Nona Esercitazione

Sommario

- Network File System (NFS)
  - Architettura
  - Lato server
    - ⇨ invocazione server
    - ⇨ file `/etc/exports`
    - ⇨ vincoli sulle condivisioni
  - Lato client
    - ⇨ opzioni di montaggio

Sommario

- Introduzione ai firewall
  - Definizione e scopo
  - Classificazione
- Firewall a filtraggio dei pacchetti
  - Informazioni associate alle regole
  - Interpretazione delle regole
- Il firewall `ipfw`
  - Configurazione
  - Impostazione delle regole

Network FileSystem (NFS)

Network Filesystem (NFS)

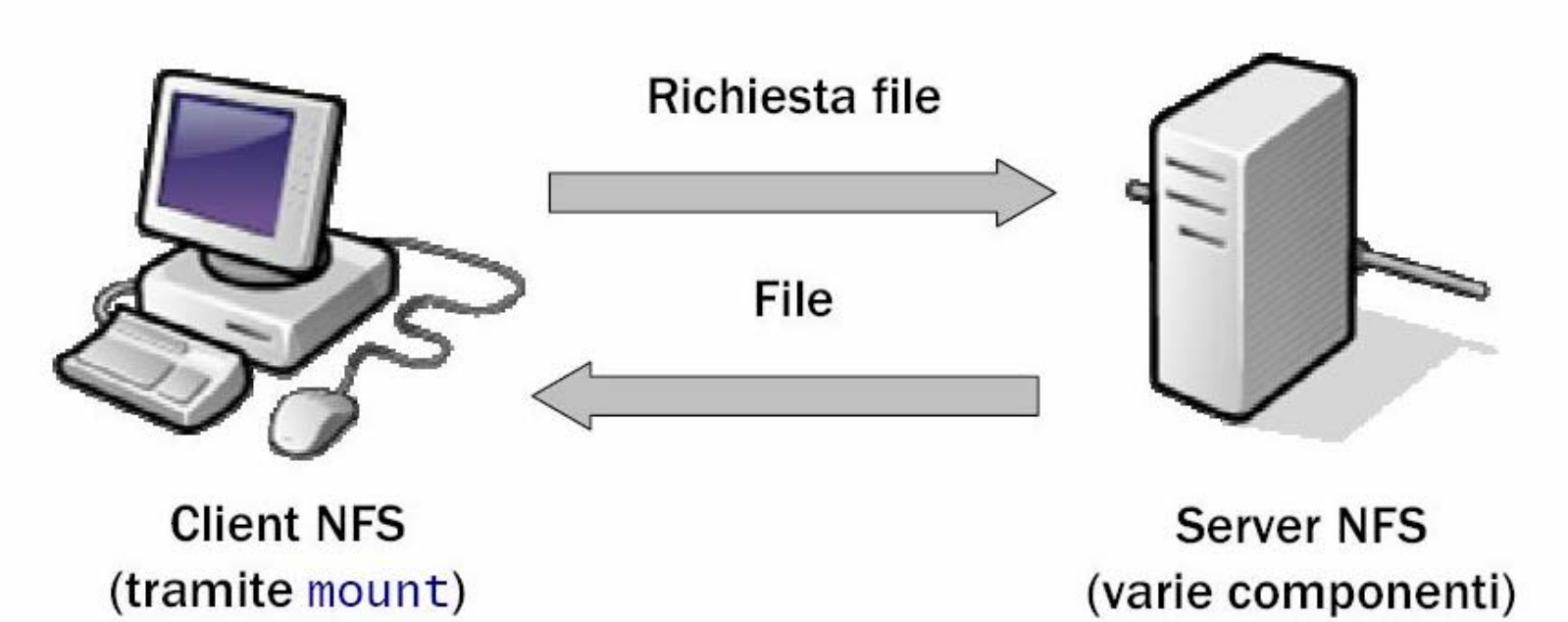
- Il servizio di NFS permette la condivisione dei file system in rete.
- Usando NFS, utenti e programmi possono accedere a file su sistemi remoti quasi come se fossero file locali.
- Funzionamento
  - Server
    - ⇨ Devono essere configurate le directory da condividere
  - Client

