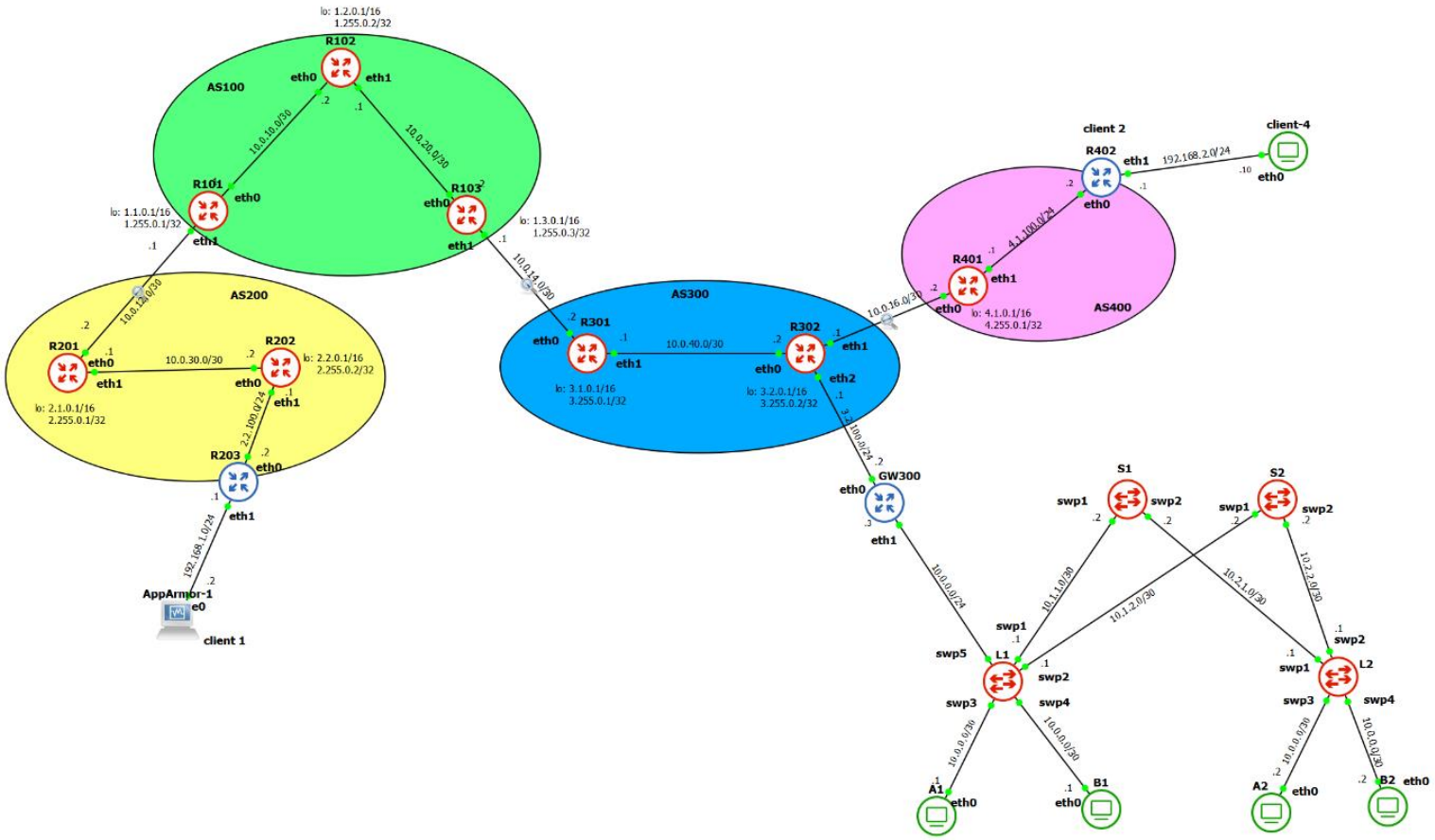
Report NSD
Di Totto Luca 0333084 – Di Marco Luca 0333083

Sommario



Configurazione degli AS

I file relativi alle configurazioni di ogni dispositivo, senza commenti, sono presenti nella cartella 'Configurazione'.

