Version : **2026.01**

Dernière mise-à-jour : 2025/11/27 11:19

LDF402 - Netfilter et Firewalld

Contenu du Module

Les Problématiques

L'IP Spoofing

L'IP Spoofing consiste en faire croire à un serveur que sa machine possède une adresse IP autre que celle réellement attribuée. Le but de cette opération est de se placer en tant que point de passage obligatoire des paquets envoyés entre le serveur et le vrai *propriétaire* de l'adresse IP spoofée. Le mécanisme est la suivante :

- L'attaquant change son adresse IP en prenant une à laquelle le serveur cible fera confiance,
- L'attaquant envoie une requête au serveur en stipulant une route de communication qui passe par l'adresse IP réelle de l'attaquant,
- L'attaquant reprend son adresse IP réelle,
- Le serveur accepte la requête car elle provient d'une adresse IP à laquelle il peux faire confiance et renvoie une réponse en utilisant la route spécifiée par l'attaquant,
- Le client utilise la route spécifiée par l'attaquant pour répondre au serveur.

Déni de Service (DoS)

Une attaque de déni de service consiste à rendre inopérable une machine en lui envoyant une grande quantité de données inutiles. Un exemple de ce type d'attaque s'appelle un *ping flood* :

- L'attaquant prend l'adresse IP de sa cible,
- Il envoie ensuite un ping à une machine de diffusion,
- La machine de diffusion envoie ce même ping à un grand nombre de clients en spécifiant l'origine de la requête,
- L'attaquant reprend son adresse IP d'origine,
- Tous les clients renvoient une réponse au ping *en même temps* à la cible.

SYN Flooding

Le **SYN Flooding**, aussi appelé un *SYN-ACK Attack*, consiste à envoyer vers une cible de multiples paquets **SYN** très rapidement. La cible répond à chaque paquet reçu avec un paquet **ACK** et attend une réponse **ACK** de l'attaquant. A ce stade pour chaque ACK renvoyé par la cible, une connexion dite *semi-ouverte* existe entre les deux machines. La cible doit réserver une petite partie de sa mémoire pour chaque connexion semi-ouverte jusqu'au *time-out* de ladite semi-connexion. Si l'attaquant envoie très rapidement des paquets SYN, le système de time-out n'a pas la possibilité d'expirer les semi-connexions précédentes. Dans ce cas la mémoire de la cible se remplit et on obtient un *buffer overflow*.

Flood

Le **Flood** consiste à envoyer très rapidement de gros paquets **ICMP** vers la cible.

La Contre-Mesure

La contre-mesure est principalement l'utilisation d'un pare-feu.

LAB #1 - La Configuration de firewalld

firewalld est à Netfilter ce que NetworkManager est au réseau. firewalld utilise des **zones** - des jeux de règles pré-définis dans lesquels sont placés les interfaces :

- **trusted** - un réseau fiable. Dans ce cas tous les ports sont autorisés,
- **work**, **home**, **internal** - un réseau partiellement fiable. Dans ce cas quelques ports sont autorisés,
- **dmz**, **public**, **external** - un réseau non fiable. Dans ce cas peu de ports sont autorisés,
- **block**, **drop** - tout est interdit. La zone drop n'envoie pas de messages d'erreurs.

Important - Une interface ne peut être que dans une zone à la fois tandis que plusieurs interfaces peuvent être dans la même zone.

Sous Debian 12, firewalld n'est pas installé par défaut :

```
root@debian12:~# apt-get -y install firewalld
```

Le service firewalld est déjà lancé et activé :

```
root@debian12:~# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-11-26 15:37:04 CET; 22s ago
       Docs: man:firewalld(1)
   Main PID: 3136 (firewalld)
```

```
      Tasks: 2 (limit: 19123)
     Memory: 29.2M
        CPU: 374ms
     CGroup: /system.slice/firewalld.service
             └─3136 /usr/bin/python3 /usr/sbin/firewalld --nofork --nopid

Nov 26 15:37:04 debian12 systemd[1]: Starting firewalld.service - firewalld - dynamic firewall daemon...
Nov 26 15:37:04 debian12 systemd[1]: Started firewalld.service - firewalld - dynamic firewall daemon.
```

