Laboratorio di Configurazione e gestione Reti Locali

Università degli studi di Roma Tor Vergata

Gabriele Biscetti

20 09 2023

Sommario

Il presente documento rappresenta un progetto puramente accademico, senza pretese di completezza nella descrizione del dominio di interesse. Per affrontare il progetto assegnato verrà utilizzata una metodologia incrementale; questa prevede una suddivisione dei requisiti in funzionalità che verranno implementate ed integrate ad ogni iterazione.

Indice

1 Raccolta e analisi dei requisiti **2**
1.1 Requisiti espressi in linguaggio naturale 2

2 Implementazione dei requisiti **5**
2.1 Indirizzamento IP 5
2.2 Diagramma di rete fisico 7
2.3 Docker 7
2.4 Routing dinamico (Quagga) 9
2.5 Dynamic Host Configuration Protocol (DHCP) 10
2.6 Firewall 11
2.7 OpenVPN 13
2.8 Domain Name Server (DNS) 13
2.9 Certification Authority (CA) 13
2.10 Web Server 13
2.11 Quality of Service (QOS) 13

Raccolta e analisi dei requisiti

L'attività di raccolta dei requisiti, che precede la fase di progettazione, fornisce una specifica informale ma completa di tutte le proprietà e le funzionalità del sistema che si vuole progettare. In sintesi, si descrive il dominio di interesse.

Generalmente i requisiti che compongono la specifica vengono acquisiti dall'interazione con gli utenti, dalla documentazione esistente e/o eventuali realizzazioni preesistenti. In questo caso il fatto che non ci siano dei possibili utenti rende qualsiasi attività della raccolta legata all'interazione con gli utenti impossibile.

Quindi assumeremo per semplicità che il testo seguente sia una specifica completa e informale ottenuta da una raccolta dei requisiti opportunamente analizzati.

1.1 Requisiti espressi in linguaggio naturale

Il progetto richiede l'emulazione su piattaforma Kathara della configurazione di rete e dei servizi di una piccola azienda collegata ad internet con l'architettura mostrata in Figura 1.1 La sede centrale dispone di due connessioni ad internet, tramite due ISP differenti. Solo attraverso ISP1 si potrà essere raggiunti, mentre la connessione attraverso ISP2 è solo "in uscita". Considerare ISP2 come "default via".

Dal committente viene richiesto quanto segue:

1. Definire tutte le sottoreti locali delle LAN utilizzando indirizzi locali e subnet /24.
2. Definire le sottoreti per collegare tra loro R1, R2, R3 ed R4. Semplificare "Internet" con un'ulteriore sottorete tra R4 ed R5.
3. Assegnare un IP statico a tutti i router (Rx), sulle rispettive interfacce.
4. Assegnare un IP statico a WS1, WS2, DNS, Printer, "Intranet Server" e "Smart TV".
5. Abilitare in R1 il servizio DHCP per assegnare indirizzi dinamici a PC1, PC2, PC3 e PC4
6. Far assegnare gli IP statici di Printer, "Intranet Server", e "Smart TV" via DHCP con un match sul MAC address.
7. Abilitare in R5 il servizio DHCP per assegnare indirizzo dinamico a PC5
8. Configurare le regole di routing su tutti i router
9. Abilitare OSPF per propagare le regole di routing interne tra i router.
10. Abilitare il servizio SSH su: "Intranet Server", DNS, R1 ed R5. PC1 deve poter accedere a "Intranet Server" senza password.
11. Abilitare il servizio HTTP su: "Printer", "Smart TV"
12. Abilitare il servizio HTTPS su WS1 e WS2, per servire la stessa coppia di pagine sicure in Virtual-Hosting https://www.azienda.net e https://www.hosted.net , con due due certificati differenti (WS1 e WS2 sono uno il mirror dell'altro).