AS – 100

Per la configurazione dei peer all'interno dell'AS100 è necessario, all'avvio di questi, eseguire '`./config.sh`', in modo che la configurazione dei moduli kernel mpls con sysctl venga effettuata prima della configurazione dei peer.

Nei dispositivi R101 e R103 configuriamo i protocolli MPLS, BGP e OSPF, mentre in R102 vengono configurati i protocolli OSPF e MPLS. Viene creta una connessione logica BGP diretta tra i nodi R101 e R103, in modo da realizzare una rete full mesh, e far sì che questi si possano scambiare informazioni sulle rotte (bypassando il problema del loop avoidance).

R101

in /etc/sysctl.conf

```
net.mpls.conf.lo.input = 1
net.mpls.conf.eth0.input = 1
net.mpls.platform_labels = 100000

cp /etc/frr/sysctl.conf /etc/sysctl.conf
sysctl -p

vtysh
conf t
interface lo
ip address 1.1.0.1/16
ip address 1.255.0.1/32
exit

interface eth0
ip address 10.0.10.1/30
mpls bgp forwarding
exit

interface eth1
ip address 10.0.12.1/30
exit

router ospf
router-id 1.255.0.1
network 1.1.0.0/16 area 0
network 1.255.0.1/32 area 0
network 10.0.10.0/30 area 0
exit


mpls ldp
router-id 1.255.0.1
ordered-control
address-family ipv4
discovery transport-address 1.255.0.1
interface eth0
exit
interface lo
exit
exit
exit

router bgp 100
network 1.1.0.0/16

neighbor 1.255.0.3 remote-as 100
neighbor 1.255.0.3 update-source 1.255.0.1

neighbor 1.255.0.2 remote-as 100
neighbor 1.255.0.2 update-source 1.255.0.1

neighbor 10.0.12.2 remote-as 200
```

```
address-family ipv4
neighbor 1.255.0.3 activate
neighbor 1.255.0.3 next-hop-self

neighbor 1.255.0.2 activate
neighbor 1.255.0.2 next-hop-self

redistribute static
exit
end
```

R102

In /etc/sysctl.conf

```
net.mpls.conf.lo.input = 1
net.mpls.conf.eth0.input = 1
net.mpls.conf.eth1.input = 1
net.mpls.platform_labels = 100000

cp /etc/frr/sysctl.conf /etc/sysctl.conf
sysctl -p

vtysh
conf t
int lo
ip address 1.2.0.1/16
ip address 1.255.0.2/32
exit
int eth0
ip address 10.0.10.2/30
mpls bgp forwarding
exit
int eth1
ip address 10.0.20.1/30
mpls bgp forwarding
exit

router ospf
router-id 1.255.0.2
network 1.2.0.0/16 area 0
network 1.255.0.2/32 area 0
network 10.0.10.0/30 area 0
network 10.0.20.0/30 area 0
exit

mpls ldp
router-id 1.255.0.2
ordered-control
address-family ipv4
discovery transport-address 1.255.0.2
interface eth0
exit
interface eth1
exit
```

```
interface lo
exit
end
```

