# Sicurezza Informatica - Prima esercitazione sui firewall

## Indice



### Avvio e verifica della rete virtuale

Il materiale per l'esercitazione è disponibile on-line sul moodle del corso e rimarrà disponibile anche in seguito. Il file è in formato *mar* ed è utilizzabile su marionnet.

L'esercitazione consiste nell'impostare regole di natting e firewalling in una rete composta da nove macchine virtuali. Lo schema della rete è rappresentato nell'immagine `network_diagram.png`, riportato anche nell'ultima pagina.

Visualizzare il diagramma di rete

```
$ ristretto network_diagram.png
```

il comando ristretto avvia un semplice programma di visualizzazione di immagini incluso di default nell'installazione di xfce.

La rete utilizzata per l'esercitazione è suddivisa in due sottoreti, DMZ e LAN. La rete LAN contiene tre macchine locali, *local1, local2* e *dhcp*, configurate con indirizzi IP privati. La macchina *dhcp* esegue un server DHCP e un server DNS a servizio della LAN e della DMZ.

La rete DMZ contiene due server, un server web (*www.fake.com*) e un server FTP (*ftp.fake.com*), configurati con due indirizzi IP pubblici. I server della DMZ utilizzano *dhcp* come server per ricevere le configurazioni di rete in modo dinamico, utilizzando il protocollo DHCP. Per fare ciò la macchina *firewall* esegue un relay DHCP, cioè un servizio in grado di intercettare richieste DHCP per inoltrarle al server *dhcp*.

Le sottoreti della DMZ sono direttamente connesse alla macchina chiamata *firewall*, che svolge funzioni di routing, filtro di pacchetti e nat. Firewall è il border router della rete che dovremo proteggere nell'esercitazione. Le macchine *www.google.com* e *remote1* rappresentano un server web e una macchina remota connesse ad Internet e dotate di IP pubblici. In questa semplice rete emulata, Internet è rappresentata da un singolo router, implementato dalla macchina *extrouter*.

Dopo aver avviato la rete in marionnet, dovrebbero comparire 9 nuovi terminali, che eseguono le rispettive macchine presenti sul diagramma di rete che si trova nell'ultima pagina di questo documento.

Tutte le macchine virtuali eseguono il boot in parallelo, litigandosi la (scarsa) capacità computazionale disponibile. Il processo di avvio impiegherà qualche minuto. La rete è pronta per essere utilizzata quando tutte le macchine virtuali hanno completato il boot e presentano il prompt di root.

**Verificare che tutte le macchine della LAN e della DMZ abbiano acquisito un indirizzo IP per l'interfaccia eth0.**

## Verificare la connettività della rete

```
remote1:~# ping www.google.com
```

per verificare la connettività della rete useremo il comando ping. Ping consente di inviare verso una macchina remota un pacchetto ICMP di tipo *echo-request*. Se non ci sono filtri di pacchetto attivi o configurazioni particolari, una macchina che riceve un *echo-request* risponde inviando un *echo-reply* al mittente. Quando la macchina mittente riceve l'*echo-reply*, il comando ping stampa una riga sullo standard output. L'output prodotto da *ping* implica la capacità delle due macchine di scambiarsi pacchetti in entrambe le direzioni.

Dal nodo *remote1* provate a pingare *www.google.com*, *www.fake.com ftp.fake.com*, *local1* e *local2*.

Dal nodo *local1* provate a pingare *www.google.com*, *www.fake.com ftp.fake.com*, *remote1* e *local2*.

I nodi con indirizzi privati non sono direttamente raggiungibili dall'esterno della rete protetta.

