Sécurité des Réseaux

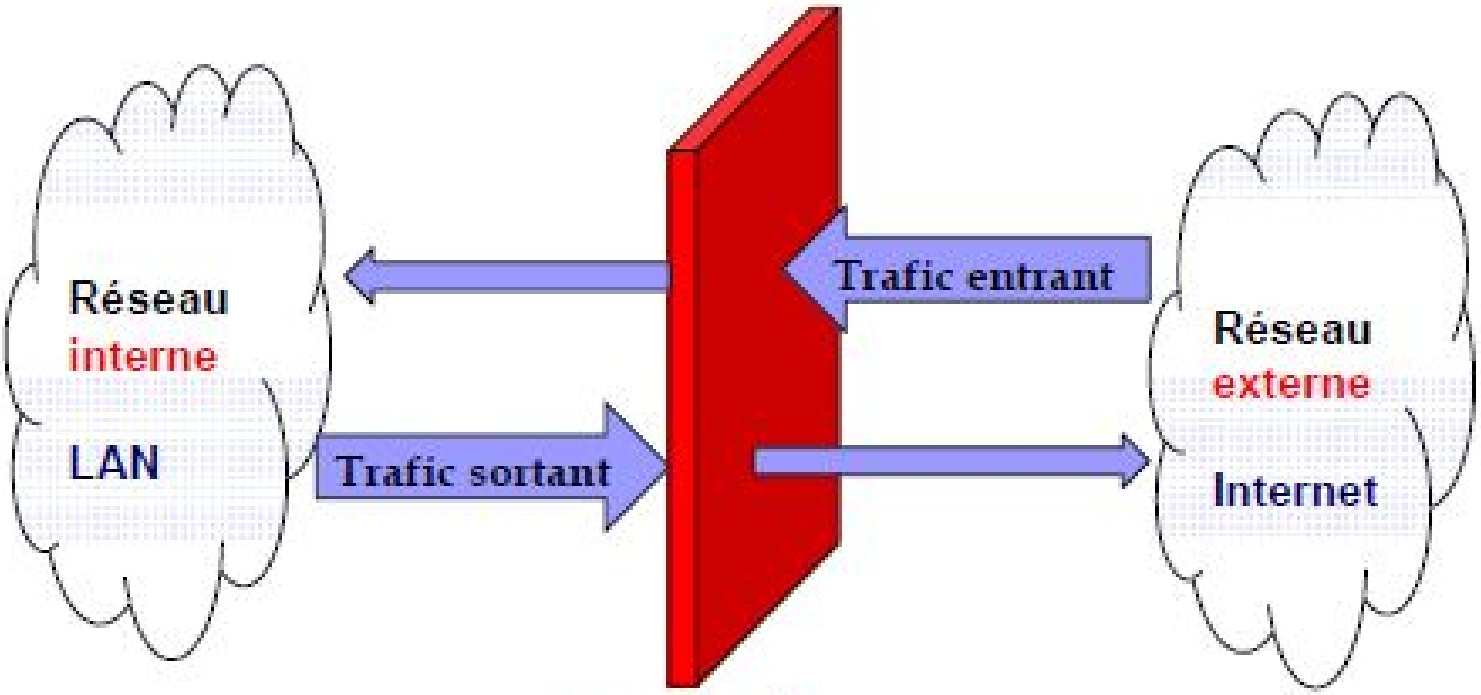
Chap 3: Les Pare-feu (firewalls)

Master I I2A

Plan

Plan

Firewall



Firewall
matériel ou logiciel

Principe de filtrage

- Le filtrage est un des outils de base de la securite. IL EST NECESSAIRE !
- Filtrage optimiste : PERMIT ALL
  - Tout est permis a part quelques services (ports)
  - Facile a installer et a maintenir : Seulement quelques regles a gerer
  - Securite faible : Ne tient pas compte des nouveaux services pouvant etre ouvert.
- Filtrage pessimiste : DENY ALL
  - Rejet systematique
  - Exception : services specifiques sur machines specifiques. Ex : Autorisations explicites pour les services HTTP, SMTP, POP3,..
  - Plus difficile a installer et a maintenir
  - Securite forte : Les nouveaux services doivent etre exmplictement declarés
- Prendre en compte les connexions entrantes et les connexions sortantes

Principe du filtrage des paquets

- Le filtrage se fait en analysant les entetes IP, TCP et UDP
- on définit une règle de filtrage en considérant :
  - @IP source
  - @IP destination
  - Port source
  - Port destination
  - Protocole encapsulé (ICMP, UDP, TCP,...)
  - Flag ACK (de TCP)
  - Interface d'accès
  - Sens du trafic filtré
- à chaque règle est associé une action :
  - laisser passer le packet OU
  - bloquer (détruire) le paquet
- Attention à l'ordre des règles : La première qui correspond est celle sélectionnée (First Matching, First Applied !)

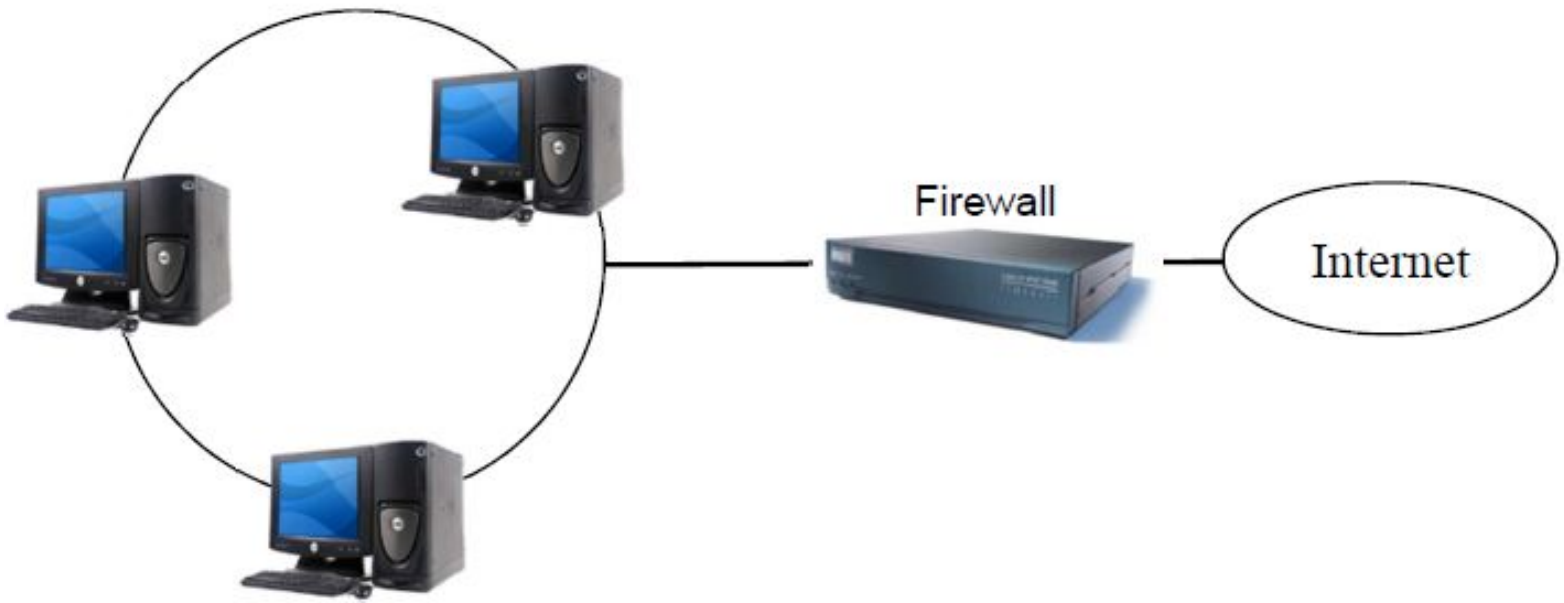
Difference entre firewall et routeur

- Un firewall ne fait pas de "IP FORWARDING"
- Un firewall peut faire du routage au niveau applicatif : Existence de mandataires (proxy) HTTP, POP3, etc . . .
- Les mandataires peuvent être intelligent : Filtrage par le contenu (informations)
- Implantation du firewall se fait par :
  - Un matériel spécialisé (Cisco PIX, ...)
  - Une machine simple avec plusieurs cartes réseaux + logiciels
  - Firewall 1 (Checkpoint), Raptor, Shorewall (Linux), ...

Plan

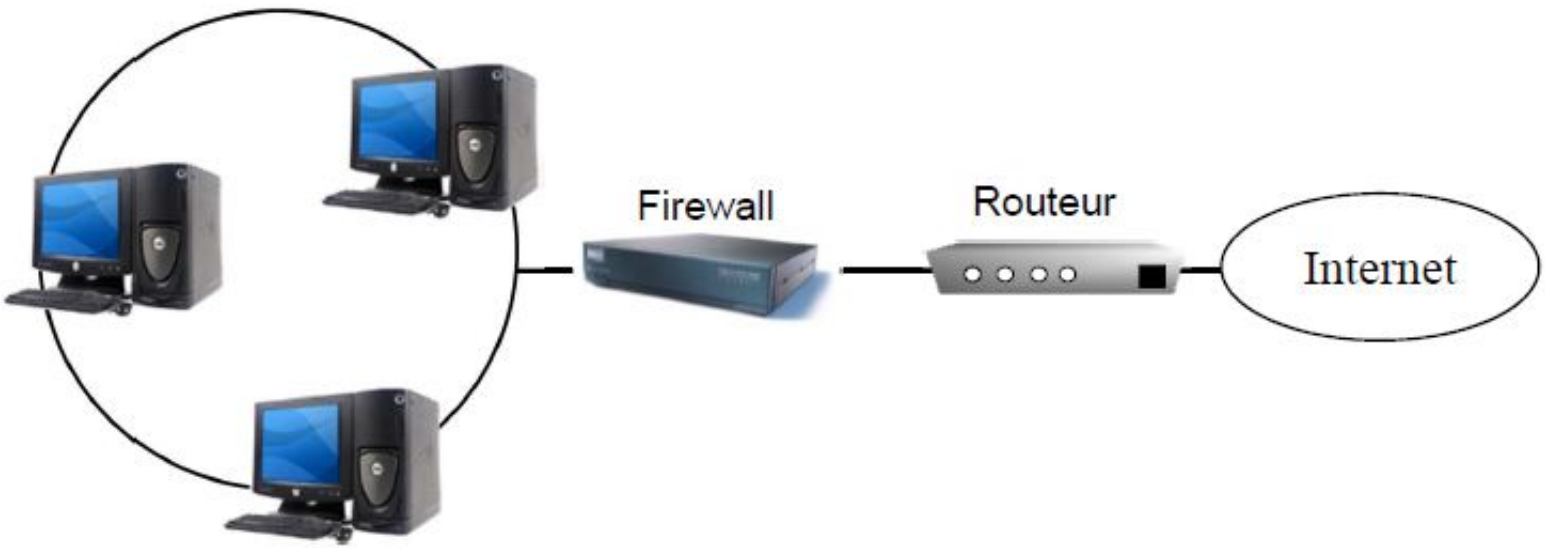
Architecture avec Firewall sans routeur

- On donne des adresses IP privées aux machines du réseau. ex : 10.1.1.1, 10.1.1.2. Donc les clients ne peuvent pas dialoguer directement avec l'extérieur
- les serveurs ont aussi des adresses publiques
- nécessicité de Passage par des mandataires internes (proxy web, ftp, smtp, pop)
- Reste une solution limitée



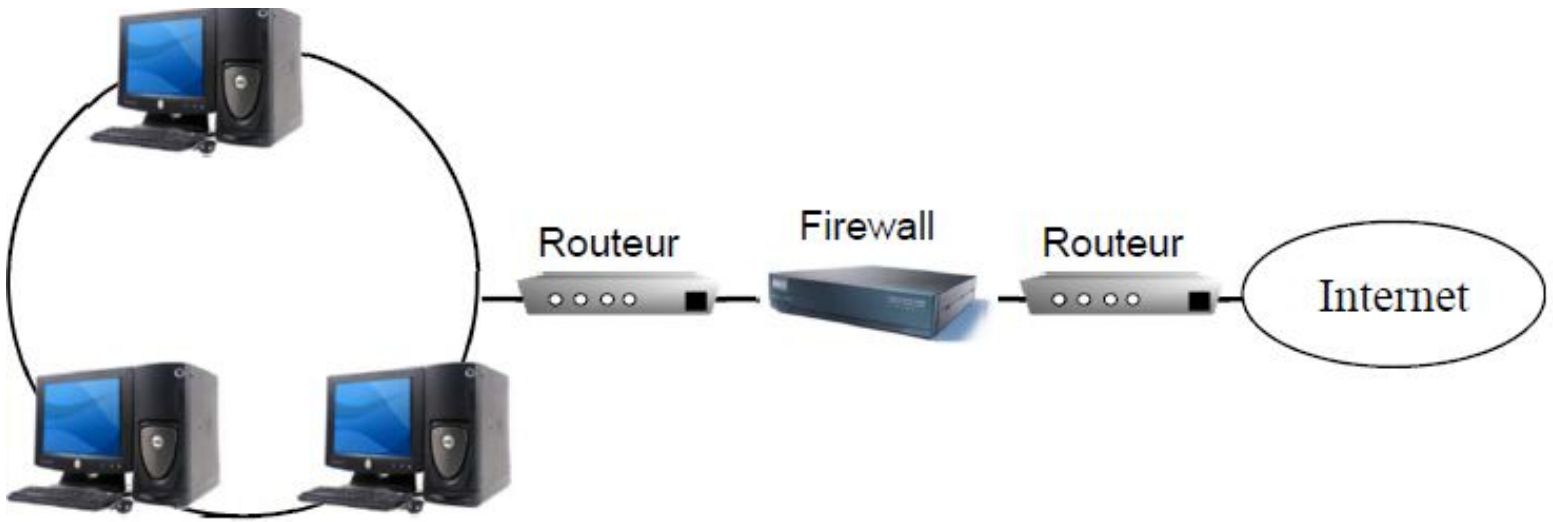
Architecture avec Firewall et routeur

- Modele avec Firewall et routage
- Le firewall est la seule machine visible de l'exterieur
  - Le firewall effectue le controle d'acces
  - Le routeur effectue le routage (translation d'adresse) $\mapsto$ NAT



