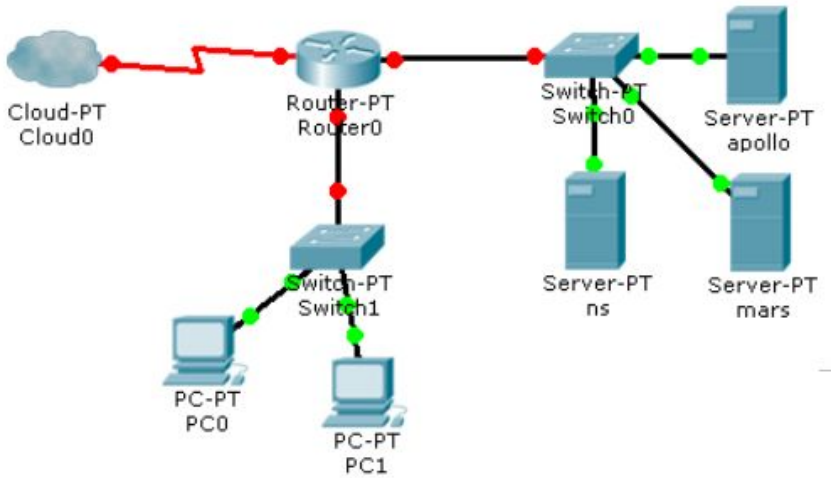
Firewall

Scopo

- Permettere agli utenti di una rete interna di iniziare connessioni solo con alcuni servizi esterni.
- Limitare la possibilità di accessi dall'esterno solo ad alcuni server appositamente dedicati. In questo caso è opportuno separare i server in una rete a parte (DMZ)



Operatività dei firewall

I firewall sono delle cosiddette middle-box, perché operano su diversi livelli. Agiscono filtrando il traffico, permettendo cioè ai pacchetti di passare da una parte all'altra in base a regole definite dall'amministratore di rete.

In particolare gli stateful firewall operano filtraggi in base a:

- porte (TCP o UDP) sorgente/destinazione,
- indirizzi ip sorgente/destinazione,
- interfaccia sorgente/destinazione,
- stato di eventuali connessioni aperte o in fase di attivazione.

Tipi di firewall

Desktop firewall:

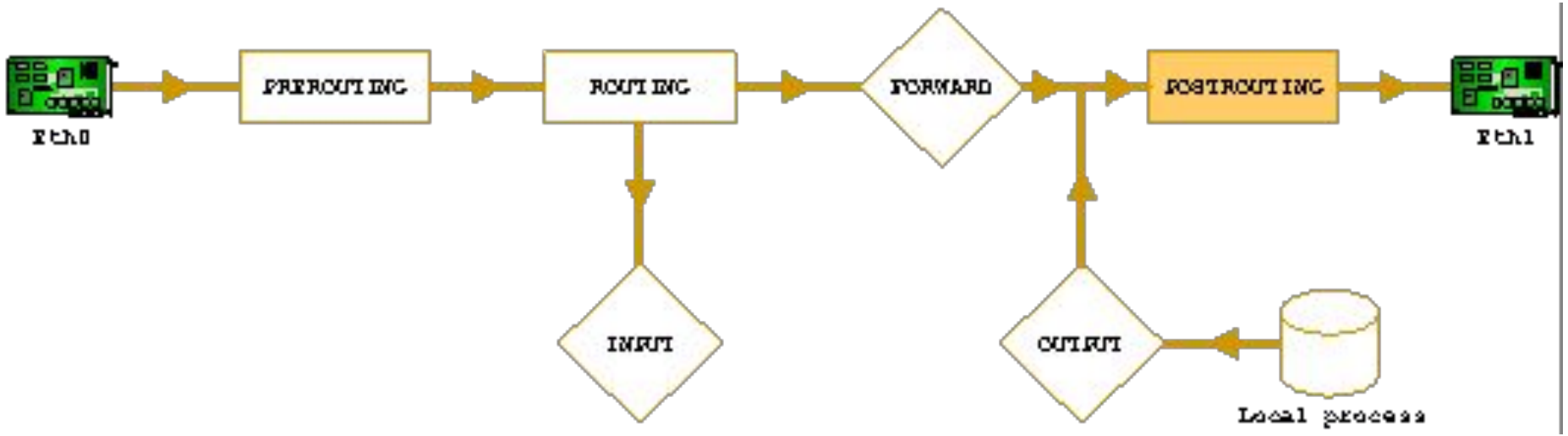
è un programma in esecuzione su computer, come ad esempio Windows Defender. Serve a proteggere il computer.

