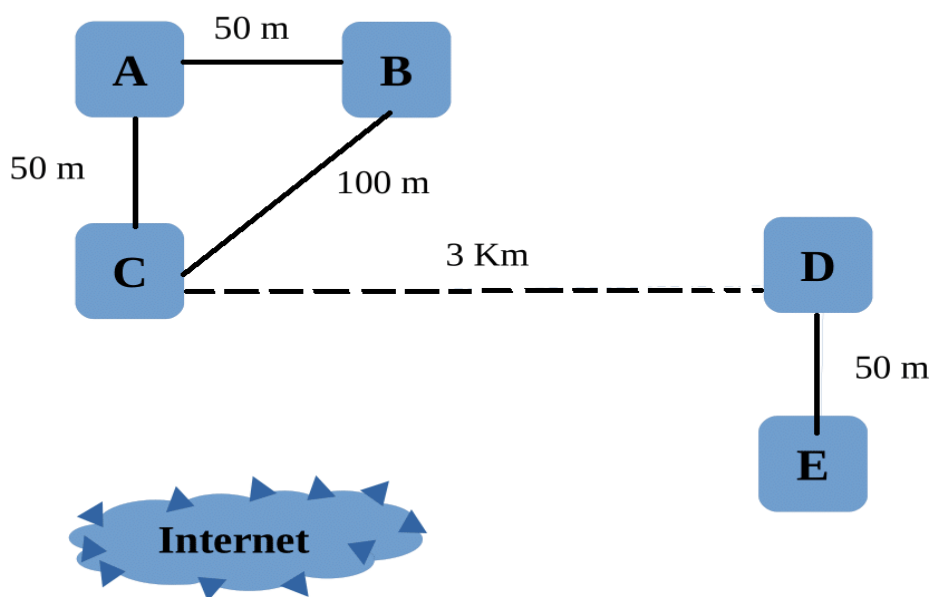


1. Descrizione del progetto

La ditta GreenPeace ha deciso di collegare in rete tutti i suoi reparti ed uffici e ci ha contattato per disegnare, installare e gestire l'intera rete. Di seguito sono illustrate le specifiche iniziali richieste in termini di topologia e di risorse.



Gli edifici sopra rappresentati, hanno le seguenti caratteristiche:

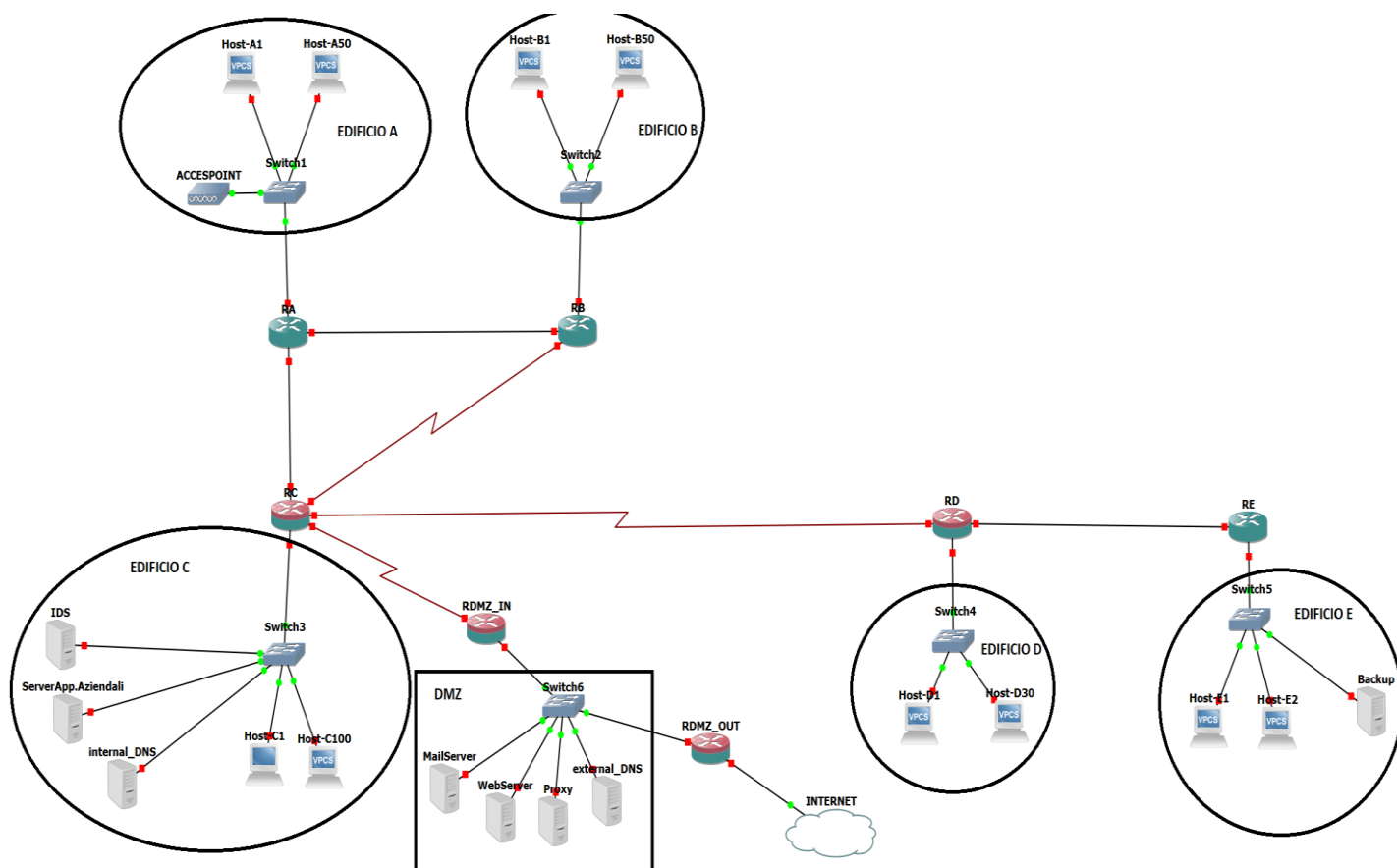
Edificio	Uffici & Reparti	Num. Utenti	Num. Server	Copertura Wi-fi
A		50		SI
B		50		NO
C		100		NO
D		30		NO
E		20		NO

All'interno dell'azienda devono essere presenti i seguenti *Server*:

X	Server di posta elettronica	N. 1		Server Proxy	N. 1
X	Server Web	N. 1		Server Fax	N. 0
X	Server DNS	N. >=2	X	Server di Backup	N. 1
X	Server per applicazioni aziendali	N. 1		Server	_____ N. 0

La rete prevede una connessione protetta ad *Internet*.

2. Schema fisico di rete



Un diagramma **di rete fisico** mostra la disposizione **fisica** effettiva **dei** componenti che compongono la **rete**, inclusi cavi e hardware. In genere, tale diagramma offre una vista dall'alto **della rete** nel suo spazio **fisico**, come una pianta.

Partendo dall'**EDIFICIO A** vediamo che in esso sono presenti i soli 50 host, relativo switch ethernet e i vari access point per fornire una copertura wi-fi completa che ricopra tutto l'edificio.

Questo edificio è connesso direttamente all'**EDIFICIO B** (il quale ha solo i relativi 50 host e switch) e all'**EDIFICIO C**, entrambi i collegamenti sono fatti mediante cavi ethernet in fibra ottica data la vicinanza a quest'ultimi.

Quest'ultimo edificio è quello che richiede il maggior numero di host (100 host). Dato questo fatto e anche la sua rispettiva **centralità nella rete**, essendo a pochi metri dai due edifici del **blocco ABC** e essendo come punto di unico collegamento al **blocco DE**, sarà l'edificio in cui oltre ai rispettivi host, vi avremo i **servizi di monitoraggio di rete IDS, il server per applicazioni aziendali e il DNS interno**. Sempre in questo edificio avremo la **DMZ che conterrà i server necessari all'erogazione di servizi verso l'Internet pubblico come il server per la posta elettronica, il web server, il proxy e il DNS esterno**.

Ovviamente la DMZ sarà isolata e controllata con due router firewall alle due estremità, uno verso l'internet pubblico e uno verso la rete dell'azienda in modo da avere una protezione massima.

Questo edificio è a sua volta collegato all'**EDIFICIO D** mediante una **VPN basata su IPsec in modalità tunnel** (in quanto è distante diversi km), questo porta quindi ad avere due router firewall anche alle due estremità della VPN. In questo modo abbiamo una doppia copertura che permette di salvaguardare la rete anche nel caso in cui vi siano intrusioni o altri problemi di sicurezza in uno dei due blocchi di edifici.

Nell'**EDIFICIO D** vi sono solo i relativi 30 host e gli switch ethernet. Quest'ultimo è direttamente collegato all'**EDIFICIO E** mediante cavo ethernet in fibra ottica.

L'**EDIFICIO E**, in quanto è il più isolato e distante da tutto il **blocco ABC** che è collegato all'internet e sarà l'edificio in cui verrà posto il **server di backup per fornirgli la massima protezione possibile**, dati anche i firewall da superare prima di arrivare a tale edificio e i pochi host dell'unico edificio a lui collegato e nell'edificio stesso(solo 20 host).

3. Schema logico

- L'indirizzo IP fornito all'azienda è di classe C in quanto le interfacce collegate direttamente in internet sono poche(100 max), mentre le restanti saranno indirizzi privati in quanto si fa uso di NAT, il quale che consente di utilizzare un singolo IP privato per affacciare una intera sottorete all'esterno (internet)
- La subnetmask che abbiamo utilizzato per la divisione degli edifici e' la 255.255.255.0 che permette di avere 254 sottoreti con un massimo di 254 host l'una

Sottoreti degli edifici:

Edificio	Sottorete
A	192.168.1.0/24
B	192.168.2.0/24
C	192.168.3.0/24
D	192.168.4.0/24
E	192.168.5.0/24
DMZ	192.168.6.0/24

Per quanto riguarda gli host dei singoli edifici verra' implementato routing statico, mentre per le connessioni tra edifici differenti verra' implementato routing dinamico.