Architecture avec Firewall et double routeurs

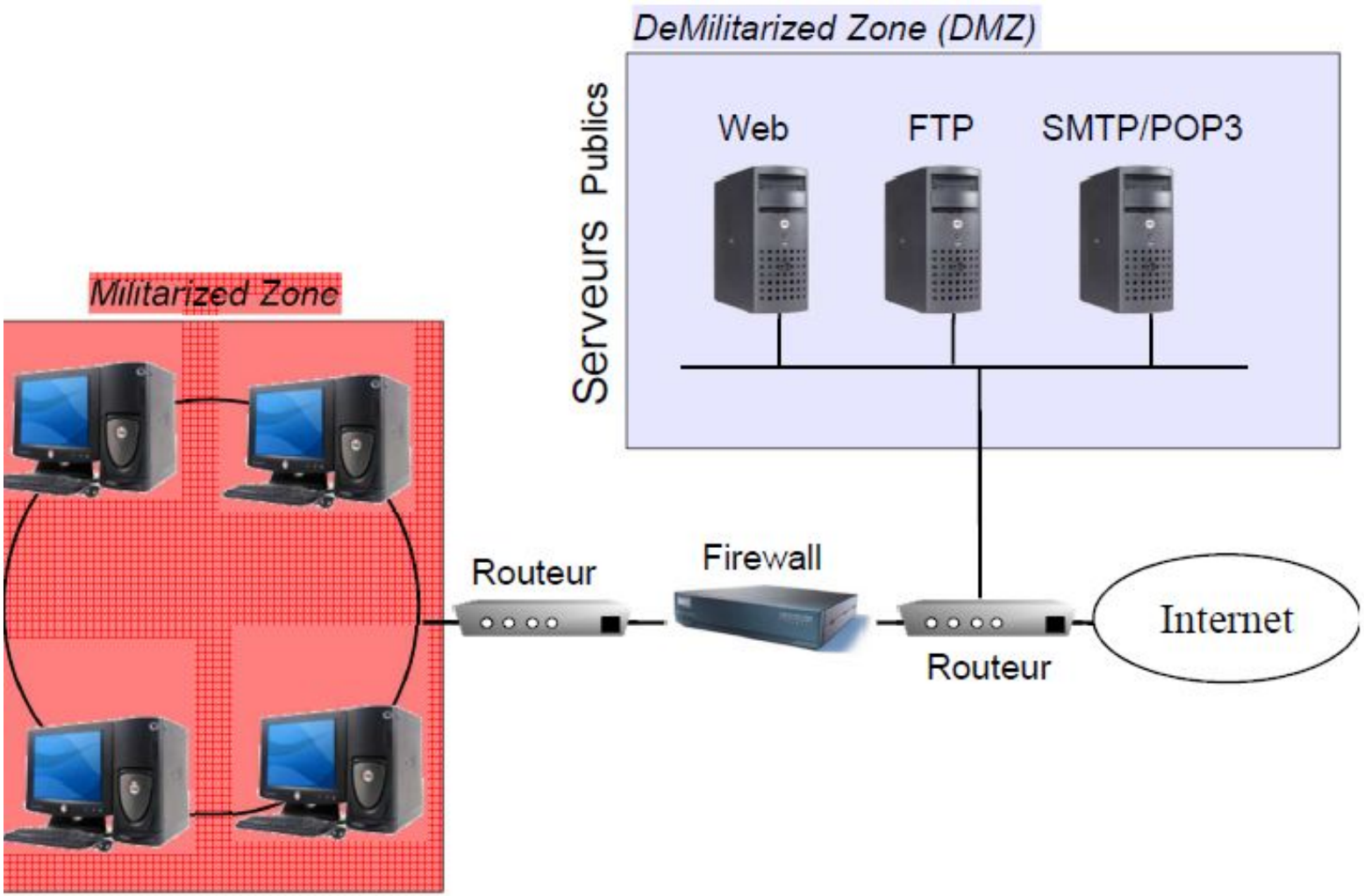
- Un routeur pour les connexions entrantes
- Un routeur pour les connexions sortantes
- Le firewall contrôle les accès entrants et sortants



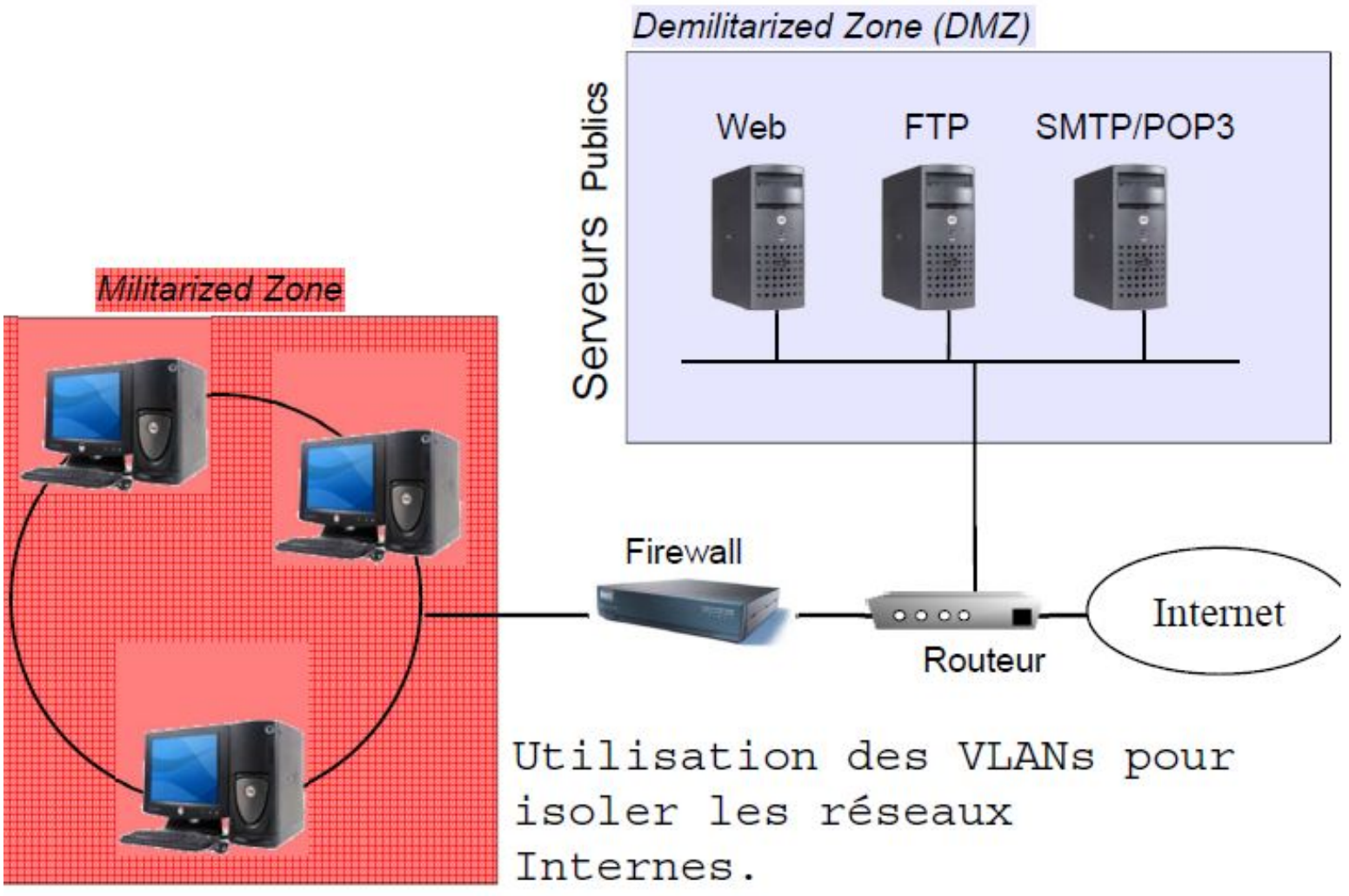
firewall et zone DMZ

- architecture DMZ : découpage du réseau interne en 2 zones isolées
- serveurs accessibles de l'extérieur situés en zone démilitarisée
- clients inaccessibles de l'extérieur situés en zone militarisée
- deux configurations possibles :
  - utiliser deux routeurs avec le firewall
  - utiliser un roiteur à 3 pattes avec le firewall

Firewall et zone démilitarisée : 2 routeurs (simple + sécurisée)

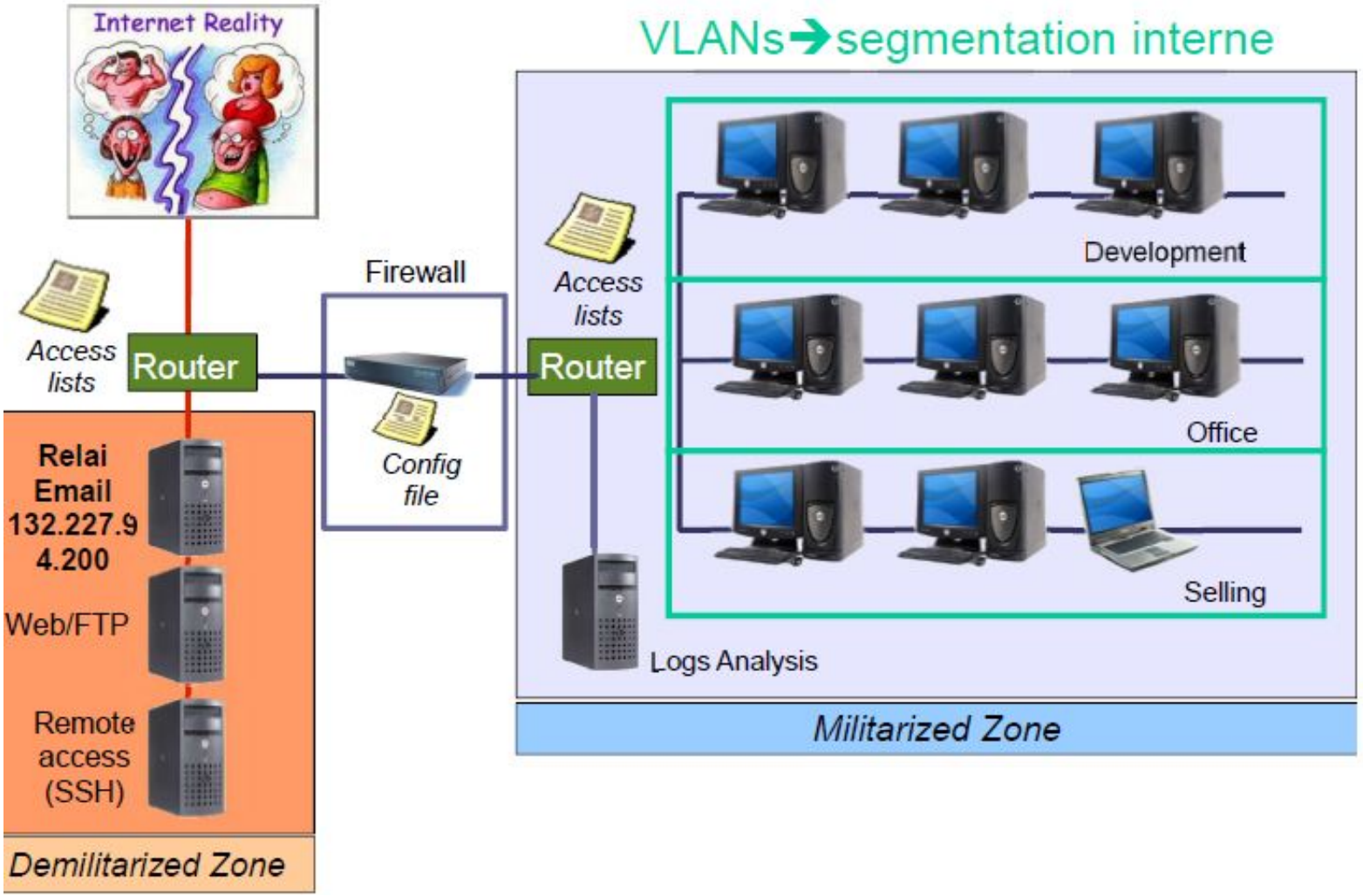


Firewall et zone démilitarisée : Routeur à 3 pattes (économique)



Utilisation des VLANs pour isoler les réseaux Internes.