NFS: funzionalità

- **Possibilità di concentrare i dati a cui spesso accedono tutti gli utenti su un host centrale e di renderli disponibili ai client al momento del boot**
- **Possibilità di conservare su un unico host tutti i dati ed i programmi che occupano molto spazio su disco.**

NFS: architettura di riferimento

- **Architettura di riferimento**
  - client/server
  - Interazione tramite RPC
  - Protocollo stateless



Lato Server

1. **Controllare se il servizio NFS è supportato dal kernel leggendo `/proc/filesystems` (`nodev nfs`).**
2. **Lanciare il demone `portmapper`**
3. **Lanciare il demone `rpc.mountd`**
   - **Gestisce il montaggio dei file system di rete dal lato del server.**
   - **Mantiene il file `/etc/rmtab` che elenca i montaggi attivi.**
4. **Lanciare il demone `rpc.nfsd`**
   - **Gestisce le richieste dei client per i servizi NFS**
   - **Il suo funzionamento dipende dal file `/etc/exports`**
5. **Configurare il file `/etc/exports`**

Lato server: `/etc/rc.conf`

- **Il file `/etc/rc.conf` deve avere le seguenti opzioni:**

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

Lato server: demoni NFS

- `mountd`
  - **gestisce le richieste di mount dei client alle cartelle condivise**
  - **usa il file di configurazione `/etc/exports`**
  - **Parte in automatico quando il NFS è attivo**
- `nfsd`
  - **realizza il servizio di trasferimento file vero e proprio**
- `rpcbind`
  - **Associa il servizio RPC al corrispondente server**
  - **Scopre quale porta usa il NFS**

Lato server: `/etc/exports` (1 di 5)

- Contiene l'indicazione delle porzioni di file system locale da concedere in condivisione alla rete NFS.
- Se il file manca o è vuoto, non viene concesso l'utilizzo di alcuna parte del file system locale all'esterno.
- Ogni record del file è composto da:
  - l'indicazione di una directory a partire dalla quale si concede la condivisione
  - una serie di nodi o reti cui viene concesso l'utilizzo di questa directory con l'eventuale specificazione di opzioni di accesso.
- Quando si fanno modifiche a questo file, è necessario riavviare il sistema di gestione del servizio NFS.

Lato server: `/etc/exports` (2 di 5)

- **Formato del file**

```
cartella_base [opzioni] [host]
```

- **Esempi**
  - **Esporta `/usr` verso 3 host**

```
/usr                    huey louie dewie
```

  - **Esporta `/home` e le sottodirectory verso 3 host**

```
/home   -alldirs    janice jimmy frank
```

  - **Esporta `/a` verso 2 host che possono accedervi con i privilegi di `root`**

```
/a      -maproot=0  bill albert
```

Lato server: `/etc/exports` (3 di 5)

`cartella_base` **`[opzioni]`** `[host]`

- `-ro`
  - Accesso in sola lettura
- `-rw`
  - Accesso in lettura e scrittura
- `-alldirs`
  - Permette alle sottodirectory di fungere da mount point.
  - **Non** monterà le sottodirectory ma permetterà al client di montare solo le directory di cui ha bisogno
- `-maproot=root`
  - Permette all'utente root sul sistema remoto di scrivere dati sul file system esportato come utente root.
  - Se non è specificato, anche se l'utente ha accesso come root sul file system remoto, non sarà in grado di modificare files

Lato server: `/etc/exports` (4 di 5)