Backbone:

Per collegare 2 router abbiamo utilizzato una sottorete di appoggio cosi' da evitare errori nel calcolo degli hop da parte degli algoritmi di routing.

Edifici	Sottorete di appoggio
Interconnessione A-B	192.168.7.0
Interconnessione A-C	192.168.8.0
Interconnessione C-B	192.168.9.0
Interconnessione C-D	192.168.10.0
Interconnessione D-E	192.168.11.0
RDMZ_IN - RC	192.168.12.0

4. Routing

Per quanto riguarda la gestione del routing, abbiamo scelto di utilizzare il protocollo RIPv2, in quanto è quello che più si appresta alla nostra situazione di rete. Questo protocollo è basato sul principio di funzionamento vettore-distanza. In accordo a tale schema ogni router è responsabile di consegnare a ciascun vicino una copia completa della propria tabella di routing. La consegna di tali annunci avviene sotto forma di pacchetti IP con indirizzo destinazione riservato multicast 224.0.0.9 precisamente per la versione 2 del protocollo RIP. Alla ricezione di un annuncio, ciascun router confronta le informazioni ricevute con quelle mantenute nella propria routing table, al fine di individuare i percorsi migliori. Abbiamo scelto proprio la versione 2 in quanto esso consente di implementare metodi di autenticazione per i messaggi ricevuti, in modo da prevenire qualsiasi possibile attacco. Lo schema utilizzato è quello di processare l'intero messaggio RIP con una funzione di hash MD5, in modo da prevenire anche ogni forma di intercettazione del traffico. Il numero massimo di hop supportato dal protocollo in questione è pari a 15. Una rotta avente metrica superiore a 15, di conseguenza, viene marcata come irraggiungibile, ma comunque nel nostro caso non è un problema degno di nota.

Esistono, inoltre, diversi metodi attraverso i quali è possibile limitare la formazione dei famigerati routing loop. Uno di questi è lo split-horizon, ovvero un router che ha ricevuto l'update di una rotta attraverso una determinata porta (ad esempio Ethernet0), non potrà rimandare indietro l'update lungo quella stessa porta. Un altro metodo è rappresentato dal route poisoning. Nella fattispecie, per fare in modo che non si formino loop, un router assegna ad una determinata rotta una metrica fittizia pari a 16 (in questo modo la rotta viene "avvelenata"), costringendo i router che hanno ricevuto l'update a scegliere percorsi alternativi. Quello da noi utilizzato sarà il secondo metodo.

Per abilitare il rip basta digitare:

```
Router(config)# router rip
```

Successivamente, se si vuole utilizzare la versione 2, basta scrivere:

```
Router(config-router)# version 2
```

Una volta fatto ciò possiamo dichiarare le reti direttamente connesse al router, le quali verranno propagate mediante gli update:

```
Router(config-router)# network 10.1.2.0  
Router(config-router)# network 172.16.2.0
```

Nel caso in cui avessimo a che fare con molte subnet simili tra loro, per evitare eventuali errori bisogna disabilitare l'auto summarization mediante il comando (vale solo per il RIPv2):

```
Router(config-router)# no auto-summary
```

Per abilitare l'autenticazione su una determinata interfaccia occorre digitare (per utilizzare md5):

```
Router(config-if)# ip rip authentication mode md5
```

Per ottenere informazioni di diagnostica sullo scambio degli update e sull'autenticazione occorre usare il comando:

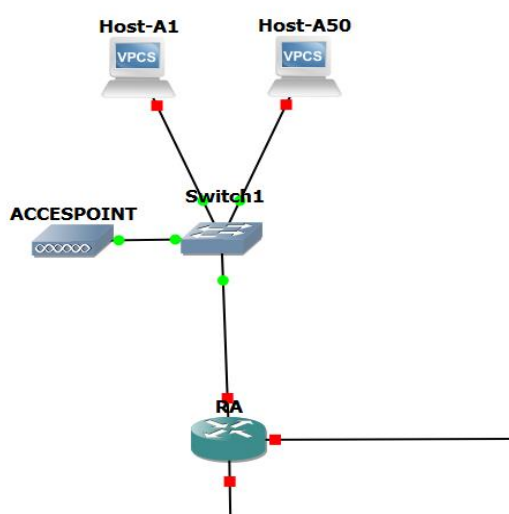
```
Router# debug ip rip
```

mentre per verificare che una determinata rotta sia stata imparata grazie a tale protocollo di routing basta digitare (le rotte marcate con R sono quelle identificate mediante RIP):

```
Router# sh ip route
```

5. Configurazioni interfacce di rete

EDIFICIO A



HOST: 50

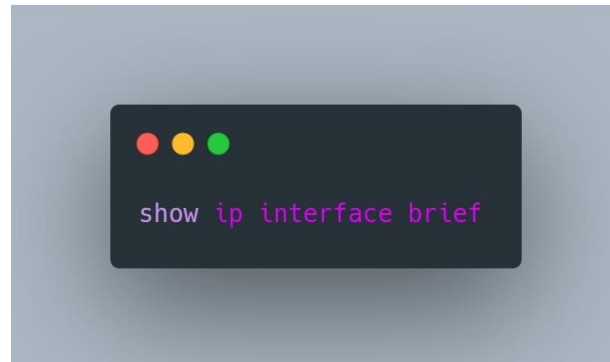
SOTTORETE: 192.168.1.0

COLLEGAMENTI: *Edifici B, C*

Codice	Dispositivo	Indirizzo IP
Host-A1	Host	192.168.1.2
.....
Host-A50	Host	192.168.1.52
RA	Router	192.168.1.1

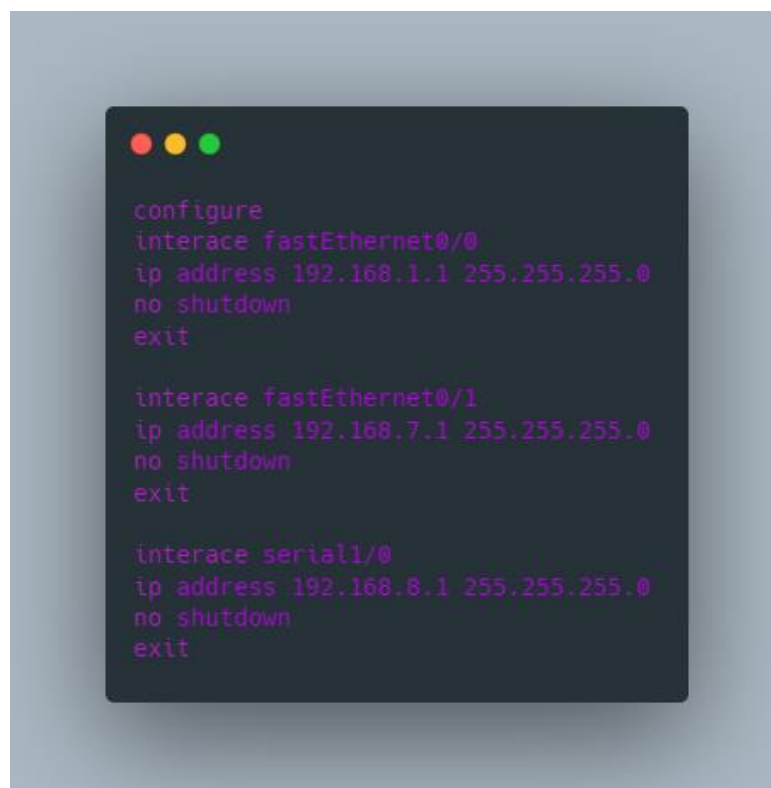
Configurazione router(RA)

Come prima cosa mostriamo tutte le interfacce disponibili e le loro impostazioni.



```
RA#show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
FastEthernet0/0    unassigned      YES unset    administratively down down
FastEthernet0/1    unassigned      YES unset    administratively down down
Serial1/0           unassigned      YES unset    administratively down down
Serial1/1           unassigned      YES unset    administratively down down
Serial1/2           unassigned      YES unset    administratively down down
Serial1/3           unassigned      YES unset    administratively down down
RA#configure
```

Come possiamo vedere tutte le interfacce sono disattivate e non impostate, passiamo ora a configurare quelle che ci occorrono e a impostare i protocolli di Routing e le altre configurazioni.



Abbiamo così acceso e impostato tutte le interfacce, procediamo con impostare il routing e il metodo di autenticazione in esso, per implementare la sicurezza.

```
router rip
version 2
network 192.168.1.0
network 192.168.7.0
network 192.168.8.0
no auto-summary
exit
interface fastEthernet0/0
ip rip authentication mode md5
exit
interface fastEthernet0/1
ip rip authentication mode md5
exit
interface fastEthernet1/0
ip rip authentication mode md5
exit

debug ip rip
sh ip route
```

Andiamo ora a settare nel router l'ip del dns interno e dato che in questo edificio abbiamo una copertura wifi, imposteremo il dhcp per assegnare gli ip agli host (impostando un aggiornamento giornaliero degli ip).

```
ip domani-lookup
ip name-server 192.168.3.103
exit

configure
no ip dhcp use vrf connected
ip dhcp excluded-address 192.168.1.250 192.168.1.254
ip dhcp pool snwA
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
lease 1
exit
exit

copy running-config startup-config
wr
```

