match-set fail2ban-sshd src
                              reject-with icmp-port-unreachable
ACCEPT     tcp  --  any  any  ssire.mib.infn.it  anywhere  tcp dpt:targus-getdata2

Chain IN_mgmt (3 references)
target         prot opt in      out     source          destination
IN_mgmt_log    all  --  any     any     anywhere       anywhere
IN_mgmt_deny   all  --  any     any     anywhere       anywhere
IN_mgmt_allow  all  --  any     any     anywhere       anywhere
ACCEPT         icmp --  any     any     anywhere       anywhere

Chain IN_mgmt_allow (1 references)
target     prot opt in     out     source         destination
ACCEPT     tcp  --  any    any     anywhere       anywhere  tcp dpt:ssh ctstate NEW

Chain IN_mgmt_deny (1 references)
target     prot opt in     out     source        destination
Chain IN_mgmt_log (1 references)
target     prot opt in     out     source        destination
```

underlying iptables layer

```
Chain IN_work (2 references)
target          prot opt in     out     source        destination
IN_work_log     all  --  any    any     anywhere      anywhere
IN_work_deny    all  --  any    any     anywhere      anywhere
IN_work_allow   all  --  any    any     anywhere      anywhere
ACCEPT          icmp --  any    any     anywhere      anywhere

Chain IN_work_allow (1 references)
target      prot opt in     out     source        destination
ACCEPT      tcp  --  any    any     anywhere      anywhere  tcp dpt:http ctstate NEW
ACCEPT      tcp  --  any    any     anywhere      anywhere  tcp dpt:https ctstate NEW

Chain IN_work_deny (1 references)
target      prot opt in     out     source      destination

Chain IN_work_log (1 references)
target      prot opt in     out     source      destination

Chain OUTPUT_direct (1 references)
target      prot opt in     out     source      destination
```

underlying iptables layer

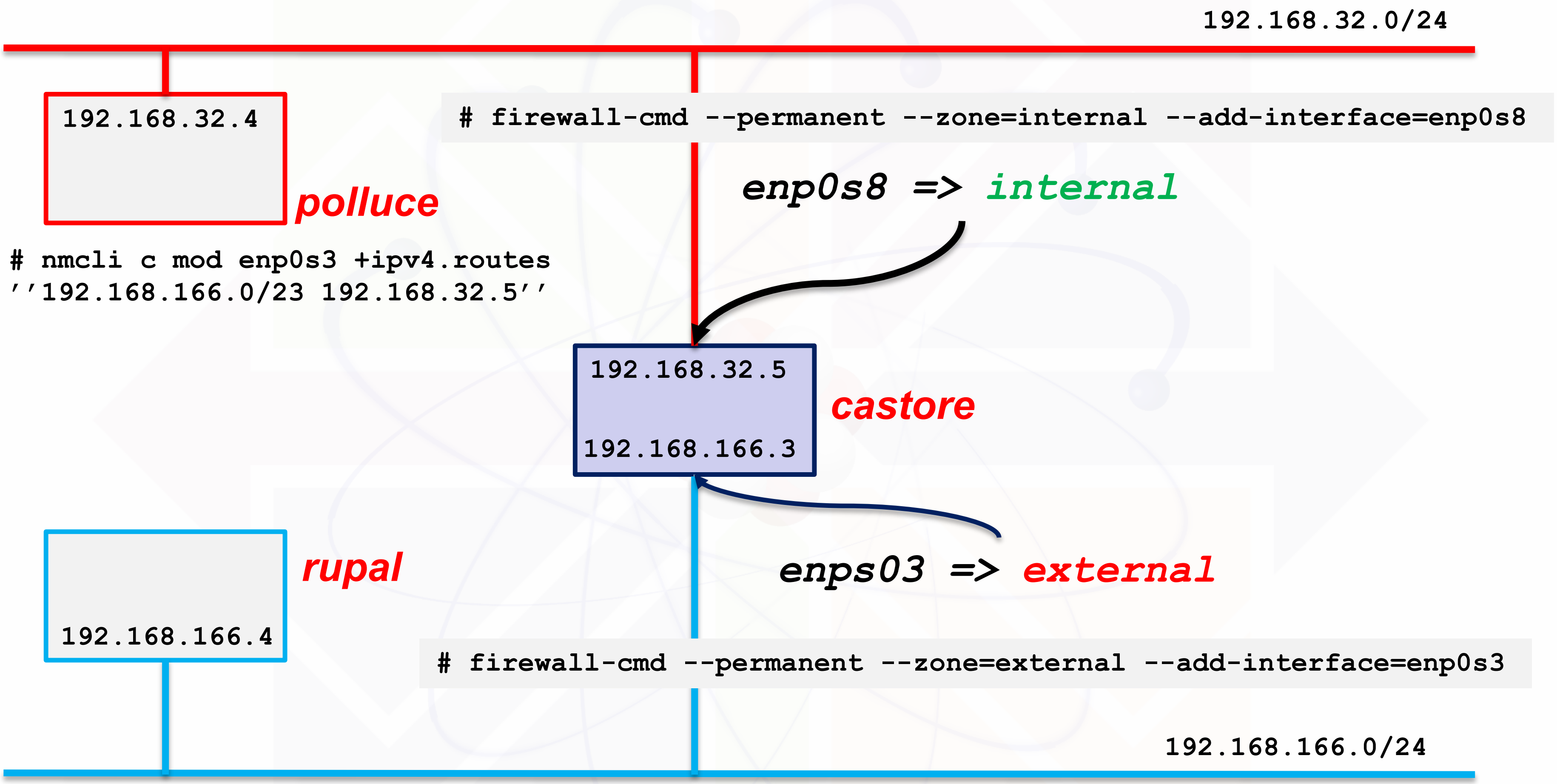
```
Chain IN_netperf (1 references)
target            prot opt in     out     source          destination
IN_netperf_log    all  --  any    any     anywhere        anywhere
IN_netperf_deny   all  --  any    any     anywhere        anywhere
IN_netperf_allow  all  --  any    any     anywhere        anywhere
ACCEPT            icmp --  any    any     anywhere        anywhere