## Verificare i servizi della rete

```
remote1:~# w3m www.google.com
```

```
remote1:~# w3m www.fake.com
```

*w3m* è un web browser con interfaccia testuale. Il comando *w3m nome_sito* visualizza la home page del sito web *nome_sito*. Per uscire da *w3m*, premere *q*, seguito da *y*. Se vengono visualizzate le homepage di *www.google.com* e di *www.fake.com*, significa che le macchine sono raggiungibili e che i loro web server (implementati con Apache2) sono avviati e configurati correttamente.

```
remote1:~# ftp fpt.fake.com
```

*ftp* è un client FTP con interfaccia testuale. Il comando *ftp nome_server* apre una sessione FTP verso il server *nome_server*. Alla richiesta di credenziali di accesso, accedere come l'utente alice (*username:alice password:alice*).

```
ftp> dir
```

usate il comando *dir* per ottenere la lista dei file dell'utente alice.

```
ftp> get file2
```

usate il comando *get nome_file* per scaricare dal server FTP il file nome_file.

Potete chiudere il comando *ftp* premendo la combinazione di tasti *Ctrl-d*

```
remote1:~# ssh www.fake.com
```

tutte le macchine della rete hanno server e client SSH. Potete aprire una connessione remota usando il comando *ssh nome_host*. Per terminare la connessione, utilizzare la combinazione di tasti *Ctrl-d*.

## Politiche di filtraggio e di nat

La rete da proteggere appartiene all'azienda FAKE. Tale azienda ha due server pubblicamente accessibili da Internet: il server web *www.fake.com* e il server FTP *ftp.fake com*. FAKE ha inoltre una rete locale con due

macchine dotate di indirizzi IP privati (FAKE non ha nessuna intenzione di pagare indirizzi IP pubblici per le macchine dei suoi dipendenti), *local1*, *local2* e un server dhcp e dns a servizio delle macchine della rete locale. Le due sottoreti di FAKE sono direttamente connesse al border router, utilizzato anche come firewall (filtro di pacchetto). Il *firewall* applica anche le regole di nat necessarie per permettere alle macchine locali con IP privato di connettersi ad altre macchine in Internet, ad esempio a *www.google.com*. La macchina *remote1* è un host appartenente al proprietario di FAKE, che vuole potersi connettere a local1 mediante ssh da casa sua. La macchina local2 è utilizzata dall'amministratore di rete, ed è l'unica macchina che può accedere in ssh al firewall.

**Politiche di filtraggio sul firewall**

- Utilizzare una policy di negazione implicita per tutti i pacchetti in transito
- Bloccare tutti i pacchetti diretti verso il firewall.
- Bloccare tutti i pacchetti in uscita dal firewall.
- Consentire flussi di comunicazione UDP provenienti dalla DMZ e diretti al server DHCP relay in esecuzione sul *firewall*.
- Consentire flussi di comunicazione UDP provenienti dal server DHCP relay in esecuzione su *firewall* verso la DMZ.
- Consentire flussi di comunicazione UDP provenienti dal *firewall* e diretti al server DHCP in esecuzione su *dhcp*.
- Consentire flussi di comunicazione UDP provenienti dal *dhcp* e diretti verso il server DHCP relay in esecuzione su *firewall*.
- Consentire flussi di comunicazione UDP provenienti dalla DMZ verso la rete LAN relative al traffico DNS (porta 53).
- Consentire le risposte delle richieste DNS generate dalla DMZ.
- Consentire le connessioni ssh generate dalla macchina *local2* verso il *firewall*.
- Consentire connessioni TCP sulla porta 80 da Internet verso il server web www.fake.com
- Consentire le risposte di *www.fake.com* a connessioni originate da Internet
- Consentire connessioni TCP sulla porta 80 dalla LAN verso il server web www.fake.com
- Consentire le risposte di *www.fake.com* a connessioni originate dalla LAN
- Consentire connessioni TCP sulla porta 21 da Internet verso il server FTP *ftp.fake.com*
- Consentire le risposte di *ftp.fake.com* a connessioni originate da Internet
- Consentire connessioni TCP sulla porta 21 dalla LAN verso il server FTP *ftp.fake.com*
- Consentire le risposte di *ftp.fake.com* a connessioni originate dalla LAN
- Consentire a tutte le macchine nella LAN di inviare pacchetti verso macchine in Internet e di ricevere i pacchetti in risposta e correlati ai pacchetti inviati