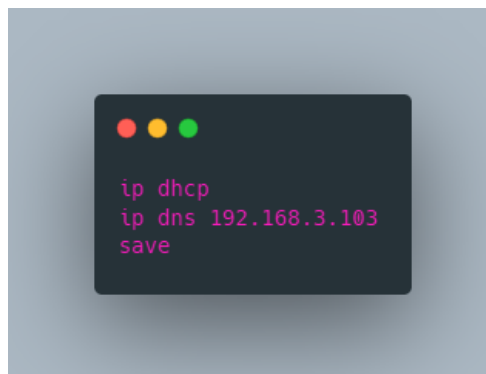

Come possiamo vedere il router manda continui aggiornamenti ai vicini per le Routing table e le rispettive interfacce sono settate.

```
Router# ip rip route
*Mar 1 00:30:25.943: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.1.1)
*Mar 1 00:30:25.943: RIP: build update entries
*Mar 1 00:30:25.943: 192.168.7.0/24 via 0.0.0.0, metric 1, tag 0
Router# ip rip route
*Mar 1 00:30:33.407: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (192.168.7.1)
*Mar 1 00:30:33.407: RIP: build update entries
*Mar 1 00:30:33.407: 192.168.1.0/24 via 0.0.0.0, metric 1, tag 0
Router# ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.7.0/24 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
Router#
*Mar 1 00:30:51.867: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.1.1)
*Mar 1 00:30:51.867: RIP: build update entries
*Mar 1 00:30:51.867: 192.168.7.0/24 via 0.0.0.0, metric 1, tag 0
Router#
*Mar 1 00:31:01.287: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (192.168.7.1)
*Mar 1 00:31:01.287: RIP: build update entries
*Mar 1 00:31:01.287: 192.168.1.0/24 via 0.0.0.0, metric 1, tag 0
Router#
*Mar 1 00:31:18.275: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.1.1)
*Mar 1 00:31:18.275: RIP: build update entries
*Mar 1 00:31:18.275: 192.168.7.0/24 via 0.0.0.0, metric 1, tag 0
Router#
*Mar 1 00:31:30.971: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (192.168.7.1)
*Mar 1 00:31:30.971: RIP: build update entries
*Mar 1 00:31:30.971: 192.168.1.0/24 via 0.0.0.0, metric 1, tag 0
Router#
*Mar 1 00:31:45.731: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.1.1)
*Mar 1 00:31:45.731: RIP: build update entries
*Mar 1 00:31:45.731: 192.168.7.0/24 via 0.0.0.0, metric 1, tag 0
Router#
*Mar 1 00:31:56.863: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (192.168.7.1)
*Mar 1 00:31:56.863: RIP: build update entries
```

Configurazione host (Host-A1)



```
RA Host-A1
Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

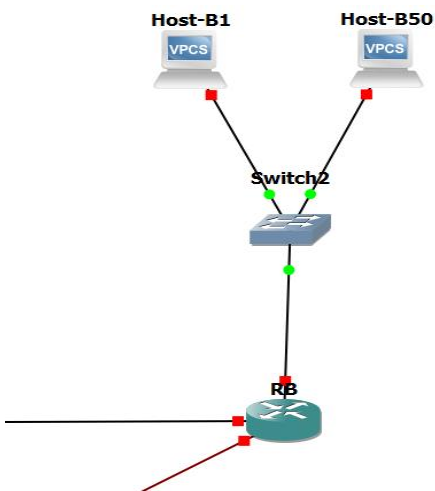
Press '?' to get help.

Executing the startup file

DDORA IP 192.168.1.2/24
GW 192.168.1.1

Host-A1>
```

EDIFICIO B



HOST: 50

SOTTORETE: 192.168.2.0

COLLEGAMENTI: *Edifici A, C*

Codice	Dispositivo	Indirizzo IP
Host-B1	Host	192.168.2.2
.....
Host-B50	Host	192.168.2.52
RB	Router	192.168.2.1

Configurazione host (Host-B1)

```
ip 192.168.2.2/24 192.168.2.1
ip dns 192.168.3.103
save
```

Configurazione router (RB)

```
configure
interface FastEthernet0/1
ip address 192.168.7.2 255.255.255.0
no shutdown
exit
interface FastEthernet0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
exit
interface Serial1/2
ip address 192.168.9.1 255.255.255.0
no shutdown
exit

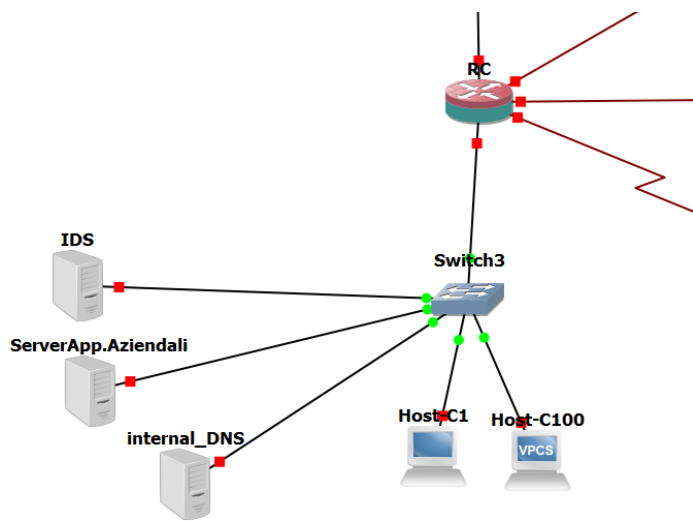
router rip
version 2
network 192.168.2.0
network 192.268.7.0
network 192.168.9.0
no auto-summary
exit

interface FastEthernet0/0
ip rip authentication mode md5
exit
interface FastEthernet0/1
ip rip authentication mode md5
exit
interface Serial1/2
ip rip authentication mode md5
exit
exit
debug ip rip
sh ip route

ip domain lookup
ip name-server 192.168.3.103
exit

copy running-config startup-config
wr
```

EDIFICIO C



HOST: 100

SOTTORETE: 192.168.3.0

COLLEGAMENTI: *Edifici B, C*

Codice	Dispositivo	Indirizzo IP
Host-C1	Host	192.168.3.2
.....
Host-C100	Host	192.168.3.102
Internal_DNS	Server	192.168.3.103
RC	Router	192.168.3.1
Server App. Aziendali	Server	192.168.3.104
IDS	Server	192.168.3.105

Configurazione host (Host-C1)

```
ip 192.168.3.2/24 192.168.3.1
ip dns 192.168.3.103
save
```

Configurazione router (RC)

```
configure
interface FastEthernet0/0
ip address 192.168.3.1 255.255.255.0
no shutdown
exit
interface FastEthernet0/1
ip address 192.168.12.1 255.255.255.0
no shutdown
exit
interface Serial1/0
ip address 192.168.8.2 255.255.255.0
no shutdown
exit
interface Serial1/1
ip address 192.168.10.1 255.255.255.0
no shutdown
exit
interface Serial1/2
ip address 192.168.9.2 255.255.255.0
no shutdown
exit

router rip
version 2
network 192.168.3.0
network 192.268.8.0
network 192.168.9.0
network 192.268.10.0
network 192.168.12.0
no auto-summary
exit

interface FastEthernet0/0
ip rip authentication mode md5
exit
interface FastEthernet0/1
ip rip authentication mode md5
exit
interface Serial1/0
ip rip authentication mode md5
exit
interface Serial1/1
ip rip authentication mode md5
exit
interface Serial1/2
ip rip authentication mode md5
exit
exit
debug ip rip
sh ip route

configure
ip domain lookup
ip name-server 192.168.3.103
exit

copy running-config startup-config
wr
```

Configurazione Internal DNS

File di configurazione **resolv.conf** del resolver, il quale comprende la lista dei **name server** da interrogare

```
domain greenPeace.it
search greenPeace.it
nameserver 192.168.3.103
nameserver 192.168.6.6
nameserver 1.1.1.1
nameserver 1.0.0.1
```

File **named.conf.local** e i relativi **zone files** per la rete locale, usati dal **daemon named** per rispondere alle richieste:

```
// DEFINIZIONE MASTER
// internal_DNS è master per edificioc.greenPeace.it
zone "edificioc. greenPeace.it" {
    type master;
    file "/etc/bind/ edificioc. greenPeace.it.db";
};
```

```
// reverse mapping per edificioc. greenPeace.it
zone "3.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/3.168.192.in-addr.arpa.db";
};
```

```
// internal_DNS è master per edificioa.greenPeace.it
zone "edificioa. greenPeace.it" {
    type master;
    file "/etc/bind/ edificioa. greenPeace.it.db";
};
```

```
// reverse mapping per edificioa. greenPeace.it
zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/1.168.192.in-addr.arpa.db";
};
```

```
// internal_DNS è master per edificiob.greenPeace.it
zone "edificiob. greenPeace.it" {
    type master;
    file "/etc/bind/ edificiob. greenPeace.it.db";
};
```

```
// reverse mapping per edificiob. greenPeace.it
zone "2.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/2.168.192.in-addr.arpa.db";
};
```

```
// internal_DNS è master per edificiod.greenPeace.it
zone "edificiod. greenPeace.it" {
    type master;
    file "/etc/bind/ edificiod. greenPeace.it.db";
};
```

```
// reverse mapping per edificiod. greenPeace.it
zone "4.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/4.168.192.in-addr.arpa.db";
};
```

```
// internal_DNS è master per edificioe.greenPeace.it
zone "edificioe. greenPeace.it" {
    type master;
    file "/etc/bind/ edificioe. greenPeace.it.db";
};
```

```
// reverse mapping per edificioe. greenPeace.it
zone "5.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/5.168.192.in-addr.arpa.db";
};
```

// DEFINIZIONE SLAVE

```
// internal_DNS è slave per greenPeace.it
zone "greenPeace.it" {
    type slave; file "/etc/bind/ greenPeace.it.bk";
    masters { 192.168.6.6; };
};
```

```
// reverse mapping per greenPeace.it
zone "168.192.in-addr.arpa" {
    type slave;
    file "/etc/bind/168.192.in-addr.arpa.bk";
    masters { 192.168.6.6; };
};
```

```
// internal_DNS è slave per dmz.greenPeace.it
```

```
zone "dmz.greenPeace.it" {  
    type slave;  
    file "/etc/bind/dmz. greenPeace.it.bk";  
    masters { 192.168.6.6; };  
};
```

```
// reverse mapping per dmz. greenPeace.it
```

```
zone "6.168.192.in-addr.arpa" {  
    type slave;  
    file "/etc/bind/6.168.192.in-addr.arpa.bk";  
    masters { 192.168.6.6; };  
};
```