Chain IN_netperf_allow (1 references)
target     prot opt in     out     source          destination
ACCEPT     tcp  --  any    any     anywhere        anywhere   tcp dpt:5001 ctstate NEW
ACCEPT     udp  --  any    any     anywhere        anywhere   udp dpt:5001 ctstate NEW
ACCEPT     tcp  --  any    any     anywhere        anywhere   tcp dpt:5201 ctstate NEW
ACCEPT     udp  --  any    any     anywhere        anywhere   udp dpt:5201 ctstate NEW

Chain IN_netperf_deny (1 references)
target     prot opt in     out     source        destination

Chain IN_netperf_log (1 references)
target     prot opt in     out     source        destination
```

Istituto Nazionale di Fisica Nucleare

underlying iptables layer

```
Chain IN_trusted (1 references)
target              prot opt in      out     source       destination
IN_trusted_log      all  --  any     any     anywhere     anywhere
IN_trusted_deny     all  --  any     any     anywhere     anywhere
IN_trusted_allow    all  --  any     any     anywhere     anywhere
ACCEPT              all  --  any     any     anywhere     anywhere

Chain IN_trusted_allow (1 references)
target     prot opt in     out     source       destination

Chain IN_trusted_deny (1 references)
target     prot opt in     out     source       destination

Chain IN_trusted_log (1 references)
target     prot opt in     out     source       destination
```

masquerading

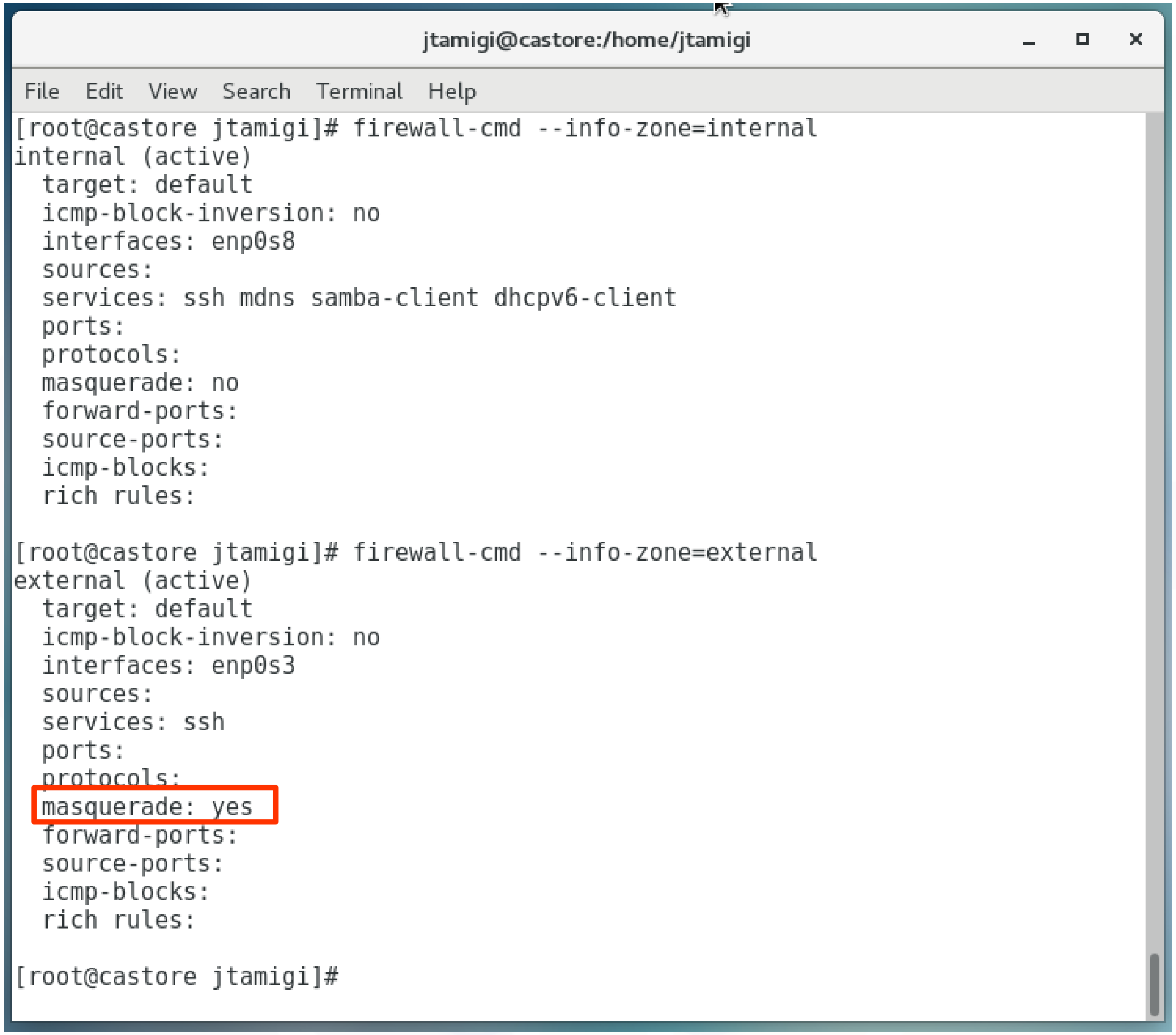