R103

```
net.mpls.conf.lo.input = 1
net.mpls.conf.eth0.input = 1
net.mpls.platform_labels = 100000

cp /etc/frr/sysctl.conf /etc/sysctl.conf
sysctl -p

vtysh
conf t
interface lo
ip address 1.3.0.1/16
ip address 1.255.0.3/32
exit

interface eth0
ip address 10.0.20.2/30
mpls bgp forwarding
exit

interface eth1
ip address 10.0.14.1/30
exit

router ospf
router-id 1.255.0.3

network 1.3.0.0/16 area 0
network 1.255.0.3/32 area 0
network 10.0.20.0/30 area 0
exit

mpls ldp
router-id 1.255.0.3
ordered-control
address-family ipv4
discovery transport-address 1.255.0.3
interface eth0
exit
interface lo
exit
exit
exit

router bgp 100
network 1.3.0.0/16
neighbor 1.255.0.1 remote-as 100
neighbor 1.255.0.1 update-source 1.255.0.3

neighbor 1.255.0.2 remote-as 100
```

```
neighbor 1.255.0.2 update-source 1.255.0.3

neighbor 10.0.14.2 remote-as 300

address-family ipv4
neighbor 1.255.0.1 activate
neighbor 1.255.0.1 next-hop-self

neighbor 1.255.0.2 activate
neighbor 1.255.0.2 next-hop-self

neighbor 10.0.14.2 next-hop-self
redistribute static
exit
end
```

AS-200

Nei dispositivi dell'AS200 vengono configurati i protocolli OSPF e BGP.
Il dispositivo R203 viene approfondito successivamente.

R201

```
vtysh
conf t
interface lo
ip address 2.1.0.1/16
ip address 2.255.0.1/32
exit

interface eth0
ip address 10.0.12.2/30
exit

interface eth1
ip address 10.0.30.1/30
exit

router ospf
router-id 2.255.0.1
network 2.1.0.0/16 area 0
network 2.255.0.1/32 area 0
network 10.0.30.0/30 area 0
exit

router bgp 200
network 2.1.0.0/16
neighbor 2.255.0.2 remote-as 200
neighbor 2.255.0.2 update-source 2.255.0.1
neighbor 2.255.0.2 next-hop-self

neighbor 10.0.12.1 remote-as 100
end
```

R202

```
vtysh
```

```
conf t

interface lo
ip address 2.2.0.1/16
ip address 2.255.0.2/32
exit

interface eth0
ip address 10.0.30.2/30
exit

interface eth1
ip address 2.2.100.1/24
exit

router ospf
router-id 2.255.0.2

network 2.2.0.0/16 area 0
network 2.255.0.2/32 area 0
network 10.0.30.0/30 area 0
exit

router bgp 200
network 2.2.0.0/16
neighbor 2.255.0.1 remote-as 200
neighbor 2.255.0.1 update-source 2.255.0.2
neighbor 2.255.0.1 next-hop-self

end
```

AS-300

Nei dispositivi dell'AS300 vengono configurati i protocolli OSPF e BGP.

R301

```
vtysh
conf t
interface lo
ip address 3.1.0.1/16
ip address 3.255.0.1/32
exit

interface eth0
ip address 10.0.14.2/30
exit
interface eth1
ip address 10.0.40.1/30
exit

router ospf
router-id 3.255.0.1
network 3.1.0.0/16 area 0
network 3.255.0.1/32 area 0
```

```
network 10.0.40.0/30 area 0
exit

router bgp 300
network 3.1.0.0/16
neighbor 3.255.0.2 remote-as 300
neighbor 3.255.0.2 update-source 3.255.0.1
neighbor 3.255.0.2 next-hop-self
neighbor 10.0.14.1 remote-as 100
end
```

R302

```
vtysh
conf t
interface lo
ip address 3.2.0.1/16
ip address 3.255.0.2/32
exit

interface eth0
ip address 10.0.40.2/30
exit

interface eth1
ip address 10.0.16.1/30
exit

interface eth2
ip address 3.2.100.1/24
exit

router ospf
router-id 3.255.0.2
network 3.2.0.0/16 area 0
network 3.255.0.2/32 area 0
network 10.0.40.0/30 area 0
exit

router bgp 300
network 3.2.0.0/16
neighbor 3.255.0.1 remote-as 300
neighbor 3.255.0.1 update-source 3.255.0.2
neighbor 3.255.0.1 next-hop-self
neighbor 10.0.16.2 remote-as 400
end
```

AS-400

Nei dispositivi dell'AS400 vengono configurati i protocolli OSPF e BGP.
Il router R402 viene approfondito nelle sezioni successive.