13. Abilitare su WS1 e WS2 una pagina privata accessibile con auth di tipo digest, che fornisca l'elenco dei file al suo interno
14. Abilitare su DNS il servizio DNS per risolvere solo internamente alla rete aziendale printer.azienda.net, ssh.azienda.net (per raggiungere "Intranet Server"), tv.azienda.net e router.azienda.net (per raggiungere R1).
15. Per poter verificare il virtual hosting HTTPS, aggiungere le entry corrette in /etc/hosts di PC5, puntando all'IP pubblico di R1 (lato ISP1).
16. Abilitare il servizio OpenVPN su R1, configurando PC5 come client per poter accedere alla rete centrale dalla connessione di casa.
17. Realizzare policy routing affinché il traffico uscente da LAN4 passi attraverso ISP1.
18. Realizzare LAN3 come VLAN, aggiungendo un elemento SW1 all'interno di LAN3 per staggare e taggare il traffico verso PC4.
19. Descrizione delle caratteristiche del firewall su R1:
    (a) Consentire traffico in forward iniziato da LAN1 verso LAN 4 (DMZ), e viceversa solo traffico RELATED
    (b) Consentire traffico in forward iniziato da LAN2 verso "Smart TV", e viceversa solo traffico RELATED
    (c) Consentire traffico in forward iniziato da LAN3 verso "Printer" e DNS, e viceversa solo traffico RELATED
    (d) Consentire traffico in forward da LAN1, LAN2 e LAN3 verso internet (ISP2), e viceversa solo traffico RELATED
    (e) Consentire traffico in forward da LAN4 verso internet, e viceversa solo traffico RELATED (se si fa policy routing al punto 17 usare ISP1, altrimenti ISP2)
    (f) Consentire traffico in forward da Internet (via ISP1) a LAN4 per il servizio HTTPS (vedi punto 19.k)
    (g) Consentire il traffico in ingresso (INPUT) per il servizio VPN da ISP1
    (h) Consentire in ingresso da ISP1 e ISP2 il ping con limite di 2 richieste al secondo
    (i) Consentire il traffico in forward tra VPN e LAN1, in entrambe le direzioni
    (j) Consentire il traffico in forward tra VPN e LAN 4, e viceversa solo traffico RELATED. Redirezionare le richieste HTTPS da ISP1 verso WS1 o WS2 in modalità round robin
    (k) Eseguire NAT (masquerade) per tutte le connessioni verso ISP1 ed ISP2
    (l) Droppare in INPUT e FORWARD (tabella filter) tutto quanto non specificato
20. Descrizione delle caratteristiche del firewall su R5:
    (a) Consentire traffico in forward iniziato dalla lan di PC5 verso internet, e viceversa solo traffico RELATED
    (b) Eseguire NAT (masquerade) per tutte le connessioni verso internet
21. Descrizioni vincoli di QoS su R1
    (a) Limitare uscita (tramite ISP2) per LAN2 e LAN3 a 10 Mbit/s
    (b) Limitare ingresso (tramite ISP2) di tutta LAN2 a 10 Mbit/s
    (c) Limitare ingresso (tramite ISP2) di tutta LAN3 a 10 Mbit/s
    (d) Limitare l'uscita (verso ISP2) e l'ingresso complessivo di tutta LAN1 a 100 Mbit/s
    (e) Limitare l'uscita (verso ISP2) e l'ingresso di tutte le macchine dentro LAN1 a 20 Mbit/s, con possibilità di usare la banda residua fino a 40 Mbit/s