1.1 - La Configuration de Base de firewalld

La configuration par défaut de firewalld se trouve dans **/usr/lib/firewalld** :

```
root@debian12:~# ls -lR /usr/lib/firewalld/
/usr/lib/firewalld/:
total 32
drwxr-xr-x 2 root root  4096 Nov 26 15:36 helpers
drwxr-xr-x 2 root root  4096 Nov 26 15:36 icmptypes
drwxr-xr-x 2 root root  4096 Nov 26 15:36 ipsets
drwxr-xr-x 2 root root  4096 Nov 26 15:36 policies
drwxr-xr-x 2 root root 12288 Nov 26 15:36 services
drwxr-xr-x 2 root root  4096 Nov 26 15:36 zones

/usr/lib/firewalld/helpers:
total 52
-rw-r--r-- 1 root root 125 Aug  2  2023 amanda.xml
-rw-r--r-- 1 root root 119 Aug  2  2023 ftp.xml
-rw-r--r-- 1 root root  85 Aug  2  2023 h323.xml
-rw-r--r-- 1 root root 134 Aug  2  2023 irc.xml
-rw-r--r-- 1 root root 141 Aug  2  2023 netbios-ns.xml
-rw-r--r-- 1 root root 136 Aug  2  2023 pptp.xml
-rw-r--r-- 1 root root  90 Aug  2  2023 proto-gre.xml
-rw-r--r-- 1 root root 122 Aug  2  2023 Q.931.xml
```

```
-rw-r--r-- 1 root root 122 Aug  2  2023 RAS.xml
-rw-r--r-- 1 root root 122 Aug  2  2023 sane.xml
-rw-r--r-- 1 root root 158 Aug  2  2023 sip.xml
-rw-r--r-- 1 root root 135 Aug  2  2023 snmp.xml
-rw-r--r-- 1 root root 120 Aug  2  2023 tftp.xml

/usr/lib/firewalld/icmptypes:
total 180
-rw-r--r-- 1 root root 385 Aug  2  2023 address-unreachable.xml
-rw-r--r-- 1 root root 258 Aug  2  2023 bad-header.xml
-rw-r--r-- 1 root root 293 Aug  2  2023 beyond-scope.xml
-rw-r--r-- 1 root root 279 Aug  2  2023 communication-prohibited.xml
-rw-r--r-- 1 root root 222 Aug  2  2023 destination-unreachable.xml
-rw-r--r-- 1 root root 173 Aug  2  2023 echo-reply.xml
-rw-r--r-- 1 root root 210 Aug  2  2023 echo-request.xml
-rw-r--r-- 1 root root 261 Aug  2  2023 failed-policy.xml
-rw-r--r-- 1 root root 280 Aug  2  2023 fragmentation-needed.xml
-rw-r--r-- 1 root root 266 Aug  2  2023 host-precedence-violation.xml
-rw-r--r-- 1 root root 257 Aug  2  2023 host-prohibited.xml
-rw-r--r-- 1 root root 242 Aug  2  2023 host-redirect.xml
-rw-r--r-- 1 root root 239 Aug  2  2023 host-unknown.xml
-rw-r--r-- 1 root root 247 Aug  2  2023 host-unreachable.xml
-rw-r--r-- 1 root root 229 Aug  2  2023 ip-header-bad.xml
-rw-r--r-- 1 root root 355 Aug  2  2023 neighbour-advertisement.xml
-rw-r--r-- 1 root root 457 Aug  2  2023 neighbour-solicitation.xml
-rw-r--r-- 1 root root 250 Aug  2  2023 network-prohibited.xml
-rw-r--r-- 1 root root 248 Aug  2  2023 network-redirect.xml
-rw-r--r-- 1 root root 239 Aug  2  2023 network-unknown.xml
-rw-r--r-- 1 root root 247 Aug  2  2023 network-unreachable.xml
-rw-r--r-- 1 root root 239 Aug  2  2023 no-route.xml
-rw-r--r-- 1 root root 328 Aug  2  2023 packet-too-big.xml
-rw-r--r-- 1 root root 225 Aug  2  2023 parameter-problem.xml
-rw-r--r-- 1 root root 233 Aug  2  2023 port-unreachable.xml
-rw-r--r-- 1 root root 256 Aug  2  2023 precedence-cutoff.xml
```

```
-rw-r--r-- 1 root root 249 Aug  2  2023 protocol-unreachable.xml
-rw-r--r-- 1 root root 185 Aug  2  2023 redirect.xml
-rw-r--r-- 1 root root 244 Aug  2  2023 reject-route.xml
-rw-r--r-- 1 root root 241 Aug  2  2023 required-option-missing.xml
-rw-r--r-- 1 root root 227 Aug  2  2023 router-advertisement.xml
-rw-r--r-- 1 root root 223 Aug  2  2023 router-solicitation.xml
-rw-r--r-- 1 root root 248 Aug  2  2023 source-quench.xml
-rw-r--r-- 1 root root 236 Aug  2  2023 source-route-failed.xml
-rw-r--r-- 1 root root 253 Aug  2  2023 time-exceeded.xml
-rw-r--r-- 1 root root 233 Aug  2  2023 timestamp-reply.xml
-rw-r--r-- 1 root root 228 Aug  2  2023 timestamp-request.xml
-rw-r--r-- 1 root root 258 Aug  2  2023 tos-host-redirect.xml
-rw-r--r-- 1 root root 257 Aug  2  2023 tos-host-unreachable.xml
-rw-r--r-- 1 root root 272 Aug  2  2023 tos-network-redirect.xml
-rw-r--r-- 1 root root 269 Aug  2  2023 tos-network-unreachable.xml
-rw-r--r-- 1 root root 293 Aug  2  2023 ttl-zero-during-reassembly.xml
-rw-r--r-- 1 root root 256 Aug  2  2023 ttl-zero-during-transit.xml
-rw-r--r-- 1 root root 259 Aug  2  2023 unknown-header-type.xml
-rw-r--r-- 1 root root 249 Aug  2  2023 unknown-option.xml

/usr/lib/firewalld/ipsets:
total 4
-rw-r--r-- 1 root root 29 Aug  2  2023 README.md

/usr/lib/firewalld/policies:
total 4
-rw-r--r-- 1 root root 649 Aug  2  2023 allow-host-ipv6.xml