**Politiche di Network Address Translation**

- Consentire alle macchine local1 e local2 di accedere a macchine in Internet condividendo l'IP pubblico associato all'interfaccia eth2 di firewall
- Consentire alla macchina remote1 di aprire una connessione TCP sulla porta 22 della macchina local1 utilizzando l'IP pubblico associato all'interfaccia eth2 di firewall come indirizzo di destinazione

## Implementazione delle politiche di filtraggio e di nat

### Prima di iniziare

Segue un piccolo elenco di comandi fondamentali.

**Questi comandi NON fanno parte della soluzione dell'esercitazione, servono solo per ricordare alcuni comandi utili e la loro sintassi**.

```
firewall:~# man iptables
```

l'unico comando di cui avete veramente bisogno. Visualizza la pagina di manuale del comando iptables. Per uscire dalla pagina di manuale, premere *q*

```
firewall:~# iptables -t filter -L -v -n
```

visualizza le regole attualmente incluse nelle catene appartenenti alla tabella filter e le loro policy di default. L'opzione *-n* evita che iptables provi ad eseguire il reverse lookup degli indirizzi IP

```
firewall:~# iptables -t filter -P FORWARD DROP
```

imposta la policy di negazione implicita (*DROP*) sulla catena *FORWARD* della tabella *filter*

```
firewall:~# iptables -t filter -A FORWARD -p tcp --dport 22 -i eth0 -j DROP
```

esempio di comando utilizzato per aggiungere una regola di packet filtering statico. Questo comando aggiunge una regola alla catena *FORWARD* della tabella filter. La regola inserita blocca (*-j DROP*) tutti i pacchetti *TCP* aventi *22* come numero di porta di destinazione e che hanno eth0 come interfaccia di ingresso

```
firewall:~# iptables -t filter -D FORWARD 2
```

esempio di eliminazione selettiva di una singola regola. Questo comando elimina la seconda regola della catena *FORWARD* nella tabella filter

```
firewall:~# iptables -t filter -F FORWARD
```

eliminazione di tutte le regole appartenenti alla catena *FORWARD*. Questo comando non modifica la policy di default della catena.

```
firewall:~# iptables -t filter -F
```

eliminazione di tutte le regole appartenenti a tutte le catene della tabella filter. Le policy di default delle catene non vengono modificate

Dopo l'inserimento di una nuova regola di filtraggio o di nat, verificate che la regola sia stata effettivamente aggiunta nella tabella e nella catena corrette utilizzando l'opzione `-L`. Verificatene inoltre gli effetti sulla raggiungibilità delle macchine e sulla fruibilità dei loro servizi.

In seguito è proposta una possibile implementazione delle policy di filtraggio e di nat. La soluzione proposta non è l'unica implementazione possibile.

## Implementazione delle politiche di filtraggio sul firewall

Effettuare login come utente *root* nella macchina *firewall*

**Policy**:

Utilizzare una policy di negazione implicita per tutti i pacchetti in transito

**Implementazione**:

```
firewall:~# iptables -t filter -P FORWARD DROP
```

Verificare l'impossibilità di comunicare tra le macchine in Internet e le macchine di FAKE Verificare che è ancora possibile aprire connessioni ssh verso il firewall

**Policy**:

Bloccare tutti i pacchetti diretti verso il firewall.

Bloccare tutti i pacchetti in uscita dal firewall.

**Implementazione**:

```
firewall:~# iptables -t filter -P INPUT DROP

firewall:~# iptables -t filter -P OUTPUT DROP
```

Verificare l'impossibilità di aprire connessioni SSH verso il firewall da local1, local2 e remote1

**Policy**:

Consentire flussi di comunicazione UDP provenienti dalla DMZ e diretti al server DHCP relay in esecuzione sul *firewall.*

Consentire flussi di comunicazione UDP provenienti dal server DHCP relay in esecuzione su *firewall* verso la DMZ.