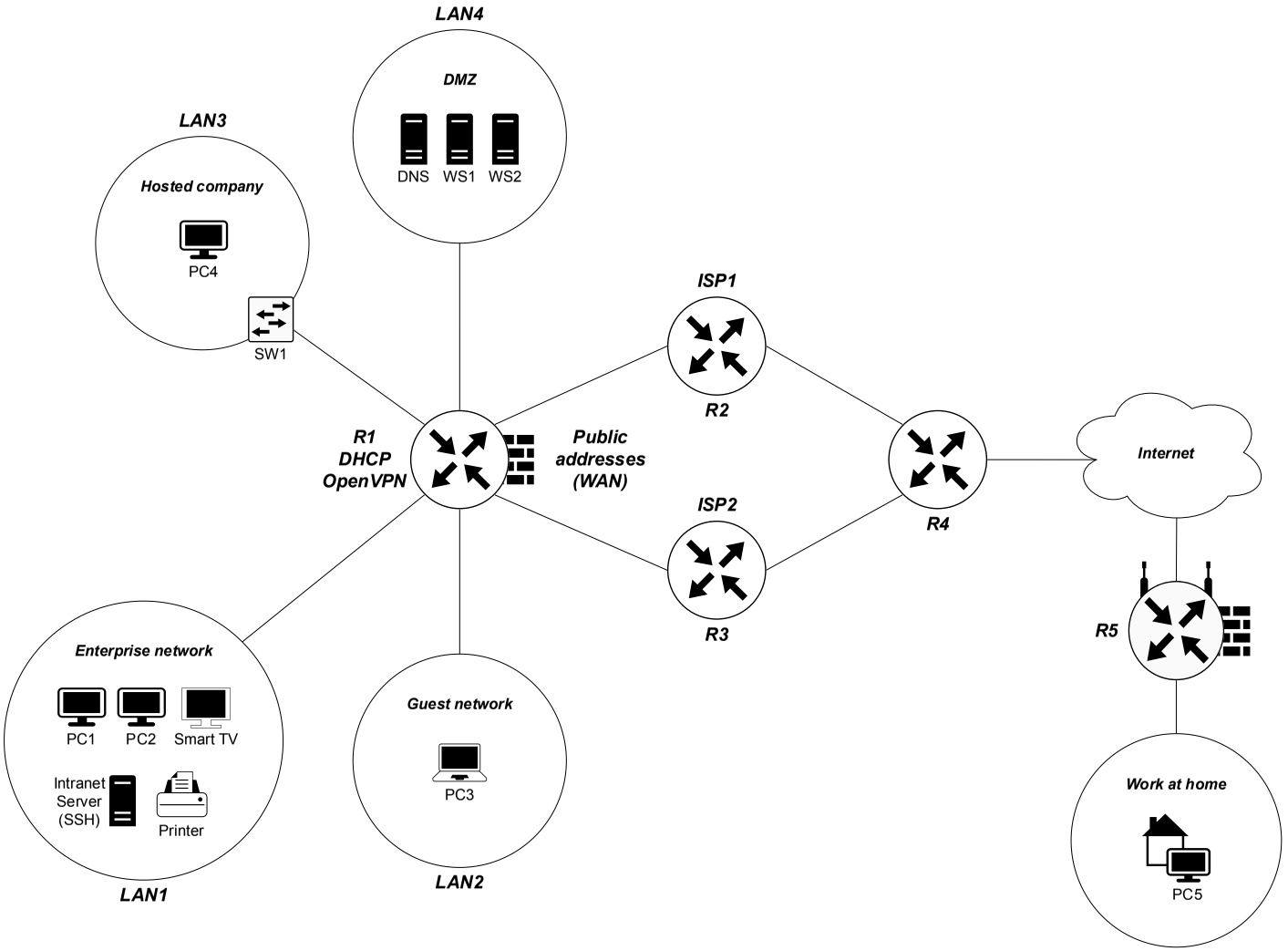

Figura 1.1: Il diagramma di rete da realizzare tramite *Kathara*

Implementazione dei requisiti

In questa fase ogni requisito, riportato dalla fase precedente, sarà soddisfatto dalla sua relativa implementazione. Queste ultime sono suddivise per sezioni permettendo quindi l'applicazione della metodologia incrementale.

2.1 Indirizzamento IP

Le subnet utilizzate sono riportate nella Tabella 2.1. Le subnet /24 sono state scelte in modo tale da poter essere ampliate nelle rispettive /23 a fronte di futuri cambiamenti nei requisiti progettuali.