/usr/lib/firewalld/services:
total 884
-rw-r--r-- 1 root root 352 Aug  2  2023 afp.xml
-rw-r--r-- 1 root root 399 Aug  2  2023 amanda-client.xml
-rw-r--r-- 1 root root 427 Aug  2  2023 amanda-k5-client.xml
-rw-r--r-- 1 root root 283 Aug  2  2023 amqps.xml
```

```
-rw-r--r-- 1 root root 273 Aug  2  2023 amqp.xml
-rw-r--r-- 1 root root 285 Aug  2  2023 apcupsd.xml
-rw-r--r-- 1 root root 301 Aug  2  2023 audit.xml
-rw-r--r-- 1 root root 436 Aug  2  2023 ausweisapp2.xml
-rw-r--r-- 1 root root 320 Aug  2  2023 bacula-client.xml
-rw-r--r-- 1 root root 346 Aug  2  2023 bacula.xml
-rw-r--r-- 1 root root 390 Aug  2  2023 bareos-director.xml
-rw-r--r-- 1 root root 255 Aug  2  2023 bareos-filedaemon.xml
-rw-r--r-- 1 root root 316 Aug  2  2023 bareos-storage.xml
-rw-r--r-- 1 root root 429 Aug  2  2023 bb.xml
-rw-r--r-- 1 root root 339 Aug  2  2023 bgp.xml
-rw-r--r-- 1 root root 275 Aug  2  2023 bitcoin-rpc.xml
-rw-r--r-- 1 root root 307 Aug  2  2023 bitcoin-testnet-rpc.xml
-rw-r--r-- 1 root root 281 Aug  2  2023 bitcoin-testnet.xml
-rw-r--r-- 1 root root 244 Aug  2  2023 bitcoin.xml
-rw-r--r-- 1 root root 410 Aug  2  2023 bittorrent-lsd.xml
-rw-r--r-- 1 root root 222 Aug  2  2023 ceph-exporter.xml
-rw-r--r-- 1 root root 294 Aug  2  2023 ceph-mon.xml
-rw-r--r-- 1 root root 329 Aug  2  2023 ceph.xml
-rw-r--r-- 1 root root 168 Aug  2  2023 cfengine.xml
-rw-r--r-- 1 root root 234 Aug  2  2023 checkmk-agent.xml
-rw-r--r-- 1 root root 211 Aug  2  2023 cockpit.xml
-rw-r--r-- 1 root root 296 Aug  2  2023 collectd.xml
-rw-r--r-- 1 root root 260 Aug  2  2023 condor-collector.xml
-rw-r--r-- 1 root root 343 Aug  2  2023 cratedb.xml
-rw-r--r-- 1 root root 296 Aug  2  2023 ctdb.xml
-rw-r--r-- 1 root root 981 Aug  2  2023 dds-multicast.xml
-rw-r--r-- 1 root root 947 Aug  2  2023 dds-unicast.xml
-rw-r--r-- 1 root root 574 Aug  2  2023 dds.xml
-rw-r--r-- 1 root root 305 Aug  2  2023 dhcpv6-client.xml
-rw-r--r-- 1 root root 234 Aug  2  2023 dhcpv6.xml
-rw-r--r-- 1 root root 227 Aug  2  2023 dhcp.xml
-rw-r--r-- 1 root root 205 Aug  2  2023 distcc.xml
-rw-r--r-- 1 root root 318 Aug  2  2023 dns-over-tls.xml
```

```
-rw-r--r-- 1 root root 346 Aug  2  2023 dns.xml
-rw-r--r-- 1 root root 374 Aug  2  2023 docker-registry.xml
-rw-r--r-- 1 root root 391 Aug  2  2023 docker-swarm.xml
-rw-r--r-- 1 root root 228 Aug  2  2023 dropbox-lansync.xml
-rw-r--r-- 1 root root 338 Aug  2  2023 elasticsearch.xml
-rw-r--r-- 1 root root 304 Aug  2  2023 etcd-client.xml
-rw-r--r-- 1 root root 304 Aug  2  2023 etcd-server.xml
-rw-r--r-- 1 root root 224 Aug  2  2023 finger.xml
-rw-r--r-- 1 root root 270 Aug  2  2023 foreman-proxy.xml
-rw-r--r-- 1 root root 408 Aug  2  2023 foreman.xml
-rw-r--r-- 1 root root 709 Aug  2  2023 freeipa-4.xml
-rw-r--r-- 1 root root 489 Aug  2  2023 freeipa-ldaps.xml
-rw-r--r-- 1 root root 488 Aug  2  2023 freeipa-ldap.xml
-rw-r--r-- 1 root root 242 Aug  2  2023 freeipa-replication.xml
-rw-r--r-- 1 root root 657 Aug  2  2023 freeipa-trust.xml
-rw-r--r-- 1 root root 361 Aug  2  2023 ftp.xml
-rw-r--r-- 1 root root 292 Aug  2  2023 galera.xml
-rw-r--r-- 1 root root 184 Aug  2  2023 ganglia-client.xml
-rw-r--r-- 1 root root 176 Aug  2  2023 ganglia-master.xml
-rw-r--r-- 1 root root 212 Aug  2  2023 git.xml
-rw-r--r-- 1 root root 406 Aug  2  2023 gpsd.xml
-rw-r--r-- 1 root root 218 Aug  2  2023 grafana.xml
-rw-r--r-- 1 root root 119 Aug  2  2023 gre.xml
-rw-r--r-- 1 root root 608 Aug  2  2023 high-availability.xml
-rw-r--r-- 1 root root 336 Aug  2  2023 http3.xml
-rw-r--r-- 1 root root 448 Aug  2  2023 https.xml
-rw-r--r-- 1 root root 353 Aug  2  2023 http.xml
-rw-r--r-- 1 root root 293 Aug  2  2023 ident.xml
-rw-r--r-- 1 root root 372 Aug  2  2023 imaps.xml
-rw-r--r-- 1 root root 327 Aug  2  2023 imap.xml
-rw-r--r-- 1 root root 315 Aug  2  2023 ipfs.xml
-rw-r--r-- 1 root root 454 Aug  2  2023 ipp-client.xml
-rw-r--r-- 1 root root 427 Aug  2  2023 ipp.xml
-rw-r--r-- 1 root root 895 Aug  2  2023 ipsec.xml
```

```
-rw-r--r-- 1 root root 255 Aug  2  2023 ircs.xml
-rw-r--r-- 1 root root 247 Aug  2  2023 irc.xml
-rw-r--r-- 1 root root 264 Aug  2  2023 iscsi-target.xml
-rw-r--r-- 1 root root 358 Aug  2  2023 isns.xml