`cartella_base [opzioni]` **`[host]`**

- **Host singolo (nome o indirizzo IP)**
- **Possono essere utilizzati i caratteri jolly `*` e `?` per indicare un gruppo di nomi di elaboratore con una sola notazione, tenendo presente che questi simboli non possono sostituirsi ai punti di un nome di dominio**
- **Con `indirizzo_ip/maschera_di_rete` è possibile indicare simultaneamente tutti gli elaboratori collocati all'interno della rete o della sottorete a cui si fa riferimento.**

```
–network <netaddr> -mask <mask>
```

Lato server: `/etc/exports` (5 di 5)

- Le cartelle possono comparire in più di una riga
  - ogni riga si riferisce ad uno o più client
  - se a client distinti si applicano opzioni diverse devono esserci righe distinte per la cartella
- Si può inserire una lista di cartelle come primo campo
  - nella forma di percorsi assoluti
  - separate da spazi
- Non è possibile esportare allo stesso client punti diversi dello stesso filesystem con opzioni diverse
  - non si possono avere due righe in cui le cartelle si trovano sullo stesso file system

Esempio

 # Invalido

```
/usr/src client
/usr/ports client
```

#Corretto

```
/usr/src /usr/ports client
```

Esempio

- **Esporta `src` e `ports` a `client01` e `client02`, ma solo `client01` ha i privilegi di `root`**

```
/usr/src /usr/ports -maproot=root client01
/usr/src /usr/ports client02
```

- **I client hanno privilegi di `root` e possono montare ovunque in `/exports`.**

```
/exports -alldirs -maproot=root client01 client02
```

- **Ognuno nel mondo può montare `/exports/obj` ma solo in lettura**

```
/exports/obj -ro
```

Esempio

```
# Sample /etc/exports file
/var/dept       -maproot=root:operator master
/var/dept       csifi
/usr/share      -ro –mapall=staff si1 si7
/opt            –ro –network 131.114.29.0 \
                –mask 255.255.255.0
/opt/extra      -alldirs
/opt /opt/sub   –ro pc1 pc2 pc3
```

Lato server: aggiornamento

- **Il demone `mountd` deve essere forzato a rileggere il file `/etc/exports` ogni volta che viene modificato**
- **Questo può essere ottenuto:**
  - **inviando un segnale HUP al processo `mountd`**

```
kill -HUP ‘cat /var/run/mountd.pid’
```

  - **invocando lo script `mountd` con i parametri appropriati:**

```
/etc/rc.d/mountd onereload
```

Comandi per avviare NFS

- **L'esecuzione dei seguenti comandi da utente `root` dovrebbe avviare tutto**
- **Sul server NFS:**

```
rpcbind
nfsd -u -t -n 4
mountd -r
```

- **Sul client NFS**

```
nfsiod -n 4
```

Lato client : `/etc/rc.conf`

- **Il file `/etc/rc.conf` deve avere l'opzione**

`nfs_client_enable="YES"`

Lato client: `mount`

- **Il montaggio di un file system di rete avviene in modo analogo a quello di una normale unità di memorizzazione, con una sintassi leggermente diversa:**

`mount -t nfs` *`host_remoto`*`:`*`dir_remota`* `dir_locale[`*`options`*`]`

- **Esempio**
  - **l'elaboratore** `mio.server.it` **concede l'utilizzo della directory** `/usr/` **e successive**
  - **l'elaboratore** `mio.client.it` **potrebbe connettersi attraverso**

```
mount -t nfs mio.server.it:/usr /usr
```

Lato client: `fstab`

- **Sul lato client si può aggiungere una riga nel file `/etc/fstab` in modo da automatizzarne la connessione.**

```
mio.server.it:/usr /usr nfs defaults
```

Lato client: opzioni `mount` & `fstab`

- `rw, ro`
  - Modalità di accesso lettura/scrittura o sola lettura
- `bg`
  - Continua a provare l'operazione di mount in background se il server non risponde
- `hard`
  - Se il server si blocca, le relative operazioni si bloccano finchè il server non è di nuovo attivo
  - Modalità di funzionamento predefinita.
- `intr`
  - Permette l'interruzione di una chiamata NFS attraverso l'uso di segnali.
  - Può essere utile per interrompere una connessione quando il server non risponde.