Rete	**Indirizzamento**
LAN1	172.16.0.0/24
LAN2	172.16.2.0/24
LAN3	172.16.4.0/24
LAN4	172.16.6.0/24
ISP1 - Azienda	154.73.45.36/30
ISP2 - Azienda	46.31.161.216/30
ISP1 - Internet	193.3.50.0/30
ISP2 - Internet	194.63.146.0/30
Home - Internet	9.0.0.0/30
Home	192.168.1.0/24

Tabella 2.1: Subnet utilizzate nell'implementazione

Per ogni router coinvolto si riportano le interfacce utilizzate con i relativi indirizzi IP.

Interfaccia	Indirizzo IP
eth0	172.16.0.1/24
eth1	172.16.2.1/24
eth2.50	172.16.4.1/24
eth3	172.16.6.1/24
eth4	154.73.45.38/30
eth5	46.31.161.218/30

Tabella 2.2: indirizzi IP di **R1** (Azienda)

Interfaccia	Indirizzo IP
eth0	154.73.45.37/30
eth1	193.3.50.2/30

Tabella 2.3: indirizzi IP di **R2** (ISP1)

Interfaccia	Indirizzo IP
eth0	46.31.161.217/30
eth1	194.63.146.2/30

Tabella 2.4: indirizzi IP di **R3** (ISP2)

Interfaccia	Indirizzo IP
eth0	193.3.50.1/30
eth1	194.63.146.1/30
eth2	*docker-bridge*

Tabella 2.5: indirizzi IP di **R4** (Internet)

Interfaccia	Indirizzo IP
eth0	192.168.1.1/24
eth1	*docker-bridge*

Tabella 2.6: indirizzi IP di **R5** (Home)

Si riportano nella Tabella 2.7 gli indirizzi IP statici dei singoli endpoint (client e server).

Endpoint	Indirizzo IP	DHCP Reservation
WS1	172.16.6.11	NO
WS2	172.16.6.12	NO
DNS	172.16.6.10	NO
Printer	172.16.0.11	SI
Intranet Server	172.16.0.10	SI
Smart TV	172.16.0.12	SI

Tabella 2.7: indirizzi IP statici

2.2 Diagramma di rete fisico

In questa sezione riportiamo il diagramma di rete dove vengono aggiunte tutte le informazioni riportate nella sezione precedente. Il tutto è illustrato in Figura 2.1

2.3 Docker

Per associare immagini in base al ruolo del dispositivo, viene utilizzata una combinazione di *Docker compose* e *Dockerfile*. In questo modo la fase costruzione delle immagini è completamente automatizzata; utilizzando il comando `docker compose build` si ottengono le immagini obiettivo. L'idea è quella di definire un'immagine di base Debian per poi riutilizzarla nei vari server, tramite il meccanismo di ereditarietà offerto illustrato in Figura 2.2. Si riportano gli estratti dei file coinvolti.

docker-compose.yml

```
services:
router:
  image: kathara/quagga:router
  container_name: router
  build: ./docker/router/
endpoint:
  image: debian:endpoint
  container_name: endpoint
  build: ./docker/endpoint/
ssh-server:
  depends_on:
    - endpoint
  image: debian:ssh-server
  container_name: ssh-server
  build: ./docker/ssh-server/
web-server:
  depends_on:
    - ssh-server
  image: debian:web-server
  container_name: web-server
  build: ./docker/web-server/
name-server:
  depends_on:
    - ssh-server
  image: debian:name-server
  container_name: name-server
  build: ./docker/name-server/
```

Dockerfile (router)

```
FROM kathara/quagga
RUN apt-get update && \
    apt-get install -y \
    isc-dhcp-server \
    openvpn \
    iperf3 && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
```

Dockerfile (endpoint)

```
FROM debian:bookworm
RUN apt-get update && \
    apt-get install -y \
    iproute2 \
    iputils-ping \
    traceroute \
    isc-dhcp-client \
    dnsutils \
    netcat-traditional \
    openssh-client \
    vim \
    lynx \
    iperf3 \
```