Firewall, VLAN et zone démilitarisée



Plan

Types de filtrage

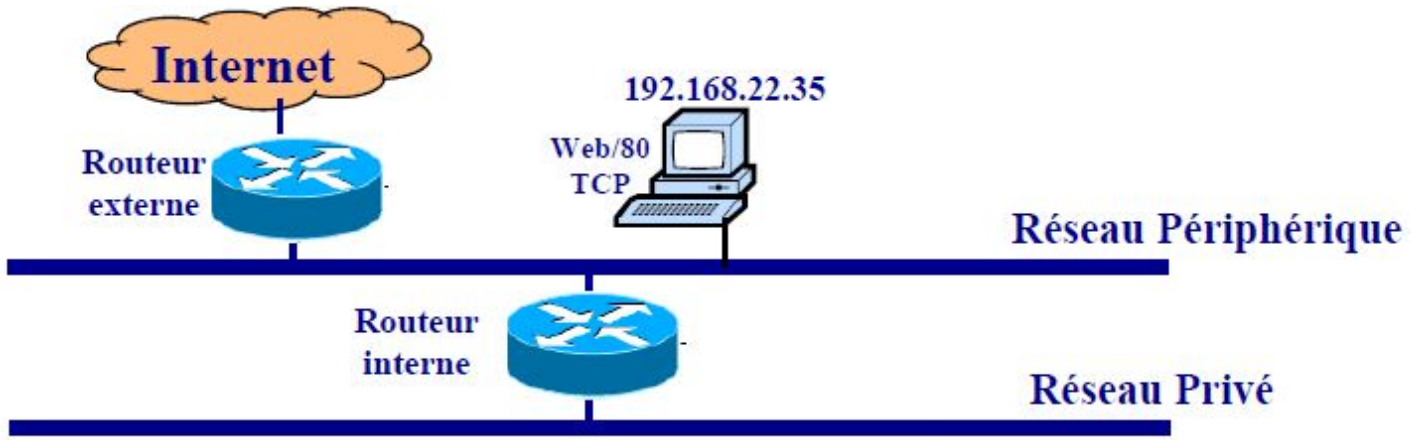
- Filtrage sans état : Stateless
  - filtrage de chaque packet et le comparer avec une liste de règles préconfigurés (ACL)
  - implémenté sur les routeurs et les systèmes d'exploitation
- Filtrage à état : Statefull
  - tracer les connexions et les sessions ds des tables d'états internes au firewall
  - décider en fonction des états des connexions
  - l'application des règles est possible sans lecture de ACL à chaque fois (les paquets d'une connexion actives seront acceptés)
- Filtrage applicatif (firewall Proxy)
  - réalisé au niveau de la couche application
  - permet d'extraire les données du protocole applicatif
  - chaque protocole est filtré par un processus dédié

Processus de développement des filtres

- définitions des régles de filtrage
  - utiliser le max des critères (@IP, port, ACK, etc)
- Pour chaque service interne et externe
  - définir les règles pour autoriser les utilisateurs internes à accéder à des services externes
  - définir les règles pour autoriser les utilisateurs externes à accéder à des serveurs ds le réseau interne
- Pour un service à autoriser
  - accepter le flux dans les deux sens (client -> serveur et serveur -> client)
- Pour un service à bloquer
  - il suffit de bloquer le flux du client->serveur

Exemple de règle de filtrage

Autoriser l'extérieur à accéder au service Web sur le réseau périphérique

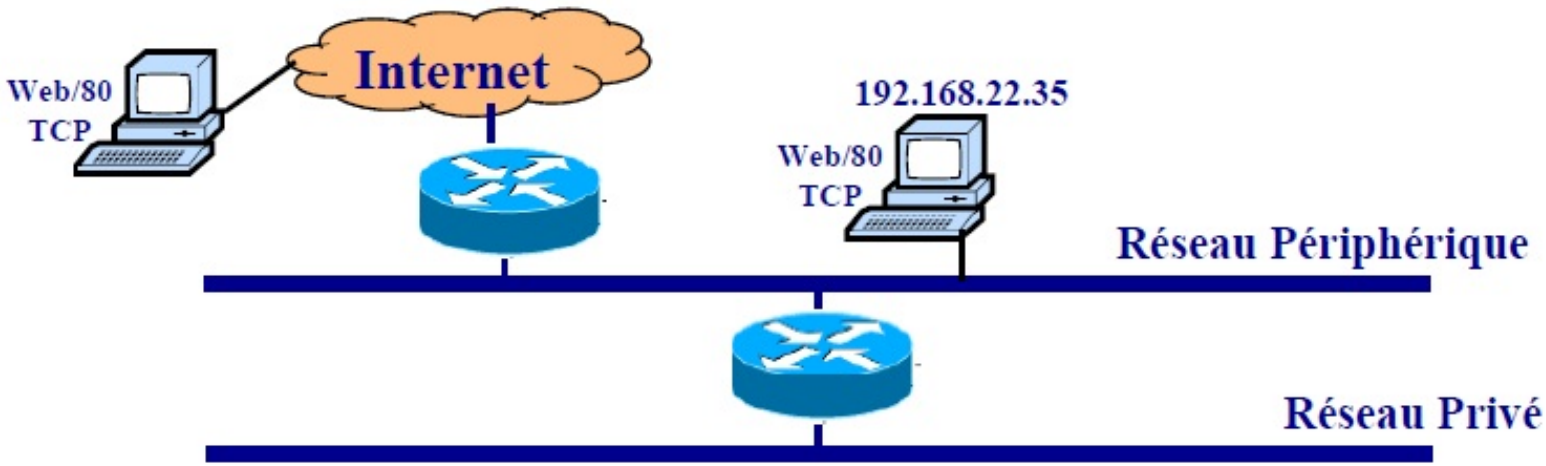


Règle	Direction paquet	IP Source	IP Dest	Prot	Port Source	Port Dest	ACK=1	Action
A	Sortant	192.168.22.35	Toutes	TCP	80	> 1023	Oui	Autoriser
B	Entrant	Toutes	192.168.22.35	TCP	> 1023	80	---	Autoriser
C	Toutes	Toutes	Toutes	Tous	Tous	Tous	---	Refuser

Processus de développement des filtres

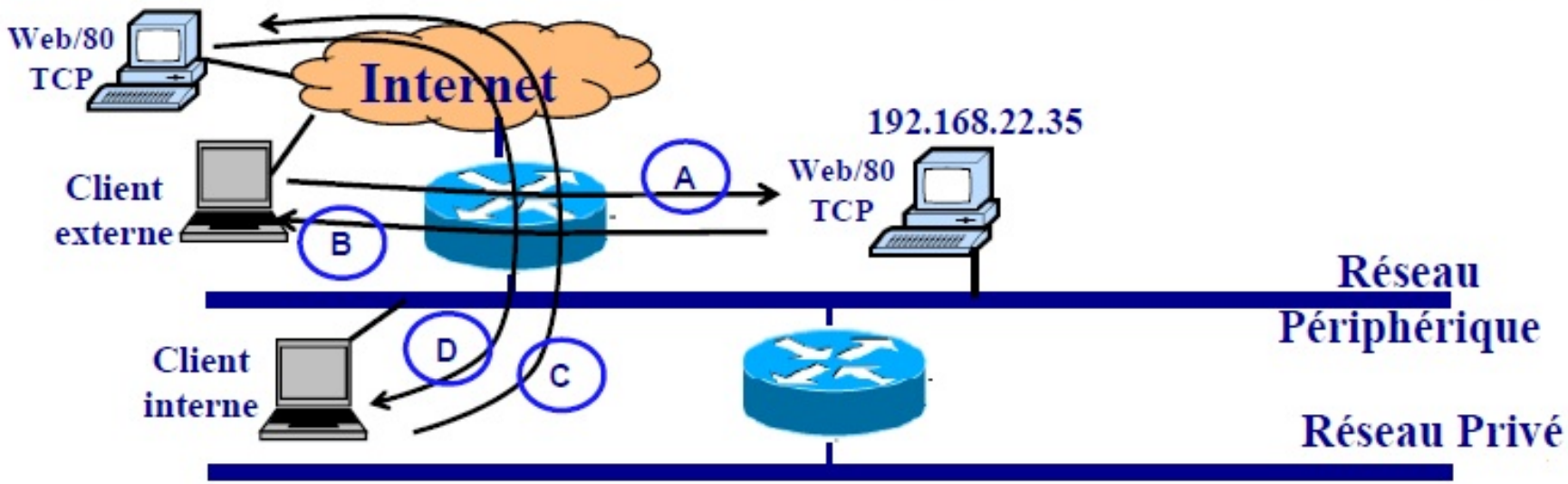
Soit la politique de sécurité :

- Accepter HTTP en entrée et en sortie et rien d'autre



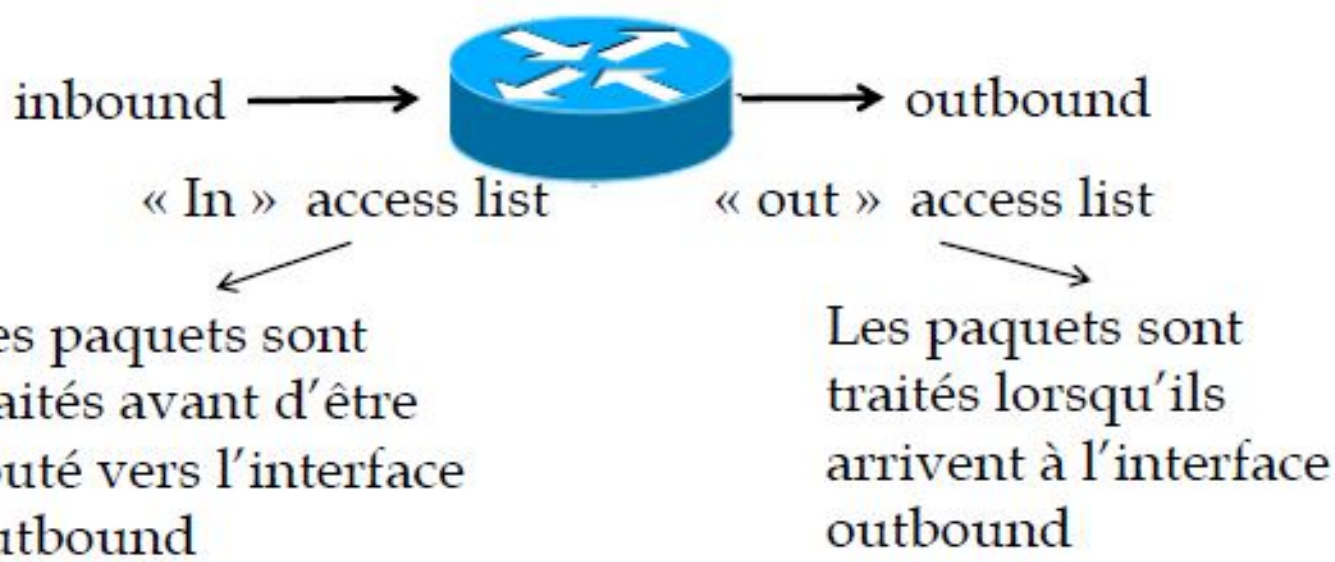
Processus de développement des filtres

Règle	Direction	@ source	@ Dest.	Protocole	Port src.	Port dest.	ACK=1	Action
A	Entrant	Externe	192.168.22.35	TCP	>1023	80	---	Autoriser
B	Sortant	192.168.22.35	Externe	TCP	80	>1023	oui	Autoriser
C	Sortant	Interne	Externe	TCP	>1023	80	---	Autoriser
D	Entrant	Externe	Interne	TCP	80	>1023	oui	Autoriser
E	Toutes	Toutes	Toutes	Tous	Tous	Tous	---	Refuser