Consentire flussi di comunicazione UDP provenienti dal *firewall* e diretti al server DHCP in esecuzione su *dhcp.*

Consentire flussi di comunicazione UDP provenienti dal *dhcp* e diretti verso il server DHCP relay in esecuzione su *firewall.*

**Implementazione**:

Verificare che effettuando un riavvio dei servizi di rete (*ifdown -a e ifup -a*) delle macchine della DMZ quest'ultime non riescono ad acquisire un indirizzo IP dal server *dhcp.*

```
firewall:~# iptables -t filter -A INPUT -i eth0 -p udp --sport 68 --dport 67 -j ACCEPT

firewall:~# iptables -t filter -A OUTPUT -o eth0 -p udp --sport 67 --dport 68 -j ACCEPT

firewall:~# iptables -t filter -A OUTPUT -o eth1 -p udp -s 155.185.1.6 --sport 67 -d
↪  192.168.1.253 --dport 67 -m state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -t filter -A INPUT -i eth1 -p udp -s 192.168.1.253 --sport 67 -d
↪  155.185.1.6 --dport 67 -m state --state ESTABLISHED -j ACCEPT
```

Verificare che ora è possibile effettuare richieste DHCP attraverso il DHCP relay in esecuzione su *firewall.*

**Policy**:

Consentire flussi di comunicazione UDP provenienti dalla DMZ verso la rete LAN relative al traffico DNS (porta 53).

Consentire le risposte delle richieste DNS generate dalla DMZ.

**Implementazione**:

```
firewall:~# iptables -A FORWARD -p udp --dport 53 -i eth0 -o eth1 -s 155.185.1.0/29 -d
↪  192.168.1.253 -m state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p udp --sport 53 -o eth0 -i eth1 -d 155.185.1.0/29 -s
↪  192.168.1.253 -m state --state ESTABLISHED -j ACCEPT
```

Verificare se è possibile effettuare richieste di risoluzione DNS dagli host presenti nella DMZ al server DNS in esecuzione su *dhcp* (es: *nslookup www.google.com*).

**Policy**:

Consentire le connessioni ssh generate dalla macchina *local2* verso il *firewall*.

**Implementazione**:

```
firewall:~# iptables -t filter -A INPUT -p tcp --dport ssh -i eth1 -s 192.168.1.2 -m
↪  state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -t filter -A OUTPUT -p tcp --sport ssh -o eth1 -d 192.168.1.2 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Verificare se è possibile aprire connessioni SSH verso il firewall da local2. Notare l'utilizzo del modulo *state*

**Policy**:

Consentire connessioni TCP sulla porta 80 da Internet verso il server web *www.fake.com*

Consentire le risposte di *www.fake.com* a connessioni originate da Internet

**Implementazione**:

```
firewall:~# iptables -A FORWARD -p tcp --dport www -i eth2 -o eth0 -d 155.185.1.1 -m
↪  state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p tcp --sport www -i eth0 -o eth2 -s 155.185.1.1 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Verificare se è possibile accedere al sito *www.fake.com* dalla macchina remote1. Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori NEW ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED?

**Policy**:

Consentire connessioni TCP sulla porta 80 dalla LAN verso il server web www.fake.com

Consentire le risposte di *www.fake.com* a connessioni originate dalla LAN

**Implementazione**:

```
firewall:~# iptables -A FORWARD -p tcp --dport www -i eth1 -o eth0 -d 155.185.1.1 -m
↪  state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p tcp --sport www -i eth0 -o eth1 -s 155.185.1.1 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Verificare se è possibile accedere al sito *www.fake.com* dalla macchina local1. Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori NEW ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED?

**Policy**:

Consentire connessioni TCP sulla porta 21 da Internet verso il server FTP *ftp.fake.com*

Consentire le risposte di *ftp.fake.com* a connessioni originate da Internet

**Implementazione**:

```
firewall:~# iptables -A FORWARD -p tcp --dport ftp -i eth2 -o eth0 -d 155.185.1.2 -m
↪  state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p tcp --sport ftp -o eth2 -i eth0 -s 155.185.1.2 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori NEW ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED?