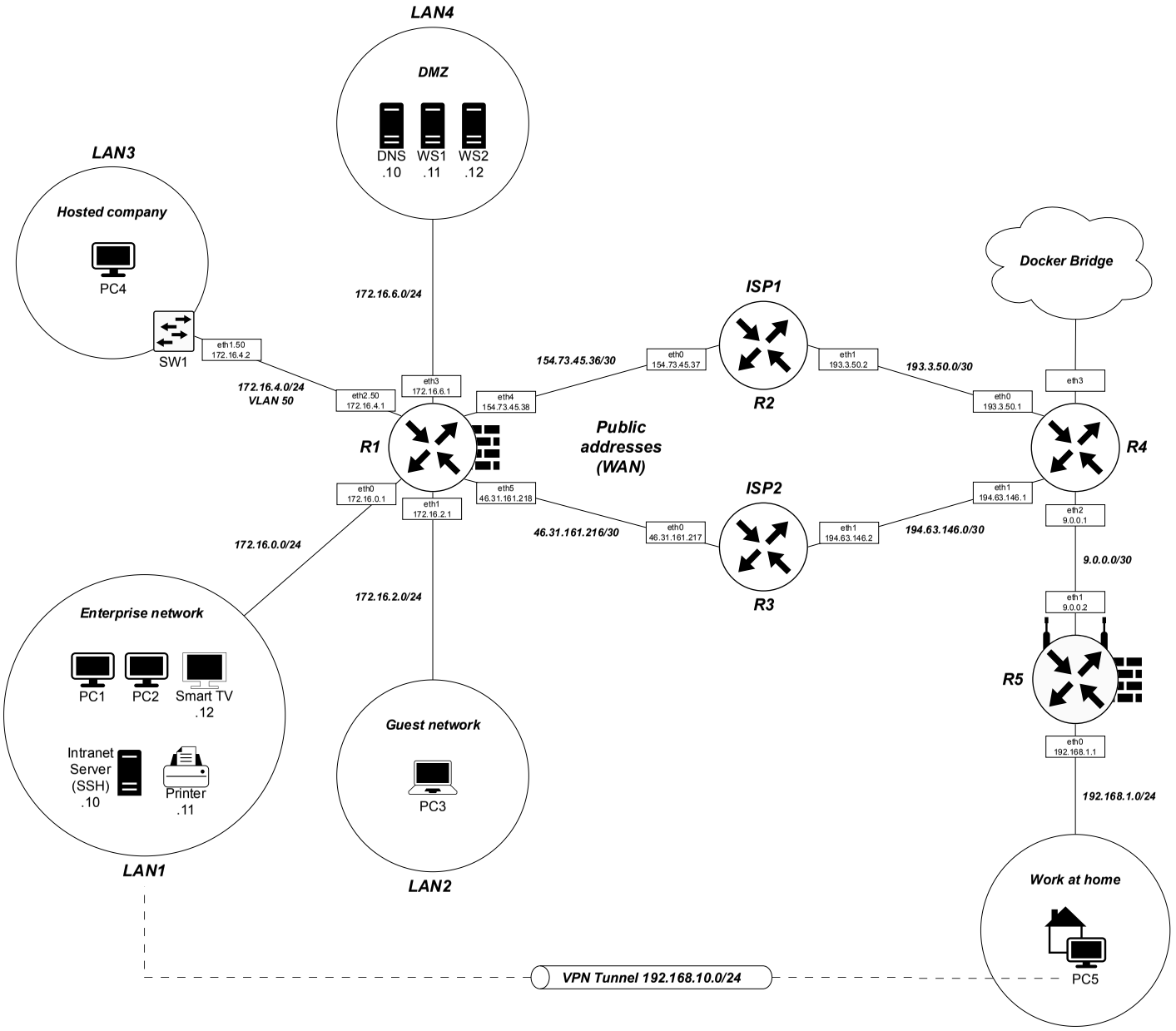

Figura 2.1: Il diagramma di rete fisico

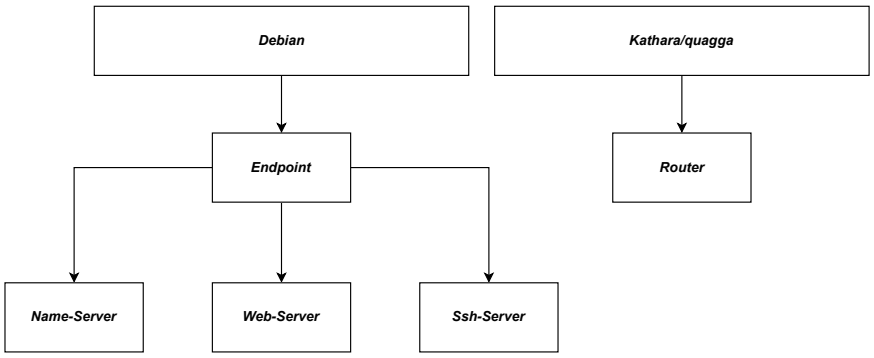

Figura 2.2: Ereditarietà dei container Docker

```
    openvpn && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
```

Dockerfile (name-server)

```
FROM debian:endpoint
RUN apt-get update && \
    apt-get install -y bind9 openssh-server && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
```

Dockerfile (ssh-server)

```
FROM debian:endpoint
RUN apt-get update && \
    apt-get install -y openssh-server && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
```

Dockerfile (ssh-server)

```
FROM debian:endpoint
RUN apt-get update && \
    apt-get install -y apache2 && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*
```

2.4 Routing dinamico (Quagga)

La costruzione della tabella di instradamento di ciascun router è affidato a Quagga utilizzando OSPF (Open Shortest Path First) offerto dal demone ospfd. Per forzare ISP2 come default gateway per l'azienda, viene utilizzata la direttiva *bandwidth* che permette di definire in maniera indiretta il costo del link. Si nota come l'utilizzo del protocollo OSPF non sia la migliore scelta per comunicazione Inter-AS (Autonomous System). Infatti questo protocollo è ottimo per comunicazioni Intra-AS. Questa scelta è solamente per scopi progettuali. In futuro si potrebbero implementare gli stessi requisiti utilizzando il BGP (Border Gateway Protocol), sempre disponibile nella suite dei protocolli offerti da Quagga.