File named.conf.option

```
acl "trusted-nameservers" {  
    localhost;  
    192.168.3.103;  
    192.168.6.6;  
};  
acl "trusted-networks" {  
    localhost;  
    192.168.3.0/24;  
    192.168.6.0/24;  
    192.168.1.0/24;  
    192.168.2.0/24;  
    192.168.4.0/24;  
    192.168.5.0/24;  
};  
options {  
    directory "/var/cache/bind";  
    dnssec-validation auto;  
    auth-nxdomain no;  
    version "Not disclosed";  
    notify yes;  
    allow-transfer { trusted-nameservers; };  
    allow-query { trusted-networks; };  
    forwarders { 1.1.1.1; };  
    recursion yes;  
};
```


File edificioc.greenPeace.i.db

\$TTL 86400

\$ORIGIN edificioc.greenPeace.it.

```
@      IN      SOA  dns. edificioc.greenPeace.it.  root. edificioc.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;    expire after 1000 hours
                                2592000 ;    default ttl )
```

; Definizione dei nameserver e dei server mail

```
      IN      NS    dns.dmz.greenPeace.it.
      IN      NS    dns. edificioc.greenPeace.it.
      IN      NS    dns.cloudflare.com.
      IN      MX    10   mail.GreenPeace.it.
```

; Host in edificioc.greenPeace.it

```
RC     IN      A     192.168.3.1
dns     IN      A     192.168.3.103
app     IN      A     192.168.3.104
ids     IN      A     192.168.3.105
```

File edificioa.greenPeace.i.db

\$TTL 86400

\$ORIGIN edificioa.greenPeace.it.

```
@      IN      SOA  dns. edificioa.greenPeace.it.  root. edificioa.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;    expire after 1000 hours
                                2592000 ;    default ttl )
```

; Definizione dei nameserver e dei server mail

```
      IN      NS    dns.dmz.greenPeace.it.
      IN      NS    dns. edificioc.greenPeace.it.
      IN      NS    dns.cloudflare.com.
      IN      MX    10   mail.GreenPeace.it.
```

; Host in edificioc.greenPeace.it

```
RA     IN      A     192.168.1.1
```

File edificiob.greenPeace.i.db

\$TTL 86400

\$ORIGIN edificiob.greenPeace.it.

```
@      IN      SOA    dns. edificiob.greenPeace.it.    root. edificiob.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;    expire after 1000 hours
                                2592000 ;    default ttl )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns. edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10      mail.GreenPeace.it.
```

; Host in edificiob.greenPeace.it

```
RB      IN      A      192.168.2.1
```

File edificiod.greenPeace.i.db

\$TTL 86400

\$ORIGIN edificiod.greenPeace.it.

```
@      IN      SOA    dns. edificiod.greenPeace.it.    root. edificiod.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;    expire after 1000 hours
                                2592000 ;    default ttl )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns. edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10      mail.GreenPeace.it.
```

; Host in edificiod.greenPeace.it

```
RD      IN      A      192.168.4.1
```

File edificioe.greenPeace.i.db

\$TTL 86400

\$ORIGIN edificioe.greenPeace.it.

```
@      IN      SOA    dns. edificioe.greenPeace.it.    root. edificioe.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;    expire after 1000 hours
```

2592000 ; default ttl)

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns.edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10    mail.GreenPeace.it.
```

; Host in edificioe.greenPeace.it

```
RE      IN      A      192.168.5.1
```

File 3.168.192.in-addr.arpa.db

\$TTL 86400

\$ORIGIN 3.168.192.in-addr.arpa.

```
@      IN      SOA      dns.edificioc.greenPeace.it. root.edificioc.greenPeace.it. (
                                2018112701 ; serial
                                43200 ; refresh
                                3600 ; retry after 1 hour
                                3600000 ; expire after 1000 hours
                                2592000 ; default ttl
                                )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns.edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10    mail.greenPeace.it.
```

; Host in edificioc.greenPeace.it

```
1      IN      PTR      RC.edificioc.greenPeace.it.
103    IN      PTR      dns.edificioc.greenPeace.it.
104    IN      PTR      app.edificioc.greenPeace.it.
105    IN      PTR      ids.Edificioc.greenPeace.it.
```

File 1.168.192.in-addr.arpa.db

\$TTL 86400

\$ORIGIN 1.168.192.in-addr.arpa.

```
@      IN      SOA      dns.edificioa.greenPeace.it. root.edificioa.greenPeace.it. (
                                2018112701 ; serial
                                43200 ; refresh
                                3600 ; retry after 1 hour
                                3600000 ; expire after 1000 hours
                                2592000 ; default ttl
                                )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
```

```
IN      NS      dns.edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10      mail.greenPeace.it.
```

; Host in edificioa.greenPeace.it

```
1      IN      PTR      RA.edificioa.greenPeace.it.
```

File 2.168.192.in-addr.arpa.db

\$TTL 86400

\$ORIGIN 2.168.192.in-addr.arpa.

```
@      IN      SOA      dns.edificiob.greenPeace.it. root.edificiob.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;   expire after 1000 hours
                                2592000 ;   default ttl
                                )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns.edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10      mail.greenPeace.it.
```

; Host in edificioa.greenPeace.it

```
1      IN      PTR      RB.edificiob.greenPeace.it.
```

File 4.168.192.in-addr.arpa.db

\$TTL 86400

\$ORIGIN 4.168.192.in-addr.arpa.

```
@      IN      SOA      dns.edificiod.greenPeace.it. root.edificiod.greenPeace.it. (
                                2018112701 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;   expire after 1000 hours
                                2592000 ;   default ttl
                                )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns.edificioc.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10      mail.greenPeace.it.
```

; Host in edificioa.greenPeace.it

```
1      IN      PTR      RD.edificiod.greenPeace.it.
```

File 5.168.192.in-addr.arpa.db

\$TTL 86400

\$ORIGIN 5.168.192.in-addr.arpa.

@ IN SOA dns.edificioe.greenPeace.it. root.edificioe.greenPeace.it. (
2018112701 ; serial
43200 ; refresh
3600 ; retry after 1 hour
3600000 ; expire after 1000 hours
2592000 ; default ttl
)

; Definizione dei nameserver e dei server mail

IN NS dns.dmz.greenPeace.it.
IN NS dns.edificioc.greenPeace.it.
IN NS dns.cloudflare.com.
IN MX 10 mail.greenPeace.it.

; Host in edificioa.greenPeace.it

1 IN PTR RE.edificioe.greenPeace.it.

Server Applicazioni Aziendali

Abbiamo utilizzato la tecnica del wrapper al fine di limitare gli accessi ai servizi in base agli indirizzi ip e agli hostname dei client . Le richieste dei client vengono intercettate e gestite da un servizio di autenticazione che sfrutta rispettivamente due file, “**host.deny**” e “**host.allow**”, per gestire i servizi presenti sui server. Viene inoltre utilizzato il demone **xinetd** per effettuare controlli a livello di applicazione e mantiene un log di ogni accesso ed anche tentativo. I vantaggi che ci offre sono: trasparenza per il client e per il servizio di rete, gestione centralizzate di più protocolli. La libreria da utilizzare è **/usr/lib/libwrap.a** . Il comando **host.allow** indica quali host hanno accesso alla rete:

/etc/hosts.allow ALL: .greePeace.it --> diamo il permesso solamente a gli host della rete.

Per negare l’accesso ad altri host:

/etc/hosts.deny

Per controllare gli accessi ai servizi di rete usiamo **Xinetd** che si basa sulle regole del controllo degli accessi scritte in TCP Wrappers. xinetd non si incarica soltanto di gestire l'accesso ai servizi di rete da parte di client che soddisfano o meno determinate regole, ma permette anche di controllare il servizio stesso a livello applicazione. Il file xinetd.conf contiene le configurazioni generali riguardanti ogni servizio sotto il suo controllo.

apt-get install xinetd

nano /etc/xinetd.conf

Configurazione:

```
defaults {  
instances=60 //numero massimo di istanze per ogni servizio  
log_type= SYSLOG authpriv  
log_on_success=HOST PID //informazioni nel log delle connessioni  
log_on_failure=HOST  
cps=25 20 } //max connessioni per secondo e wait time
```

Includedir /etc/xinetd.d

La directory xinetd.d contiene la configurazione di ogni servizio con relativo nome su cui è attivo Xinetd. Esempio con il servizio **SSH**. Dobbiamo configurare:

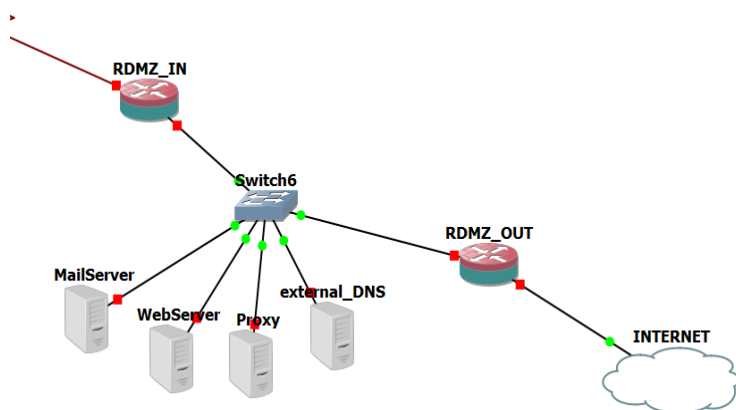
nano /etc/xinetd.d/ssh

Configurazione dentro sshd:

```
service ssh {  
  disable = no  
  flags = REUSE  
  socket_type = stream  
  wait = no  
  user = root  
  server = /usr/sbin/sshd  
  server_args = - i  
  log_on_failure += USERID  
  log_on_success += PID HOST EXIT  
  access_times = 09:45-16.15 }
```

DMZ (EDIFICIO C)

SOTTORETE: 192.168.6.0



Codice	Dispositivo	Indirizzo IP
RDMZ_IN	Router	192.168.6.1
RDMZ_OUT	Router	192.168.6.2
MailServer	Server	192.168.6.3
Proxy	Server	192.168.6.4
WebServer	Server	192.168.6.5
External_DNS	Server	192.168.6.6

Configurazione router (RDMZ_IN)

```
configure
interface FastEthernet0/0
ip address 192.168.6.1 255.255.255.0
no shutdown
exit
interface FastEthernet0/1
ip address 192.168.12.2 255.255.255.0
no shutdown
exit

router rip
version 2
network 192.168.6.0
network 192.168.12.0
no auto-summary
exit
exit
debug ip rip
sh ip route


interface FastEthernet0/0
ip rip authentication mode md5
exit
interface FastEthernet0/1
ip rip authentication mode md5
exit

ip domain-lookup
ip name-server 192.168.3.103
exit

copy running-config startup-config
wr--
```


Configurazione router (RDMZ_OUT)

Per questo router sarà compito del server DHCP dell'ISP scegliere un indirizzo IP per il router.