Provate ad aprire una connessione FTP verso *ftp.fake.com* dalla macchina remote1. Riuscite ad effettuare login? E ad ottenere la lista dei file? Perché?

```
firewall:~# iptables -A FORWARD -p tcp --sport ftp-data -o eth2 -i eth0 -s 155.185.1.2 -m
↪  state --state RELATED,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p tcp --dport ftp-data -i eth2 -o eth0 -d 155.185.1.2 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori RELATED ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED?

Provate ad aprire una connessione FTP verso *ftp.fake.com* dalla macchina remote1. Riuscite ad effettuare login? E ad ottenere la lista dei file? Perché?

**Policy**:

Consentire connessioni TCP sulla porta 21 dalla LAN verso il server FTP *ftp.fake.com*

Consentire le risposte di *ftp.fake.com* a connessioni originate dalla LAN

**Implementazione**:

```
firewall:~# iptables -A FORWARD -p tcp --dport ftp -i eth1 -o eth0 -d 155.185.1.2 -m
↪  state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p tcp --sport ftp -o eth1 -i eth0 -s 155.185.1.2 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori NEW ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED?

Provate ad aprire una connessione FTP verso *ftp.fake.com* dalla macchina local1. Riuscite ad effettuare login? E ad ottenere la lista dei file? Perché?

```
firewall:~# iptables -A FORWARD -p tcp --sport ftp-data -i eth0 -o eth1 -s 155.185.1.2 -m
↪  state --state RELATED,ESTABLISHED -j ACCEPT

firewall:~# iptables -A FORWARD -p tcp --dport ftp-data -o eth0 -i eth1 -d 155.185.1.2 -m
↪  state --state ESTABLISHED -j ACCEPT
```

Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori NEW ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED?

Provate ad aprire una connessione FTP verso *ftp.fake.com* dalla macchina local1. Riuscite ad effettuare login? E ad ottenere la lista dei file? Perché?

**Policy**:

Consentire a tutte le macchine nella LAN di inviare pacchetti verso macchine in Internet e di ricevere i pacchetti in risposta e correlati ai pacchetti inviati

**Implementazione**:

```
firewall:~# iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT

firewall:~# iptables -A FORWARD -o eth1 -i eth2 -m state --state ESTABLISHED,RELATED -j
↪  ACCEPT
```

Notare le differenze tra i due comandi. Solo il secondo introduce delle regole dinamiche, mentre il primo è una regola statica. Perchè non uso il qualificatore NEW? Dalla macchina local1 provate a pingare remote1 e ad accedere al sito web *www.google.com*. Funziona? Perché?

## Implementazione delle politiche di Network Address Translation

**Policy**:

Consentire alle macchine local1 e local2 di accedere a macchine in Internet condividendo l'IP pubblico associato all'interfaccia eth2 di firewall

**Implementazione**:

```
firewall:~# iptables -t nat -A POSTROUTING -o eth2 -s 192.168.1.0/24 -j MASQUERADE
```

Notare l'utilizzo della tabella nat. Perchè aggiungo una regola alla catena POSTROUTING, invece di PREROUTING? Dalla macchina local1 provate a pingare remote1 e ad accedere al sito web *www.google.com*. Funziona? Perché?

Supporre di non aver specificato l'opzione *-s 192.168.1.0/24*, cosa sarebbe cambiato?