Network firewall:

è interposto all'estremità della rete (dispositivo ad hoc o regole impostate nel router) che protegge l'intera rete.

Anche quando la rete è protetta da firewall, è opportuno attivare i desktop firewall, per aumentare il livello di protezione e per proteggersi da attacchi provenienti dall'interno.

Linux firewall: iptables



Linux firewall: iptables

È l'applicazione Linux che gestisce le regole di filtraggio. Impostando le regole di filtraggio nel router, otteniamo un firewall di rete.

Le regole di filtraggio possono essere granulari quanto si vuole, ma esistono delle "catene" a cui si applicano regole predefinite.

Le catene predefinite sono:

PREROUTING

> contiene direttive che vengono applicate prima del routing ai pacchetti che "entrano" da un'interfaccia. In questa fase si applicano le le regole per la gestione del destination NAT (DNAT).

Linux firewall: iptables

FORWARD:

definisce le regole da applicare ai pacchetti che devono essere inoltrati verso altre interfacce di rete, e quindi “uscire” dal router.

INPUT:

definisce le regole da applicare ai pacchetti che hanno come la macchina locale, ossia sono destinati a un processo locale.

Linux firewall: iptables

OUTPUT:

definisce le regole da applicare ai pacchetti che hanno come sorgente la macchina locale, ossia sono stati generati da un processo locale.

POSTROUTING:

definisce regole applicate, prima di uscire dalla scheda di rete interna, ai pacchetti provenienti dalle catene forward e output. In questa fase si applicano le regole per il source NAT (SNAT).

Linux firewall: iptables

In ognuno di questi passaggi ogni direttiva si chiede sostanzialmente: “se l’header del pacchetto verifica certe condizioni, che cosa devo fare del pacchetto” ?

La risposta a questa domanda può essere o di accettare il pacchetto che continua nel suo percorso all’interno delle altre direttive e degli altri step o rigettare il pacchetto che viene definitivamente buttato via.