```
configure
interface FastEthernet0/0
ip address 192.168.6.2 255.255.255.0
no shutdown
exit
interface FastEthernet0/1
ip address dhcp
no shutdown
exit

router rip
version 2
network 192.168.6.0
default-information originate
exit

ip domain-lookup
ip name-server 192.168.3.103
exit

copy running-config startup-config
wr
```

Configurazione Firewall

Nella nostra configurazione sono presenti 4 firewall. RDMZ_IN che funge anche da estremità della DMZ verso l'interno della rete aziendale e RDMZ-OUT, posto all'altra estremità della DMZ, quindi verso l'internet pubblico. Infine RC e RD presenti nei router di frontiera dell'edificio C e D a rappresentare le estremità del tunnel VPN tra i due edifici.

RDMZ-OUT (*interfaccia esterna eth1 interna eth0*)

```
//pulisco le catene
```

```
iptables -F FORWARD
iptables -F INPUT
iptables -F OUTPUT
```

```
//imposto default policy
```

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

```
//accetto tutto dall'interno verso l'esterno per creare connessioni
```

```
iptables -A -s (ip pubblico) FORWARD -I eth0 -j ACCEPT
```

```
//accetto tutto ciò che appartiene a connessioni attive (permesse tramite la regola di prima) => stateful inspection firewall
```

```
iptables -A FORWARD -s (ip pubblico) RELATED, ESTABLISHED -j ACCEPT
```

```
//accettiamo in input le connessioni ssh
```

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
//Accetta pacchetti smtp, pop3, imap, dns (ciò che è destinato ai server nella DMZ)
```

```
iptables -A FORWARD -p tcp --dport 25 -j ACCEPT
iptables -A FORWARD -p tcp --dport 110 -j ACCEPT
iptables -A FORWARD -p tcp --dport 143 -j ACCEPT
iptables -A FORWARD -p tcp --dport 53 -j ACCEPT
iptables -A FORWARD -p tcp --dport 80 -j ACCEPT
```

```
// permetto connessioni ipsec
```

```
iptables -A FORWARD -p esp -j ACCEPT
iptables -A FORWARD -p udp --dport 4500 -j ACCEPT
iptables -A FORWARD -p udp --dport 500 -j ACCEPT
iptables -A FORWARD -p udp --dport 1701 -j ACCEPT
```

```
iptables -t nat -P POSTROUTING DROP
```

```
iptables -t nat -A POSTROUTING !-d 192.168.0.0/16 -o eth1 -j MASQUERADE
```

RDMZ-IN (*interfaccia esterna eth1 interna eth0*)

//pulisco le catene

```
iptables -F FORWARD
```

```
iptables -F INPUT
```

```
iptables -F OUTPUT
```

//imposto default policy

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
```

```
iptables -P FORWARD DROP
```

//accetto tutto dall'interno verso l'esterno per creare connessioni

```
iptables -A FORWARD -i eth0 -j ACCEPT
```

//accetto tutto ciò che appartiene a connessioni attive (permesse tramite la regola di prima) => stateful inspection firewall

```
iptables -A FORWARD -i eth1 -m state --state RELATED, ESTABLISHED -j ACCEPT
```

// accetto i messaggi rip (porta 520)

```
iptables -A INPUT -o eth0 -p udp --dport 520 -j ACCEPT
```

```
iptables -A OUTPUT -o eth0 -p udp --dport 520 -j ACCEPT
```

// permetto connessioni ipsec

```
iptables -A INPUT -p udp --dport 4500 -j ACCEPT
```

```
iptables -A INPUT -p udp --dport 500 -j ACCEPT
```

```
iptables -A INPUT -p udp --dport 1701 -j ACCEPT
```

```
iptables -A OUTPUT -p udp --dport 4500 -j ACCEPT
```

```
iptables -A OUTPUT -p udp --dport 500 -j ACCEPT
```

```
iptables -A OUTPUT -p udp --dport 1701 -j ACCEPT
```

```
iptables -P PREROUTING ACCEPT
```

```
iptables -P INPUT ACCEPT
```

```
iptables -P POSTROUTING ACCEPT
```

```
iptables -P OUTPUT ACCEPT
```

RC (interfaccia esterna eth1 interna eth1)

//pulisco le catene

```
iptables -F FORWARD
iptables -F INPUT
iptables -F OUTPUT
```

//imposto default policy

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

//accetto tutto dall'interno verso l'esterno per creare connessioni

```
iptables -A FORWARD -i eth0 -j ACCEPT
```

//accetto tutto ciò che appartiene a connessioni attive (permesse tramite la regola di prima) => stateful inspection firewall

```
iptables -A FORWARD -i eth1 -m state --state RELATED, ESTABLISHED -j ACCEPT
```

// accetto i messaggi rip (porta 520)

```
iptables -A INPUT -o eth0 -p udp --dport 520 -j ACCEPT
iptables -A OUTPUT -o eth0 -p udp --dport 520 -j ACCEPT
```

// permetto connessioni ipsec

```
iptables -A INPUT -p udp --dport 4500 -j ACCEPT
iptables -A INPUT -p udp --dport 500 -j ACCEPT
iptables -A INPUT -p udp --dport 1701 -j ACCEPT
iptables -A OUTPUT -p udp --dport 4500 -j ACCEPT
iptables -A OUTPUT -p udp --dport 500 -j ACCEPT
iptables -A OUTPUT -p udp --dport 1701 -j ACCEPT
```

```
iptables -P PREROUTING ACCEPT
iptables -P INPUT ACCEPT
iptables -P POSTROUTING ACCEPT
iptables -P OUTPUT ACCEPT
```

RD (*interfaccia esterna eth1 interna eth1*)

//pulisco le catene

```
iptables -F FORWARD
iptables -F INPUT
iptables -F OUTPUT
```

//imposto default policy

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

//accetto tutto dall'interno verso l'esterno per creare connessioni

```
iptables -A FORWARD -i eth0 -j ACCEPT
```

//accetto tutto ciò che appartiene a connessioni attive (permesse tramite la regola di prima) => stateful inspection firewall

```
iptables -A FORWARD -i eth1 -m state --state RELATED, ESTABLISHED -j ACCEPT
```

// accetto i messaggi rip (porta 520)

```
iptables -A INPUT -o eth0 -p udp --dport 520 -j ACCEPT
iptables -A OUTPUT -o eth0 -p udp --dport 520 -j ACCEPT
```

// permetto connessioni ipsec

```
iptables -A INPUT -p udp --dport 4500 -j ACCEPT
iptables -A INPUT -p udp --dport 500 -j ACCEPT
iptables -A INPUT -p udp --dport 1701 -j ACCEPT
iptables -A OUTPUT -p udp --dport 4500 -j ACCEPT
iptables -A OUTPUT -p udp --dport 500 -j ACCEPT
iptables -A OUTPUT -p udp --dport 1701 -j ACCEPT
```

```
iptables -P PREROUTING ACCEPT
iptables -P INPUT ACCEPT
iptables -P POSTROUTING ACCEPT
iptables -P OUTPUT ACCEPT
```

Configurazione external_DNS

File di configurazione **resolv.conf** del resolver, il quale comprende la lista dei **name server** da interrogare

```
domain greenPeace.it
search greenPeace.it
nameserver 192.168.6.6
nameserver 192.168.3.103
nameserver 1.1.1.1
nameserver 1.0.0.1
```

File **named.conf.local** e i relativi **zone files** per la rete locale, usati dal **daemon named** per rispondere alle richieste:

```
// DEFINIZIONE MASTER
// external_DNS è master per greenPeace.it
zone "greenPeace.it" {
    type master;
    file "/etc/bind/azienda.greenPeace.it.db";
};
```

```
// reverse mapping per greenPeace.it
zone "168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/168.192.in-addr.arpa.db";
};
```

```
// external_DNS è master per dmz.greenPeace.it
zone "dmz.greenPeace.it" {
    type master;
    file "/etc/bind/dmz.greenPeace.it.db";
};
```

```
// reverse mapping per dmz.greenPeace.it
zone "6.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/6.168.192.in-addr.arpa.db";
};
```

// DEFINIZIONE SLAVE

```
// external_DNS è slave per edificioc.greenPeace.it
zone "edificioc.greenPeace.it" {
    type slave; file "/etc/bind/greenPeace.it.bk";
    masters { 192.168.3.103; };
};
```

```
// reverse mapping per edificioc.greenPeace.it
zone "3.168.192.in-addr.arpa" {
    type slave;
    file "/etc/bind/3.168.192.in-addr.arpa.bk";
    masters { 192.168.3.103; };
};
```

// DEFINIZIONE SLAVE

```
// external_DNS è slave per edificioa.greenPeace.it
zone "edificioa.greenPeace.it" {
    type slave; file "/etc/bind/greenPeace.it.bk";
    masters { 192.168.3.103; };
};
```

```
// reverse mapping per edificioa.greenPeace.it
zone "1.168.192.in-addr.arpa" {
    type slave;
    file "/etc/bind/1.168.192.in-addr.arpa.bk";
    masters { 192.168.3.103; };
};
```

// DEFINIZIONE SLAVE

```
// external_DNS è slave per edificioc.greenPeace.it
zone "edificiob.greenPeace.it" {
    type slave; file "/etc/bind/greenPeace.it.bk";
    masters { 192.168.3.103; };
};
```

```
// reverse mapping per edificioc.greenPeace.it
zone "2.168.192.in-addr.arpa" {
    type slave;
    file "/etc/bind/2.168.192.in-addr.arpa.bk";
    masters { 192.168.3.103; };
};
```

// DEFINIZIONE SLAVE

