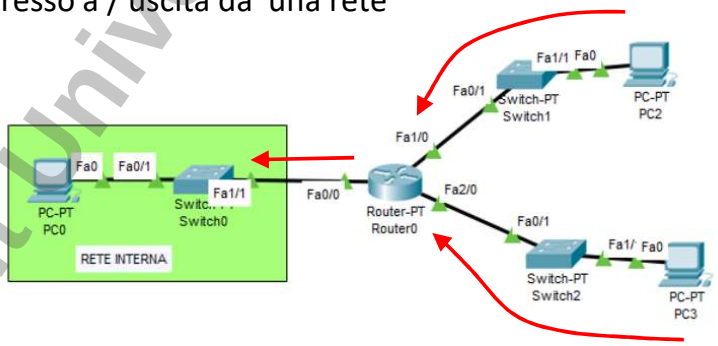
LABORATORIO DI RETI DI CALCOLATORI

Configurazione router Cisco: ACL e NAT

router: Access Control List

- sul router esiste la possibilità di limitare il traffico in ingresso a / uscita da una rete



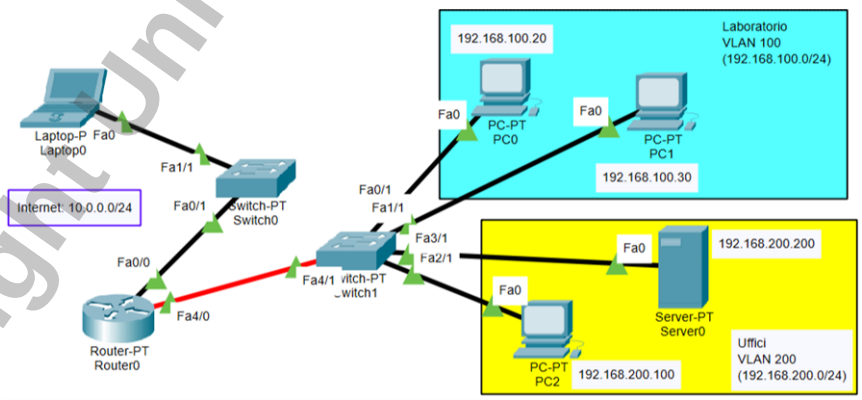
- IN "rete interna" = (IN Fa1/0 ∨ IN Fa2/0) ∧ **OUT** Fa0/0
  - **VOI SIETE IL ROUTER!**

Tipi di ACL disponibili

- ACL *standard* *<ID 1-99>*: si può solo selezionare lo IP sorgente dei pacchetti da (non) far passare
- ACL *extended* *<ID 100-199>*: si possono selezionare
  - protocollo di livello Network o Trasporto
  - indirizzo sorgente e/o destinazione /* wildcard **`any`** */
  - (insiemi di) porte /* e quindi servizi well-known */
  - modalità: **`permit`** / **`deny`**
  - **`established`**: segmenti TCP con **`ack`** flag a 1
    - *dal secondo passo del three-way handshake*
- ACL *named extended* *<ID 100-199>*: possibilità di modifica successiva

ACL extended: esempio

- Si permette accesso a Web Server in VLAN Uffici
- ma **no** ping (ICMP) nelle due VLAN (es. DoS attack)



ACL extended: esempio <cont.>

I. **creazione ACL – paradigma:**

- access-list acl# permit|deny protocol source|wildcard [port] destination|wildcard [port] [established] [log]

- applicazione a esempio:

```
Router(config)# access-list 110 permit TCP any host
192.168.200.200 eq 80
Router(config)# access-list 110 deny ICMP any any
```

- *"ACL 110 permette il passaggio di segmenti TCP che arrivano da qualunque sorgente e che hanno destinazione con indirizzo 192.168.200.200 e port# uguale a 80"*
- *"ACL 110 nega il passaggio di pacchetti ICMP da qualunque host a qualunque altro"*

