Réseaux et Systèmes (MIN 15112)

2025 - 2026

Zhiyi Zhang

zhiyi.zhang@uvsq.fr

https://www.zhiyizhang.com

Adresse globale unicast

2.5.4. Global Unicast Addresses

The general format for IPv6 Global Unicast addresses is as follows:

```
|         n bits         |   m bits  |       128-n-m bits          |
+------------------------+-----------+-----------------------------+
| global routing prefix  | subnet ID |       interface ID          |
+------------------------+-----------+-----------------------------+
```

where the global routing prefix is a (typically hierarchically-structured) value assigned to a site (a cluster of subnets/links), the subnet ID is an identifier of a link within the site, and the interface ID is as defined in Section 2.5.1.

All Global Unicast addresses other than those that start with binary 000 have a 64-bit interface ID field (i.e., n + m = 64), formatted as described in Section 2.5.1. Global Unicast addresses that start with binary 000 have no such constraint on the size or structure of the interface ID field.

Examples of Global Unicast addresses that start with binary 000 are the IPv6 address with embedded IPv4 addresses described in Section 2.5.5. An example of global addresses starting with a binary value other than 000 (and therefore having a 64-bit interface ID field) can be found in [GLOBAL].

UVSQ

université PARIS-SACLAY

Routage statique

Route statique

Une route statique est une information de routage qui est configurée et maintenue manuellement par l'administrateur.

C'est stable et facile à configurer, mais il ne se met pas à jour automatiquement. Des modifications manuelles sont nécessaires si la topologie change.

Les routes statiques sont adaptées aux petits réseaux et aux situations où les chemins restent relativement stables, comme les routes par défaut.

Pour le routage dynamique -> Semestre 2, option IRS - Réseaux étendus

Route par défaut

La route par défaut est un type particulier de route statique utilisée lorsqu'un routeur ne trouve aucune correspondance plus spécifique dans sa table de routage. Tout trafic destiné à un réseau inconnu est envoyé vers le prochain saut indiqué par cette route.

Router(config)# ip route 0.0.0.0 0.0.0.0 *next-hop*

Elle simplifie la configuration du routage en évitant d'avoir à définir une route pour chaque réseau externe.

- Sur un routeur de sortie, la route par défaut est vers le FAI.
- Sur les équipements domestiques, la route par défaut est vers la box.
- Dans les réseaux d'entreprise, la route par défaut des routeurs de niveau inférieur est vers un routeur de niveau supérieur.

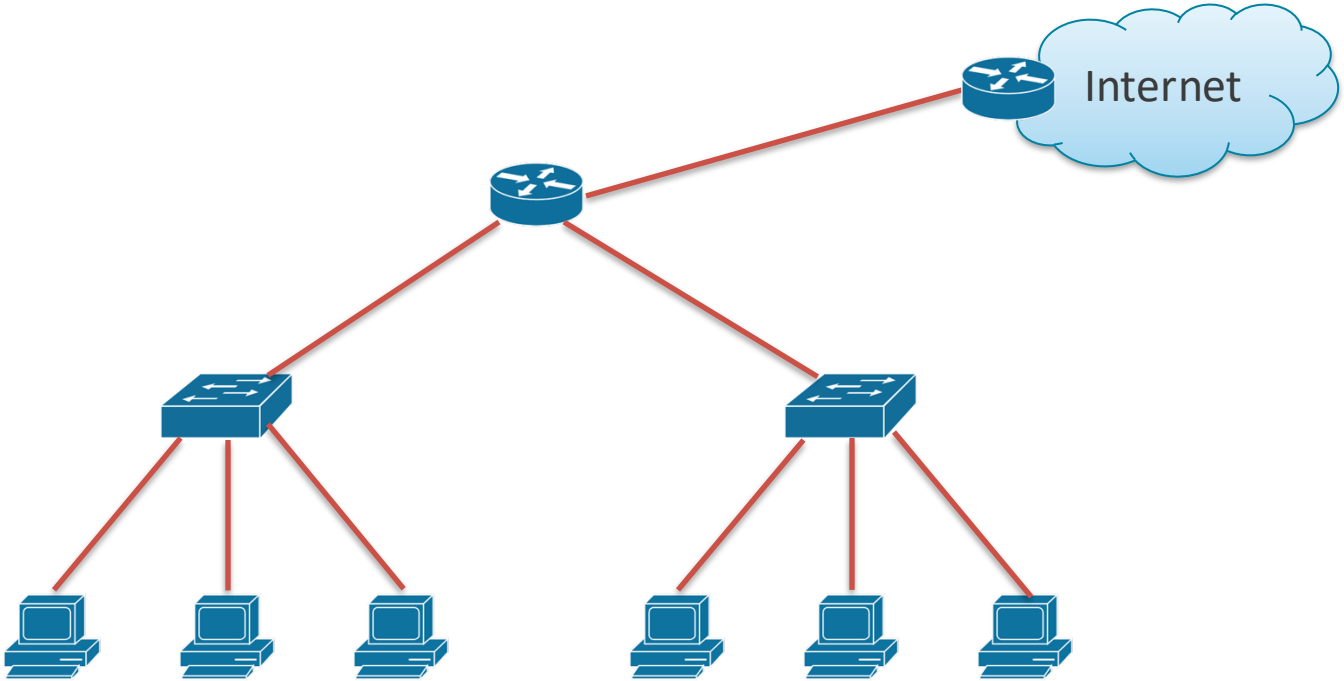
Config route statique

Pour ajouter une route statique :

Router(config)# ip route *@réseau_destination masque next-hop*

En routage, ce qui nous intéresse sont le réseau de destination et le prochain saut. Après avoir transmis le paquet à ce prochain routeur, c'est ce routeur qui doit prendre en charge la suite du chemin.