```
// external_DNS è slave per edificioc.greenPeace.it
zone "edificiod.greenPeace.it" {
```

```
type slave; file "/etc/bind/greenPeace.it.bk";
masters { 192.168.3.103; };
};
```

```
// reverse mapping per edificioc.greenPeace.it
zone "4.168.192.in-addr.arpa" {
type slave;
file "/etc/bind/4.168.192.in-addr.arpa.bk";
masters { 192.168.3.103; };
};
```

// DEFINIZIONE SLAVE

```
// external_DNS è slave per edificioc.greenPeace.it
zone "edificioe.greenPeace.it" {
type slave; file "/etc/bind/greenPeace.it.bk";
masters { 192.168.3.103; };
};
```

```
// reverse mapping per edificioc.greenPeace.it
zone "5.168.192.in-addr.arpa" {
type slave;
file "/etc/bind/5.168.192.in-addr.arpa.bk";
masters { 192.168.3.103; };
};
```

File named.conf.option

```
acl "trusted-nameservers" {
localhost;
192.168.6.6;
192.168.3.103;
};
acl "trusted-networks" {
localhost;
192.168.3.0/24;
192.168.6.0/24;
192.168.1.0/24;
192.168.2.0/24;
192.168.4.0/24;
192.168.5.0/24;
};
options {
directory "/var/cache/bind";
dnssec-validation auto;
auth-nxdomain no;
version "Not disclosed";
```



```
notify yes;  
allow-transfer { trusted-nameservers; };  
allow-query { "any" };  
forwarders { 1.1.1.1; };  
recursion yes;  
Allow-recursion {"any"}  
};
```

File dmz.greenPeace.i.db

```
$TTL 86400
$ORIGIN dmz.greenPeace.it.
@ IN SOA dns.dmz.greenPeace.it.root.dmz.greenPeace.it. (
    2018112903 ; serial
    43200 ; refresh
    3600 ; retry after 1 hour
    3600000 ; expire after 1000 hours
    2592000 ; default ttl )
```

; Definizione dei nameserver e dei server mail

```
IN NS dns.dmz.greenPeace.it.
IN NS dns.azienda.greenPeace.it.
IN NS dns.cloudflare.com.
IN MX 10 mail.dmz.greenPeace.it.
```

; Host in dmz.greenPeace.it

```
Rdmz_out IN A 192.168.6.2
External_dns IN A 192.168.6.6
www IN A 192.168.6.5
mail IN A 192.168.6.3
proxy IN A 192.168.6.4
```

File greenPeace.it.db

```
$TTL 86400
$ORIGIN greenPeace.it.
@ IN SOA dns.greenPeace.it. root.greenPeace.it. (
    2018112902 ; serial
    43200 ; refresh
    3600 ; retry after 1 hour
    3600000 ; expire after 1000 hours
    2592000 ; default ttl )
```

; Definizione dei nameserver e dei server mail

```
IN NS dns.greenPeace.it.
IN NS dns.cloudflare.com.
IN MX 10 mail.greenPeace.it.
```

; Sottodomini in greenPeace.it

```
;dmz IN A 198.168.6.0
;azienda IN A 198.168.3.0
```

; Host in greenPeace.it

mail	IN	A	198.168.6.3	
dns	IN	A	198.168.6.6	
@.	IN	A	192.168.6.5	
www	IN	CNAME	@	//nome del dominio
app	IN	A	198.168.3.104	
proxy	IN	A	198.168.6.4	

File 168.192.in.addr.arpa.db

\$TTL 86400

\$ORIGIN 168.192.in-addr.arpa.

@	IN	SOA	dns.greenPeace.it.	root.dmz.greenPeace.it. (
			2018112902 ; serial	
			43200 ; refresh	
			3600 ; retry after 1 hour	
			3600000 ; expire after 1000 hours	
			2592000 ; default ttl	
)	

; Definizione dei nameserver e dei server mail

	IN	NS	dns.greenPeace.it.
	IN	NS	dns.cloudflare.com.
	IN	MX	10 mail.greenPeace.it.

; Sottodomini in greenPeace.it

0.6	IN	PTR	dmz.greenPeace.it.
0.3	IN	PTR	azienda.greenPeace.it.

; Host in greenPeace.it

3.6	IN	PTR	mail.greenPeace.it.
6.6	IN	PTR	dns.greenPeace.it.
5.6	IN	PTR	www.greenPeace.it.
104.3	IN	PTR	app.greenPeace.it.
4.6	IN	PTR	proxy.greenPeace.it.

File 6.168.192.in-addr.arpa.db

\$TTL 86400

\$ORIGIN 6.168.192.in-addr.arpa.

```
@      IN      SOA      dns.dmz.greenPeace.it.      root.dmz.greenPeace.it. (
                                2018112903 ; serial
                                43200 ;      refresh
                                3600 ;      retry after 1 hour
                                3600000 ;    expire after 1000 hours
                                2592000 ;    default ttl )
```

; Definizione dei nameserver e dei server mail

```
IN      NS      dns.dmz.greenPeace.it.
IN      NS      dns.azienda.greenPeace.it.
IN      NS      dns.cloudflare.com.
IN      MX      10      mail.dmz.greenPeace.it.
```

; Host in dmz.greenPeace.it

```
2      IN      PTR      rdmzout.dmz. greenPeace.it.
3      IN      PTR      mail.dmz. greenPeace.it.
6      IN      PTR      dns.dmz. greenPeace.it.
5      IN      PTR      www.dmz. greenPeace.it.
4      IN      PTR      proxy.dmz. greenPeace.it.
```

Server Web

Per la configurazione del server Web andremo prima di tutto ad installare Apache2
apt install apache2

Creiamo un file PHP dentro la cartella di default predefinita del server Web: **/var/www/html/**

```
nano /var/www/html/phpinfo.php
<?php
    Phpinfo();
?>
```

Installiamo ora il php
apt install php

Per configurare Apache accediamo al file **apache.conf**
nano /etc/apache2/apache2.conf

All'interno di questa cartella possiamo gestire varie opzioni , che di default sono tutte attive. **AllowOverride** disabilita la possibilità di utilizzare file denominati **.htaccess** all'interno di un sito web. Può tornare utile abilitare questa opzione per permettere ad un webmaster, che non ha accesso alla macchina, di apportare determinate modifiche come redirect. **L'htaccess** è utilizzato anche dai CMS, pertanto, se se ne vogliono utilizzare, è necessario attivare questa opzione cambiandola a "**AllowOverride All**".

```
Options Indexes
FollowSymLinks
AllowOverride None
Require all granted
```

Per attivare **l'HTTPS** bisogna modificare alcune configurazioni di Apache2, come ad esempio mettere in ascolto alla **porta 443** il servizio del protocollo in questione e inoltrare tutte le richieste HTTP dalla porta 80 verso la porta dell'HTTPS. Bisogna inoltre installare un certificato da allegare ai domini del server (es **www.greenPeace.it** e **greenPeace.it**) e infine farlo utilizzare da Apache2. Queste configurazioni possono essere fatte utilizzando **CERTBOT** che permette di installare gratuitamente un certificato di **LetsEncrypt** e configura automaticamente il server web utilizzato.

Per procedere alla configurazione accediamo alla cartella "sitesavailable"

```
nano /etc/apache2/sites-available/000-default.conf
ServerName www.greenPeace.it
ServerName greenPeace.it*greenPeace.it
```

Entriamo nella cartella

```
/etc/host cd /etc/hosts
```

e aggiungiamo quello che sarà l'indirizzo assegnato all'azienda dall'ISP e il dominio (esempio: **220.8.140.23 www.greenPeace.it greenPeace.it**)

Tale cartella utilizzerà un virtual host(metodo per mascherare l'indirizzo ip dell'utente)

```
mkdir /var/www/greenPeace.it
```

```
echo "<?php echo 'Sito di greenPeace.it' ? >"
```

```
/ var/www/greenPeace.it/index.php
```

I VHOST permettono di hostare più siti sulla stessa macchina e sullo stesso indirizzo IP. In base a quale dominio si richiede Apache2 restituisce la directory del relativo VHOST. In questa maniera si possono creare siti differenti senza dover configurare più macchine e risparmiare risorse. Domini possono essere come **www.greenPeace.it** e **supporto.greenPeace.it**, oppure direttamente qualcosa di diverso, come **greenPeace.com**.

Per rendere utilizzabile il Server Web modifichiamo il file e le cartelle create come root.

*(**chmod 755** permette di settare i permessi , l'user/owner può leggere scrivere ed eseguire , mentre gli altri utenti possono solo leggere ed eseguire. **chmod 644** è come il 755 ma gli utenti esterni possono solo leggere. Useremo chmod 755 per le directory e chmod 644 per i file)*

```
cd /var/www/greenPeace.it/
```

```
find . -type d -exec chmod 755 {} \;
```

```
find . -type f -exec chmod 644 {} \;
```

```
chown -R www-data:www-data (chown = change owner)
```

```
/var/www/ greenPeace.it
```

Cartelle per salvare i log: (**chmod750** – user può leggere,scrivere,eseguire mentre gli utenti esterni non possono fare nessuna delle tre;i gruppi possono leggere ed eseguire)

```
Mkdir/var/log/apache2/ greenPeace.it
```

```
Chmod 750 /var/log/apache2/ greenPeace.it
```

```
chown root:adm/var/log/apache2/ greenPeace.it
```

File all'interno della cartella **"sites-available"** :

```
nano /etc/apache2/sites-available/ greenPeace.it.conf
<Virtual Host*: 80>
    ServerAdmin webmaster@ greenPeace.it
    ServerName greenPeace.it
    ServerAlias greenPeace.it
    DocumentRoot /var/www/ greenPeace.it/
    ErrorLog
    ${APACHE_LOG_DIR}/ greenPeace.it/error.log
    CustomLog
    ${APACHE_LOG_DIR}/ greenPeace.it/access.log
    Combined
</Virtual Host>
```

Abilitiamo ora il sito

```
a2ensite greenPeace.it
```

Alternativamente, per abilitare un sito basta creare un link del suo file di configurazione in **"sites-enabled"** *(i siti elencati in sites-enabled sono serviti da apache, mentre i siti presenti in sites-available sono quelli presenti nel nostro server e non sono visibili perchè non sono stati ancora resi disponibili)*

```
ln -s
/etc/apache2/sites-available/ greenPeace.it.conf
/etc/apache2/sites-enabled/ greenPeace.it.conf
```

**Infine per rendere effettive le modifiche riavviamo
service apache2 restart**

Adesso, visitando il dominio sarà soltanto il nuovo sito ad essere raggiunto. Visitando l'IP invece verrà caricato il contenuto di /var/www/html/

Server Posta Elettronica

Installiamo sendmail

```
apt install sendmail
```

Vediamo le principali impostazioni e **configuriamo sendmail**

```
nano /etc/mail/access
```

Il file access permette di configurare da chi accettare email, da chi accettare l'inoltro e a chi non inviare email. La nostra LAN è 192.168.0.0/24, pertanto andremo a togliere il commento ("#") a:

```
Connect:192.168
GreetPause:192.168
ClientRate:192.168
ClientConn:192.168
```

Infondo al file access scriviamo ora le regole di ricezione, inoltro e invio.