**Policy**:

Consentire alla macchina remote1 di aprire una connessione TCP sulla porta 22 della macchina local1 utilizzando l'IP pubblico associato all'interfaccia eth2 di firewall come indirizzo di destinazione

**Implementazione**:

```
firewall:~# iptables -t filter -A FORWARD -i eth2 -o eth1 -s 11.22.33.211 -p tcp --dport
↪  ssh -m state --state NEW,ESTABLISHED -j ACCEPT

firewall:~# iptables -t filter -A FORWARD -o eth2 -i eth1 -d 11.22.33.211 -p tcp --sport
↪  ssh -m state --state ESTABLISHED -j ACCEPT
```

Notare le differenze tra i due comandi. Perchè il primo comando usa i qualificatori NEW ed ESTABLISHED, mentre il secondo comando usa solo ESTABLISHED? Provate ad aprire una connessione SSH da remote1 a local1. Funziona? Perchè? Provate ad aprire una connessione SSH da remote1 a 155.185.1.9 (indirizzo IP associato all'interfaccia di rete eth2 del firewall). Funziona? Perchè?

```
firewall:~# iptables -t nat -A PREROUTING -p tcp --dport ssh -i eth2 -s 11.22.33.211 -d
↪  155.185.1.9 -j DNAT --to-destination 192.168.1.1
```

Notare l'utilizzo della tabella nat. Perchè aggiungo una regola alla catena PREROUTING, invece di POSTROUTING? Provate ad aprire una connessione SSH da remote1 a 155.185.1.9 (indirizzo IP associato all'interfaccia di rete eth2 del firewall). Verso quale macchina viene stabilita la connessione? Perchè?

## Per concludere

Verificate la lista di regole di filtraggio con il comando

```
firewall:~# iptables -t filter -L -v -n
```

Analizzate la lista e ricostruite i comandi di iptables necessari per creare le regole di filtraggio

Verificate la lista di regole di nat con il comando

```
firewall:~# iptables -t nat -L -v -n
```

Analizzate la lista e ricostruite i comandi di iptables necessari per creare le regole di nat

## Se vi rimane tempo...

Il proprietario di FAKE, Bob, non vuole che i suoi famigliari (Alice, Carl, Dave e Eve) possano utilizzare remote1 per accedere ad Internet. Implementare le opportune regole di filtraggio per fare in modo che solo i processi appartenenti all'utente bob possano inviare pacchetti verso Internet.

Suggerimenti:

- impostare DROP come policy di default per la catena OUTPUT
- leggere la pagina di manuale di iptables, facendo particolare attenzione alla descrizione del modulo *owner*
- implementare le regole di filtraggio usando l'account di root su remote1. Verificare l'efficacia delle regole provando ad accedere a *www.google.com* con l'account di alice.

  Perchè questa regola di filtro viene impostata sulla macchina remote1, e non sul firewall?

## Errori comuni

State attenti a non commettere i seguenti errori nella scrittura dei comandi di iptables. Sono errori veniali e abbastanza comuni, ma spesso difficili da trovare.

I nomi delle interfacce di rete devono essere scritti correttamente. Se scrivete una regola con una interfaccia di rete inesistente, iptables non genera nessun messaggio di errore. Quindi:

- scrivete *eth0*, e non *etho* o *ethO*
- scrivete *eth1*, e non *ethl*
- scrivete *eth*, e non *eht*

Quando scrivete le regole di filtro, ricordatevi sempre di aggiungere l'obiettivo con l'opzione -j. Le regole con solo l'espressione di confronto e senza l'obiettivo sono ancora sintatticamente valide (iptables non si lamenta) ma ovviamente non hanno l'effetto desiderato.

Quando scrivete regole che contengono i qualificatori relativi alle interfacce di rete di ingresso e di uscita dei pacchetti, verificate sempre il percorso dei pacchetti utilizzando il diagramma di rete.

Quando scrivete una regola per autorizzare il flusso dei pacchetti in una direzione copiando e incollando la regola scritta precedentemente per autorizzare i pacchetti che viaggiano in direzione opposta, ricordatevi di sostituire porta sorgente e porta destinazione, interfaccia sorgente e interfaccia destinazione, ip sorgente e ip destinazione.