-rw-r--r-- 1 root root 213 Aug  2  2023 jenkins.xml
-rw-r--r-- 1 root root 182 Aug  2  2023 kadmin.xml
-rw-r--r-- 1 root root 272 Aug  2  2023 kdeconnect.xml
-rw-r--r-- 1 root root 233 Aug  2  2023 kerberos.xml
-rw-r--r-- 1 root root 384 Aug  2  2023 kibana.xml
-rw-r--r-- 1 root root 249 Aug  2  2023 klogin.xml
-rw-r--r-- 1 root root 221 Aug  2  2023 kpasswd.xml
-rw-r--r-- 1 root root 182 Aug  2  2023 kprop.xml
-rw-r--r-- 1 root root 242 Aug  2  2023 kshell.xml
-rw-r--r-- 1 root root 308 Aug  2  2023 kube-apiserver.xml
-rw-r--r-- 1 root root 204 Aug  2  2023 kube-api.xml
-rw-r--r-- 1 root root 289 Aug  2  2023 kube-controller-manager-secure.xml
-rw-r--r-- 1 root root 280 Aug  2  2023 kube-controller-manager.xml
-rw-r--r-- 1 root root 560 Aug  2  2023 kube-control-plane-secure.xml
-rw-r--r-- 1 root root 537 Aug  2  2023 kube-control-plane.xml
-rw-r--r-- 1 root root 244 Aug  2  2023 kubelet-readonly.xml
-rw-r--r-- 1 root root 212 Aug  2  2023 kubelet-worker.xml
-rw-r--r-- 1 root root 239 Aug  2  2023 kubelet.xml
-rw-r--r-- 1 root root 224 Aug  2  2023 kube-nodeport-services.xml
-rw-r--r-- 1 root root 328 Aug  2  2023 kube-scheduler-secure.xml
-rw-r--r-- 1 root root 319 Aug  2  2023 kube-scheduler.xml
-rw-r--r-- 1 root root 374 Aug  2  2023 kube-worker.xml
-rw-r--r-- 1 root root 232 Aug  2  2023 ldaps.xml
-rw-r--r-- 1 root root 199 Aug  2  2023 ldap.xml
-rw-r--r-- 1 root root 385 Aug  2  2023 libvirt-tls.xml
-rw-r--r-- 1 root root 389 Aug  2  2023 libvirt.xml
-rw-r--r-- 1 root root 269 Aug  2  2023 lightning-network.xml
-rw-r--r-- 1 root root 468 Aug  2  2023 llmnr-client.xml
-rw-r--r-- 1 root root 410 Aug  2  2023 llmnr-tcp.xml
-rw-r--r-- 1 root root 463 Aug  2  2023 llmnr-udp.xml
```

```
-rw-r--r-- 1 root root 519 Aug  2  2023 llmnr.xml
-rw-r--r-- 1 root root 349 Aug  2  2023 managesieve.xml
-rw-r--r-- 1 root root 432 Aug  2  2023 matrix.xml
-rw-r--r-- 1 root root 424 Aug  2  2023 mdns.xml
-rw-r--r-- 1 root root 245 Aug  2  2023 memcache.xml
-rw-r--r-- 1 root root 334 Aug  2  2023 minidlna.xml
-rw-r--r-- 1 root root 237 Aug  2  2023 mongodb.xml
-rw-r--r-- 1 root root 473 Aug  2  2023 mosh.xml
-rw-r--r-- 1 root root 211 Aug  2  2023 mountd.xml
-rw-r--r-- 1 root root 296 Aug  2  2023 mqtt-tls.xml
-rw-r--r-- 1 root root 287 Aug  2  2023 mqtt.xml
-rw-r--r-- 1 root root 170 Aug  2  2023 mssql.xml
-rw-r--r-- 1 root root 180 Aug  2  2023 ms-wbt.xml
-rw-r--r-- 1 root root 242 Aug  2  2023 murmur.xml
-rw-r--r-- 1 root root 171 Aug  2  2023 mysql.xml
-rw-r--r-- 1 root root 250 Aug  2  2023 nbd.xml
-rw-r--r-- 1 root root 309 Aug  2  2023 nebula.xml
-rw-r--r-- 1 root root 262 Aug  2  2023 netbios-ns.xml
-rw-r--r-- 1 root root 243 Aug  2  2023 netdata-dashboard.xml
-rw-r--r-- 1 root root 342 Aug  2  2023 nfs3.xml
-rw-r--r-- 1 root root 324 Aug  2  2023 nfs.xml
-rw-r--r-- 1 root root 293 Aug  2  2023 nmea-0183.xml
-rw-r--r-- 1 root root 247 Aug  2  2023 nrpe.xml
-rw-r--r-- 1 root root 389 Aug  2  2023 ntp.xml
-rw-r--r-- 1 root root 368 Aug  2  2023 nut.xml
-rw-r--r-- 1 root root 335 Aug  2  2023 openvpn.xml
-rw-r--r-- 1 root root 260 Aug  2  2023 ovirt-imageio.xml
-rw-r--r-- 1 root root 343 Aug  2  2023 ovirt-storageconsole.xml
-rw-r--r-- 1 root root 235 Aug  2  2023 ovirt-vmconsole.xml
-rw-r--r-- 1 root root 869 Aug  2  2023 plex.xml
-rw-r--r-- 1 root root 433 Aug  2  2023 pmcd.xml
-rw-r--r-- 1 root root 474 Aug  2  2023 pmproxy.xml
-rw-r--r-- 1 root root 544 Aug  2  2023 pmwebapis.xml
-rw-r--r-- 1 root root 460 Aug  2  2023 pmwebapi.xml
```

```
-rw-r--r-- 1 root root 357 Aug  2  2023 pop3s.xml
-rw-r--r-- 1 root root 348 Aug  2  2023 pop3.xml
-rw-r--r-- 1 root root 181 Aug  2  2023 postgresql.xml
-rw-r--r-- 1 root root 509 Aug  2  2023 privoxy.xml
-rw-r--r-- 1 root root 226 Aug  2  2023 prometheus-node-exporter.xml
-rw-r--r-- 1 root root 213 Aug  2  2023 prometheus.xml
-rw-r--r-- 1 root root 261 Aug  2  2023 proxy-dhcp.xml
-rw-r--r-- 1 root root 262 Aug  2  2023 ps2link.xml
-rw-r--r-- 1 root root 173 Aug  2  2023 ps3netsrv.xml
-rw-r--r-- 1 root root 424 Aug  2  2023 ptp.xml
-rw-r--r-- 1 root root 414 Aug  2  2023 pulseaudio.xml
-rw-r--r-- 1 root root 297 Aug  2  2023 puppetmaster.xml
-rw-r--r-- 1 root root 273 Aug  2  2023 quassel.xml