```
# firewall-cmd --permanent --zone=internal --add-interface=enp0s8
```

```
# nmcli c mod enp0s3 +ipv4.routes
''192.168.166.0/23 192.168.32.5''
```

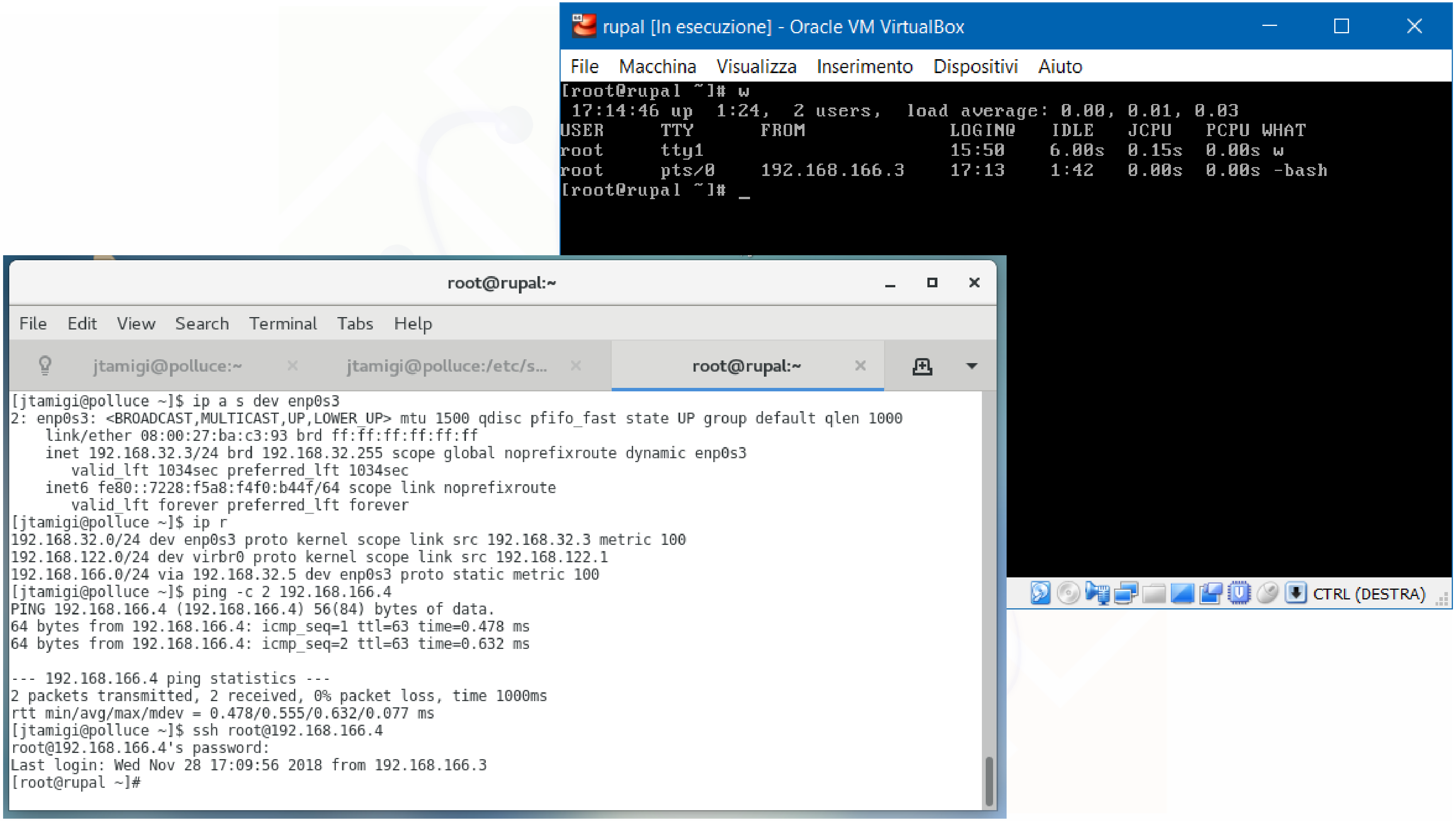
```
# firewall-cmd --permanent --zone=external --add-interface=enp0s3
```

masquerading - cont’d 1


jtamigi@castore:/home/jtamigi
File Edit View Search Terminal Help

```
[root@castore jtamigi]# firewall-cmd --info-zone=internal
internal (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s8
  sources:
  services: ssh mdns samba-client dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@castore jtamigi]# firewall-cmd --info-zone=external
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@castore jtamigi]#
```

masquerading - cont’d 2


rupal [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
CTRL (DESTRA)
root@rupal:~
File Edit View Search Terminal Tabs Help
jtamigi@polluce:~
jtamigi@polluce:/etc/s...
root@rupal:~

```
[root@rupal ~]# w
 17:14:46 up  1:24,  2 users,  load average: 0.00, 0.01, 0.03
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1                      15:50    6.00s  0.15s  0.00s w
root     pts/0    192.168.166.3    17:13    1:42   0.00s  0.00s -bash
[root@rupal ~]# _
```