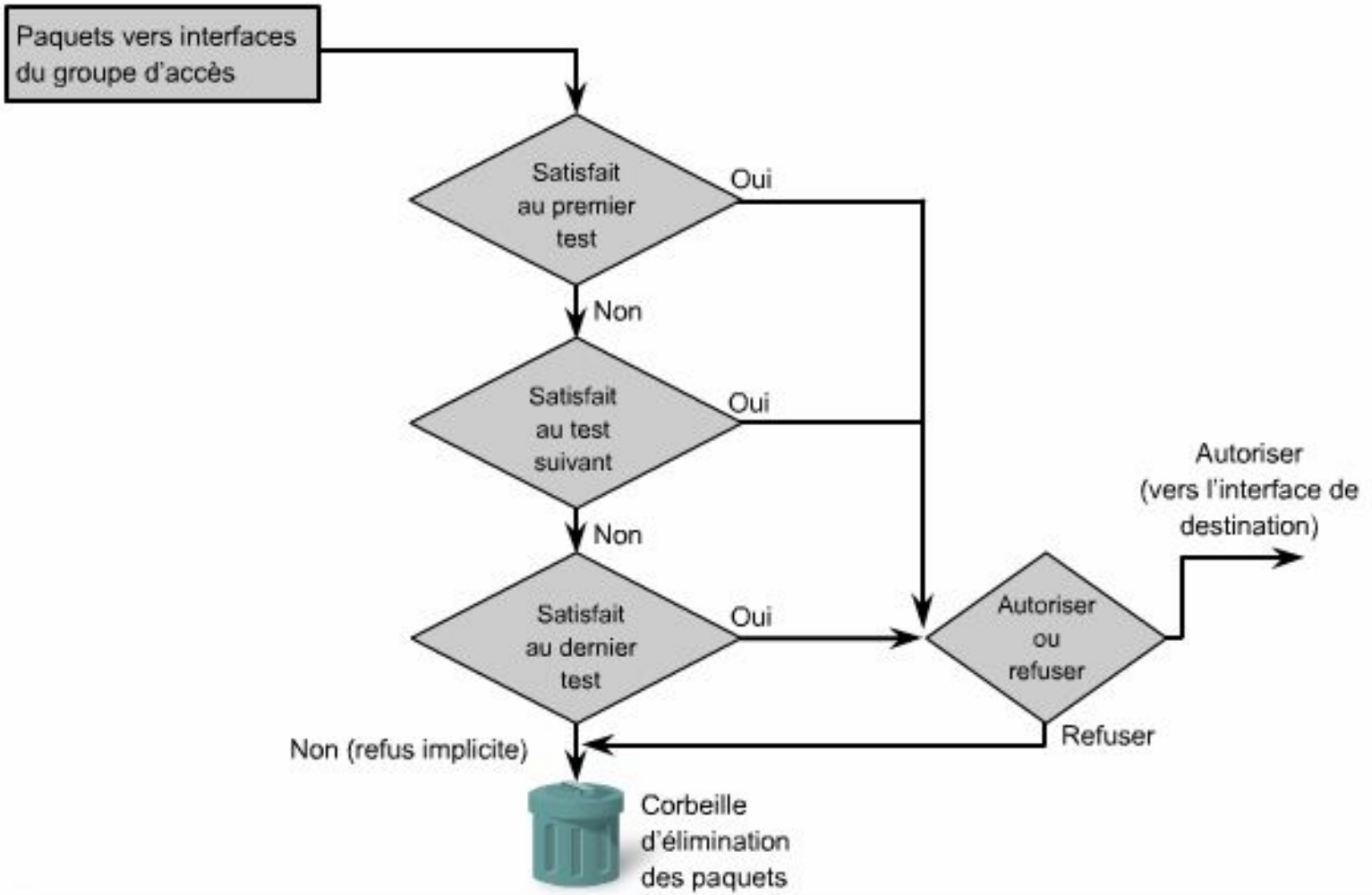


Access Control List ACL

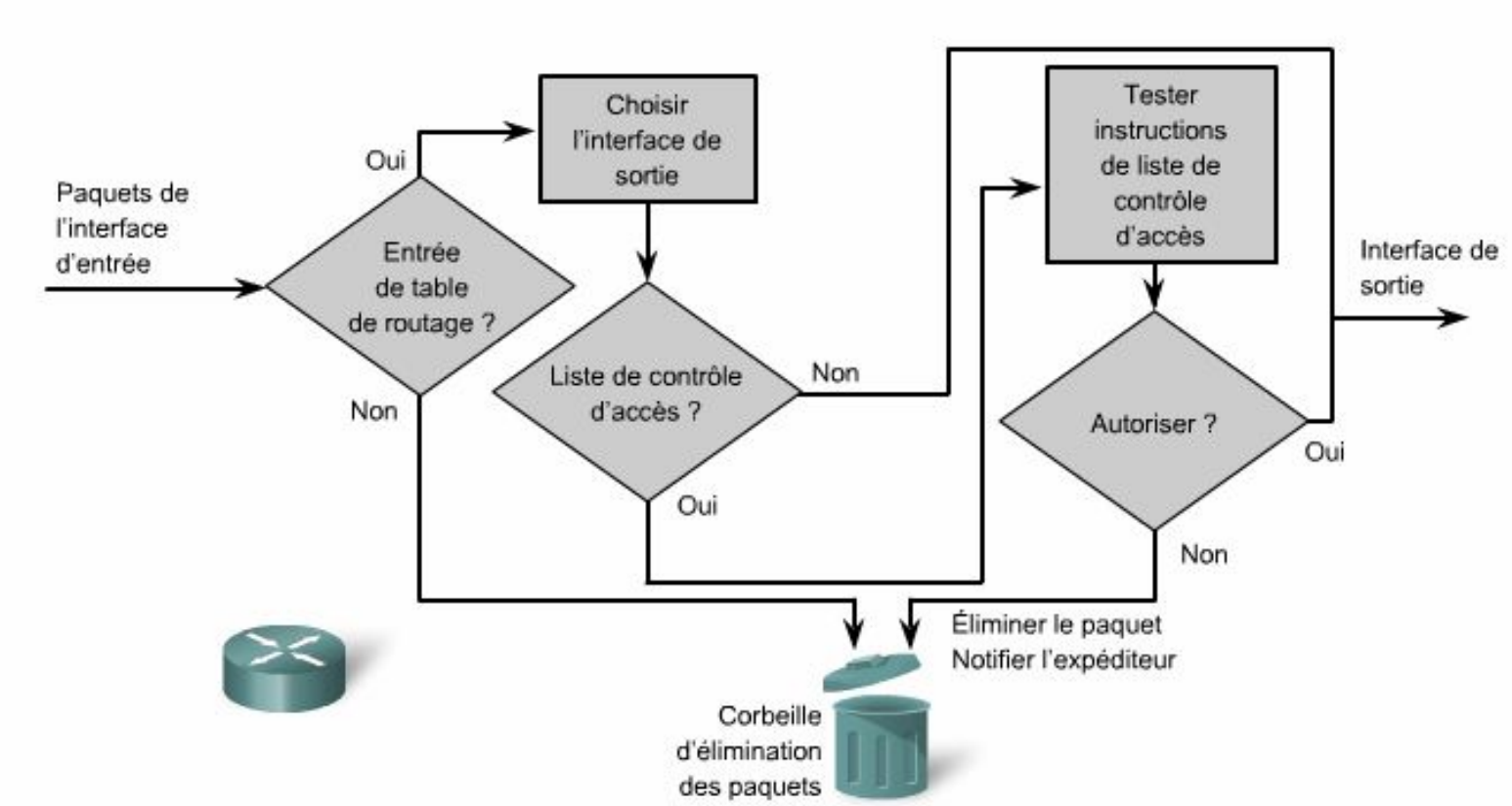
- une ACL doit etre associée à une interface du filtre routeur
  - interface **in** : paquets entrants avant routage ds l'interface du routeur
  - interface **out** : paquets sortant après routage à l'interface du routeur



Listes de contrôle d'accès entrantes



Listes de contrôle d'accès sortantes



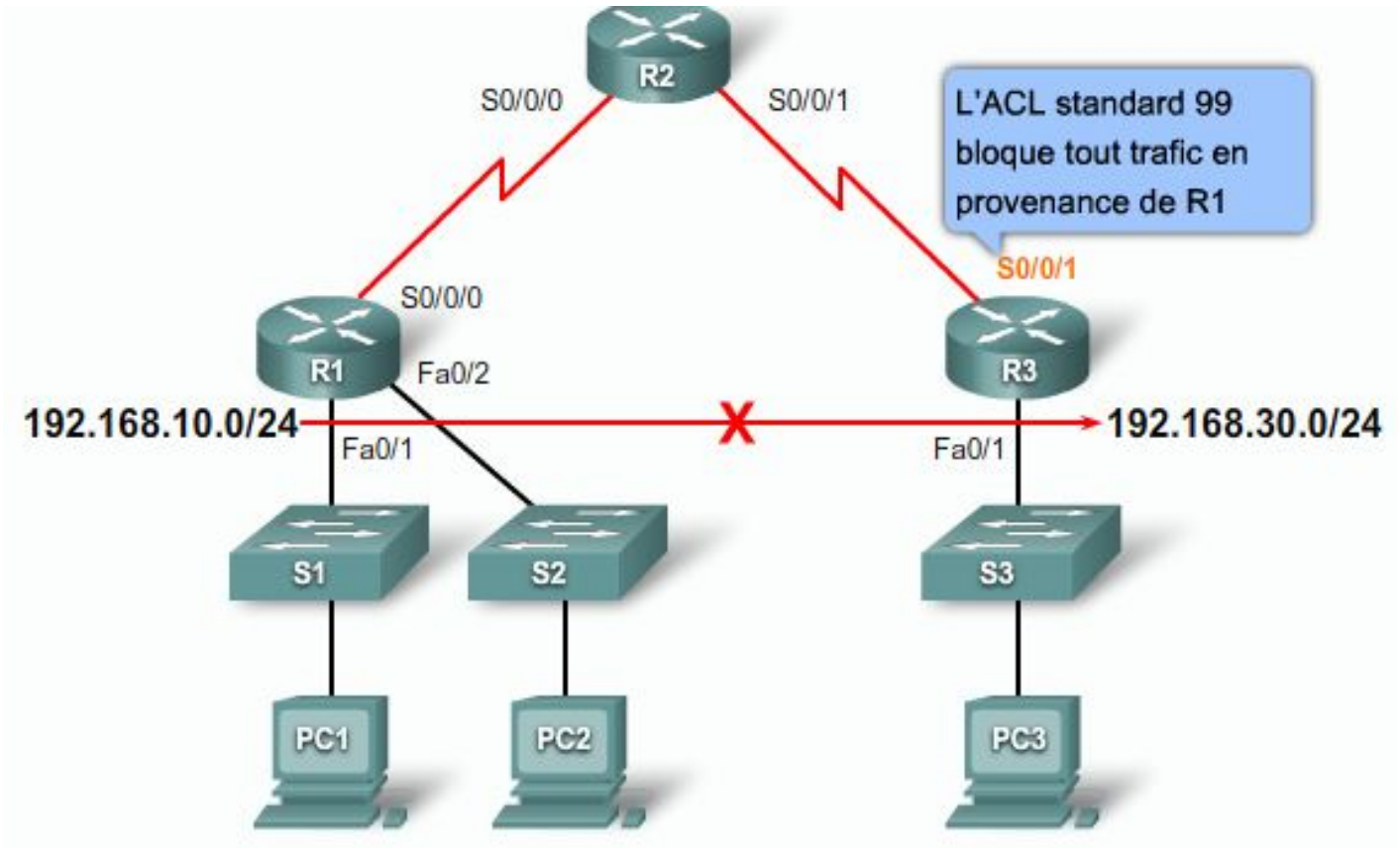
Positionnement des ACL

Les règles de base sont les suivantes :

- Placez les listes de contrôle d'accès étendues le plus près possible de la source du trafic refusé. Ainsi, le trafic indésirable est filtré sans traverser l'infrastructure réseau.
- Étant donné que les listes de contrôle d'accès standard ne précisent pas les adresses de destination, placez-les le plus près possible de la destination.

Positionnement des ACL standard

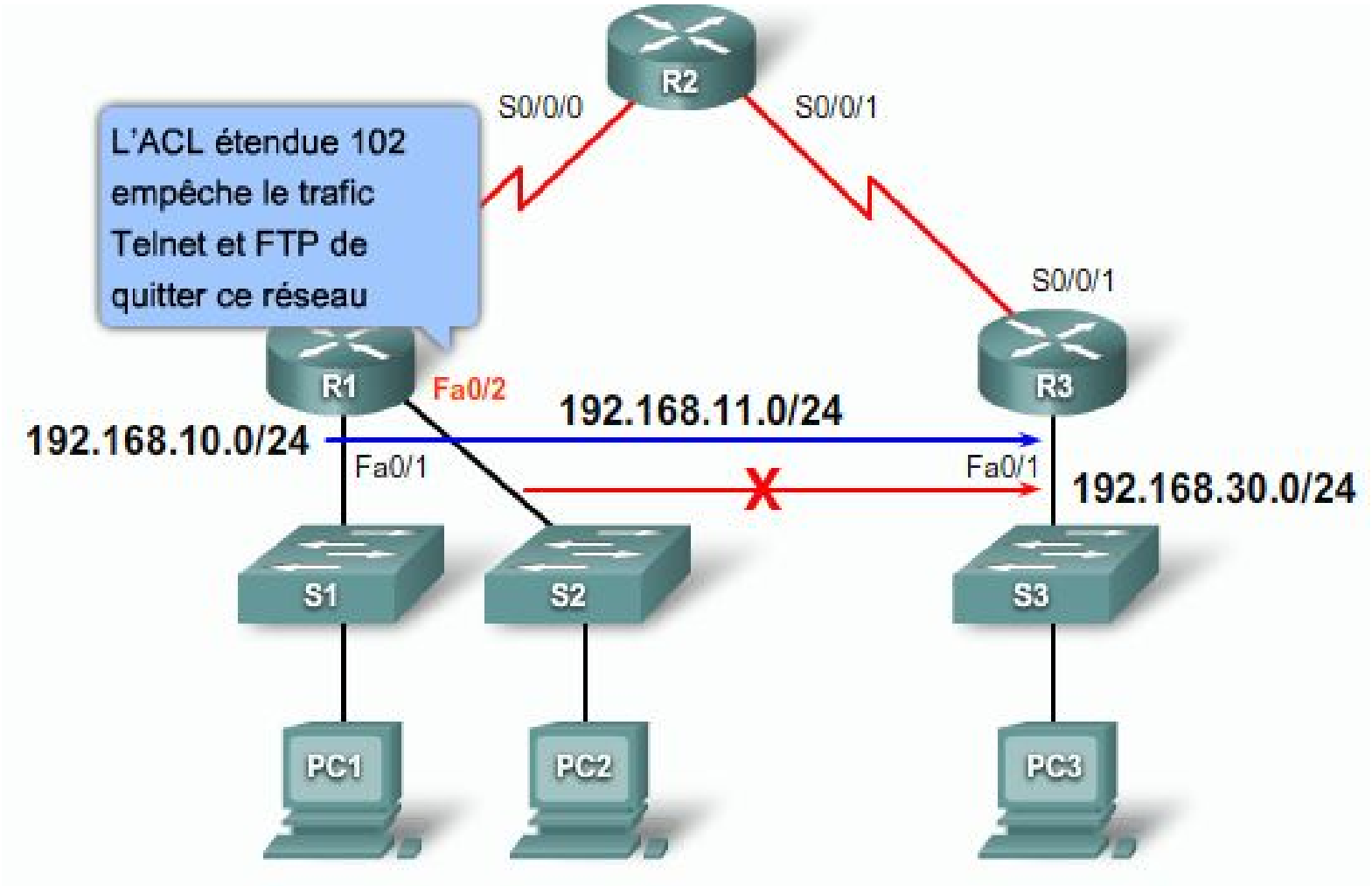
- l'administrateur souhaite empêcher l'accès du trafic provenant du réseau 192.168.10.0/24 au réseau 192.168.30.0/24.
- Une ACL sur l'interface de sortie de R1 l'empêche d'envoyer le trafic vers toute autre destination.
- La solution consiste à placer une ACL standard sur l'interface d'entrée de R3 pour arrêter tout trafic provenant de l'adresse source 192.168.10.0/24.