Exemple

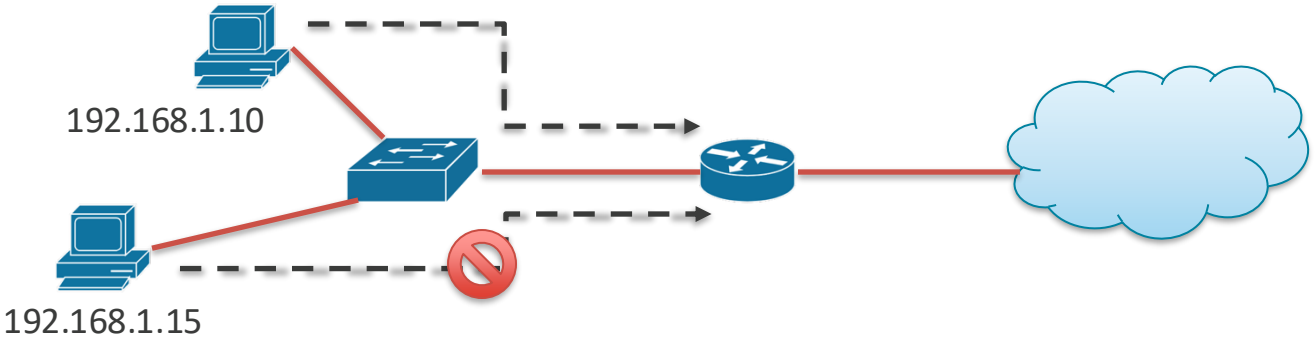


UVSQ

université PARIS-SACLAY

ACL
(Access Control List)

ACL



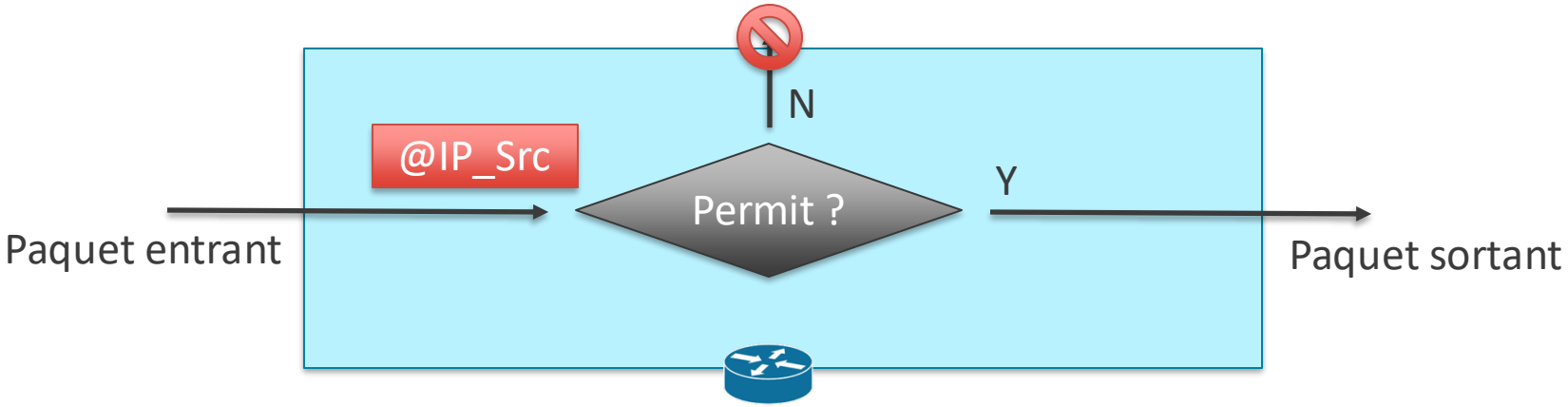
Deux fonctions principales :

- Contrôle du trafic
- Effectuer la correspondance des flux

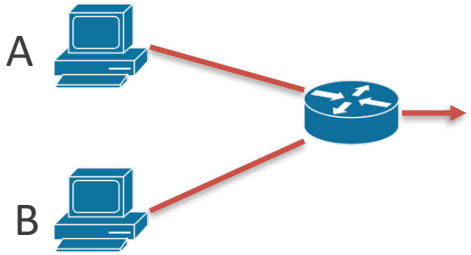
ACL standard

ACL standard :

- Filtrer le trafic selon l'adresse source
- Applique une action (autoriser ou refuser) à l'ensemble du protocole

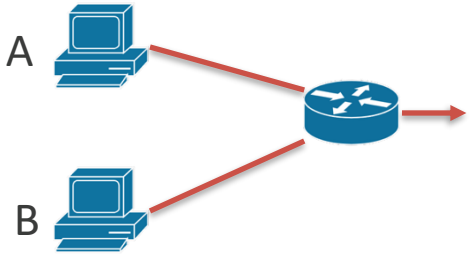


ACL standard

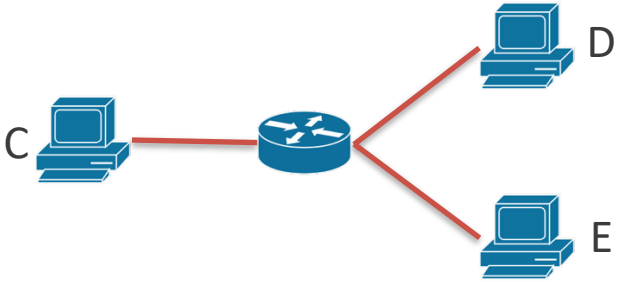


On autorise le trafic provenant de A,
et on bloque le trafic provenant de B.
C'est **faisable** avec ACL standard.