Si ricorda come OSPF offra caratteristiche gerarchiche; le reti sono raggruppate in *aree* interconnesse da un'area *dorsale*; quest'ultima è quella che viene chiamata *area 0* (oppure *0.0.0.0*) ed è qui che i router appartenenti a quest'area scambiano fra loro le informazioni. La suddivisione in aree permette di limitare la distribuzione delle rotte; un router all'interno di un'area deve gestire la sola topologia per l'area a cui appartiene.

Per la semplicità, non si è adottata la divisione in aree (un ottimo esempio si trova presso [1] capitolo 17). In seguito si riportano gli estratti dei file di router 1; gli altri router hanno configurazioni simili (R4 in aggiunta comunica ai suoi vicini che sarà il default via per internet).

zebra.conf (r1))

```
hostname r1
password zebra
```

```
enable password zebra

interface eth4
ip address 154.73.45.38/30
bandwidth 5000
link-detect

interface eth5
ip address 46.31.161.218/30
bandwidth 100000
link-detect
```

ospfd.conf (r1))

```
hostname r1
password zebra
log file /var/log/quagga/ospfd.log

interface eth4
ospf hello-interval 1

interface eth5
ospf hello-interval 1

router ospf
  network 154.73.45.36/30 area 0.0.0.0
  network 46.31.161.216/30 area 0.0.0.0
```

2.5 Dynamic Host Configuration Protocol (DHCP)

Il server DHCP utilizzato è quello offerto dal pacchetto *isc-dhcp-server* per la sua semplicità nell'utilizzo. Si nota però, come riportato in questa pagina, la suite ISC sia deprecata in favore di Kea (tramite il pacchetto *kea-dhcp-ddns-server*). In seguito si riportano gli estratti dei file utilizzati per la configurazione.