-rw-r--r-- 1 root root 520 Aug  2  2023 radius.xml
-rw-r--r-- 1 root root 183 Aug  2  2023 rdp.xml
-rw-r--r-- 1 root root 212 Aug  2  2023 redis-sentinel.xml
-rw-r--r-- 1 root root 268 Aug  2  2023 redis.xml
-rw-r--r-- 1 root root 381 Aug  2  2023 RH-Satellite-6-capsule.xml
-rw-r--r-- 1 root root 556 Aug  2  2023 RH-Satellite-6.xml
-rw-r--r-- 1 root root 214 Aug  2  2023 rpc-bind.xml
-rw-r--r-- 1 root root 213 Aug  2  2023 rquotad.xml
-rw-r--r-- 1 root root 310 Aug  2  2023 rsh.xml
-rw-r--r-- 1 root root 311 Aug  2  2023 rsyncd.xml
-rw-r--r-- 1 root root 350 Aug  2  2023 rtsp.xml
-rw-r--r-- 1 root root 329 Aug  2  2023 salt-master.xml
-rw-r--r-- 1 root root 339 Aug  2  2023 samba-client.xml
-rw-r--r-- 1 root root 782 Aug  2  2023 samba-dc.xml
-rw-r--r-- 1 root root 382 Aug  2  2023 samba.xml
-rw-r--r-- 1 root root 324 Aug  2  2023 sane.xml
-rw-r--r-- 1 root root 283 Aug  2  2023 sips.xml
-rw-r--r-- 1 root root 496 Aug  2  2023 sip.xml
-rw-r--r-- 1 root root 299 Aug  2  2023 slp.xml
-rw-r--r-- 1 root root 231 Aug  2  2023 smtp-submission.xml
-rw-r--r-- 1 root root 577 Aug  2  2023 smtps.xml
```

```
-rw-r--r-- 1 root root 550 Aug  2  2023 smtp.xml
-rw-r--r-- 1 root root 359 Aug  2  2023 snmptls-trap.xml
-rw-r--r-- 1 root root 390 Aug  2  2023 snmptls.xml
-rw-r--r-- 1 root root 308 Aug  2  2023 snmptrap.xml
-rw-r--r-- 1 root root 342 Aug  2  2023 snmp.xml
-rw-r--r-- 1 root root 405 Aug  2  2023 spideroak-lansync.xml
-rw-r--r-- 1 root root 275 Aug  2  2023 spotify-sync.xml
-rw-r--r-- 1 root root 173 Aug  2  2023 squid.xml
-rw-r--r-- 1 root root 421 Aug  2  2023 ssdp.xml
-rw-r--r-- 1 root root 463 Aug  2  2023 ssh.xml
-rw-r--r-- 1 root root 631 Aug  2  2023 steam-streaming.xml
-rw-r--r-- 1 root root 287 Aug  2  2023 svdrp.xml
-rw-r--r-- 1 root root 231 Aug  2  2023 svn.xml
-rw-r--r-- 1 root root 297 Aug  2  2023 syncthing-gui.xml
-rw-r--r-- 1 root root 414 Aug  2  2023 syncthing-relay.xml
-rw-r--r-- 1 root root 350 Aug  2  2023 syncthing.xml
-rw-r--r-- 1 root root 496 Aug  2  2023 synergy.xml
-rw-r--r-- 1 root root 444 Aug  2  2023 syslog-tls.xml
-rw-r--r-- 1 root root 329 Aug  2  2023 syslog.xml
-rw-r--r-- 1 root root 393 Aug  2  2023 telnet.xml
-rw-r--r-- 1 root root 252 Aug  2  2023 tentacle.xml
-rw-r--r-- 1 root root 424 Aug  2  2023 tftp.xml
-rw-r--r-- 1 root root 221 Aug  2  2023 tile38.xml
-rw-r--r-- 1 root root 336 Aug  2  2023 tinc.xml
-rw-r--r-- 1 root root 771 Aug  2  2023 tor-socks.xml
-rw-r--r-- 1 root root 244 Aug  2  2023 transmission-client.xml
-rw-r--r-- 1 root root 264 Aug  2  2023 upnp-client.xml
-rw-r--r-- 1 root root 593 Aug  2  2023 vdsm.xml
-rw-r--r-- 1 root root 475 Aug  2  2023 vnc-server.xml
-rw-r--r-- 1 root root 443 Aug  2  2023 warpinator.xml
-rw-r--r-- 1 root root 310 Aug  2  2023 wbem-https.xml
-rw-r--r-- 1 root root 352 Aug  2  2023 wbem-http.xml
-rw-r--r-- 1 root root 285 Aug  2  2023 wireguard.xml
-rw-r--r-- 1 root root 355 Aug  2  2023 ws-discovery-client.xml
```

```
-rw-r--r-- 1 root root 320 Aug  2  2023 ws-discovery-tcp.xml
-rw-r--r-- 1 root root 375 Aug  2  2023 ws-discovery-udp.xml
-rw-r--r-- 1 root root 357 Aug  2  2023 ws-discovery.xml
-rw-r--r-- 1 root root 323 Aug  2  2023 wsmans.xml
-rw-r--r-- 1 root root 316 Aug  2  2023 wsman.xml
-rw-r--r-- 1 root root 329 Aug  2  2023 xdmcp.xml
-rw-r--r-- 1 root root 509 Aug  2  2023 xmpp-bosh.xml
-rw-r--r-- 1 root root 488 Aug  2  2023 xmpp-client.xml
-rw-r--r-- 1 root root 264 Aug  2  2023 xmpp-local.xml
-rw-r--r-- 1 root root 545 Aug  2  2023 xmpp-server.xml
-rw-r--r-- 1 root root 314 Aug  2  2023 zabbix-agent.xml
-rw-r--r-- 1 root root 315 Aug  2  2023 zabbix-server.xml
-rw-r--r-- 1 root root 242 Aug  2  2023 zerotier.xml

/usr/lib/firewalld/zones:
total 40
-rw-r--r-- 1 root root 312 Aug  2  2023 block.xml
-rw-r--r-- 1 root root 306 Aug  2  2023 dmz.xml
-rw-r--r-- 1 root root 304 Aug  2  2023 drop.xml
-rw-r--r-- 1 root root 317 Aug  2  2023 external.xml
-rw-r--r-- 1 root root 382 Aug  2  2023 home.xml
-rw-r--r-- 1 root root 397 Aug  2  2023 internal.xml
-rw-r--r-- 1 root root 729 Mar 22  2025 nm-shared.xml
-rw-r--r-- 1 root root 328 Aug  2  2023 public.xml
-rw-r--r-- 1 root root 175 Aug  2  2023 trusted.xml
-rw-r--r-- 1 root root 324 Aug  2  2023 work.xml
```