showmount

- **Per verificare l'utilizzo effettivo del servizio da parte dei client**

```
showmount [opzioni] [host]
```

- **Sul lato client serve per conoscere le directory esportate da un server NFS**
- **Opzioni**
  - `-a`
    - ⇨ **Elenca i client che utilizzano il proprio servizio e le directory che questi hanno montato.**
  - `-e`
    - ⇨ **Elenca le directory esportate dal server locale o dal server remoto**

Introduzione ai firewall

Problema: sicurezza di una rete

- Internet è un ambiente insicuro
- Necessità di proteggere reti interne collegate ad Internet
  - imporre restrizioni sul tipo di traffico ammesso
  - definire delle policy di sicurezza
  - filtrare il traffico entrante e uscente

Definizione di firewall

- Firewall
  - dispositivo di sicurezza utilizzato in campo informatico per accettare, bloccare o mediare il traffico dati
  - può essere hardware o software
  - è configurato secondo le policy di sicurezza dell'organizzazione in cui si trova

Classificazione dei firewall

- Si possono individuare tre categorie contraddistinte da:
  - modalità di filtraggio delle comunicazioni
    - ⇨ tra un nodo e la rete o tra reti diverse
  - modalità di gestione dei pacchetti
    - ⇨ livello ISO/OSI dello stack di protocolli
  - capacità di tenere traccia dello stato delle connessioni

Classificazione: filtraggio comunicazione

- **Personal firewall**
  - filtra il traffico che transita tra un singolo nodo e una rete
  - applicazione utilizzata in ambito desktop/office
    - ⇨ in esecuzione sullo stesso PC dell'utente
    - ⇨ esempi: Windows Firewall, Zone Alarm, Kerio PF
- **Network firewall**
  - filtra il traffico che transita tra le diverse reti che connette insieme
    - ⇨ dispositivo/computer dedicato
    - ⇨ situato al bordo di una rete (collegamento Internet)
    - ⇨ in genere indicato con il solo termine 'firewall'

Classificazione: gestione pacchetti

- Firewall a filtraggio di pacchetto (packet filtering)
  - operano a livello network/transport
    - ⇨ utilizzano gli header dei pacchetti IP/ICMP/TCP/UDP
- Gateway di applicazione (application gateway)
  - opera a livello applicazione
    - ⇨ proxy server, servizio che permette ai client di effettuare connessioni indirette ad altri servizi
  - tutti i dati sono vincolati a passare attraverso il gateway

Classificazione: stato della connessione

- Firewall stateless
  - ogni pacchetto viene trattato considerandolo singolarmente
  - semplice ma poco potente
- Firewall stateful
  - tiene traccia dello stato delle connessioni che lo attraversano
    - ⇨ flussi TCP, comunicazioni UDP
  - potente ma più complesso e lento
    - ⇨ richiede allocazione di risorse in memoria

Firewall a filtraggio dei pacchetti

Firewall a filtraggio dei pacchetti

- Funzionamento
  - accede alle intestazioni dei pacchetti
  - consulta una sequenza di regole (rule chain)
- Insieme delle regole
  - ogni regola
    - ⇨ è individuata da una serie di informazioni
  - specifica l'azione da intraprendere quando le intestazioni dei pacchetti corrispondono alle informazioni specificate
  - azioni possibili:
    - ⇨ accettare
    - ⇨ scartare senza notifica al mittente
    - ⇨ scartare con notifica al mittente

Informazioni associate alle regole

- **Informazioni fondamentali utilizzate**
  - indirizzo e porta mittente
  - indirizzo e porta destinatario
  - esempio