```
FREE.STEALTH.MAILER@ 550 Non accettiamo spam
miodominio.it RELAY
192.168 RELAY
```

Creiamo ora degli **aliases**, ovvero alias di utenti esistenti. Può tornare utile utilizzarli poiché si può scrivere un'email generica, come "webmaster@miodominio.it", e assegnare a webmaster l'attuale utente che svolge quella mansione. **(I nomi sono solo indicativi e verrebbero specificati con il cliente in fase di configurazione reale.)**

```
nano /etc/mail/aliases postmaster: cristian
```

```
admin: cristian, fabrizio
dmz: admindmz
azienda: adminazienda
admindmz: cristian
adminrete1: cristian, fabrizio
```

Configuriamo ora il file per indicare a sendmail per quali domini tale server deve ricevere la posta. Ad esempio, nel nostro caso

```
nano /etc/mail/local-host-names
localhost
mail.greePeace.it
greenPeace.it
dmz.greenPeace.it
azienda.greenPeace.it
```


Andiamo ora a modificare il file di configurazione. Il file di configurazione è sendmail.cf, ma questo viene generato attraverso il comando "make" e prende le impostazioni dal file sendmail.mc. Pertanto andremo a modificare quest'ultimo:

```
nano /etc/mail/sendmail.mc
```

Vogliamo far processare a sendmail tutte le email di greenPeace.it, quindi rimuoveremo da tale riga "Addr=127.0.0.1", otterremo dunque

```
DAEMON_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp')dnl
```

Di default solo gli host indicati come RELAY nel db sono abilitati all'inoltro. Permettiamo dunque ad ogni host di fare l'inoltro. Dopo l'ultimo include del file scriviamo dunque

```
FEATURE(`relay_entire_domain')dnl
```

Creiamo ora qualche utente per le mail

```
useradd --create-home -s /sbin/nologin cristian; passwd cristian
useradd --create-home -s /sbin/nologin fabrizio; passwd fabrizio
useradd --create-home -s /sbin/nologin utente1; passwd utente1
```

Mappiamo ora gli indirizzi email con le mailbox reali. Tali mailbox possono essere locali, remote, alias definiti nel file aliases oppure singoli file. Andiamo a fare le mailbox per ciascun alias e utente creato

```
nano /etc/mail/virtusertable
root@greenPeace.it root
postmaster@greenPeace.it postmaster
admin@greenPeace.it admin
dmz@greenPeace.it dmz
azienda@greenPeace.it rete1
cristian@greenPeace.it cristian
fabrizio@greenPeace.it fabrizio
utente1@greenPeace.it utente1
```

Entriamo in /etc/mail e facciamo "make" per aggiornare i database cd /etc/mail make.

Riavviare ora sendmail

```
service sendmail restart
```

Server Proxy

Uno dei metodi più efficaci per limitare attacchi maliziosi è quello di utilizzare un **Proxy server**. Un server Proxy opera come un **application Firewall**, quindi:

- intercetta le richieste di servizi internet fatte dalle applicazioni e le reindirizza al giusto servizio in accordo con le regole impostate
- si interpone come gateway di comunicazione tra gli utenti (applicazioni) interne e quelle esterne alla rete, agendo come un filtro
- è totalmente trasparente all'utente che richiede il servizio
- è in grado di operare un controllo sull'operato dell'utente, ad esempio è possibile negare il metodo GET o PUT del servizio FTP per un determinato sito
- il proxy è un prodotto software (ad es. **SQUID**, che è quello utilizzato da noi nello specifico) che lavora in congiunzione con il packet filtering

Quello che faremo sarà quindi utilizzare un server proxy anche per accedere dalla rete interna ai servizi “critici” nella DMZ, oltre che per gli altri servizi della rete esterna. Devolveremo anche al proxy il compito di assolvere alle richieste di servizi dall'esterno.

Installazione Squid

```
pat-get install squid
```

Per configurare squid andiamo a modificare il file **squid.conf**

```
nano /etc/squid/squid.conf
```

Di default nessuno può accedere a squid quindi andremo ad aggiungere le reti che possano accedere a Squid

```
acl localnet src 192.168.1.0/24
acl localnet src 192.168.2.0/24
acl localnet src 192.168.3.0/24
acl localnet src 192.168.4.0/24
acl localnet src 192.168.5.0/24
acl localnet src 192.168.6.0/24
```

Riavvio del proxy server

```
systemctl restart squid
```

Ora creiamo una lista di siti da **bloccare** (sono puramente esplicativi, verranno poi accordati in fase di configurazione con il cliente)

```
nano /etc/squid/blocked_sites
```

Scrivere i siti dentro **blocked_sites**

facebook.com
twitter.com
instagram.com

Riandiamo su **squid.conf** e aggiungiamo le seguenti righe

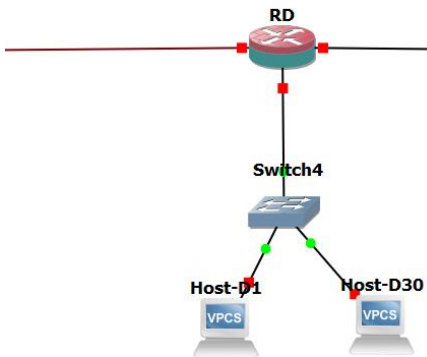
```
nano /etc/squid/squid.conf  
acl blocked_sites dstdomain "/etc/squid/blocked_sites"  
http_access deny blocked_sites
```

Riavviamo Squid

```
systemctl restart squid
```

per usare squid configuriamo il proxy nel browser del client.

EDIFICIO D



HOST: 30

SOTTORETE: 192.168.4.0

COLLEGAMENTI: *Edifici C, E*

Codice	Dispositivo	Indirizzo IP
Host-D1	Host	192.168.4.3
.....
Host-D30	Host	192.168.4.32
RD	Router	192.168.4.1

Configurazione host (Host-D1)

```
ip 192.168.4.2/24 192.168.4.1
ip dns 192.168.3.103
save
```

Configurazione router (RD)

```
configure
interface FastEthernet0/0
ip address 192.168.4.1 255.255.255.0
no shutdown
exit
interface FastEthernet0/1
ip address 192.168.11.1 255.255.255.0
no shutdown
exit
interface Serial1/1
ip address 192.168.10.2 255.255.255.0
no shutdown
exit

router rip
version 2
network 192.168.4.0
network 192.268.10.0
network 192.168.11.0
no auto-summary
exit

interface FastEthernet0/0
ip rip authentication mode md5
exit
interface FastEthernet0/1
ip rip authentication mode md5
exit
interface Serial1/1
ip rip authentication mode md5
exit
exit
debug ip rip
sh ip route

configure
ip domain lookup
ip name-server 192.168.3.103
exit

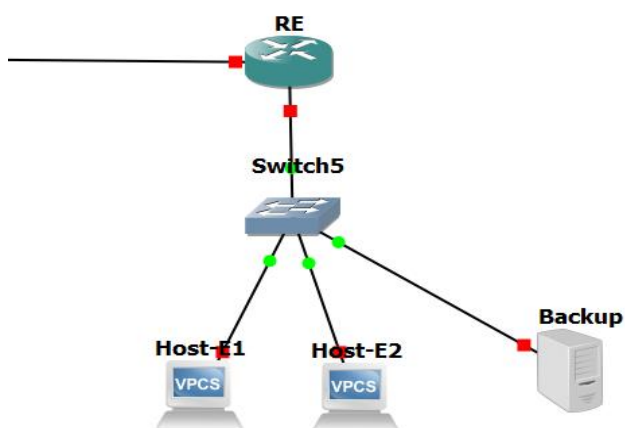
copy running-config startup-config
wr
```

EDIFICIO E

HOST: 20

SOTTORETE: 192.168.5.0

COLLEGAMENTI: *Edificio D*



Codice	Dispositivo	Indirizzo IP
Host-E1	Host	192.168.5.2
.....
Host-E20	Host	192.168.5.22
RE	Router	192.168.5.1
Server Backup	Server	192.168.5.23

Configurazione host (Host-E1)

```
ip 192.168.5.2/24 192.168.5.1
ip dns 192.168.3.103
save
```

Configurazione router (RE)

```
configure
interface FastEthernet0/0
ip address 192.168.5.1 255.255.255.0
no shutdown
exit
interface FastEthernet0/1
ip address 192.168.11.2 255.255.255.0
no shutdown
exit

router rip
version 2
network 192.168.5.0
network 192.168.11.0
no auto-summary
exit

interface FastEthernet0/0
ip rip authentication mode md5
exit
interface FastEthernet0/1
ip rip authentication mode md5
exit
exit
debug ip rip
sh ip route

configure
ip domain lookup
ip name-server 192.168.3.103
exit

copy running-config startup-config
wr
```

Per completezza abbiamo riportato qui sotto le schermate per mostrare che la simulazione della configurazione è funzionante e ad esempio mediante il protocollo di routing vengono aggiornate costantemente le routing table con le giuste metriche.