Ces fichiers sont au format **xml**, par exemple :

```
root@debian12:~# cat /usr/lib/firewalld/zones/home.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Home</short>
  <description>For use in home 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="mdns"/>
  <service name="samba-client"/>
  <service name="dhcpv6-client"/>
  <forward/>
</zone>
```

La configuration de firewalld ainsi que les définitions et règles personnalisées se trouvent dans **/etc/firewalld** :

```
root@debian12:~# ls -lR /etc/firewalld/
/etc/firewalld/:
total 32
-rw-r--r-- 1 root root 2483 Aug  2  2023 firewalld.conf
drwxr-xr-x 2 root root 4096 Aug  2  2023 helpers
drwxr-xr-x 2 root root 4096 Aug  2  2023 icmptypes
drwxr-xr-x 2 root root 4096 Aug  2  2023 ipsets
-rw-r--r-- 1 root root  268 Aug  2  2023 lockdown-whitelist.xml
drwxr-xr-x 2 root root 4096 Aug  2  2023 policies
drwxr-xr-x 2 root root 4096 Aug  2  2023 services
drwxr-xr-x 2 root root 4096 Aug  2  2023 zones

/etc/firewalld/helpers:
total 0

/etc/firewalld/icmptypes:
total 0

/etc/firewalld/ipsets:
total 0

/etc/firewalld/policies:
total 0
```

```
/etc/firewalld/services:
total 0

/etc/firewalld/zones:
total 0
```

Le fichier de configuration de firewalld est **/etc/firewalld/firewalld.conf** :

```
root@debian12:~# cat /etc/firewalld/firewalld.conf
# firewalld config file

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

# Clean up on exit
# If set to no or false the firewall configuration will not get cleaned up
# on exit or stop of firewalld.
# Default: yes
CleanupOnExit=yes

# Clean up kernel modules on exit
# If set to yes or true the firewall related kernel modules will be
# unloaded on exit or stop of firewalld. This might attempt to unload
# modules not originally loaded by firewalld.
# Default: no
CleanupModulesOnExit=no

# 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.
# Note: This feature has a performance impact. See man page FIREWALLD.CONF(5)
# for details.
# Default: yes
IPv6_rpfilter=yes

# IndividualCalls
# Do not use combined -restore calls, but individual calls. This increases the
# time that is needed to apply changes and to start the daemon, but is good for
# debugging.
# Default: no
IndividualCalls=no

# LogDenied
# Add logging rules right before reject and drop rules in the INPUT, FORWARD
# and OUTPUT chains for the default rules and also final reject and drop rules
# in zones. Possible values are: all, unicast, broadcast, multicast and off.
# Default: off
LogDenied=off

# FirewallBackend
# Selects the firewall backend implementation.
# Choices are:
#       - nftables (default)
#       - iptables (iptables, ip6tables, ebtables and ipset)
# Note: The iptables backend is deprecated. It will be removed in a future
# release.
FirewallBackend=nftables
```

```
# FlushAllOnReload
# Flush all runtime rules on a reload. In previous releases some runtime
# configuration was retained during a reload, namely; interface to zone
# assignment, and direct rules. This was confusing to users. To get the old
# behavior set this to "no".
# Default: yes
FlushAllOnReload=yes

# RFC3964_IPv4
# As per RFC 3964, filter IPv6 traffic with 6to4 destination addresses that
# correspond to IPv4 addresses that should not be routed over the public
# internet.
# Defaults to "yes".
RFC3964_IPv4=yes
```