```
[jtamigi@polluce ~]$ ip a s dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:ba:c3:93 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.3/24 brd 192.168.32.255 scope global noprefixroute dynamic enp0s3
       valid_lft 1034sec preferred_lft 1034sec
    inet6 fe80::7228:f5a8:f4f0:b44f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[jtamigi@polluce ~]$ ip r
192.168.32.0/24 dev enp0s3 proto kernel scope link src 192.168.32.3 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
192.168.166.0/24 via 192.168.32.5 dev enp0s3 proto static metric 100
[jtamigi@polluce ~]$ ping -c 2 192.168.166.4
PING 192.168.166.4 (192.168.166.4) 56(84) bytes of data.
64 bytes from 192.168.166.4: icmp_seq=1 ttl=63 time=0.478 ms
64 bytes from 192.168.166.4: icmp_seq=2 ttl=63 time=0.632 ms

--- 192.168.166.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.478/0.555/0.632/0.077 ms
[jtamigi@polluce ~]$ ssh root@192.168.166.4
root@192.168.166.4's password:
Last login: Wed Nov 28 17:09:56 2018 from 192.168.166.3
[root@rupal ~]#
```

smorgasboard

- ✓ **logging denied packets** to `/var/log/messages (--reload` not needed)

```
# firewall-cmd --get-log-denied
off
# firewall-cmd --set-log-denied=all (unicast|multicast|broadcast)
success
```

- ✓ **dropping all packets**: panic mode – all incoming and outgoing packets are dropped, active connections will be terminated after a period of inactivity (depending on individual session timeout)

```
# firewall-cmd --panic-on
success
# firewall-cmd --panic-off
success
# firewall-cmd --query-panic
no|yes
```

smorgasboard

- ✓ **reloading firewall** *not losing state information*

```
# firewall-cmd --reload
success
```

- ✓ **reloading firewall** *discarding state information and interrupting active connection*

```
# firewall-cmd --complete-reload
success
```

- ✓ **one shot runtime to permanent**: save active runtime configuration and overwrite permanent configuration with it. The way this is supposed to work is that when configuring firewalld you do runtime changes only and once you're happy with the configuration and you tested that it works the way you want, you save the configuration to disk. Note: was reported to be a litte bit buggy – as an alternative double-typing every command (with/without `--permanent` option) is annoying but works ☺