ACL extended: esempio <cont.>

- le regole vengono analizzate in ordine
  - se un pacchetto non fa match con nessuna delle regole, allora *vale l'implicita* **`deny IP any any`** *anche se non scritta*
- port number: altri qualificatori sono

`lt` (less than)	**`gt`** (greater than)
`neq` (not equal)	**`range`** (intervallo)

II. **applicazione ACL a interfaccia:**

```
Router(config)# interface fastEthernet 4/0.200
Router(config-if)# ip access-group 110 out
```

- applico la ACL a tutto il traffico che *esce* da quella sub-interface, ovvero che *entra* nella VLAN Uffici

Proviamo!

ACL extended: esempio <cont.>

- si possono ottenere statistiche su controllo traffico

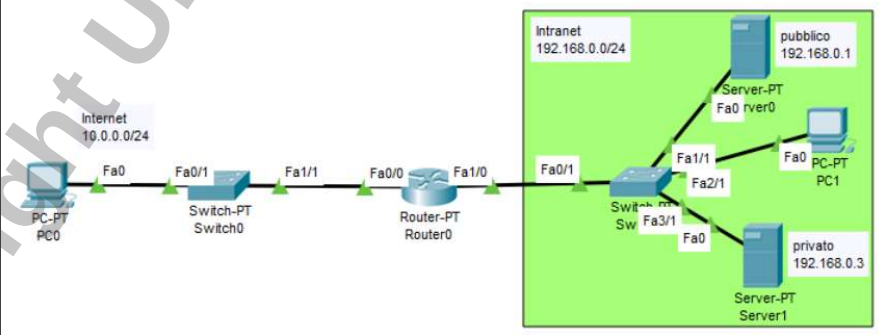
```
Router#show access-lists ?
  <1-199>  ACL number
  WORD     ACL name
  |        Output Modifiers
  <cr>
Router#show access-lists
Extended IP access list 110
    10 permit tcp any host 192.168.200.200 eq www (5 match(es))
    20 deny icmp any any (1 match(es))
Extended IP access list 120
    10 deny icmp any any (1 match(es))

Router#
```

- si può controllare configurazione

```
Router#show ip interface fast 4/0.200
FastEthernet4/0.200 is up, line protocol is up (connected)
  Internet address is 192.168.200.254/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is 110
  Inbound  access list is not set
  Proxy ARP is enabled
```


ACL named extended: esempio

- tutti possono accedere al Server Web pubblico
- solo gli host della VLAN verde possono accedere anche al Server Web privato



ACL named extended: esempio

- definizione ACL:

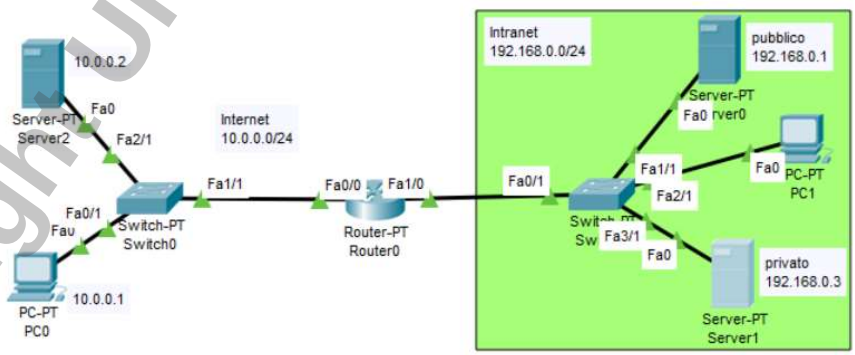
```
Router(config)#  ip access-list extended 100
Router(config-ext-nacl)#  permit TCP any host 192.168.0.1 eq www
Router(config-ext-nacl)# deny IP any any
Router(config-ext-nacl)#  exit
Router(config)#  interface fastEthernet 0/0
Router(config-if)#  ip access-group 100 in
```

- *"ACL 100 permette l'ingresso dall'interfaccia fastEthernet 0/0 di tutti i segmenti TCP generati da qualunque sorgente, e destinati allo host 192.168.0.1 su porta 80"*
- *"ACL 100 non accetta alcun altro tipo di traffico IP"*

Proviamo!