Indice	IP sorgente	IP destinatario	Azione
1	131.114.0.0/16	131.114.29.9	Blocca

- **Informazioni aggiuntive**
  - numero della regola (ordine)
  - tipo protocollo e stato della connessione (stateful inspection)

Interpretazione delle regole

- Il firewall
  - controlla la corrispondenza delle intestazioni alle regole impostate
  - quando una regola viene soddisfatta allora viene applicata l'azione corrispondente
  - le regole sono processate nell'ordine in cui sono inserite all'interno della catena
  - solo la prima corrispondenza ha effetto

Importanza dell'ordine delle regole (1 di 2)

- **L'amministratore di una rete aziendale con indirizzo `222.22.0.0/16` desidera**
  - **impedire l'accesso da Internet alla rete aziendale**
  - **consentire l'accesso dalla rete `111.11.0.0/16` alla sottorete interna `222.22.22.0/24`**
  - **impedire alla singola sottorete `111.11.11.0/24` di poter accedere alla sottorete interna `222.22.22.0/24`**

Importanza dell'ordine delle regole (2 di 2)

Errato!

Indice	IP sorgente	IP destinatario	Azione
1	111.11.0.0/16	222.22.22.0/24	Consenti
2	111.11.11.0/24	222.22.0.0/16	Blocca
3	0.0.0.0/0	0.0.0.0/0	Blocca

Corretto

Indice	IP sorgente	IP destinatario	Azione
1	111.11.11.0/24	222.22.0.0/16	Blocca
2	111.11.0.0/16	222.22.22.0/24	Consenti
3	0.0.0.0/0	0.0.0.0/0	Blocca

Regole default

- Caso in cui nessuna regola è soddisfatta
  - firewall inclusivo (inclusive)
    - ⇨ blocca tutto il traffico che non soddisfa le regole
    - ⇨ corrisponde ad avere come ultima regola ‘blocca tutto’
    - ⇨ sicuro ma scomodo: senza definire le regole non si può accedere all’esterno
  - firewall esclusivo (exclusive)
    - ⇨ accetta tutto il traffico che non soddisfa le regole
    - ⇨ corrisponde ad avere come ultima regola ‘accetta tutto’
    - ⇨ comodo ma insicuro

Il firewall `ipfw`

ipfw

- `ipfw` versione 2
  - firewall a filtraggio dei pacchetti con stateful inspection
  - firewall presente in FreeBSD
    - ⇨ modulo del kernel
    - ⇨ utility a riga di comando `ipfw`
  - caratteristiche aggiuntive
    - ⇨ accounting
    - ⇨ traffic shaping

Configurazione del modulo `ipfw`

- In fase di compilazione del kernel
  - opzioni di logging
  - comportamento default
    - ⇨ firewall inclusivo o esclusivo
    - ⇨ in assenza di direttive esplicite il firewall è inclusivo (la regola di default è 'blocca tutto')

File di configurazione

- `/etc/rc.conf`
  - **direttiva** `firewall_enable="YES"`
  - **direttiva** `firewall_type=valore`
- `/etc/rc.firewall`
  - **Contiene delle configurazioni prestabilite del firewall che si possono far partire all'avvio del sistema attraverso il** `file rc.conf`

Visualizzazione regole

- Il programma `ipfw` definisce le regole di accesso per il firewall elencandole in una lista in cui ad ogni regola è associato un numero di linea compreso tra 1 e 65534.
- Questo elenco è contenuto all'interno del principale file di configurazione del firewall

```
ipfw show
```

- Ogni riga ha la struttura

```
<numero linea> <azione> <pattern> <flag>
```

Numero di regola

- E' importante fare attenzione al numero di regola scelto perché nel momento in cui il firewall è attivo e arriva un pacchetto dalla rete esterna, `ipfw` inizia a scorrere l'elenco delle regole partendo da quella con il numero di riga più basso e via via salendo.
- Appena incontra una direttiva che interessa quel determinato pacchetto, esegue l'azione impostata e smette di scorrere l'elenco, anche se più avanti ci sono altre regole che lo riguardano; passa poi a controllare altri pacchetti.
- Regola di default non modificabile (firewall esclusivo)

```
65535 deny all from any to any
```