dhcpd.conf (r1))

```
####################
#                  #
# GLOBAL SETTINGS #
#                  #
####################

option domain-name "azienda.net";
option domain-name-servers 172.16.6.10;

default-lease-time 600;
max-lease-time 7200;

ddns-update-style none;

authoritative;

#########################
#                       #
#   DHCP SUBNET LEASE   #
#                       #
#########################

subnet 172.16.0.0 netmask 255.255.255.0 {
        range 172.16.0.50 172.16.0.100;
        option routers 172.16.0.1;
}

subnet 172.16.2.0 netmask 255.255.255.0 {
        range 172.16.2.50 172.16.2.100;
        option routers 172.16.2.1;
}
```

```
subnet 172.16.4.0 netmask 255.255.255.0 {
        range 172.16.4.50 172.16.4.100;
        option routers 172.16.4.1;
}

#########################
#                       #
#   DHCP RESERVATION    #
#                       #
#########################

host intranet_server {
        hardware ethernet 18:66:DA:34:29:25;
        fixed-address 172.16.0.10;
}

host printer {
        hardware ethernet 08:00:72:03:02:51;
        fixed-address 172.16.0.11;
}

host smart_tv {
        hardware ethernet 00:23:C2:72:1C:6A;
        fixed-address 172.16.0.12;
}
```

2.6 Firewall

Il firewall perimetrale dell'azienda coincide con *R1*. In questo è installato *iptables* il quale è utilizzato per implementare i vicoli richiesti. Il seguente estratto riporta l'implementazione di tali vincoli, dove sono stati aggiunte ulteriori utili funzionalità extraprogettuali.

dhcpd.conf (r1))

```
################
# FILTER TABLE #
################

# Permit ospf
iptables -A INPUT -i eth4 -p ospf -m state --state NEW -j ACCEPT
iptables -A INPUT -i eth5 -p ospf -m state --state NEW -j ACCEPT
iptables -A FORWARD -i eth4 -p ospf -m state --state NEW -j ACCEPT
iptables -A FORWARD -i eth5 -p ospf -m state --state NEW -j ACCEPT

# Requirements: 19g,19h
iptables -A INPUT -i eth4 -p udp --dport 1194 -m state --state NEW -j ACCEPT
iptables -A INPUT -i eth4 -p icmp --icmp-type 8 -m limit --limit 2/sec -j ACCEPT
iptables -A INPUT -i eth5 -p icmp --icmp-type 8 -m limit --limit 2/sec -j ACCEPT
iptables -A INPUT -i tun0 -p icmp --icmp-type 8 -m limit --limit 2/sec -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Extra: this open ssh access to enterprise network
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW -j ACCEPT

iptables -A INPUT -j DROP

# Requirements: 19a-19f
iptables -A FORWARD \
  -i eth0 -o eth3 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth1 -o eth0 -d 172.16.0.12/32 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth2.50 -o eth0 -d 172.16.0.11/32 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth2.50 -o eth3 -d 172.16.6.10/32 -m state --state NEW -j ACCEPT

iptables -A FORWARD \
  -i eth0 -s 172.16.0.0/24 -o eth5 -m state --state NEW -j ACCEPT
```

```
iptables -A FORWARD \
  -i eth1 -s 172.16.2.0/24 -o eth5 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth2.50 -s 172.16.4.0/24 -o eth5 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth3 -s 172.16.6.0/24 -o eth4 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth4 -o eth3 -d 172.16.6.11/32 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth4 -o eth3 -d 172.16.6.12/32 -m state --state NEW -j ACCEPT

# Requirements: 19i-19j
iptables -A FORWARD \
  -i tun0 -s 192.168.10.0/24 -o eth0 -d 172.16.0.0/24 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i eth0 -s 172.16.0.0/24 -o tun0 -d 192.168.10.0/24 -m state --state NEW -j ACCEPT
iptables -A FORWARD \
  -i tun0 -s 192.168.10.0/24 -o eth3 -d 172.16.6.0/24 -m state --state NEW -j ACCEPT

# EXTRA RULE : Permit dns queries from guest network to ns.azienda.net.
# This is useful to resolv record like tv.azienda.net
iptables -A FORWARD \
  -i eth1 -s 172.16.2.0/24 -o eth3 -d 172.16.6.10/32 -p udp --dport 53 \
  -m state --state NEW -j ACCEPT

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -j DROP

##############
# NAT TABLE #
##############

# Requirements: 19l
iptables -t nat -A POSTROUTING -o eth4 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth5 -j MASQUERADE


# Requirements: 19k
# Another way is use Random balanging with
# -m statistic --mode random --probability .5
# In general, it works but there are problems with stateful protocol like TLS

#iptables -t nat -A PREROUTING -i eth4 -d 154.73.45.38/32 \
#         -p tcp --dport 80 \
#         -m state --state NEW \
#         -j DNAT --to 172.16.6.11:80

iptables -t nat -A PREROUTING -i eth4 -d 154.73.45.38/32 \
-p tcp --dport 443 \
-m state --state NEW \
-j DNAT --to 172.16.6.11:443

iptables -t nat -A PREROUTING -i eth4 -d 154.73.45.38/32 \
-p tcp --dport 80 \
-m state --state NEW \
-m statistic --mode nth --every 2 --packet 0 \
-j DNAT --to 172.16.6.11:80

iptables -t nat -A PREROUTING -i eth4 -d 154.73.45.38/32 \
-p tcp --dport 80 \
-m state --state NEW \
-m statistic --mode nth --every 1 --packet 0 \
-j DNAT --to 172.16.6.12:80

#iptables -t nat -A PREROUTING -i eth4 -d 154.73.45.38/32 \
#         -p tcp --dport 443 \
#         -m state --state NEW \
#         -m statistic --mode nth --every 2 --packet 0 \
#         -j DNAT --to 172.16.6.11:443
```

```
#iptables -t nat -A PREROUTING -i eth4 -d 154.73.45.38/32 \
#         -p tcp --dport 443 \
#         -m state --state NEW \
#         -m statistic --mode nth --every 1 --packet 0 \
#         -j DNAT --to 172.16.6.12:443
```