ACL standard



On autorise le trafic provenant de A, et on bloque le trafic provenant de B.
C'est **faisable** avec ACL standard.

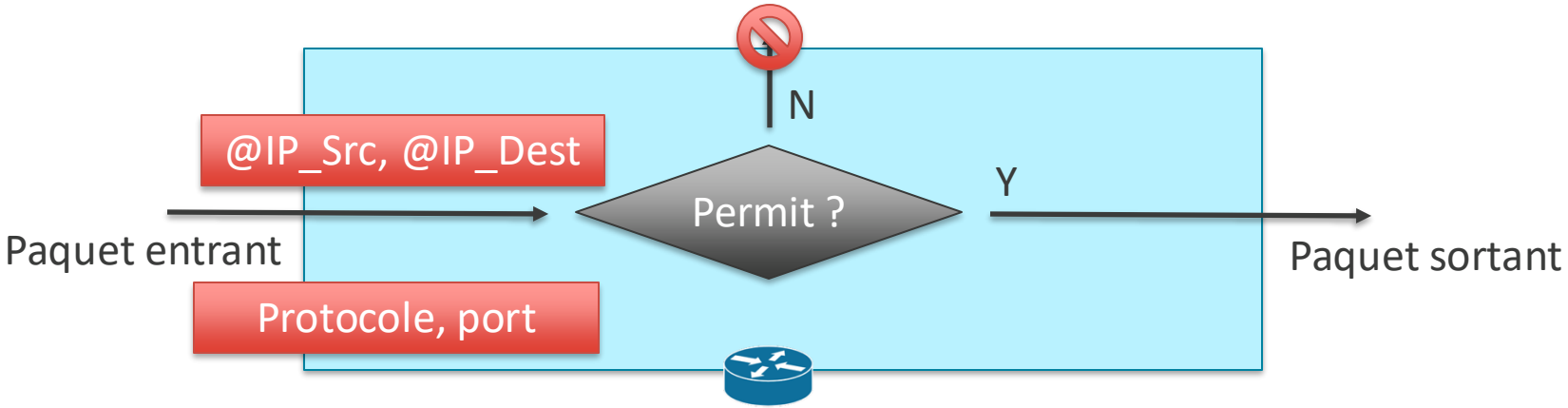


On autorise le trafic de C vers D, mais on bloque le trafic de C vers E.
C'est **pas faisable** avec ACL standard.

ACL étendue

ACL étendue :

- Filtrer le trafic en fonction de l'adresse source, de l'adresse de destination, du numéro de port …
- Permettre/refuser des protocoles spécifiques

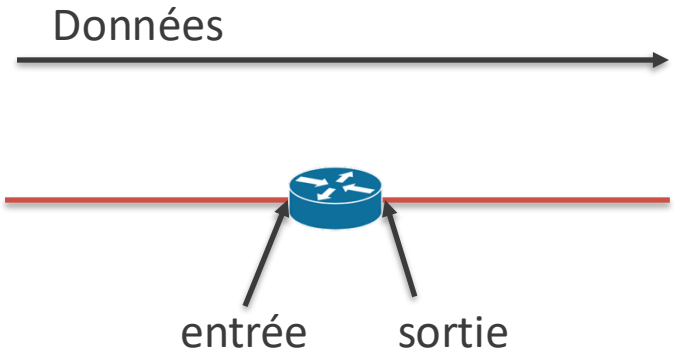


Types ACL

- **ACL standard**
  - Vérifie **uniquement l'adresse source**
  - Autorise/refuse **l'ensemble de la suite de protocoles**
- **ACL étendue**
  - Vérifie **l'adresse source et l'adresse de destination**
  - Autorise/refuse généralement **des protocoles et des applications spécifiques**
- Deux méthodes pour identifier les ACL
  - ACL numérotée : utilise un numéro pour l'identification
  - ACL nommée : utilise un nom descriptif pour l'identification