```
RD RE
RE(config)#interface fa
RE(config)#interface FastEthernet0/1
RE(config-if)#ip address 192.168.11.2 255.255.255.0
RE(config-if)#no shutdown
RE(config-if)#exit
*Mar 1 00:52:11.947: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:52:12.947: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
RE(config-if)#exit
RE(config)#router rip
RE(config-router)#version 2
RE(config-router)#network 192.168.5.0
RE(config-router)#network 192.168.11.0
RE(config-router)#no auto-summary
RE(config-router)#exit
RE(config)#interface FastEthernet0/0
RE(config-if)#ip rip authentication mode md5
RE(config-if)#exit
RE(config)#interface FastEthernet0/1
RE(config-if)#ip rip authentication mode md5
RE(config-if)#exit
RE(config)#exit
RE#
*Mar 1 00:53:37.275: %SYS-5-CONFIG_I: Configured from console by console
RE#debug ip rip
RIP protocol debugging is on
RE#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R 192.168.12.0/24 [120/2] via 192.168.11.1, 00:00:03, FastEthernet0/1
R 192.168.8.0/24 [120/2] via 192.168.11.1, 00:00:03, FastEthernet0/1
R 192.168.9.0/24 [120/2] via 192.168.11.1, 00:00:03, FastEthernet0/1
R 192.168.10.0/24 [120/1] via 192.168.11.1, 00:00:03, FastEthernet0/1
C 192.168.11.0/24 is directly connected, FastEthernet0/1
R 192.168.4.0/24 [120/1] via 192.168.11.1, 00:00:03, FastEthernet0/1
C 192.168.5.0/24 is directly connected, FastEthernet0/0
R 192.168.7.0/24 [120/3] via 192.168.11.1, 00:00:05, FastEthernet0/1
R 192.168.1.0/24 [120/3] via 192.168.11.1, 00:00:05, FastEthernet0/1
R 192.168.2.0/24 [120/3] via 192.168.11.1, 00:00:05, FastEthernet0/1
R 192.168.3.0/24 [120/2] via 192.168.11.1, 00:00:05, FastEthernet0/1
RE#
```

```
RD RE
*Mar 1 00:54:03.083: 192.168.7.0/24 via 0.0.0.0, metric 4, tag 0
*Mar 1 00:54:03.087: 192.168.8.0/24 via 0.0.0.0, metric 3, tag 0
*Mar 1 00:54:03.087: 192.168.9.0/24 via 0.0.0.0, metric 3, tag 0
*Mar 1 00:54:03.087: 192.168.10.0/24 via 0.0.0.0, metric 2, tag 0
RE#
*Mar 1 00:54:03.087: 192.168.11.0/24 via 0.0.0.0, metric 1, tag 0
*Mar 1 00:54:03.091: 192.168.12.0/24 via 0.0.0.0, metric 3, tag 0
RE#
*Mar 1 00:54:11.887: RIP: received v2 update from 192.168.11.1 on FastEthernet0/1
*Mar 1 00:54:11.887: 192.168.1.0/24 via 0.0.0.0 in 3 hops
*Mar 1 00:54:11.891: 192.168.2.0/24 via 0.0.0.0 in 3 hops
*Mar 1 00:54:11.891: 192.168.3.0/24 via 0.0.0.0 in 2 hops
*Mar 1 00:54:11.891: 192.168.4.0/24 via 0.0.0.0 in 1 hops
*Mar 1 00:54:11.891: 192.168.7.0/24 via 0.0.0.0 in 3 hops
*Mar 1 00:54:11.895: 192.168.8.0/24 via 0.0.0.0 in 2 hops
*Mar 1 00:54:11.895: 192.168.9.0/24 via 0.0.0.0 in 2 hops
*Mar 1 00:54:11.895: 192.168.10.0/24 via 0.0.0.0 in 1 hops
*Mar 1 00:54:11.895: 192.168.12.0/24 via 0.0.0.0 in 2 hops
RE#
*Mar 1 00:54:27.811: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (192.168.11.2)
*Mar 1 00:54:27.811: RIP: build update entries
*Mar 1 00:54:27.811: 192.168.5.0/24 via 0.0.0.0, metric 1, tag 0
RE#
*Mar 1 00:54:29.915: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.5.1)
*Mar 1 00:54:29.915: RIP: build update entries
*Mar 1 00:54:29.915: 192.168.1.0/24 via 0.0.0.0, metric 4, tag 0
*Mar 1 00:54:29.915: 192.168.2.0/24 via 0.0.0.0, metric 4, tag 0
*Mar 1 00:54:29.919: 192.168.3.0/24 via 0.0.0.0, metric 3, tag 0
*Mar 1 00:54:29.919: 192.168.4.0/24 via 0.0.0.0, metric 2, tag 0
*Mar 1 00:54:29.919: 192.168.7.0/24 via 0.0.0.0, metric 4, tag 0
*Mar 1 00:54:29.923: 192.168.8.0/24 via 0.0.0.0, metric 3, tag 0
*Mar 1 00:54:29.923: 192.168.9.0/24 via 0.0.0.0, metric 3, tag 0
*Mar 1 00:54:29.923: 192.168.10.0/24 via 0.0.0.0, metric 2, tag 0
RE#
*Mar 1 00:54:29.923: 192.168.11.0/24 via 0.0.0.0, metric 1, tag 0
*Mar 1 00:54:29.927: 192.168.12.0/24 via 0.0.0.0, metric 3, tag 0
RE#
*Mar 1 00:54:38.867: RIP: received v2 update from 192.168.11.1 on FastEthernet0/1
*Mar 1 00:54:38.867: 192.168.1.0/24 via 0.0.0.0 in 3 hops
*Mar 1 00:54:38.871: 192.168.2.0/24 via 0.0.0.0 in 3 hops
*Mar 1 00:54:38.871: 192.168.3.0/24 via 0.0.0.0 in 2 hops
*Mar 1 00:54:38.871: 192.168.4.0/24 via 0.0.0.0 in 1 hops
*Mar 1 00:54:38.871: 192.168.7.0/24 via 0.0.0.0 in 3 hops
*Mar 1 00:54:38.875: 192.168.8.0/24 via 0.0.0.0 in 2 hops
*Mar 1 00:54:38.875: 192.168.9.0/24 via 0.0.0.0 in 2 hops
*Mar 1 00:54:38.875: 192.168.10.0/24 via 0.0.0.0 in 1 hops
*Mar 1 00:54:38.875: 192.168.12.0/24 via 0.0.0.0 in 2 hops
RE#
```


Mostriamo anche che è possibile pingare gli host tra di loro, nell'esempio il ping è effettuato tra i 2 più lontani Host-A1 e l'Host-E1.

```

Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

Checking for duplicate address...
PC1 : 192.168.5.2 255.255.255.0 gateway 192.168.5.1

Host-E1> ping 192.168.1.2
192.168.1.2 icmp_seq=1 timeout
192.168.1.2 icmp_seq=2 timeout
84 bytes from 192.168.1.2 icmp_seq=3 ttl=60 time=122.166 ms
84 bytes from 192.168.1.2 icmp_seq=4 ttl=60 time=122.334 ms
84 bytes from 192.168.1.2 icmp_seq=5 ttl=60 time=121.773 ms

Host-E1>
```

Mostriamo anche il ping tra l'Host-A1 e l'Host-C1

```

Welcome to Virtual PC Simulator
, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:2
0
Copyright (c) 2007-2014, Paul M
eng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distribu
ted under the terms of the "BSD
" licence.
Source code and license can be
found at vpcs.sf.net.
For more information, please vi
sit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

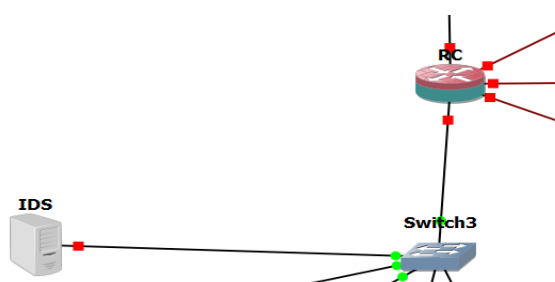
DORA IP 192.168.1.2/24 GW 192.1
68.1.1

Host-A1> ping 192.168.3.2
192.168.3.2 icmp_seq=1 timeout
192.168.3.2 icmp_seq=2 timeout
84 bytes from 192.168.3.2 icmp_seq=3 ttl=62 time=39.199 ms
84 bytes from 192.168.3.2 icmp_seq=4 ttl=62 time=33.989 ms
84 bytes from 192.168.3.2 icmp_seq=5 ttl=62 time=40.998 ms

Host-A1>
```

6. Monitoraggio della rete

Al fine di garantire una maggiore sicurezza, affidabilità e integrità alla rete dell'azienda, è stata adottata la soluzione di implementare un sistema di monitoraggio continuo. Questo verrà fatto utilizzando dei **Sistemi di Intercettazione delle Intrusioni (Intrusion Detection System)**. Vi sarà quindi un **network-based IDS** che permetterà di scandire tutto il traffico della rete mediante il quale saranno segnalati eventuali pacchetti sospetti. Avremo anche un **host-based IDS** che andrà a verificare periodicamente i log dei servizi di rete e di verificare l'integrità dei dati e dei filesystem. Con questa implementazione avremo così un monitoraggio su tutti gli ambiti e aspetti della rete aziendale e una maggiore sicurezza, essendo quindi in grado di prevenire e individuare eventuali attacchi o intrusioni malevole.



Per convenzione l'IDS sarà messo nell'**EDIFICIO C** in quanto esso è il più centrale ed è quello più grande e che avrà il maggior numero di host, oltre ad avere il server per le applicazioni aziendali.

7. Preventivo di spesa

Componente	Prezzo Unità'	Quantità'	Prezzo Totale
Cavo Fibra Ottica	3.00€/metro	300m	900€
Router	300€	7	2100€
Vpn	15€/mese	//	//
Dominio	20€/anno	//	//
Switich 50 porte	450€	6	2700€
Altre Spese	//	//	3000€
Access Point	150€	1	150€
Progettazione	//	//	5000€
Installazione	//	//	4000€

TOTALE: 17850€