Positionnement des ACL etendues

- on souhaite refuser l'accès du trafic Telnet et FTP depuis le réseau Onze au réseau 192.168.30.0/24. Les autres types de trafic doivent être autorisés à quitter le réseau Dix.
- Une ACL étendue sur R3 pourrait bloquer Telnet et FTP sur le réseau Onze. Mais cette solution autorise le passage du trafic indésirable sur le réseau, pour le bloquer uniquement une fois arrivé à destination. Cette situation affecte les performances réseau globales.
- on peut utiliser une ACL étendue sortante, qui précise des adresses source et de destination (Dix et Trente respectivement) et qui dit « Le trafic Telnet et FTP du réseau Onze n'est pas autorisé vers le réseau Trente ». Placez cette liste de contrôle d'accès étendue sur le port S0/0/0 sortant de R1. L'inconvénient de ce choix est que le trafic du réseau Dix sera également soumis à un traitement par l'ACL, bien que le trafic Telnet et FTP soit autorisé.
- Le mieux est par conséquent de vous rapprocher de la source et de placer une liste de contrôle d'accès étendue sur l'interface d'entrée Fa0/2 de R1. Ainsi, les paquets du réseau Dix n'accèdent pas à R1. En d'autres termes, ils ne traversent pas le réseau Onze et n'accèdent pas au routeur R2 ou R3.

Positionnement des ACL etendues



Plan

Standard IP Access Lists (1-99)

- Filtrage en se basant sur **l'@ IP source uniquement**
- se placent près de la destination
- syntaxe
  - création de l'ACL :
    Router(config)# *access-list num-list {deny|permit} source [wilcard mask][log]*
  - Associer l'ACL à une interface du routeur :
    Router(config)# *interface [port-du-routeur]*
    Router(config-if)# *ip access-group num-list {in/out}*

Standard IP Access Lists (1-99)

- Le champ source :
  - A.B.C.D : @IP réseau ou sous-réseau (interpretation faite avec le wilcard mask)
  - any : n'importe quel hôte (pour remplacer 0.0.0.0 255.255.255.255
  - Host A.B.C.D : adresse particuliere d'une machine (generalment pour remplacé A.B.C.D 0.0.0.0)
- Le champ Wilcard mask : 32 bits
  - Les bits '0' signifient que les ces positions de bits doivent etre vérifiés (match)
  - Les bits '1' signifient que les ces positions de bits sont ignorés

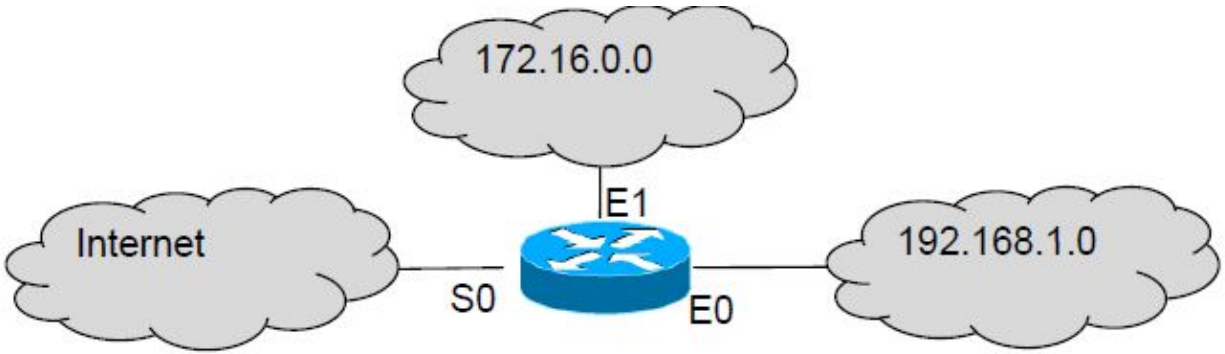
Router(config)# *access-list 14 deny 192.168.16.0 0.0.0.255* (tous les hôtes)
Router(config)# *access-list 14 deny 192.168.16.0 0.0.0.127* (la 1ere moitié des hôtes)
Router(config)# *access-list 14 deny 192.168.16.128 0.0.0.127* (la 2eme moitié des hôtes)

Standard IP Access Lists (1-99)

Exemple :

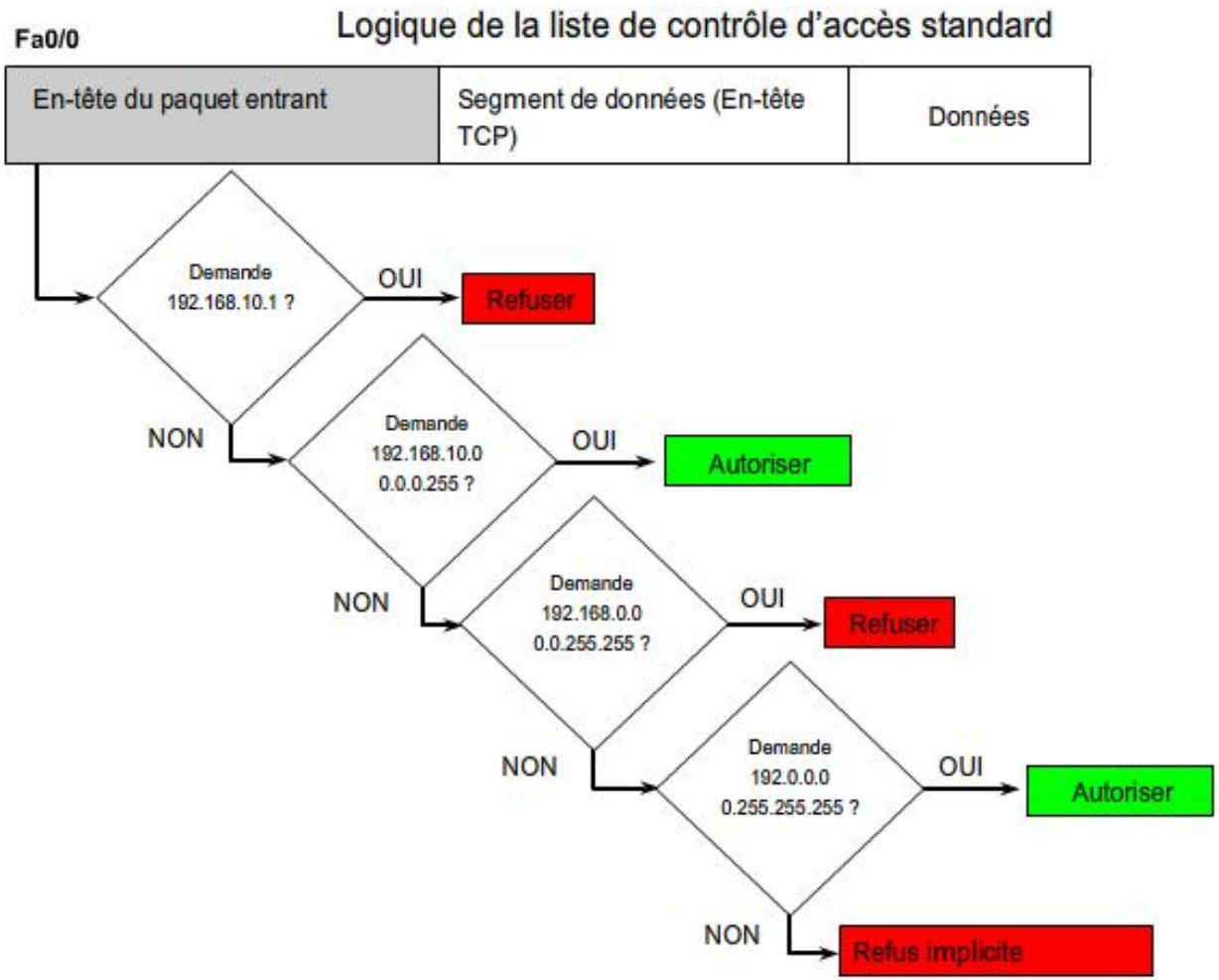
- Permettre l'acheminement du trafic du réseau 192.168.1.0 vers Internet et vers le réseau 172.16.0.0.

Router(config)# *access-list 11 permit 192.168.1.0 0.0.0.255*
Router(config)# *int S0*
Router(config-if)# *ip access-group 11 out*
Router(config)# *int E1*
Router(config-if)# *ip access-group 11 out*



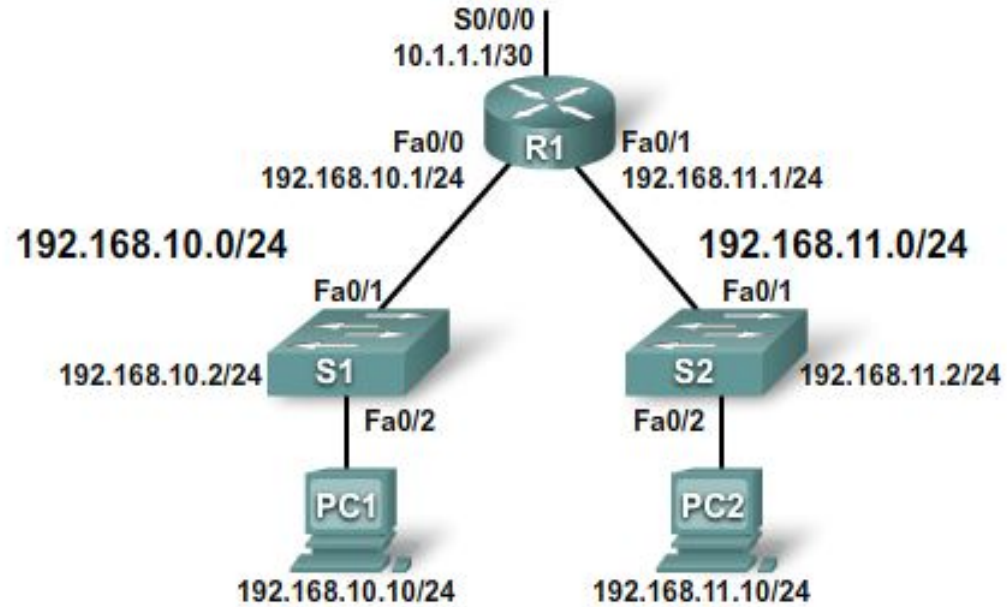
Logique de l'ACL standard

access-list 2 deny 192.168.10.1
access-list 2 permit 192.168.10.0 0.0.0.255
access-list 2 deny 192.168.0.0 0.0.255.255
access-list 2 permit 192.0.0.0 0.255.255.255



exemple 1 : autoriser un réseau et bloquer un autre

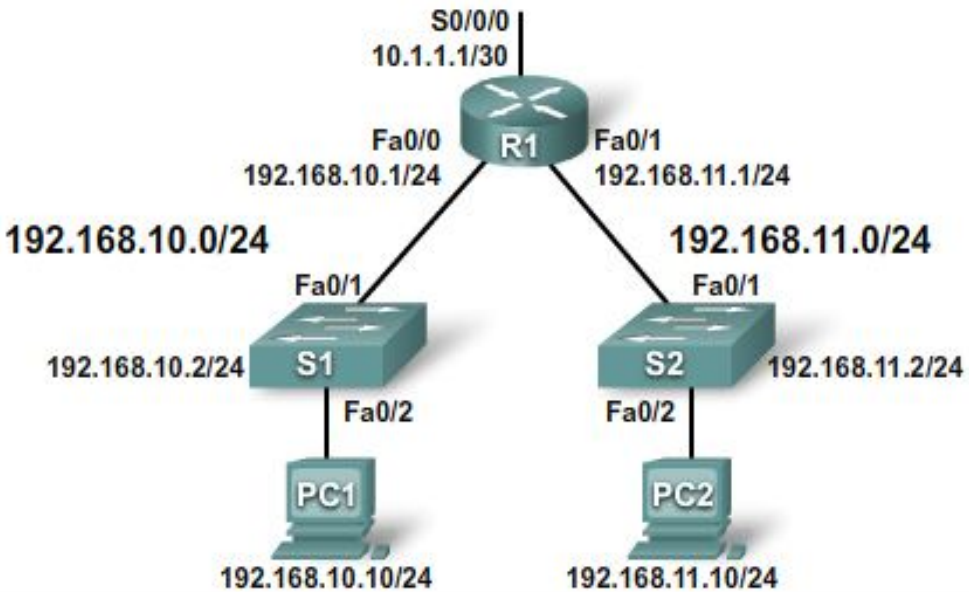
autorise uniquement le transfert du trafic du réseau 192.168.10.0 vers S0/0/0. Le trafic provenant du réseau 192.168.11.0 est bloqué.




```
R1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)# interface S0/0/0
R1(config-if)# ip access-group 1 out
```

exemple 2 : refuser un ensemble de machines

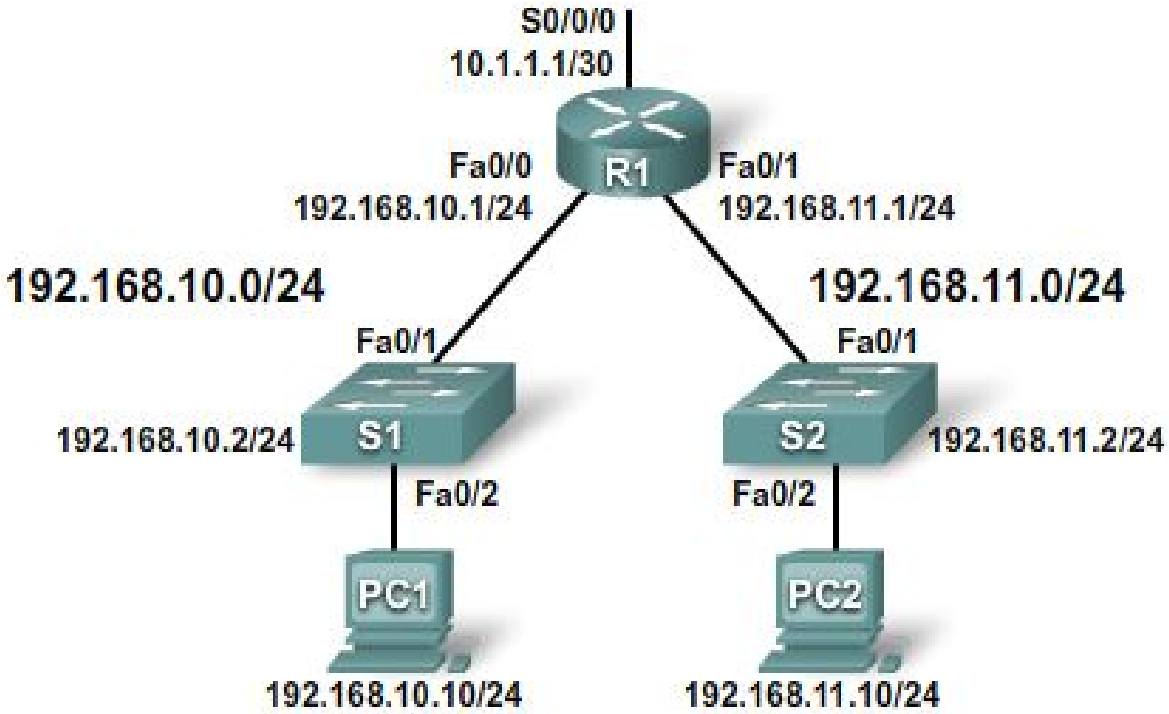
- refuse PC1 et il ya un refus implicite du reseau 192.168.11.0.




```
R1(config)#no access-list 1
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.10.0 0.0.0.255
R1(config)#interface S0/0/0
R1(config-if)#ip access-group 1 out
```

exemple 3 : refuser uniquement une machine

- les deux réseaux locaux attachés au routeur R1 peuvent quitter l'interface S0/0/0 à l'exception du PC1 hôte.




```
R1(config)#no access-list 1
R1(config)#access-list 1 deny 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#interface S0/0/0
R1(config-if)#ip access-group 1 out
```