Opération des ACL

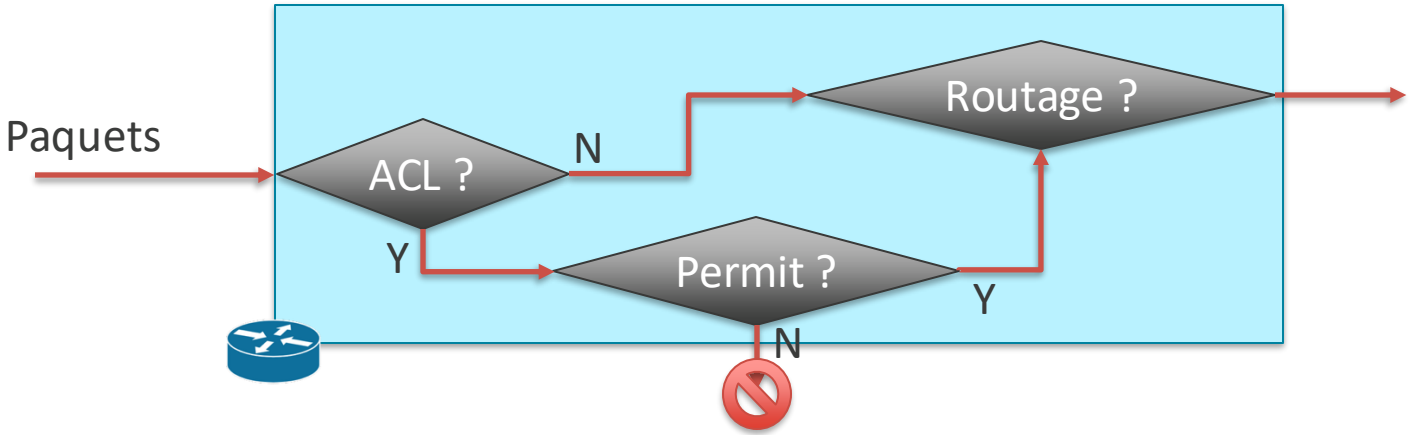
Les ACL s'appliquent généralement sur les interfaces du routeur, en entrée ou en sortie.



ACL en entrée (Inbound ACL)

Une ACL appliquée en entrée filtre les paquets au moment où ils arrivent sur l'interface du routeur.

Plus efficace, sécurité renforcée



ACL en entrée (Inbound ACL)

ACL appliquée sur l'interface d'entrée ?

Non -> le paquet est traité normalement par le routeur : il consulte la table de routage pour déterminer l'interface de sortie.

Oui -> le paquet est d'abord comparé aux règles de l'ACL

ACL en entrée (Inbound ACL)

ACL appliquée sur l'interface d'entrée ?

Non -> le paquet est traité normalement par le routeur : il consulte la table de routage pour déterminer l'interface de sortie.

Oui -> le paquet est d'abord comparé aux règles de l'ACL

S'il existe une entrée correspondante et que l'action est ***permit***, le paquet est autorisé et poursuit les étapes suivantes (consultation de la table de routage …)

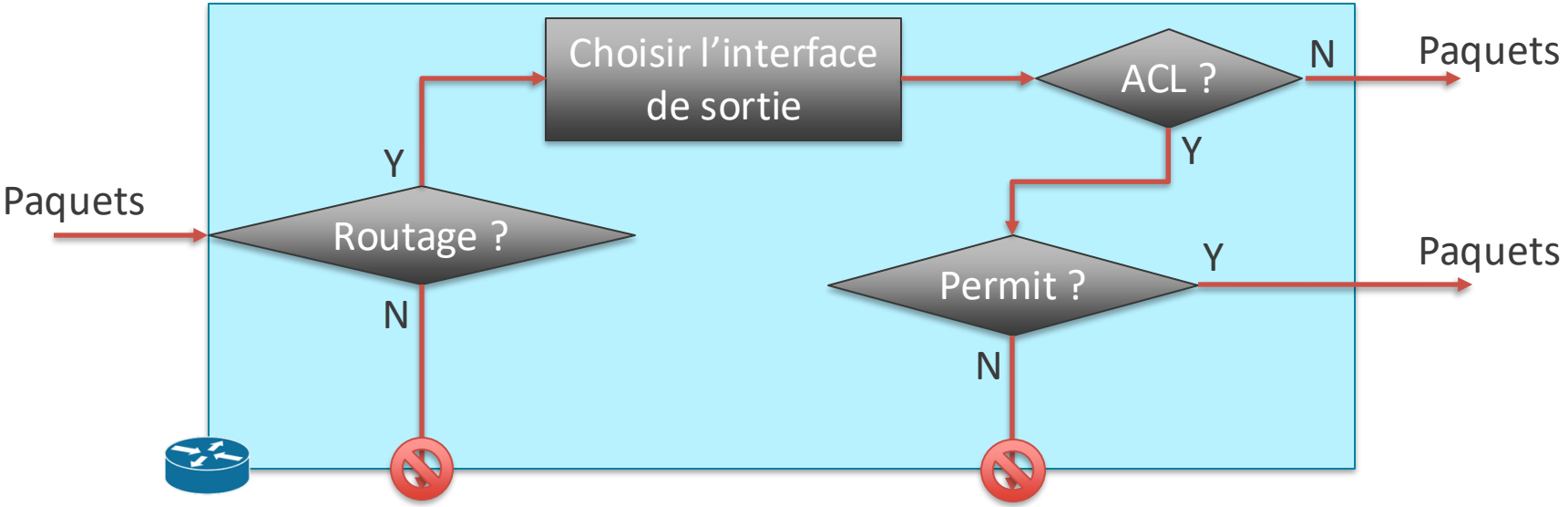
Si l'entrée correspondante indique ***deny***, le paquet est supprimé

Si **aucune entrée correspondante** dans l'ACL, le paquet est également supprimé

Toute ACL se termine implicitement par *deny any*

ACL en sortie (Outbound ACL)

Une ACL en sortie est appliquée au moment où le paquet quitte l'interface du routeur.



ACL en sortie (Outbound ACL)

Pour un paquet destiné à sortir du routeur, le routeur consulte d'abord la table de routage.

S'il n'existe pas d'entrée correspondante, le paquet est supprimé.