R401

```
vtysh
conf t

interface lo
```

```
ip address 4.1.0.1/16
exit

interface eth0
ip address 10.0.16.2/30
exit

interface eth1
ip address 4.1.100.1/24

router bgp 400
network 4.1.0.0/16

neighbor 10.0.16.1 remote-as 300
end
```

Datacenter

Per la realizzazione della rete DC viene utilizzato il modello leaf – spine. I dispositivi leaf e spine sono stati creati con una macchina virtuale CumulusLinux.

Vengono realizzate due VNI (di tipo L2VNI), di cui la VNI100 dedicata al collegamento con i tenants A e la VNI200 dedicata al collegamento dei tenants B.

Per il collegamento del Leaf1 verso l'esterno, viene configurata la porta swp5 del Leaf1 come facente parte anch'essa della VNI100. In questo modo entrambi i tenants A riescono a raggiungere l'esterno tramite il gateway300.

S1 (vm1)

```
net del all

net add interface swp1 ip add 10.1.1.2/30
net add interface swp2 ip add 10.2.1.2/30
net add loopback lo ip add 4.4.4.4/32

net add ospf router-id 4.4.4.4
net add ospf network 0.0.0.0/0 area 0

net commit

net add bgp autonomous-system 65000
net add bgp router-id 4.4.4.4
net add bgp neighbor swp1 remote-as external
net add bgp neighbor swp2 remote-as external
net add bgp evpn neighbor swp1 activate
net add bgp evpn neighbor swp2 activate
net commit
```

S2 (vm2)

```
net del all

net add interface swp1 ip add 10.1.2.2/30
net add interface swp2 ip add 10.2.2.2/30
net add loopback lo ip add 5.5.5.5/32
```

```
net add ospf router-id 5.5.5.5
net add ospf network 0.0.0.0/0 area 0

net commit

net add bgp autonomous-system 65000
net add bgp router-id 5.5.5.5
net add bgp neighbor swp1 remote-as external
net add bgp neighbor swp2 remote-as external
net add bgp evpn neighbor swp1 activate
net add bgp evpn neighbor swp2 activate
net commit
```

In tutti i server A1/A2/B1/B2:

```
>> cd
>> ./config.sh
```

Nei *config.sh* troviamo:

A1

```
ip addr add 10.0.0.1/24 dev eth0
```

B1

```
ip addr add 10.0.0.1/24 dev eth0
```

A2

```
ip addr add 10.0.0.2/24 dev eth0
```

B2

```
ip addr add 10.0.0.2/24 dev eth0
```

A1 e A2

```
ip route add default via 10.0.0.3
```

L1 (vm3)

```
net del all
net commit

net add bridge bridge ports swp3,swp4
net add interface swp3 bridge access 10
net add interface swp4 bridge access 20
net commit

net add interface swp1 ip add 10.1.1.1/30
net add interface swp2 ip add 10.1.2.1/30
net add loopback lo ip add 1.1.1.1/32
net commit

net add ospf router-id 1.1.1.1
net add ospf network 10.1.1.0/30 area 0
net add ospf network 10.1.2.0/30 area 0
net add ospf network 1.1.1.1/32 area 0
net add ospf passive-interface swp3,swp4
```

```
net commit

net add vxlan vni100 vxlan id 100
net add vxlan vni100 vxlan remoteip 2.2.2.2
net add vxlan vni100 vxlan local-tunnelip 1.1.1.1
net add vxlan vni100 bridge access 10
net add vxlan vni200 vxlan id 200
net add vxlan vni200 vxlan remoteip 2.2.2.2
net add vxlan vni200 vxlan local-tunnelip 1.1.1.1
net add vxlan vni200 bridge access 20

net commit

net del vxlan vni100 vxlan remoteip 2.2.2.2
net del vxlan vni200 vxlan remoteip 2.2.2.2

net add bgp autonomous-system 65001
net add bgp router-id 1.1.1.1
net add bgp neighbor swp1 remote-as 65000
net add bgp neighbor swp2 remote-as 65000
net add bgp evpn neighbor swp1 activate
net add bgp evpn neighbor swp2 activate
net add bgp evpn advertise-all-vni
net commit
```