Manipolazione delle regole

- Attraverso il comando `ipfw`
- Operazioni principali
  - aggiunta/modifica di una regola
  - visualizzazione delle regole
  - cancellazione di una regola/dell'intera catena
- Insieme delle regole
  - valido finché la macchina rimane attiva
  - per sopravvivere al riavvio deve essere salvato in un file (in genere uno script)

Aggiunta/modifica di regole

- **Sintassi per l'aggiunta di regole**

```
ipfw [-N] add [index] action [log] protocol
  pattern options
```

  - `-N`
    - ⇨ **per risolvere gli indirizzi numerici nell'output**
  - `index`
    - ⇨ **indice della regola specificata**
  - `log`
    - ⇨ **stampa sulla console le regole soddisfatte**
  - `action`
    - ⇨ **comportamento da adottare in caso di validità della regola**
  - `protocollo`
    - ⇨ **pacchetti su cui agire**
  - `opzioni`

ipfw: index

```
ipfw [-N] add [index] action [log]
  protocol pattern options
```

- indica la posizione da assegnare alla regola specificata all'interno della catena
- sono disponibili 216 possibili posizioni nella catena
  - la regola 65535 è la policy di default
- se omesso la regola viene collocata 100 posizioni sotto l'ultima regola inserita (esclusa la regola default)

ipfw: action

```
ipfw [-N] add [index] action [log] protocol pattern
   options
```

- allow (accept, pass, permit)
  - lascia passare il pacchetto
  - termina la ricerca
- deny (drop)
  - scarta il pacchetto
  - termina la ricerca
- unreach
  - scarta il pacchetto
  - invia al mittente un pacchetto ICMP host o port unreachable
  - termina la ricerca
- reset
  - scarta il pacchetto
  - invia al mittente un messaggio di reset della connessione
  - applicato solo a pacchetti TCP
  - termina la ricerca

ipfw: protocol

```
ipfw [-N] add [index] action [log] protocol
  pattern options
```

- `all`
  - **tutti i pacchetti**
  - **altre opzioni IP nel campo** `options`
- `icmp`
  - **singoli tipi ICMP nel campo** `options`
- `udp`
- `tcp`
  - **opzioni relative allo stato nel campo** `options`
- **I servizi e le rispettive porte well-known possono essere ricavate dal file**

`/etc/services`

ipfw: pattern (1 di 2)

- **Insieme di coppie host-porta**

```
ipfw [-N] add [index] action [log] protocol
   pattern options
```

- **ha la seguente forma**

from <mittente> **to** <destinatario> [via interface]

- <mittente> **e** <destinatario> **sono espressi da:**
  - **un indirizzo o una rete**
  - **una o un insieme di porte**
- interface
  - **descrive l'interfaccia da considerare**

ipfw:pattern (2 di 2)

- Formato dell'indirizzo
  - indirizzo singolo
    - ⇨ es. `131.114.29.9`
  - valori speciali
    - ⇨ `any` (`0.0.0.0`): qualunque host
    - ⇨ `me`: mio host
  - rete con maschera (numero di bit)
    - ⇨ address/mask-bits, es. `192.216.222.1/24`
  - rete con maschera numerica
    - ⇨ address:mask-pattern, es. `192.216.222.1:255.255.255.0`
- Formato della porta
  - porta singola o range di porte
    - ⇨ es. `112, 113` oppure `1-1024`

ipfw: options

```
ipfw [-N] add [index] action [log] protocol
  pattern options
```

- **direzione del pacchetto**
  - `in` **entrante**
  - `out` **uscente**
- **stato della connessione TCP e flag**
  - `setup` (**inizializzazione**)
  - `established` (**già attiva**)
  - `tcpflags flags` (`fin, syn, rst, psh, urg, ack`)
- **tipo ICMP (numero)**
  - `icmptypes types`
  - `0 : echo reply`
  - `8 : echo request`
- **altre opzioni IP**