S'il existe une route, le routeur vérifie ensuit si une ACL est appliquée sur l'interface de sortie.

ACL en sortie (Outbound ACL)

Pour un paquet destiné à sortir du routeur, le routeur consulte d'abord la table de routage.

S'il n'existe pas d'entrée correspondante, le paquet est supprimé.

S'il existe une route, le routeur vérifie ensuit si une ACL est appliquée sur l'interface de sortie.

S'il n'y a pas d'ACL, le paquet est transmis normalement.

S'il y a une ACL :

Si une entrée correspondante indique ***permit***, le paquet peut sortir;

Si l'entrée correspondante indique ***deny***, le paquet est supprimé;

S'il n'existe **aucune correspondance**, le paquet est supprimé.

Format ACL

access-list *X*

Ligne 1	condition 1	Action (*permit / deny*)
Linge 2	condition 2	Action (*permit / deny*)
Linge 3	condition 3	Action (*permit / deny*)
…		
Ligne N	condition N	Action (*permit / deny*)

Les lignes sont évaluées de haut en bas, dès qu'une condition correspond, l'action est appliquée et le traitement s'arrête.

Si aucune condition correspond, le paquet est rejeté !

Identification ACL

- ACL standard numérotée : **1-99**, 1300-1999
- ACL étendue numérotée : **100-199**, 2000-2699
- ACL nommée (standard ou étendue) : identifie par un nom descriptif

Config ACL standard

- Définir ACL
  Router(config)# access-list *numéro_ACL {permit/deny} source [masque inversé]*
  numéro_ACL : 1-99
  masque inversé

- Appliquer ACL
  Router(config-if)# ip access-group *numéro_ACL {in/out}*
  Application d'une ACL sur une interface
  Direction *in*/*out*
  Attention : pour une interface donnée, seule une ACL peut être appliquée par direction.

Config ACL standard

Il n’est pas possible d’insérer ou de modifier une ligne au milieu d’une ACL numérotée.

Pour changer l’ordre ou le contenu d’une ligne, il faut supprimer toute l’ACL puis la recréer.

Pour supprimer l’ACL :

Router(config)# no access-list *numéro_ACL*

Masque inversé (wildcard mask)

Un masque inversé est un masque utilisé dans les ACL pour indiquer quelles parties d'une adresse IP doivent correspondre exactement et lesquelles peuvent être ignorées.

Règle :

- 0 -> correspondance obligatoire
  Le bit doit être identique à celui de l'adresse spécifiée
- 1 -> valeur indifférente
  Le bit peut être n'importe quoi, il n'est pas vérifié

Masque inversé (wildcard mask)

Le mot-clé ***host*** permet de représenter une adresse unique :

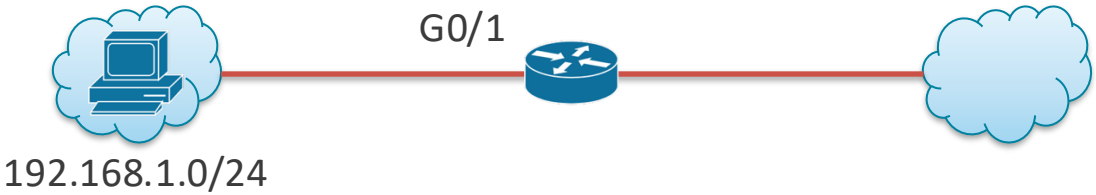
192.168.10.1 0.0.0.0

Cette syntaxe est équivalente à : host 192.168.10.1

Le mot-clé ***any*** correspond à n'importe quelle adresse IP

0.0.0.0 255.255.255.255 -> any

Exemple ACL standard



Interdire 192.168.1.100 d'accéder à Internet, autoriser les autres.

Router(config)# access-list 1 deny host 192.168.1.100

Router(config)# access-list 1 permit any **(très important, sinon on bloque tout !)**

Router(config)# interface g0/1

Router(config-if)# ip access-group 1 in

Comportement ACL

Les ACL ne s'appliquent pas au trafic émis par le routeur lui-même. Elles filtrent uniquement les paquets qui entrent ou sortent d'une interface.

Pourquoi ?

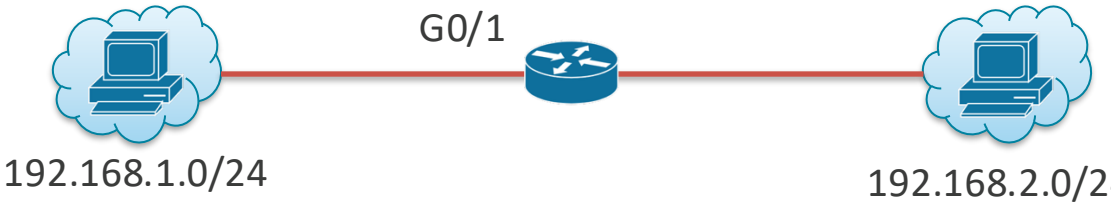
- Les ACL sont conçues pour filtrer le trafic en transit (data plane), pas le trafic local (control plane).
- Filtrer le trafic local risquerait de perturber les fonctions du routeur (protocoles de routage, gestion …)

Config ACL étendue

- Définir ACL
  Router(config)# access-list *numéro_ACL {permit/deny} protocole source source_masque_inversé [operator port] destination destination_masque_inversé [operator port] [established] [log]*
  numéro_ACL : 100-199

- Appliquer ACL
  Router(config-if)# ip access-group *numéro_ACL {in/out}*

Exemple ACL étendue



Interdire 192.168.1.100 d'accéder à 192.168.2.100, autoriser tout le reste.