ACL named extended: esempio

- aggiungiamo un server web pubblico esterno (Server2)
  - *nessuno degli end system nella VLAN verde può accedervi!*
  - dedurre la ragione con simulazione passo-passo (filtri ARP, ICMP, TCP e HTTP)



ACL named extended: esempio

- *three-way handshake*: dal 2° segmento il flag **`ack`** è sempre abilitato
  - da Teoria: per riscontrare il 1° segmento SYN
  - e poi per riscontrare i segmenti dati precedenti
  - e infine per riscontrare tutti i dati e il FIN
- modifica ACL a interfaccia:

```
Router(config)#  ip access-list extended 100
Router(config-ext-nacl)#  15 permit TCP any any established
```

  - con inserimento della nuova regola in posizione opportuna
- verificare che ora la rete funziona come desiderato!

router: Network Address Translation

- per la traduzione degli indirizzi è necessario:

1. configurare le interfacce di ingresso e uscita alla rete abilitando il servizio NAT
2. creare una ACL che determini quali indirizzi devono essere tradotti
3. configurare il router stabilendo per quale traffico devono essere tradotti gli indirizzi (determinato dalla ACL) e qual è l'indirizzo presentato all'esterno

- ... consideriamo la medesima topologia dell'esempio precedente

configurazione NAT

```
Router(config)#ip nat ?
  inside   Inside address translation
  outside  Outside address translation
  pool     Define pool of addresses
Router(config)#ip nat inside ?
  source   Source address translation
Router(config)#ip nat inside source ?
  list     Specify access list describing local addresses
  static   Specify static local->global mapping
Router(config)#ip nat outside ?
  source   Source address translation
Router(config)#ip nat outside source ?
  list     Specify access list describing local addresses
  static   Specify static global->local mapping
Router(config)#ip nat outside source list ?
  <1-199>  Access list number for local addresses
  WORD     Access list name for local addresses
Router(config)#ip nat outside source list 110 ?
  pool  Name pool of global addresses
```

scelta interfacce interna/esterna

pool indirizzi nel codominio

traduzione indirizzo della sorgente

per server

- ACL: bisogna definire il traffico a cui applicare la traduzione

NAT: configurazione esempio

- configurazione interfacce e ACL:

```
Router(config)# interface fastEthernet 1/0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface fastEthernet 0/0
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# access-list 110 permit ip any any
```

- la ACL *non* deve essere associata ad alcuna interfaccia

NAT: configurazione esempio

- configurazione servizio:

```
Router(config)# ip nat inside source list 110 interface fastEthernet 0/0
```

- *"quando si fa NAT degli indirizzi interni si usi come criterio degli indirizzi da tradurre quello indicato nella ACL 110 e si traducano gli indirizzi usando l'indirizzo IP dell'interfaccia FastEthernet 0/0"*

- come funziona ora la rete?
  - ping interno → esterno funziona
  - *il server web pubblico nella Intranet non è più accessibile da Internet!* *Per forza: non ne vediamo l'indirizzo...*

NAT: configurazione esempio

accesso a servizi interni:

- mapping tra indirizzo esterno e porta → indirizzo interno e porta servizio: ***port forwarding***

```
Router(config)# ip nat inside source static tcp IP_interno porta_interna
IP_esterno porta_esterna
```

- nel nostro esempio:
  - *IP_interno = 192.168.0.1* *porta_interna = 80*
  - *IP_esterno = 10.0.0.254* *porta esterna = 80*
- ora tutto funziona come atteso!
  - simulazione passo-passo di accesso a web server pubblico per osservare la rimarcatura pacchetti