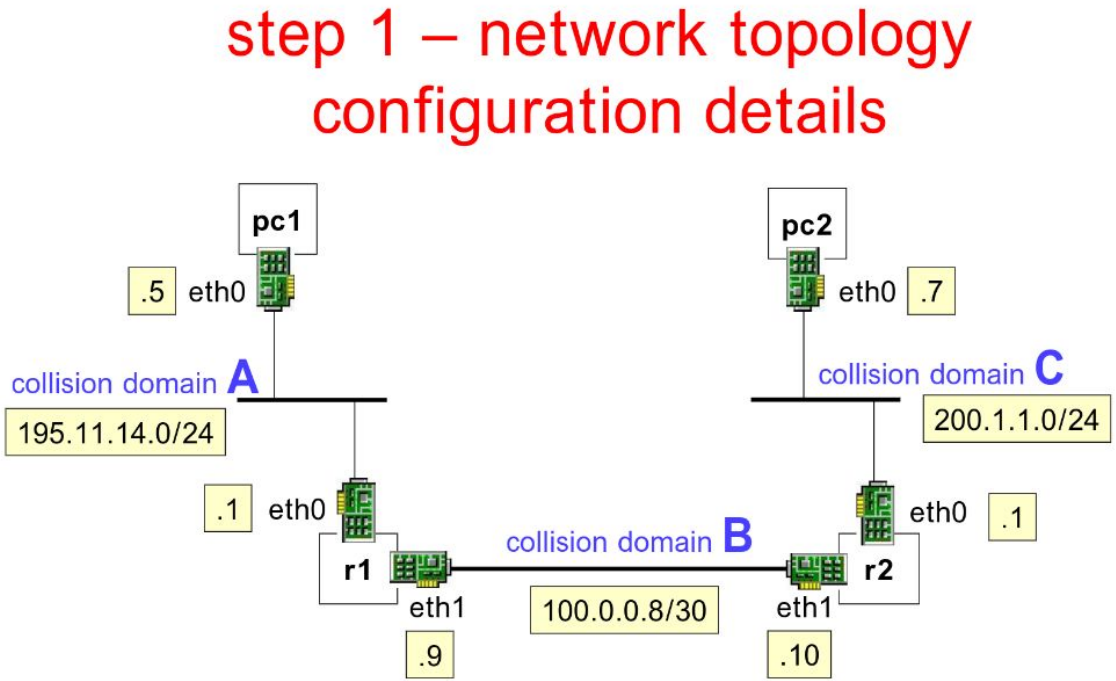
Linux firewall: iptables

In ogni passaggio, se il pacchetto non verifica nessuna delle condizioni impostate, sono definite delle regole di default (policies) da applicare ai pacchetti.

Le regole di default per le catene INPUT, FORWARD E OUTPUT sono preimpostate ad ACCEPT, implementando quindi una politica di default permit.

È possibile (e conveniente per semplificare le regole) definire delle catene personalizzate.

Linux iptables: un esempio



Linux iptables: un esempio

Il router R1 collega la rete A all'esterno.

Le sue schede di rete sono eth0 verso la rete interna, eth1 verso il resto del mondo.

Per prima cosa, impostiamo le policy per una politica di default deny:

```
iptables -P INPUT DROP

iptables -P FORWARD DROP

iptables -P OUTPUT DROP
```

Linux iptables: un esempio

In questo modo tutto il traffico è bloccato. Da qui in poi aggiungeremo le regole che consentiranno di accedere ai servizi che si vogliono attivare.

Per semplificare le regole, è utile definire due nuove catene:

```
iptables -N lan_inet
```

```
iptables -A FORWARD -i eth0 -o eth1 -j lan_inet
```

creo una nuova catena che si chiama lan_inet e su questa catena “dirotto” tutto il traffico in ingresso dalla scheda eth0 e diretto verso la scheda eth1 (cioè il traffico che dalla LAN va verso Internet).

Linux iptables: un esempio

```
iptables -N inet_lan

iptables -A FORWARD -i eth1 -o eth0 -j inet_lan
```

creo una nuova catena che si chiama inet_lan e su questa catena “dirotto” tutto il traffico in ingresso dalla scheda eth1 e diretto verso la scheda eth0 (cioè il traffico che da Internet va verso la LAN).

Utilizzando regole basate sulle interfacce fisiche da cui provengono o verso cui sono instradati i pacchetti protegge da situazioni di IP spoofing.

Linux iptables: un esempio

Queste regole abilitano l'uscita verso l'esterno di pacchetti per l'utilizzo dei servizi HTTP, HTTPS, DNS:

```
iptables -A lan_inet -p tcp --dport 80 -j ACCEPT

iptables -A lan_inet -p tcp --dport 443 -j ACCEPT

iptables -A lan_inet -p udp --dport 53 -j ACCEPT
```

Linux iptables: un esempio

Questa regola (ultima della catena) serve a inviare un segmento TCP di reset negli altri casi, così le applicazioni ricevono un “porta chiusa” e non fanno ulteriori tentativi.

```
iptables -A lan_inet -p tcp -j REJECT --reject-with tcp-reset
```

Queste regole (in questo ordine) consentono di ricevere le risposte dai server contattati, rispondendo con reset ad altri tentativi di accesso dall’esterno

```
iptables -A inet_lan -m state --state ESTABLISHED, RELATED -j ACCEPT
```

```
iptables -A inet_lan -p tcp -j REJECT --reject-with tcp-reset
```

Linux iptables: un esempio

Infine, se si deve abilitare il NAT sui pacchetti che escono verso Internet, supponendo che la rete interna sia la 192.168.1.0/24:

```
iptables -t nat -I POSTROUTING -s 192.168.1.0/24 -o eth1 -j MASQUERADE
```

Per consentire invece l'accesso dall'esterno a un servizio interno alla rete si deve abilitare il D-NAT. Ad esempio, per un server Web in ascolto sulla porta 80:

```
iptables -t nat -A PREROUTING -p tcp -d <IP PUBBLICO>
--dport 80 -j DNAT --to-destination <IP PRIVATO>
```