Plan

Extended IP access list

- Filtrage en se basant sur :
  - @IP source et @IP destination
  - Port source et port destination (filtrage par service)
  - Type de protocole de transport
- Se placent près de la source
- Syntaxe :
  - création de la liste d'accès :
    Router(config)# *access-list num-list-access {deny|permit} protocol source [source-mask] [operator operand] destination [destination-mask] [operator operand] [established]*
  - Associer la liste d'accès à une interface du routeur filtre :
    Router(config)# *interface [port-du-routeur]*
    Router(config-if)# *ip access-group num-list {in/out}*

Le champ "protocol"

- Le champ "protocol" peut avoir plusieurs valeurs :

<0-255>	An IP protocol number
eigrp	Cisco's EIGRP routing protocol
gre	Cisco's GRE tunneling
icmp	Internet Control Message Protocol
igmp	Internet Gateway Message Protocol
igrp	Cisco's IGRP routing protocol
ip	Any Internet Protocol
ipinip	IP in IP tunneling
nos	KA9Q NOS compatible IP over IP tunneling
ospf	OSPF routing protocol
tcp	Transmission Control Protocol
udp	User Datagram Protocol

Exemple :

Router(config)# *access-list 112 permit tcp . . .*

Les champs "source" et "destination"

- Source :
  - A.B.C.D : @IP réseau ou sous-réseau (interpretation faite avec le masque générique)
  - any : n'importe quel hôte (pour remplacer 0.0.0.0 255.255.255.255
  - Host A.B.C.D : adresse particuliere d'une machine (generalment pour remplacé A.B.C.D 0.0.0.0)
  - exemple :
    Router(config)# *access-list 112 permit tcp 192.168.2.1 . . .*
- Destination :
  - A.B.C.D : @IP réseau ou sous-réseau (interpretation faite avec le masque générique)
  - any : n'importe quel hôte (pour remplacer 0.0.0.0 255.255.255.255
  - Host A.B.C.D : adresse particuliere d'une machine (generalment pour remplacé A.B.C.D 0.0.0.0)
  - exemple :
    Router(config)# *access-list 112 permit tcp 192.168.2.1 any . . .*

Le champ "operator"

- Le champ "operator" peut prendre plusieurs valeurs :

eq	Match only packets on a given port number
established	Match established connections
fragments	Check fragments
gt	Match only packets with a greater port number
log	Log matches against this entry
log-input	Log matches against this entry, including input interface
lt	Match only packets with a lower port number
neq	Match only packets not on a given port number
precedence	Match packets with given precedence value
range	Match only packets in the range of port numbers
tos	Match packets with given TOS value

- Exemple :
  Router(config)# *access-list 112 permit tcp 192.168.2.1 any eq . . .*

Le champ "operand"

- Le champ "operand" peut prendre plusieurs valeurs :

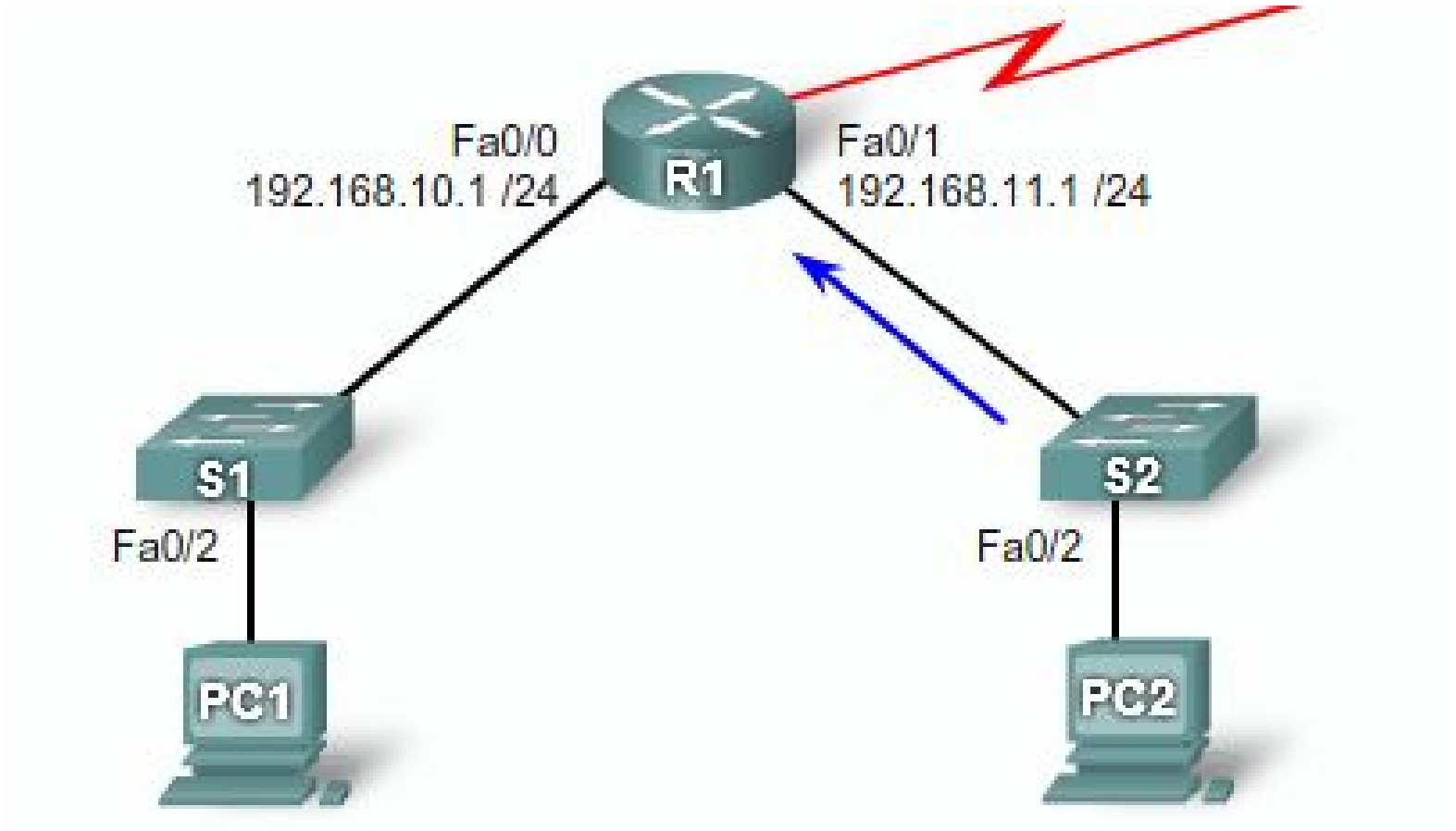
daytime	Daytime (13)
domain	Domain Name Service (53)
echo	Echo (7)
ftp File	Transfer Protocol (21)
irc	Internet Relay Chat (194)
lpd	Printer service (515)
smtp	Simple Mail Transport Protocol (25)
sunrpc	Sun Remote Procedure Call (111)
telnet	Telnet (23)
www	World Wide Web HTTP,80)

.....etc

- Exemple :
Router(config)# *access-list 112 permit tcp 192.168.2.1 any eq 25*

exemple ACL etendu

- Refuser FTP de 192.168.11.0 à destination du sous-réseau 192.168.10.0 et autoriser le reste.
- FTP utilise les ports 21 et 20



Solution

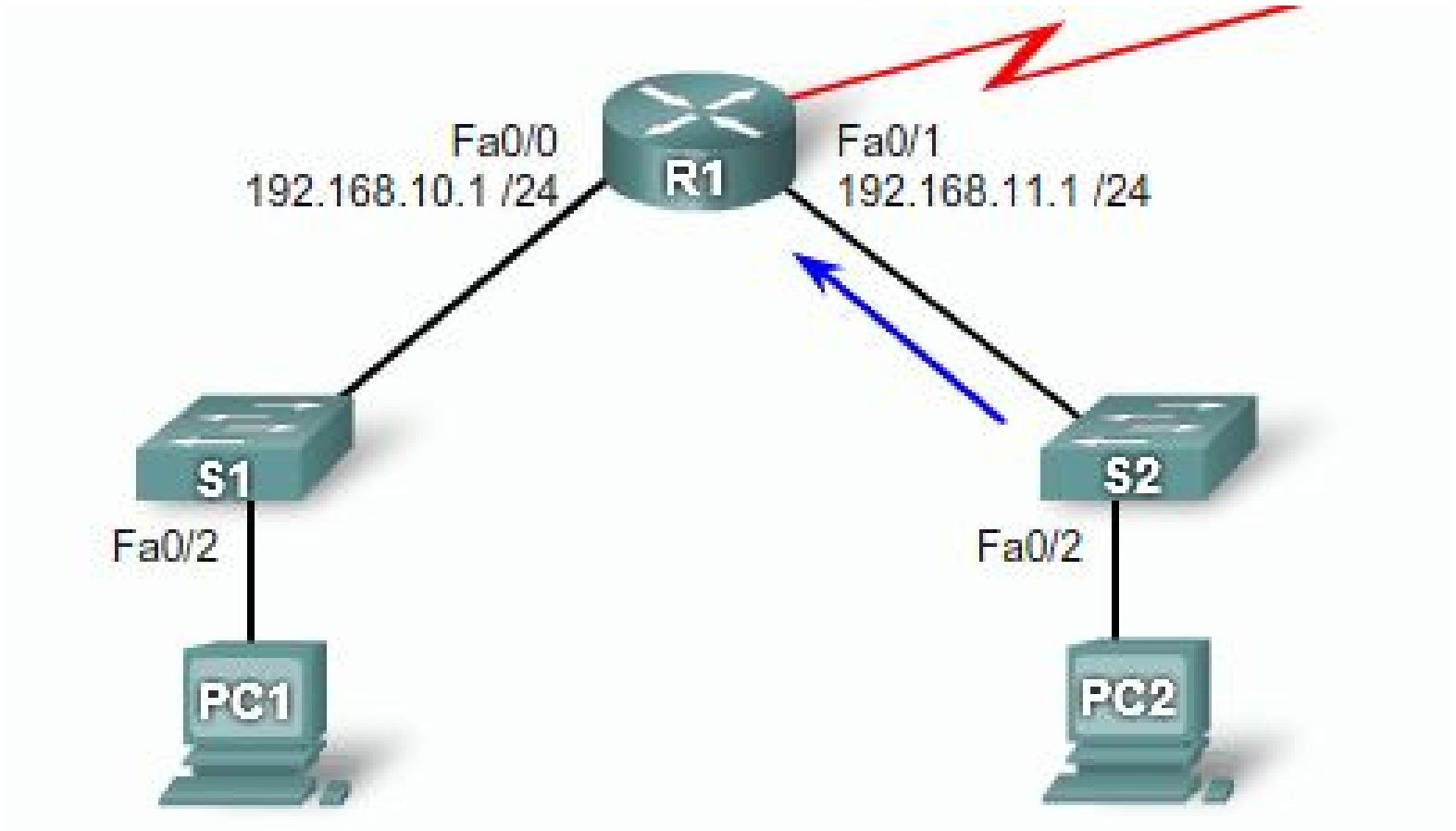
```
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0
0.0.0.255 eq 21
R1(config)# access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0
0.0.0.255 eq 20
R1(config)# access-list 101 permit ip any any
R1(config)# interface Fa0/1
R1(config-if)# ip access-group 101 in
```

- on pourrait faire autrement en utilisant les noms des ports au lieu de leurs numeros :

access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp-data

autre exemple ACL etendu

- Refuser Telnet du 192.168.11.0 à destination du sous-réseau 192.168.10.0 et autoriser le reste.
- Telnet utilise num de port 23



Solution

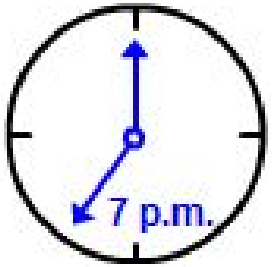
- On peut l'appliquer aussi dans l'interface in dans ce cas (ce qui est mieux)