1.2 - La Commande firewall-cmd

Important - firewall-cmd est le front-end de firewalld en ligne de commande. Il existe aussi la commande **firewall-config** qui lance un outil de configuration graphique.

Pour obtenir la liste de toutes les zones prédéfinies, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
```

Pour obtenir la liste de toutes les services prédéfinis, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd 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 cockpit collectd condor-
collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-
registry 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 https 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-control-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-client llmnr-tcp llmnr-udp managesieve matrix mdns
memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-
dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd
pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link
ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel 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-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui
syncthing-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-client xmpp-local xmpp-server zabbix-agent zabbix-server
zerotier
```

Pour obtenir la liste de toutes les types ICMP prédéfinis, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --get-icmptypes
address-unreachable bad-header beyond-scope communication-prohibited destination-unreachable echo-reply echo-
request failed-policy fragmentation-needed host-precedence-violation host-prohibited host-redirect host-unknown
host-unreachable ip-header-bad neighbour-advertisement neighbour-solicitation network-prohibited network-redirect
network-unknown network-unreachable no-route packet-too-big parameter-problem port-unreachable precedence-cutoff
protocol-unreachable redirect reject-route required-option-missing router-advertisement router-solicitation
source-quench source-route-failed time-exceeded timestamp-reply timestamp-request tos-host-redirect tos-host-
unreachable tos-network-redirect tos-network-unreachable ttl-zero-during-reassembly ttl-zero-during-transit
unknown-header-type unknown-option
```

Pour obtenir la liste des zones de la configuration courante, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --get-active-zones
public
  interfaces: ens18
```

Pour obtenir la liste des zones de la configuration courante pour une interface spécifique, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --get-zone-of-interface=ens18
public
```

Pour obtenir la liste des services autorisés pour la zone public, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=public --list-services
dhcpv6-client ssh
```

Pour obtenir toute la configuration pour la zone public, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Pour obtenir la liste complète de toutes les zones et leurs configurations, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --list-all-zones
block
  target: %%REJECT%%
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

dmz
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

drop
  target: DROP
  icmp-block-inversion: no
```

```
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  forward: yes
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client mdns samba-client ssh
  ports:
```

```
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

internal
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

nm-shared
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcp dns ssh
  ports:
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
```

```
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject

public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

trusted
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Pour changer la zone par défaut de public à work, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --set-default-zone=work
success

root@debian12:~# firewall-cmd --get-active-zones
work
  interfaces: ens18
```

Créez un profil Network Manager, dénommez **ip_fixe** et rattaché au périphérique **ens18** :

```
root@debian12:~# nmcli connection add con-name ip_fixe ifname ens18 type ethernet ip4 10.0.2.46/24 gw4 10.0.2.1
Connection 'ip_fixe' (33c26470-0968-4646-a88a-a22f10fab6da) successfully added.

root@debian12:~# nmcli c show
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  77c569e6-3176-4c10-8008-40d7634d2504  ethernet  ens18
lo                  c4172990-a224-464f-a1de-9820ca5e83c8  loopback  lo
ip_fixe             33c26470-0968-4646-a88a-a22f10fab6da  ethernet  --
```

```
root@debian12:~# nmcli connection up ip_fixe
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)

root@debian12:~# nmcli c show
NAME                UUID                                  TYPE      DEVICE
ip_fixe             33c26470-0968-4646-a88a-a22f10fab6da  ethernet  ens18
lo                  c4172990-a224-464f-a1de-9820ca5e83c8  loopback  lo
Wired connection 1  77c569e6-3176-4c10-8008-40d7634d2504  ethernet  --

root@debian12:~# nmcli connection mod ip_fixe ipv4.dns 8.8.8.8

root@debian12:~# systemctl restart NetworkManager.service
```

Pour ajouter le profil ip_fixe à la zone work, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --add-interface=ip_fixe
success

root@debian12:~# firewall-cmd --get-active-zones
work
  interfaces: ens18 ip_fixe
```

Pour supprimer l'interface ip_fixe à la zone work, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --remove-interface=ip_fixe
success

root@debian12:~# firewall-cmd --get-active-zones
work
  interfaces: ens18
```

Pour ajouter le service **http** à la zone **work**, utilisez la commande suivante :

```
root@debian11:~# firewall-cmd --zone=work --add-service=http
```

```
success
root@debian11:~# firewall-cmd --zone=work --list-services
dhcpv6-client http ssh
```

Pour supprimer le service **http** de la zone **work**, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --add-service=http
success

root@debian12:~# firewall-cmd --zone=work --list-services
dhcpv6-client http ssh
```

Pour ajouter un nouveau bloc ICMP, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --add-icmp-block=echo-reply
success

root@debian12:~# firewall-cmd --zone=work --list-icmp-blocks
echo-reply
```

Pour supprimer un bloc ICMP, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --remove-icmp-block=echo-reply
success

root@debian12:~# firewall-cmd --zone=work --list-icmp-blocks

root@debian12:~#
```

Pour ajouter le port 591/tcp à la zone work, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --add-port=591/tcp
success
```

```
root@debian12:~# firewall-cmd --zone=work --list-ports
591/tcp
```

Pour supprimer le port 591/tcp à la zone work, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --zone=work --remove-port=591/tcp
success

root@debian12:~# firewall-cmd --zone=work --list-ports

root@debian12:~#
```

Pour créer un nouveau service, il convient de :

- copier un fichier existant se trouvant dans le répertoire **/usr/lib/firewalld/services** vers **/etc/firewalld/services**,
- modifier le fichier,
- recharger la configuration de firewalld,
- vérifier que firewalld voit le nouveau service.