Per connetterci a GW300:

```
net add bridge bridge ports swp3,swp4,swp5
net add interface swp5 bridge access 10
net add ospf passive-interface swp5
```

L2 (vm4)

```
net del all
net commit

net add bridge bridge ports swp3,swp4
net add interface swp3 bridge access 10
net add interface swp4 bridge access 20
net commit

net add interface swp1 ip add 10.2.1.1/30
net add interface swp2 ip add 10.2.2.1/30
net add loopback lo ip add 2.2.2.2/32
net commit

net add ospf router-id 2.2.2.2
net add ospf network 10.2.1.0/30 area 0
net add ospf network 10.2.2.0/30 area 0
net add ospf network 2.2.2.2/32 area 0
net add ospf passive-interface swp3,swp4
net commit
```

```
net add vxlan vni100 vxlan id 100
net add vxlan vni100 vxlan remoteip 1.1.1.1
net add vxlan vni100 vxlan local-tunnelip 2.2.2.2
net add vxlan vni100 bridge access 10
net add vxlan vni200 vxlan id 200
net add vxlan vni200 vxlan remoteip 1.1.1.1
net add vxlan vni200 vxlan local-tunnelip 2.2.2.2
net add vxlan vni200 bridge access 20

net commit

net del vxlan vni100 vxlan remoteip 1.1.1.1
net del vxlan vni200 vxlan remoteip 1.1.1.1

net add bgp autonomous-system 65002
net add bgp router-id 2.2.2.2
net add bgp neighbor swp1 remote-as 65000
net add bgp neighbor swp2 remote-as 65000
net add bgp evpn neighbor swp1 activate
net add bgp evpn neighbor swp2 activate
net add bgp evpn advertise-all-vni
net commit
```

R203

Il router 203 viene configurato per comunicare direttamente con l'R202 ed effettuare NAT dinamico. Su questo dispositivo è presente una configurazione per il firewall approfondita successivamente.

```
>> cd
>> ./config.sh
```

Nel *config.sh* troviamo:

```
ip addr add 192.168.1.1/24 dev eth1
ip addr add 2.2.100.2/24 dev eth0
ip route add default via 2.2.100.1
iptables -t nat -F
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -F
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

AppArmor (client – 200)

Il dispositivo client 200, denominato AppArmor, è realizzato mediante l'immagine di una macchina Lubuntu. In questa sezione vengono specificati gli indirizzi; per la configurazione del MAC AppArmor, rimandiamo la lettura alla sezione dedicata; per la configurazione del client200 come client openVPN rimandiamo la lettura alla sezione dedicata.

```
>> cd
>> ./config.sh
```

Nel *config.sh* abbiamo configurato i seguenti indirizzi:

```
ip addr add 192.168.1.2/24 dev enp0s8
ip route add default via 192.168.1.1
```

R402

Il router R402 viene configurato per comunicare direttamente con l'R401 ed effettuare NAT dinamico. Per la configurazione di questo router come client openVPN rimandiamo la lettura alla sezione dedicata.

```
>> cd
>> ./config.sh
```

Nel *config.sh* troviamo:

```
ip addr add 192.168.2.1/24 dev eth1
ip addr add 4.1.100.2/24 dev eth0
ip route add default via 4.1.100.1
iptables -t nat -F
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -F
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Client – 4

```
>> cd
>> ./config.sh
```

Nel *config.sh* troviamo:

```
ip addr add 192.168.2.10/24 dev eth0
ip route add default via 192.168.2.1
```

GW – 300

Il dispositivo GW300 ha la funzionalità di esporre verso l'esterno tutta la rete DC. Viene configurato per comunicare direttamente con l'R302 ed effettuare NAT dinamico. Questo dispositivo ha inoltre il ruolo di server, nell'architettura openVPN realizzata. Per la configurazione di questa rimandiamo la lettura alla sezione dedicata.

```
>> cd
>> ./config.sh
```