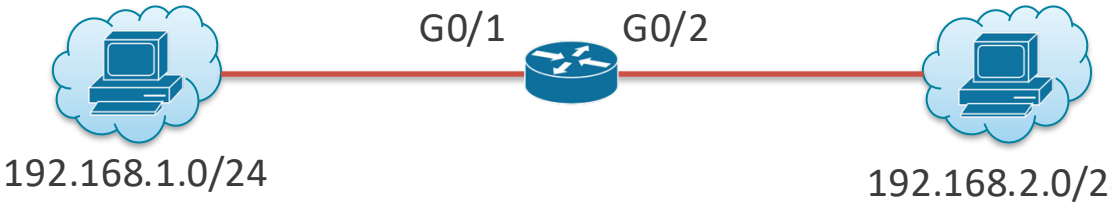
Router(config)# access-list 100 deny ip host 192.168.1.100 host 192.168.2.100

Router(config)# access-list 100 permit ip any any

Router(config)# interface g0/1

Router(config-if)# ip access-group 100 in

Exemple ACL étendue



Interdire l'accès Telnet (TCP 23) au serveur 192.168.2.200, autoriser le reste

Router(config)# access-list 100 deny tcp any host 192.168.2.200 eq 23

Router(config)# access-list 100 permit ip any any

Router(config)# interface g0/2

Router(config-if)# ip access-group 100 out

ACL nommée

- Une fois qu'une ACL numérotée est configurée, elle ne peut pas être modifiée. Il faut supprimer entièrement puis la recréer.
- De plus, lorsque plusieurs ACL numérotées sont utilisées, les numéros ne sont pas très parlants et il devient difficile d'identifier clairement l'objectif de chaque ACL.

Avec ACL nommée :

- Le nom d'une ACL nommée peut être explicite et descriptif.
- On peut modifier, ajouter ou insérer des lignes sans recréer l'ACL.

Config ACL nommée

- Définir ACL
  Router(config)# ip access-list *{standard/extended} nom_ACL*
  Router(config-{std-/ext-}nacl)# [séquence] *{permit/deny} conditions*

- Appliquer ACL
  Router(config-if)# ip access-group *nom_acl {in/out}*

ACL nommée

- Si aucun numéro de séquence n'est spécifié lors de la configuration d'une ACL nommée, les entrées commencent par défaut à 10, et chaque nouvelle ligne est incrémentée de 10.
- Il est possible d'insérer de nouvelles lignes entre les entrées existantes grâce aux numéros de séquence.
- La commande *no séquence* permet de supprimer une ligne spécifique, ce qui rend possible la réécriture ou la modification d'une ligne individuelle.

Conclusion ACL

- Pour chaque interface et pour chaque direction, une seule ACL peut être appliquée.
- Il est essentiel d'organiser correctement l'ordre des règles, une fois qu'une ligne correspond, le traitement s'arrête et les lignes suivantes ne sont pas évaluées.
- Les ACL numérotées ne permettent ni l'insertion ni la modification d'une ligne spécifique, toute modification nécessite la suppression complète de l'ACL.
- Une ACL se termine toujours par *deny any* implicite, il faut s'assurer qu'au moins une règle *permit* soit présente, sinon l'ACL n'aura aucune utilité.
- Après avoir créé une ACL, il ne faut pas oublier de l'appliquer sur l'interface.
- Les ACL ne filtrent que les paquets qui transitent à travers le routeur, elles ne s'appliquent pas au trafic généré localement par le routeur lui-même.

UVSQ

université PARIS-SACLAY

NAT
(Network Address Translation)

Cmd NAT statique

- Créer la correspondance entre une adresse privée et une adresse publique
  Router(config)# ip nat inside source static *@ip_local @ip_globale*

- Définir l'interface interne et externe
  Router(config-if)# ip nat *{inside/outside}*

- Afficher les traductions NAT actives
  Router# show ip nat translations

Cmd NAT dynamique

- Définir un pool d'adresses globales qui seront attribuées
  Router(config)# ip nat pool *nom @ip_début @ip_fin {netmask masque/prefix-length longueur_préfixe}*
- Créer une ACL permettant les adresses privées internes devant être traduites
  Router(config)# access-list *numéro_ACL* permit *@ip masque_inversé*
- Définir l'interface interne et externe
  Router(config-if)# ip nat *{inside/outside}*
- Établir la traduction dynamique de l'adresse source en utilisant l'ACL définie précédemment et le pool d'adresses globales
  Router(config)# ip nat inside source list *numéro_ACL* pool *nom*

Cmd PAT

- Créer une ACL permettant les adresses privées internes devant être traduites
  Router(config)# access-list *numéro_ACL* permit *@ip masque_inversé*

- Définir l'interface interne et externe
  Router(config-if)# ip nat *{inside/outside}*

- Activer le PAT en utilisant l'adresse de l'interface externe
  Router(config)# ip nat inside source list *numéro_ACL* interface *nom_interface* overload

UVSQ

université PARIS-SACLAY

BPDU Guard

Qu’est-ce que BPDU Guard ?

Rappel : Les BPDUs sont des messages STP échangés entre les commutateurs pour détecter les boucles réseau et configurer une topologie sans boucle.