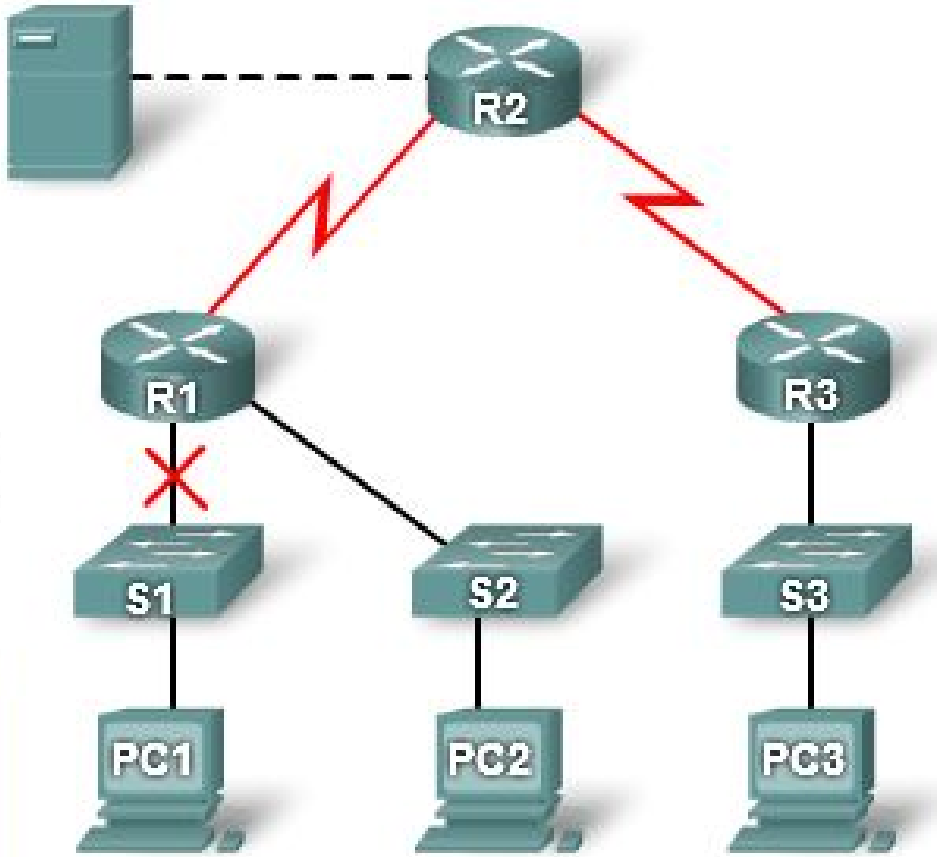
```
R1(config)#access-list 101 deny tcp 192.168.11.0 0.0.0.255  any eq 23
R1(config)#access-list 101 permit ip any any

R1(config)# interface Fa0/0
R1(config-if)#ip access-group 101 out
```

Plan

Liste d'accès basé sur le temps

une connexion Telnet est autorisée depuis le réseau intérieur au réseau extérieur les lundis, mercredis et vendredis pendant les heures ouvrables.





Listes de contrôle d'accès basées sur le temps : autorisation du contrôle d'accès en fonction du jour et de la semaine

- Définir la plage horaire pour implémenter une ACL et appeler-la EVERYOTHERDAY, par exemple.

Étape 1

```
R1(config)#time-range EVERYOTHERDAY
R1(config-time-range)#periodic Monday Wednesday Friday 8:00 to
17:00
```

- Appliquez la plage horaire à la liste de contrôle d'accès.

Étape 2

```
R1(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255
any eq telnet time-range EVERYOTHERDAY
```

- Appliquez la liste de contrôle d'accès à l'interface.

Étape 3

```
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 101 out
```

Plan

Tout d'abord : Masque générique

Donnez l'ensemble des adresses IP concernées par les notations suivantes :

1. 192.168.10.0 0.0.0.255
2. 172.16.0.0 0.0.255.255
3. 10.0.0.0 0.255.255.255
4. 192.168.50.1 0.0.0.254
5. 192.168.0.0 0.0.254.255
6. 192.168.10.61 0.0.0.95

Solution

1. de 192.168.10.0 à 192.168.10.255
2. de 172.16.0.0 à 172.16.255.255
3. de 10.0.0.0 à 10.255.255.255
4. toutes les machines impaires du réseau 192.168.50.0/24
5. 192.168.0.0 à 192.168.0.255
   et 192.168.2.0 à 192.168.2.255
   et 192.168.4.0 à 192.168.4.255
   et . . . . . .
   et 192.168.250.0 à 192.168.250.255
   et 192.168.252.0 à 192.168.252.255
   et 192.168.254.0 à 192.168.254.255
6. 192.168.10.32 à 192.168.10.63
   et 192.168.10.96 à 192.168.10.127

Masque générique chemin inverse

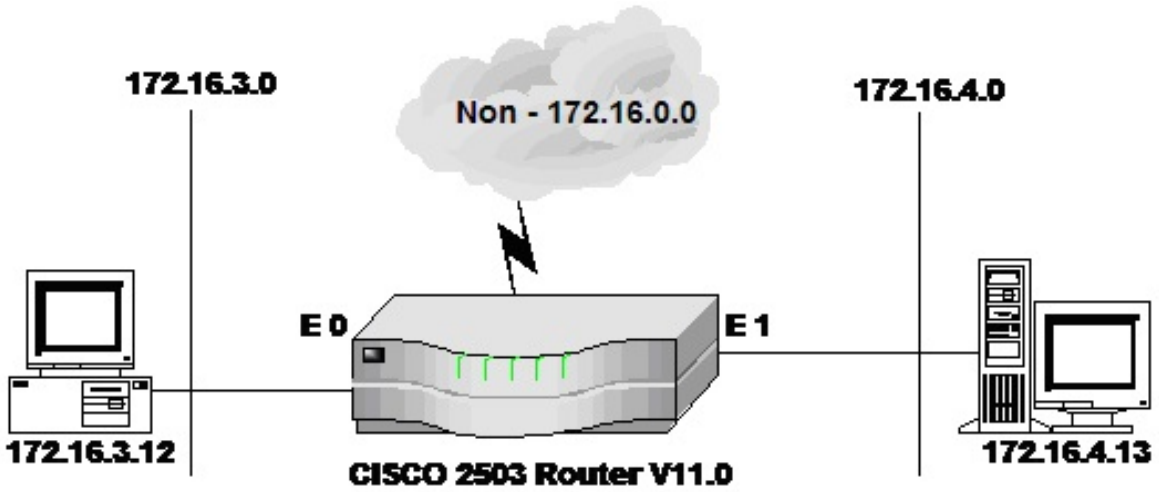
Trouvez les notations "masque générique" qui correspondent aux réseaux suivants :

1. 192.168.16.0 à 192.168.16.127
2. 172.250.16.32 à 172.250.16.63
3. 192.168.10.128 à 192.168.10.159 et 192.168.10.192 à 192.168.10.223

Solution

- 192.168.16.32 0.0.0.127
- 172.250.16.32 0.0.0.31
- 192.168.10.128 0.0.0.95

Exemple 2




```
Router(config)# access-list 1 deny 172.16.4.13 0.0.0.0
Router(config)# interface ethernet 0
Router(config)# ip access-group 1 out
```

- A quoi sert cette ACL ?
- Proposez une modification pour qu'elle produise effectivement l'effet attendu

Solution

- cette ACL est fausse, elle interdit en fait tout le trafic sortant de E0 (c'est-à-dire vers le réseau 172.16.3.0/24. L'idée de l'administrateur était de n'interdire que le trafic qui vient du serveur 172.16.4.13
- il faut ajouter : access-list 1 permit ip any any
- donc l'ACL correcte est la suivante :

```
Router(config)# access-list 1 deny 172.16.4.13 0.0.0.0
Router(config)# access-list 1 permit ip any any
Router(config)# interface ethernet 0
Router(config)# ip access-group 1 out
```

Exemple 3

Meme réseau avec l'ACL suivante

```
Router(config)#access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 21
Router(config)#access-list 101 deny tcp 172.16.4.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 20
Router(config)#access-list 101 permit ip any any
Router(config)#interface ethernet 0
Router(config-if)#access-group 101 out
```

- A quoi sert cette ACL ?
- Pourquoi est-il utile de filtrer le port 21 et le port 20 ?
- Trouvez une ACL standard qui produit le même effet

Solution

- elle empêche les machines du réseau de droite d'utiliser ftp sur le réseau de gauche. Il faut remarquer que cette ACL aurait pu avantageusement être placée en entrée de l'interface E1. (in de E1)
- il faut filtrer les deux ports car l'application ftp les utilise tous les deux (21 : contrôle ; 20 : données)
- impossible, car les ACLs standards ne permettent pas de spécifier un numéro de port

Exemple 4

un autre réseau et une autre ACL :

```
Router(config)#access-list 101 deny tcp 192.168.10.0 0.0.0.255 any neg 80
Router(config)#access-list 101 deny tcp 192.168.10.0 0.0.0.255 any neg 21
Router(config)#access-list 101 permit ip any any
Router(config)#interface ethernet 0
Router(config-if)#access-group 101 in
```

- Quel est l'effet de cette ACL ?
- En devinant l'intention de l'administrateur, proposez une ACL correcte

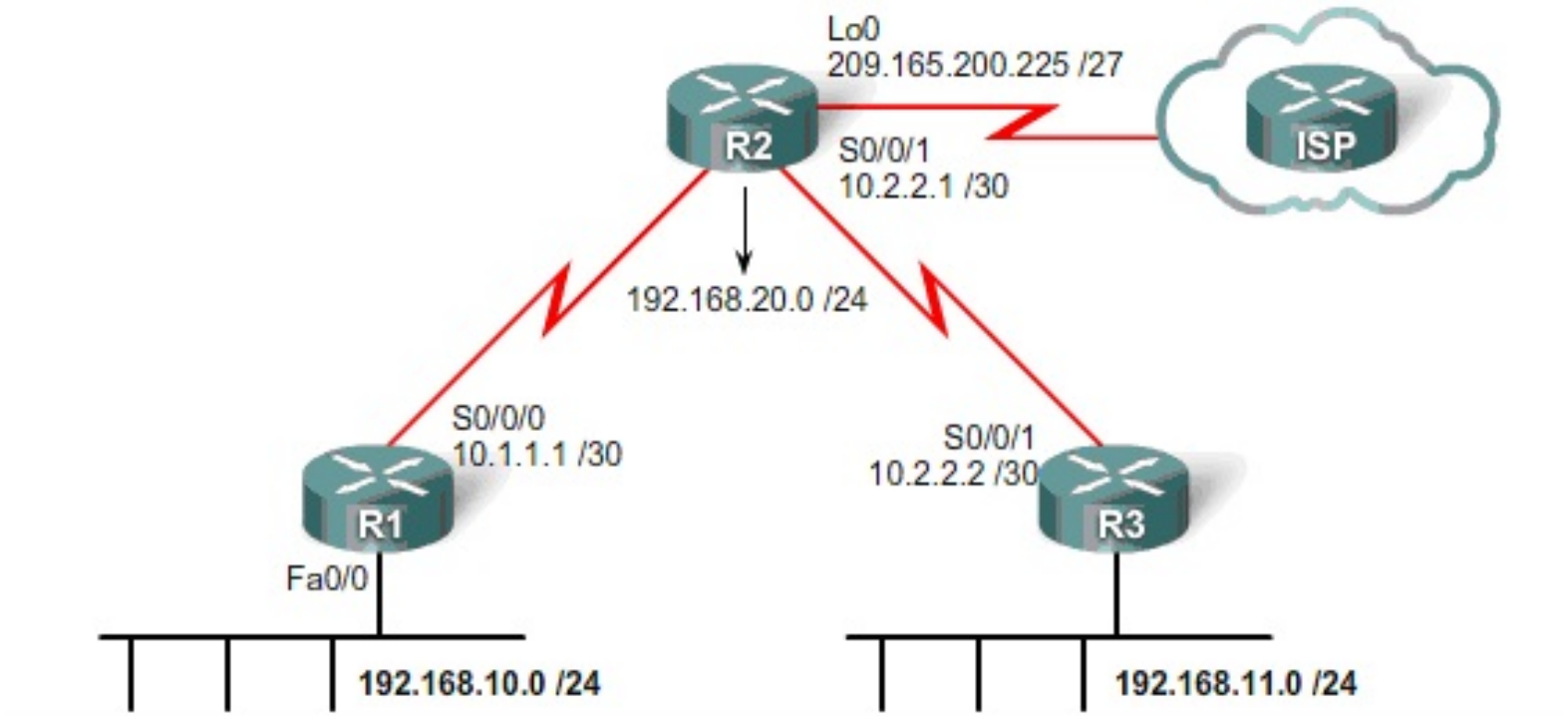
Solution

- elle interdit tout car : si le paquet est à destination du port 21, il est refusé par la première ligne, si le paquet est à destination du port 80, il est refusé par la deuxième ligne. Et on arrivera jamais à la dernière ligne
- On devine que l'administrateur voulait interdire tout, sauf les ports 80 et 21. Il aurait dû écrire :

```
Router(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 80
Router(config)#access-list 101 permit tcp 192.168.10.0 0.0.0.255 any eq 21
Router(config)#access-list 101 deny ip any any
Router(config)#interface ethernet 0
Router(config-if)#access-group 101 in
```