## Diagramma di rete

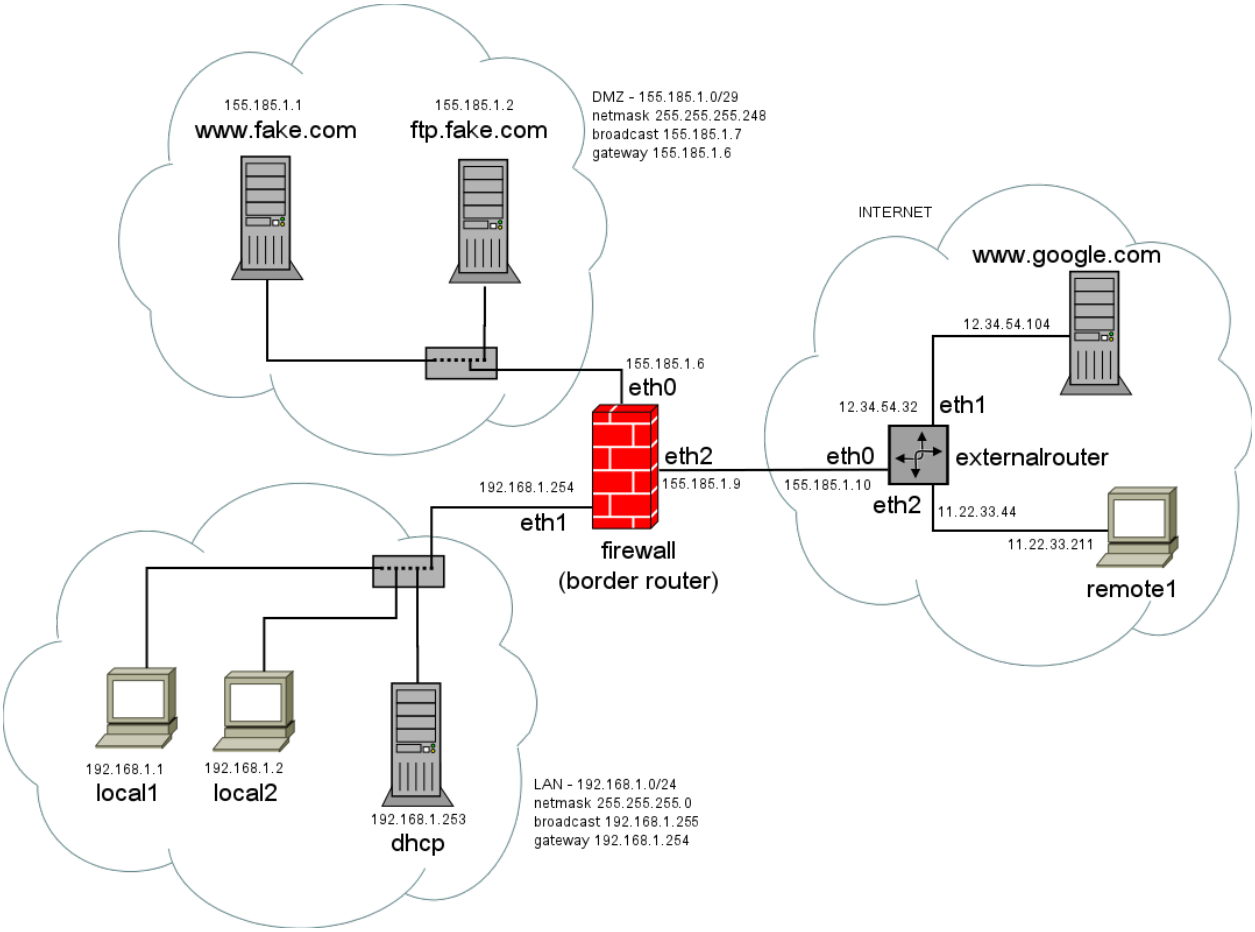


Figure 1: Diagramma di rete