Nel config.sh troviamo:

```
ip addr add 3.2.100.2/24 dev eth0
ip addr add 10.0.0.3/24 dev eth1
ip route add default via 3.2.100.1
iptables -t nat -F
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -F
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Firewall – R203

Il firewall implementato nel router R203 ha il compito di permettere il transito solo delle connessioni avviate dall'interno della LAN

```
>> cd
>> ./firewall.sh
```

Nel *firewall.sh* troviamo:

```
#!/bin/bash
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -i eth1 -m state --state NEW -j DROP
```

AppArmor profile

Nel profilo AppArmor presente sulla VM "AppArmor" nella cartella /etc/apparmor.d troviamo un confinamento per l'applicazione Wireshark:

- Vengono dati permessi per accedere ai soli file necessari per l'esecuzione corretta di Wireshark;
- Vengono negati gli accessi in lettura e scrittura nelle cartelle Desktop, Documenti, Immagini, Pubblici, Scaricati e Video;
- Vengono concesse capability e network strettamente necessarie al corretto funzionamento di Wireshark.

Il profilo è il seguente:

```
# Last Modified: Thu Feb  8 16:00:24 2024
include <tunables/global>

# vim:syntax=apparmor
# AppArmor policy for wireshark
# ###AUTHOR###
# ###COPYRIGHT###
# ###COMMENT###


/usr/bin/wireshark {
  include <abstractions/X>
  include <abstractions/base>
  include <abstractions/bash>
```

```
include <abstractions/consoles>
include <abstractions/dbus-session>
include <abstractions/gnome>
include <abstractions/kde-open5>
include <abstractions/kde>
include <abstractions/nameservice>
include <abstractions/opencl-pocl>
include <abstractions/user-write>

capability dac_override,
capability dac_read_search,
capability fowner,
capability net_admin,
capability net_raw,

network bluetooth raw,
network packet dgram,
network packet raw,
network unix stream,

deny /home/*/Desktop/** rw,
deny /home/*/Documenti/** rw,
deny /home/*/Immagini/** rw,
deny /home/*/Pubblici/** rw,
deny /home/*/Scaricati/** rw,
deny /home/*/Video/** rw,

/dev/ r,
/etc/ethers r,
/etc/pango/pango.modules r,
/etc/wireshark/init.lua r,
/etc/xdg/xdg-Lubuntu/lxqt/lxqt.conf r,
/home/*/ r,
/home/*/.Xauthority r,
/home/*/.bash_logout r,
/home/*/.xsession-errors r,
/home/mac/capture/ rw,
/home/mac/capture/* rw,
/proc/*/net/dev r,
/run/user/1000/at-spi/bus_0 rw,
/sys/devices/pci0000:00/** r,
/usr/bin/dbus-daemon mrix,
/usr/bin/dumpcap mrix,
/usr/bin/wireshark mrix,
/usr/lib/firefox/firefox.sh rPx,
/usr/lib/gtk-*/*/loaders/* mr,
/usr/lib/x86_64-linux-gnu/wireshark/extcap/androiddump mrix,
/usr/lib/x86_64-linux-gnu/wireshark/extcap/ciscodump mrix,
/usr/lib/x86_64-linux-gnu/wireshark/extcap/dpauxmon mrix,
/usr/lib/x86_64-linux-gnu/wireshark/extcap/randpktdump mrix,
/usr/lib/x86_64-linux-gnu/wireshark/extcap/sdjournal mrix,
```