Exemple5

- Autoriser la navigation web http et securisée (https) du réseau 192.168.10.0



Solution

```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
```

- L'ACL 103 s'applique au trafic provenant du réseau 192.168.10.0 ;
- l'ACL 104 s'applique au trafic à destination du réseau 192.168.10.0.
- L'ACL 103 autorise le trafic en provenance de toute adresse sur le réseau 192.168.10.0 à accéder à n'importe quelle destination, à condition que le trafic soit transféré vers les ports 80 (HTTP) et 443 (HTTPS).
- La liste de contrôle d'accès 104 procède ainsi en bloquant tout trafic entrant, à l'exception des connexions établies. HTTP établit des connexions en commençant par la demande initiale avant de passer aux échanges de messages ACK, FIN et SYN.
- Ce paramètre autorise des réponses au trafic provenant du réseau 192.168.10.0 /24 pour un retour entrant sur s0/0/0. Une correspondance survient si les bits ACK ou RST (réinitialisation) du datagramme TCP sont activés, ce qui indique que le paquet appartient à une connexion existante. Sans paramètre established dans l'instruction de la liste de contrôle d'accès, les clients peuvent envoyer le trafic vers un serveur Web mais ils ne peuvent pas le recevoir.

suite solution exemple 5

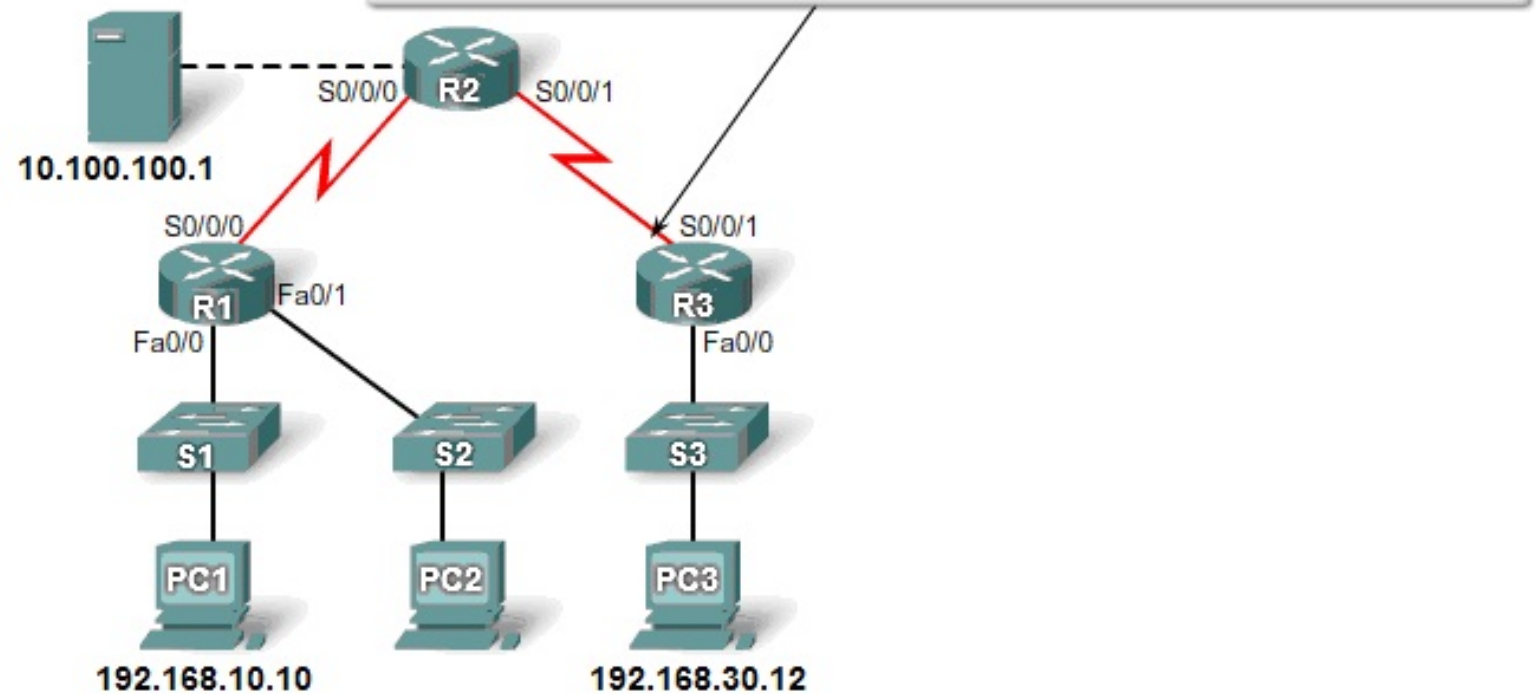
```
R1(config)# interface S0/0/0
R1(config-if)# ip access-group 103 out
R1(config-if)# ip access-group 104 in
```

- le routeur R1 comprend deux interfaces. Il comprend un port série S0/0/0 et un port Fast Ethernet Fa0/0.
- Le trafic Internet entrant accède à l'interface S0/0/0 mais quitte l'interface Fa0/0 à destination de PC1.
- Cet exemple applique la liste de contrôle d'accès à l'interface série dans les deux sens.

Exemple 6

- Analyser le réseau suivant et les erreurs qui découlent de mauvaises configurations des ACLs :
- Erreur : L'hôte 192.168.10.10 n'a établi aucune connectivité avec 192.168.30.12.

```
# show access-lists 110
Extended IP access list 110
    10 deny tcp 192.168.10.0 0.0.0.255 any
    20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
    30 permit ip any any
```

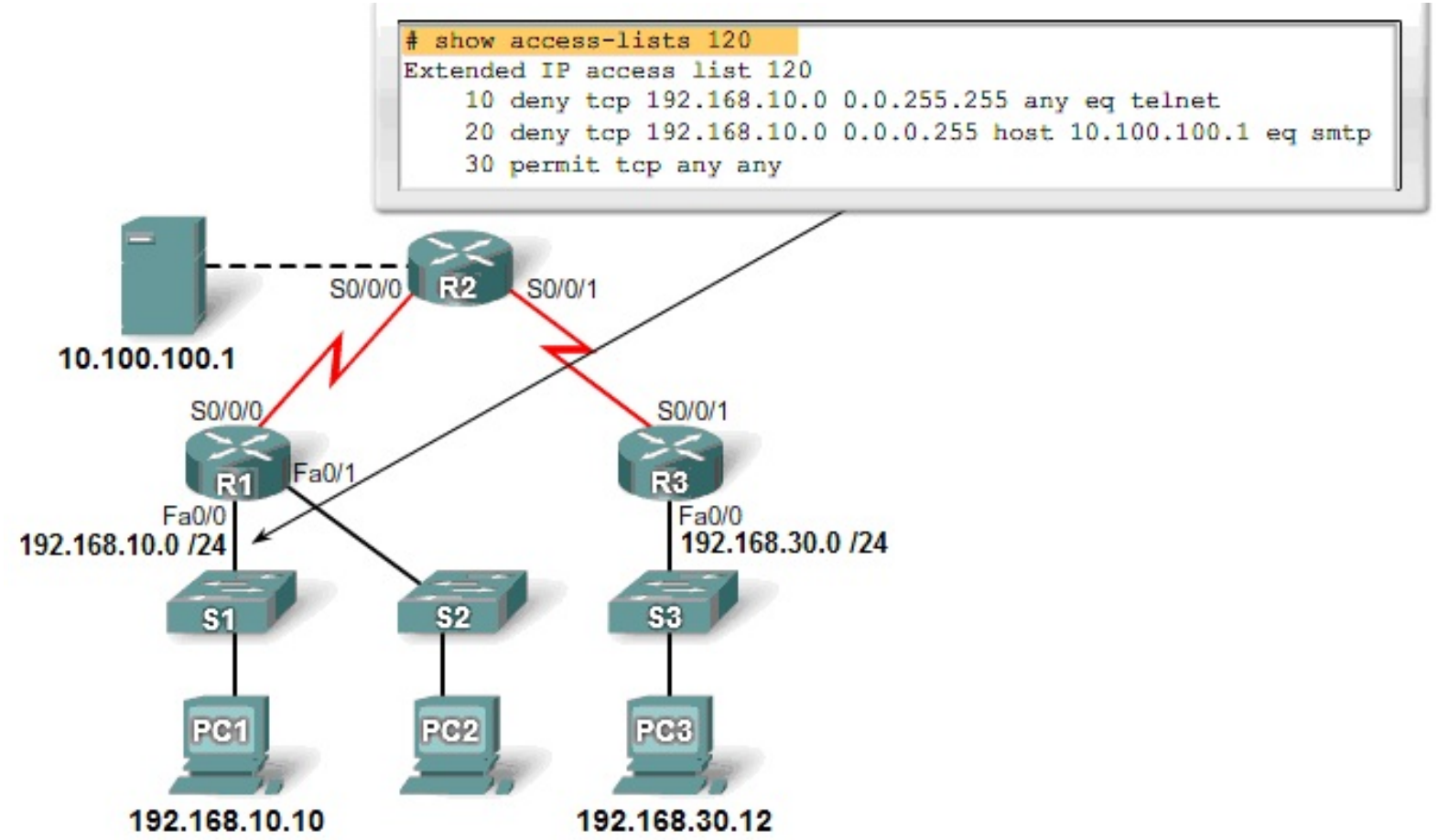


Solution

- Consultez l'ordre des instructions de la liste de contrôle d'accès. L'hôte 192.168.10.10 n'a établi aucune connectivité avec 192.168.30.12 à cause de l'ordre de la règle 10 dans la liste de contrôle d'accès.
- Sachant que le routeur traite les listes de contrôle d'accès de haut en bas, l'instruction 10 refuse l'hôte 192.168.10.10, donc l'instruction 20 n'est pas traitée.
- Les instructions 10 et 20 doivent être inversées.
- La dernière ligne autorise tout autre trafic non TCP correspondant au protocole IP (ICMP, UDP et ainsi de suite).

Exemple 7

- Erreur : Le réseau 192.168.10.0 /24 ne peut pas utiliser TFTP pour se connecter au réseau 192.168.30.0 /24.



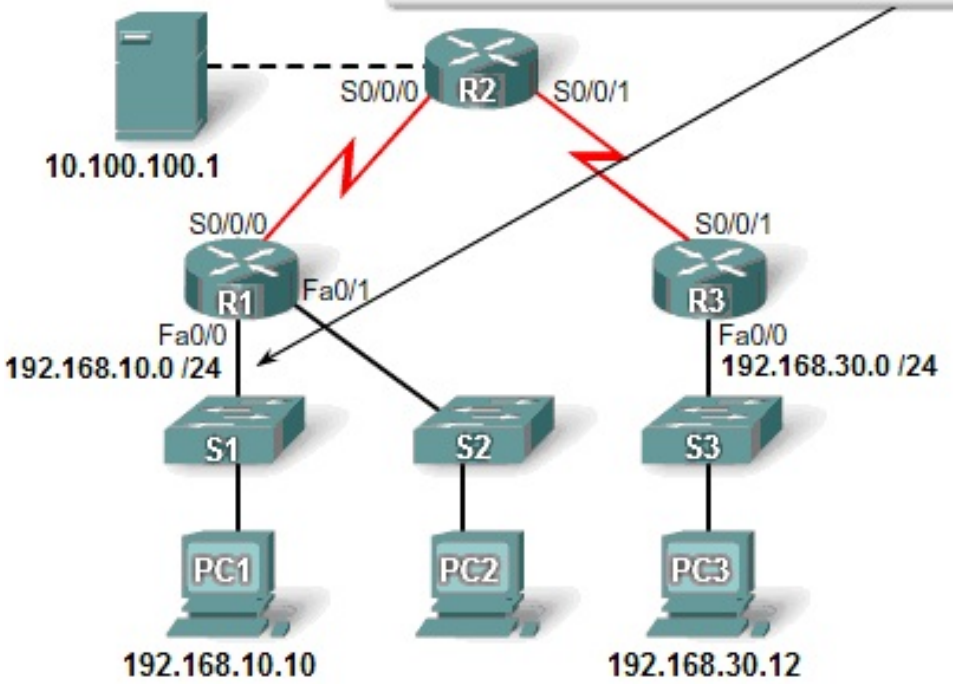
Solution

- Le réseau 192.168.10.0 /24 ne peut pas utiliser TFTP pour se connecter au réseau 192.168.30.0 /24 car TFTP utilise le protocole de transport UDP. L'instruction 30 dans la liste de contrôle d'accès 120 autorise tout autre trafic TCP. Sachant que TFTP utilise UDP, il est refusé implicitement. L'instruction 30 doit être ip any any.
- Cette liste de contrôle d'accès fonctionne qu'elle soit appliquée à Fa0/0 (routeur R1), à S0/0/1 (routeur R3), ou à S0/0/0 (routeur R2) dans le sens entrant. Néanmoins, conformément à la règle voulant que les listes de contrôle d'accès étendues soient placées le plus près de la source, le meilleur emplacement est sur Fa0/0 (routeur R1) car tout trafic indésirable y est filtré sans traverser l'infrastructure réseau.

Exemple 8

- 3) Erreur : Le réseau 192.168.10.0 /24 peut utiliser Telnet pour se connecter à 192.168.30.0 /24 alors que cette connexion doit être interdite.

```
# show access-lists 130
Extended IP access list 130
    10 deny tcp any eq telnet any
    20 deny tcp 192.168.1.0 0.0.0.255 host 192.168.30.0 eq smtp
    30 permit ip any any
```



Solution

- Le réseau 192.168.10.0/24 peut utiliser Telnet pour se connecter au réseau 192.168.30.0/24 car le numéro du port Telnet dans l'instruction 10 de la liste de contrôle d'accès 130 est mal placé. L'instruction 10 refuse actuellement toute source avec un numéro de port égal à Telnet qui essaie d'établir une connexion à une adresse IP.
- Si vous souhaitez refuser le trafic Telnet entrant sur S0, refusez le numéro de port de destination équivalent à Telnet, par exemple **deny tcp any any eq telnet**.