BPDU Guard est une fonctionnalité de sécurité qui désactive un port lorsqu’il reçoit des BPDUs.

Lorsque BPDU Guard est activé sur un port, ce port est automatiquement shutdown (err-disabled) s’il reçoit un BPDU.

Pourquoi utiliser BPDU Guard ?

- Protection des ports PortFast
  Les ports configurés avec PortFast ne doivent jamais recevoir de BPDUs, car ils sont censés être connectés à des machines, et non à d'autres commutateurs. BPDU Guard assure que ces ports sont désactivés s'ils reçoivent des BPDUs.

- Sécurité
  BPDU Guard empêche les commutateurs non autorisés de participer au STP.

- Stabilité du réseau
  En désactivant immédiatement les ports problématiques, BPDU Guard contribue à maintenir une topologie réseau stable et sans boucle.

Config BPDU Guard

BPDU Guard peut être configuré globalement pour tous les ports PortFast, ou individuellement pour des ports spécifiques.

- Configuration globale
  Switch(config)# spanning-tree portfast bpduguard default

- Configuration sur un port spécifique
  Switch(config-if)# spanning-tree bpduguard enable

Récupération des ports désactivés

Lorsqu'un port est shutdown (err-disabled) en raison de BPDU Guard, il peut être réactivé manuellement ou automatiquement.

- Manuellement
  Mettre le port en shutdown, puis le réactiver avec *no shutdown*

- Automatiquement
  Switch(config)# errdisable recovery cause bpduguard
  Switch(config)# errdisable recovery interval *nb_seconds*

UVSQ

université PARIS-SACLAY

DHCP Snooping

DHCP snooping

Le DHCP snooping est une fonctionnalité de sécurité du protocole DHCP, qui permet de garantir que les clients obtiennent leur adresse IP auprès d'un serveur DHCP légitime.

Dans le pratique, le protocole DHCP présente plusieurs vulnérabilités de sécurité, telles que :

- L'usurpation d'un serveur DHCP
- Les attaques

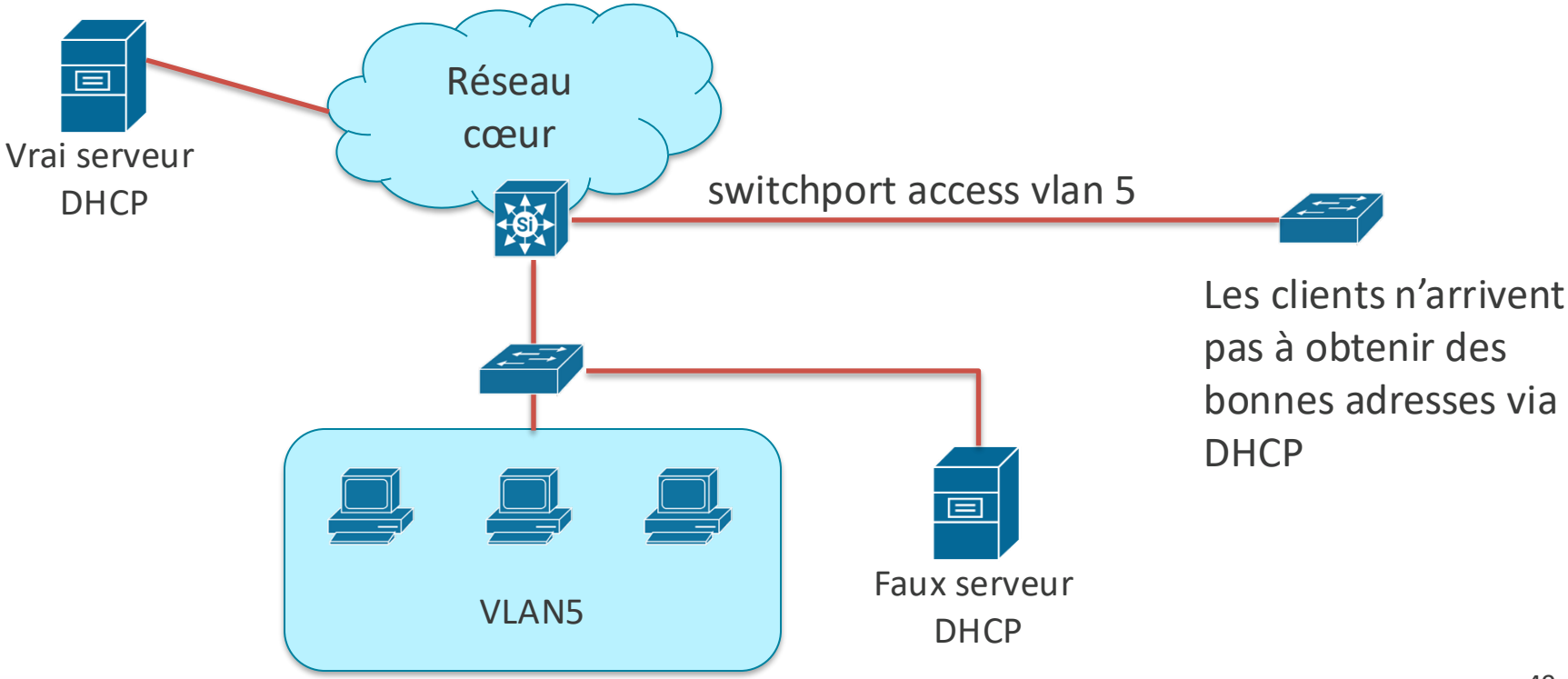
Le DHCP snooping renforce la sécurité principalement grâce à la notion de ports de confiance (trusted/untrusted).