```
# firewall-cmd --runtime-to-permanent
```

firewall lockdown

locking down firewall: the firewall configuration can be locked so that either no applications or only applications in the *lockdown whitelist* are able to request firewall changes. This behavior is controlled by the `Lockdown` parameter of the firewalld configuration file `/etc/firewalld/firewalld.conf`:

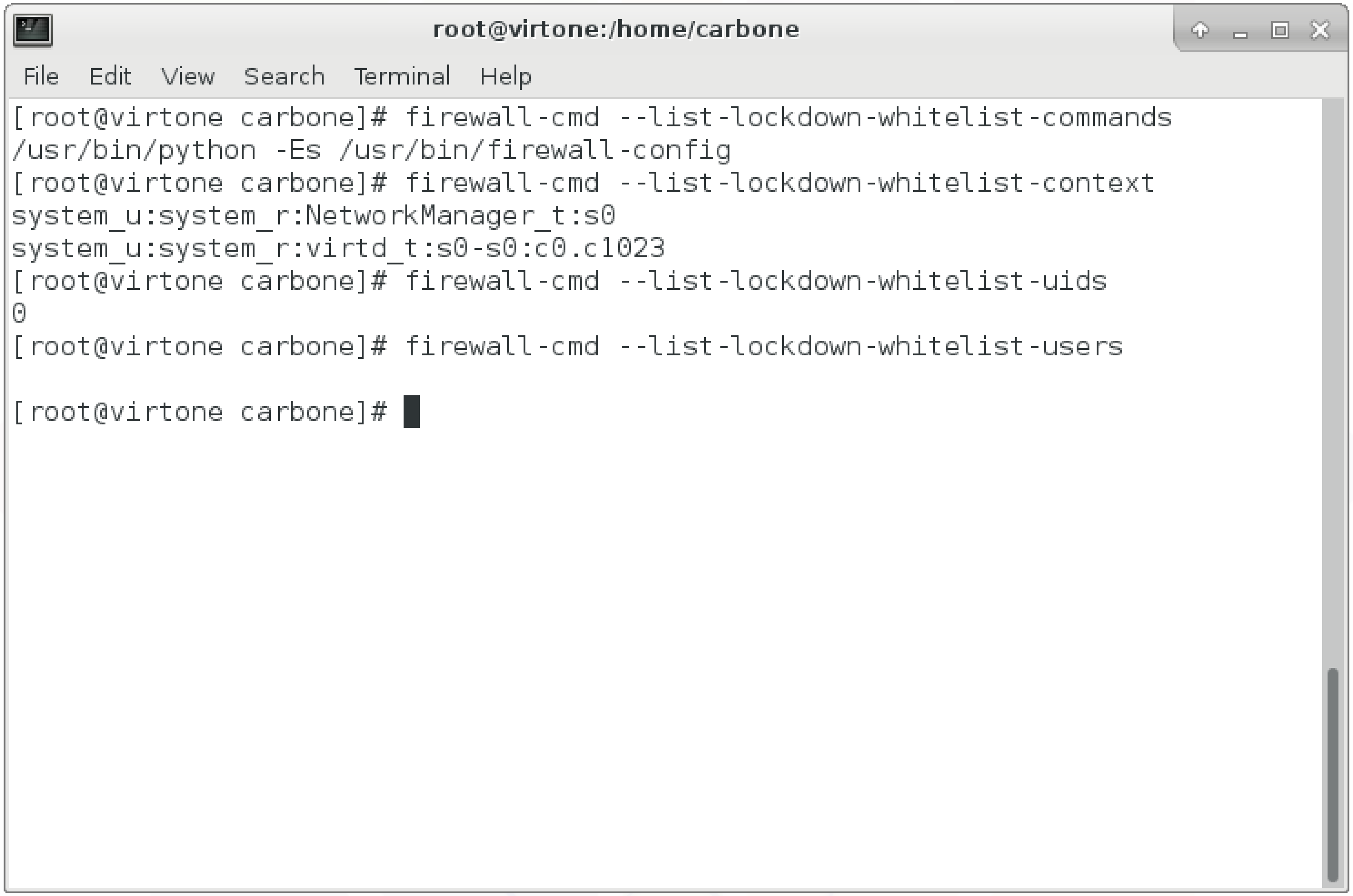
```
# If set to enabled, changes with the D-Bus interface will be limited
# to applications that are listed in the lockdown whitelist.
# The lockdown whitelist file is lockdown-whitelist.xml
# Default: no
Lockdown=no
```

or by firewall-cmd `Lockdown` and `Lockdown Whitelist` options:

```
[--permanent] --lockdown-on | --lockdown-off | --query-lockdown
[--permanent] --list-lockdown-whitelist-commands=command
[--permanent] --add-lockdown-whitelist-commands=command
[--permanent] --remove-lockdown-whitelist-commands=command
[--permanent] --query-lockdown-whitelist-commands=command
[--permanent] --list-lockdown-whitelist-context
[--permanent] --add-lockdown-whitelist-context=context
…
```

more on this later… ☺

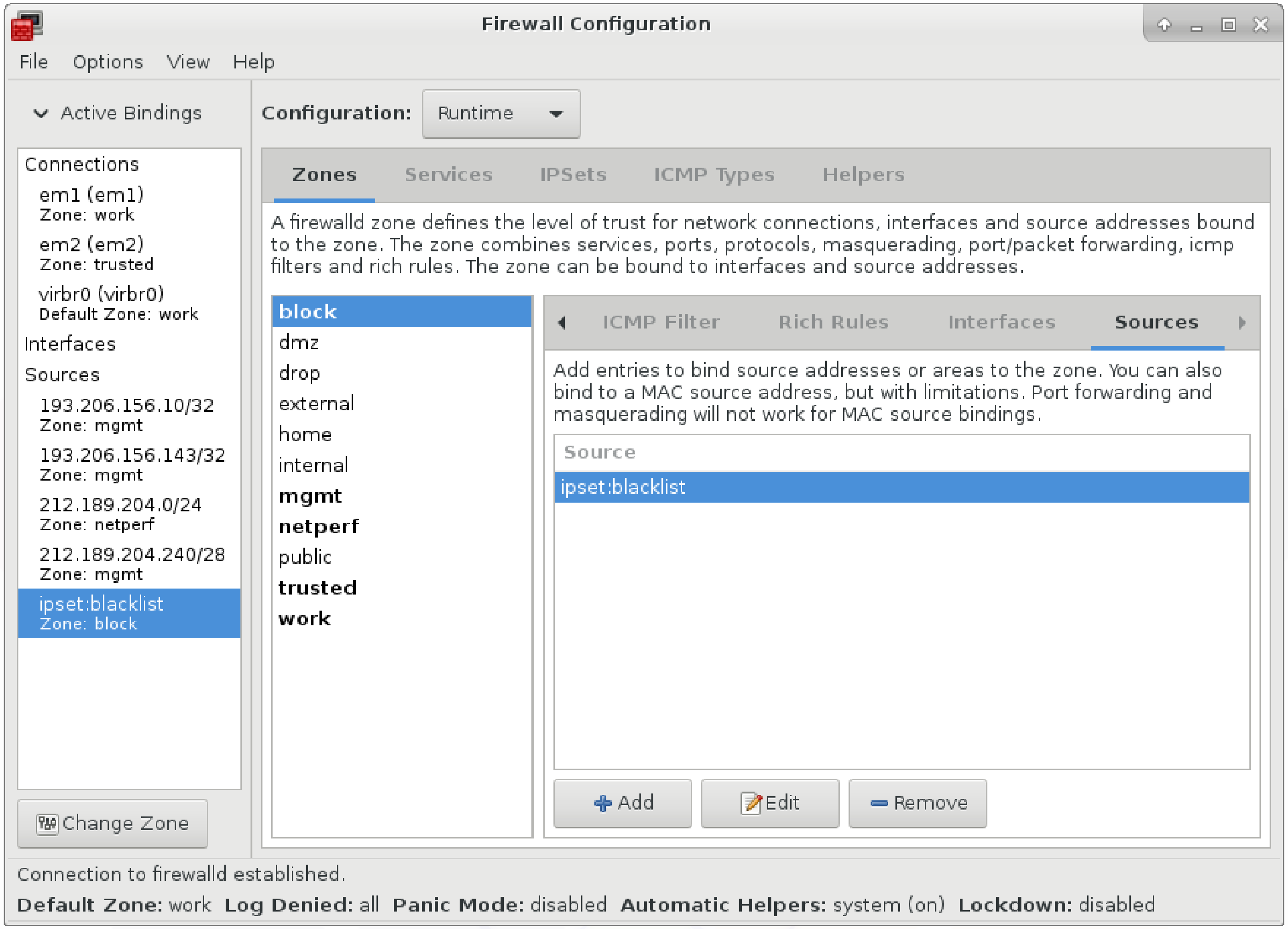
firewall lockdown



root@virtone:/home/carbone
File Edit View Search Terminal Help

```
[root@virtone carbone]# firewall-cmd --list-lockdown-whitelist-commands
/usr/bin/python -Es /usr/bin/firewall-config
[root@virtone carbone]# firewall-cmd --list-lockdown-whitelist-context
system_u:system_r:NetworkManager_t:s0
system_u:system_r:virtd_t:s0-s0:c0.c1023
[root@virtone carbone]# firewall-cmd --list-lockdown-whitelist-uids
0
[root@virtone carbone]# firewall-cmd --list-lockdown-whitelist-users

[root@virtone carbone]# 
```

firewall-config



Connections
em1 (em1)
Zone: work
em2 (em2)
Zone: trusted
virbr0 (virbr0)
Default Zone: work
Interfaces
Sources
193.206.156.10/32
Zone: mgmt
193.206.156.143/32
Zone: mgmt
212.189.204.0/24
Zone: netperf
212.189.204.240/28
Zone: mgmt
ipset:blacklist
Zone: block

A firewalld zone defines the level of trust for network connections, interfaces and source addresses bound to the zone. The zone combines services, ports, protocols, masquerading, port/packet forwarding, icmp filters and rich rules. The zone can be bound to interfaces and source addresses.

block
dmz
drop
external
home
internal
mgmt
netperf
public
trusted
work

Add entries to bind source addresses or areas to the zone. You can also bind to a MAC source address, but with limitations. Port forwarding and masquerading will not work for MAC source bindings.

ipset:blacklist

Connection to firewalld established.

Default Zone: work **Log Denied:** all **Panic Mode:** disabled **Automatic Helpers:** system (on) **Lockdown:** disabled