```
  /usr/lib/x86_64-linux-gnu/wireshark/extcap/sshdump mrix,
  /usr/lib/x86_64-linux-gnu/wireshark/extcap/udpdump mrix,
  /usr/share/* r,
  /usr/share/icons r,
  /usr/share/icons/** r,
  /usr/share/libfm-qt/translations/libfm-qt_it.qm r,
  /usr/share/lxqt/lxqt.conf r,
  /usr/share/mime/** r,
  /usr/share/snmp/mibs/** r,
  /usr/share/snmp/mibs/.index rw,
  /usr/share/thumbnailers/ r,
  /usr/share/thumbnailers/** r,
  /usr/share/wireshark/* r,
  /usr/share/wireshark/** r,
  @{HOME}/.fonts.cache-* r,
  @{HOME}/.wireshark/* rw,
  owner /etc/ r,
  owner /etc/dbus-1/session.d/ r,
  owner /etc/fstab r,
  owner /home/* w,
  owner /home/*/.config/#786513 rw,
  owner /home/*/.config/#789761 rw,
  owner /home/*/.config/* rwlk,
  owner /home/*/.config/QtProject* lk,
  owner /home/*/.config/QtProject* rw,
  owner /home/*/.config/lxqt/lxqt.conf r,
  owner /home/*/.config/wireshark/** rw,
  owner /home/*/.local/share/gvfs-metadata/** r,
  owner /proc/*/attr/apparmor/current r,
  owner /proc/*/attr/current r,
  owner /proc/*/cmdline r,
  owner /proc/*/fd/ r,
  owner /proc/*/mountinfo r,
  owner /root/.config/#* rw,
  owner /root/.config/QtProject* lk,
  owner /root/.config/QtProject* rwk,
  owner /root/.config/wireshark/recent rw,
  owner /root/.config/wireshark/recent_common rw,
  owner /root/.dbus/session-bus/** w,
  owner /run/user/1000/ rw,
  owner /run/user/1000/* rw,
  owner /run/user/1000/gvfsd/socket-* rw,
  owner /sys/kernel/security/apparmor/.access rw,
  owner /sys/kernel/security/apparmor/features/dbus/mask r,
  owner /usr/share/dbus-1/** r,
  owner /var/lib/snapd/dbus-1/services/ r,

}
```

OpenVPN

Indirizzi della rete VPN:

- client1 AppArmor: 192.168.100.10
- client2 R402: 192.168.100.6
- client3 A1: 192.168.100.14
- server GW300: 192.168.100.1

Dopo aver inizializzato la CA all'interno del server (GW300), aver generato tutte le chiavi pubbliche e private ed i certificati ed averli copiati nei rispettivi client abbiamo inserito i seguenti file di configurazione OpenVPN nei dispositivi:

File ovpn client2 – R402

```
client
dev tun
proto udp
remote 3.2.100.2 1194
resolv-retry infinite
ca ca.crt
cert client2.crt
key client2.key
remote-cert-tls server
cipher AES-256-GCM
```

File ovpn client1 (VM AppArmor)

```
client
dev tun
proto udp
remote 3.2.100.2 1194
resolv-retry infinite
ca ca.crt
cert client1.crt
key client1.key
remote-cert-tls server
cipher AES-256-GCM
```

File ovpn client3 – A1

```
client
dev tun
proto udp
remote 10.0.0.3 1194
resolv-retry infinite
ca ca.crt
cert client3.crt
key client3.key
remote-cert-tls server
cipher AES-256-GCM
```

File ovpn server – GW300

```
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
server 192.168.100.0 255.255.255.0
push "route 192.168.1.0 255.255.255.0"
push "route 192.168.2.0 255.255.255.0"
route 192.168.2.0 255.255.255.0
client-config-dir ccd
client-to-client
keepalive 10 120
cipher AES-256-GCM
```

Nel server, in ccd/client2

```
iroute 192.168.2.0 255.255.255.0
```

Per tutti i dispositivi della VPN:

```
>> cd
>> cd ovpn
```

>> `openvpn client1.ovpn` oppure `client2.ovpn` oppure `client3.ovpn` oppure `server.ovpn` (a seconda del dispositivo)

Osservazioni

Nella cartella “catture” sono presenti alcune catture Wireshark effettuate sul sistema. Nella cartella “screenshot” sono presenti immagini degli output del terminale di alcune delle catture prima specificate.