Visualizzazione delle regole

- **Sintassi per la visualizzazione delle regole**

`ipfw [-a] [-c] [-d] [-t] [-N] list`

- `-a`
  - mostra il contatore associato alla regola specificata
- `-c`
  - utilizza la forma compatta
- `-t`
  - mostra il timestamp relativo all'ultimo match della regola specificata
- `-N`
  - risolve il nome degli host/servizi

Cancellazione delle regole

- **Cancellazione di una regola**

```
ipfw [-q] delete index
```

  - `-q`
    - ⇨ **disabilita l'output dell'operazione**

- **Cancellazione dell'intera catena**

```
ipfw [-f] [-q] flush
```

  - `-f`
    - ⇨ **forza la cancellazione**
  - **rimuove tutte le regole tranne la regola default**

Esempi (1 di 2)

- **Bloccare il traffico** `telnet` **proveniente dal sito** `evil.crackers.ru` **verso l'host** `trusted.host.org`

```
ipfw add deny tcp from evil.crackers.ru to
  trusted.host.org 23
ipfw add deny tcp from evil.crackers.ru to
  trusted.host.org telnet
```

- **Bloccare l'intero traffico proveniente dalla rete** `169.16.0.0/16` **verso la macchina locale**

```
ipfw add deny all from 169.16.0.0/16 to me
```

Esempi (2 di 2)

- **Regole con opzioni stateful**
  - schema generale

```
ipfw add allow tcp from any to any established
ipfw add allow tcp from trusted.net to my.net
ports setup
…
ipfw add deny tcp from any to any
```

- **prima regola**
  - soddisfatta per tutti i pacchetti TCP su connessione già stabilita
- **seconda regola**
  - soddisfatta per connessioni TCP iniziate da `trusted.net` verso l'host `my.net` alle porte specificate
- **ultima regola**
  - blocca il resto

Esercizi

- **Mandare in esecuzione il server Apache mettendolo in ascolto sulla porta `8080` e visualizzare una pagina `html`**
- **Rispetto alla macchina locale**
  1. **bloccare tutto il traffico `TCP` in ingresso ad esclusione di quello diretto verso il `webserver` (supponendo che si trovi sulla porta `8080`)**
  2. **consentire tutto il traffico `TCP` diretto in ingresso ad esclusione di quello diretto al `webserver`, bloccando la fase di `setup` della connessione `TCP`**
  3. **Ripetere il punto 2 bloccando le connessioni `established` (notare il diverso messaggio del browser)**
  4. **bloccare il traffico `ICMP` in ingresso garantendo il funzionamento del comando `ping` sull'interfaccia locale**

Soluzioni (1 di 3)

- `cd /usr/local/etc/apache22`

  `cp httpd.conf /tmp/`

  Modifico il file in `tmp`
  - `Listen 8080`

  `httpd –f /tmp/httpd.conf`

Soluzioni (2 di 3)

```
ipfw -f -q flush
ipfw add allow tcp from any to me 8080
ipfw add allow tcp from me to any
ipfw add deny tcp from any to me
ipfw add allow ip from any to any
```

```
ipfw -f -q flush
ipfw add deny tcp from any to me 8080 setup
ipfw add allow ip from any to any
```

Soluzioni (2 di 3)

```
ipfw -f -q flush
ipfw add deny tcp from any to me 8080
  established
ipfw add allow ip from any to any
```

```
ipfw -f -q flush
ipfw add allow icmp from any to me icmptype 8
ipfw add allow icmp from me to any icmptype 0
ipfw add deny icmp from any to any
ipfw add allow ip from any to any
```

(**provare i comandi** `ping` **e** `traceroute` **verso il proprio host e vedere il diverso comportamento**)