Un'alternativa per bilanciare https è quello di utilizzare software denominati *Reverse Proxy*. Questi si comportano come intermediari nella comunicazione, I più noti sono sicuramente squid e modulo di apache mod_proxy. Un esempio con *mod_proxy*:

dhcpd.conf (r1))

```
<VirtualHost *:443>
  <Proxy balancer://mycluster>
      BalancerMember http://127.0.0.1:8080
      BalancerMember http://127.0.0.1:8081
  </Proxy>

  ProxyPreserveHost On

  ProxyPass / balancer://mycluster/
  ProxyPassReverse / balancer://mycluster/
</VirtualHost>
```

2.7 OpenVPN

2.8 Domain Name Server (DNS)

2.9 Certification Authority (CA)

Per semplicità nell'implementazione, il firewall aziendale (R1) è anche la Certification Authority (CA) interna; in uno scenario reale, questa non è una buona idea ... L'emissione dei certificati per il servizio *OpenVPN* e *Web Server* è affidata ad *EASY-RSA*; una collezione di script per semplificare la gestione di una CA.

2.10 Web Server

wget –no-check-certificate –user gabriele –password lcgrl https://www.azienda.net/internal

2.11 Quality of Service (QOS)

Il controllo del traffico è implementato grazie a *tc*; software che permette di configurare l'organizzazione dei pacchetti gestiti dal Kernel.

Bibliografia

[1] Bert Hubert et al. *Linux Advanced Routing and Traffic Control HOWTO*. 2012.

[2] Simone Piccardi. *Amministrazione dei servizi web*. 2012.

[3] Simone Piccardi. *Ammnistrare GNU/Linux*. 2023.

[4] Simone Piccardi. *La sicurezza con GNU/Linux*. 2003.

[5] Debian Project. *wiki.debian.org*. 2023.