Par exemple :

```
root@debian12:~# cp /usr/lib/firewalld/services/http.xml /etc/firewalld/services/filemaker.xml

root@debian12:~# vi /etc/firewalld/services/filemaker.xml

root@debian12:~# cat /etc/firewalld/services/filemaker.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>FileMakerPro</short>
  <description>fichier de service firewalld pour FileMaker Pro</description>
  <port protocol="tcp" port="591"/>
</service>

root@debian12:~# firewall-cmd --reload
```

```
success

root@debian12:~# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd 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 cockpit collectd condor-
collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-
registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server filemaker 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 https 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-control-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-client llmnr-tcp llmnr-udp managesieve matrix mdns
memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-
dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd
pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link
ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel 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-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui
syncthing-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-client xmpp-local xmpp-server zabbix-agent zabbix-server
zerotier
```

1.3 - La Configuration Avancée de firewalld

La configuration de base de firewalld ne permet que la configuration des zones, services, blocs ICMP et les ports non-standard. Cependant, firewalld peut également être configuré avec des **Rich Rules** ou ***Règles Riches***. Rich Rules ou Règles Riches évaluent des **critères** pour ensuite entreprendre une **action**.

Les **Critères** sont :

- **source address=“<adresse_IP>“**
- **destination address=”<adresse_IP>“**,
- **rule port port=”<numéro_du_port>“**,
- **service name=<nom_d'un_sevice_prédéfini>**.

Les **Actions** sont :

- **accept**,
- **reject**,
  - une Action reject peut être associée avec un message d'erreur spécifique par la clause **type=”<type_d'erreur>**,
- **drop**.

Saisissez la commande suivante pour ouvrir le port 80 :

```
root@debian12:~# firewall-cmd --add-rich-rule='rule port port="80" protocol="tcp" accept'
success
```

Important - Notez que la Rich Rule doit être entourée de caractères '.

Important - Notez que la Rich Rule a créé deux règles, une pour IPv4 et une deuxième pour IPv6. Une règle peut être créée pour IPv4 seul en incluant le Critère **family=ipv4**. De la même façon, une règle peut être créée pour IPv6 seul en incluant le Critère **family=ipv6**.

Cette nouvelle règle est écrite en mémoire mais non pas sur disque. Pour l'écrire sur disque dans le fichier zone se trouvant dans **/etc/firewalld**, il faut ajouter l'option **–permanent** :

```
root@debian12:~# firewall-cmd --add-rich-rule='rule port port="80" protocol="tcp" accept' --permanent
success
```

```
root@debian12:~# cat /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"/>
  <rule>
    <port port="80" protocol="tcp"/>
    <accept/>
  </rule>
  <forward/>
</zone>
```

Important - Attention ! La règle ajoutée avec l'option –permanent n'est pas prise en compte immédiatement mais uniquement au prochain redémmarge. Pour qu'une règle soit appliquée immédiatement **et** écrite sur disque, il faut saisir la commande deux fois dont une avec l'option –permanent et l'autre sans l'option –permanent.

Pour visualiser cette règle dans la configuration de firewalld, il convient de saisir la commande suivante :

```
root@debian12:~# firewall-cmd --reload
success

root@debian12:~# firewall-cmd --zone=work --list-all
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
```

```
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule port port="80" protocol="tcp" accept
```

Notez que la Rich Rule est créée dans la Zone par Défaut. Il est possible de créer une Rich Rule dans une autre zone en utilisant l'option **–zone=<zone>** de la commande firewall-cmd :

```
root@debian12:~# firewall-cmd --zone=public --add-rich-rule='rule port port="80" protocol="tcp" accept'
success

root@debian12:~# firewall-cmd --reload
success

root@debian12:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
```

```
  icmp-blocks:
  rich rules:
root@debian12:~# firewall-cmd --zone=public --add-rich-rule='rule port port="80" protocol="tcp" accept' --
permanent
success

root@debian12:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
root@debian12:~# firewall-cmd --reload
success

root@debian12:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
```

```
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule port port="80" protocol="tcp" accept
```

Pour supprimer une Rich Rule, il faut copier la ligne entière la concernant qui se trouve dans la sortie de la commande **firewall-cmd –list-all-zones** :

```
root@debian12:~# firewall-cmd --zone=public --remove-rich-rule='rule port port="80" protocol="tcp" accept'
success

root@debian12:~# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

1.4 - Le mode Panic de firewalld

Le mode Panic de firewalld permet de bloquer tout le trafic avec une seule commande. Pour connaître l'état du mode Panic, utilisez la commande suivante :

```
root@debian12:~# firewall-cmd --query-panic
no
```

Pour activer le mode Panic, il convient de saisir la commande suivante :

Important - Veuillez ne PAS saisir la commande suivante !!

```
# firewall-cmd --panic-on
```

Pour désactiver le mode Panic, il convient de saisir la commande suivante :

```
# firewall-cmd --panic-off
```