Fonction de confiance

La fonction de confiance de DHCP snooping consiste à classifier les ports du commutateur comme fiables ou non fiables pour accepter ou bloquer les messages provenant d'un serveur DHCP.

Cela permet de contrôler qui a le droit d'envoyer des réponses DHCP (Offer/Ack) dans le réseau.

Scénario


Réseau
cœur
Vrai serveur
DHCP
switchport access vlan 5
Les clients n'arrivent pas à obtenir des bonnes adresses via DHCP
VLAN5
Faux serveur
DHCP

Port fiable (Trusted Port)

Un port fiable est un port sur lequel le commutateur autorise les messages DHCP provenant d'un serveur DHCP légitime :

- DHCP Offer
- DHCP Ack
- DHCP Nak

Les ports fiables sont généralement :

- Les ports connectés au vrai serveur DHCP
- Les liaisons montantes vers le vrai serveur DHCP

Port non fiable (Untrusted Port)

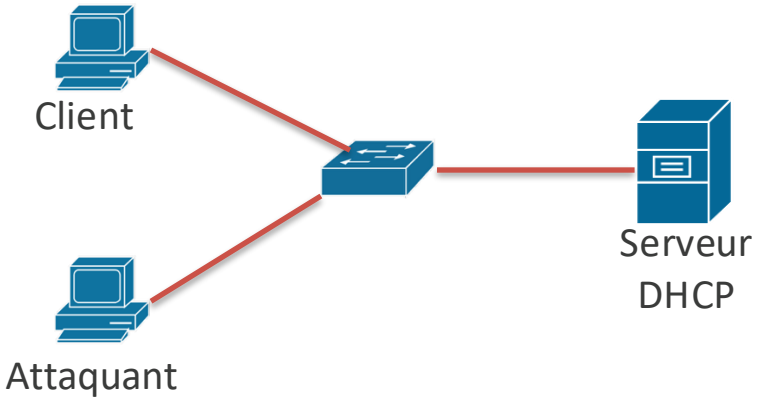
Un port non fiable bloque automatiquement toute réponse DHCP.

Si un DHCP Offer arrive d'un port non fiable, il est immédiatement rejeté.

Les ports non fiables sont :

- Les ports d'accès connectés aux PC, imprimantes …
- Les zones où un utilisateur pourrait brancher un faux serveur DHCP

Scénario attaque



L'attaquant envoie rapidement un grand nombre de messages DHCP Discover en utilisant des adresses MAC différentes.

- Le serveur DHCP attribue toutes les adresses disponibles à de faux clients.
- Le pool DHCP se retrouve épuisé.
- Les clients légitimes ne peuvent plus obtenir d'adresse IP.

Scénario attaque

- On peut limiter le taux de paquets DHCP.
  Switch(config-if)# ip dhcp snooping limit rate *nb_pkt/s*

- Si un port dépasse la limite, on désactive le port.

- On peut aussi limiter le nombre d'adresses MAC autorisée par port. Si le port dépasse la limite, le port sera shutdown.
  Switch(config-if)# switchport port-security maximum *nb_@MAC*
  Switch(config-if)# switchport port-security violation shutdown

UVSQ

université PARIS-SACLAY

IEEE 802.1X

IEEE 802.1X

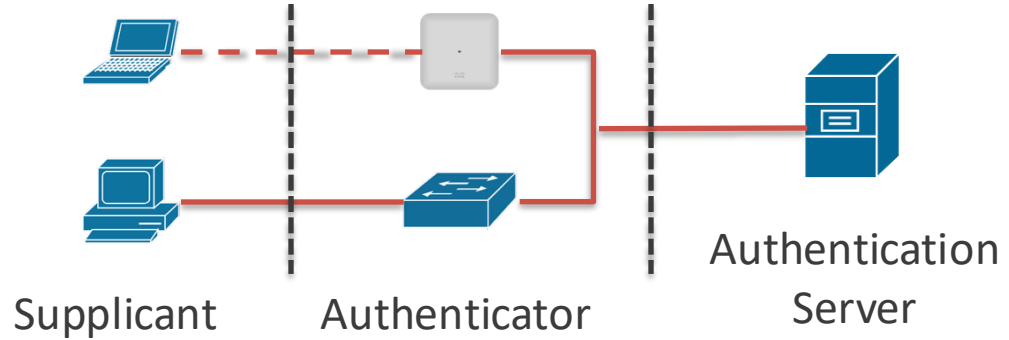
- Utiliser l'authentification pour contrôler l'accès au port, limiter l'accès physique au réseau
- Un protocole de couche 2 permettant l'encapsulation de l'EAP (Extensible Authentication Protocol), appelé EAPoL (Extensible Authentication Protocol over LAN)
- Un protocole de couche 2 permettant de transporter les messages d'authentification (EAP) entre le supplicant (utilisateur) et l'authentificateur (commutateur/point d'accès).

Rôles 802.1X

Trois rôles dans 802.1X

Avec l'authentification basée sur les ports IEEE 802.1X, les équipements du réseau ont des rôles :

- Client (Supplicant)
- Commutateur ou point d'accès (Authenticator)
- Serveur d'authentification



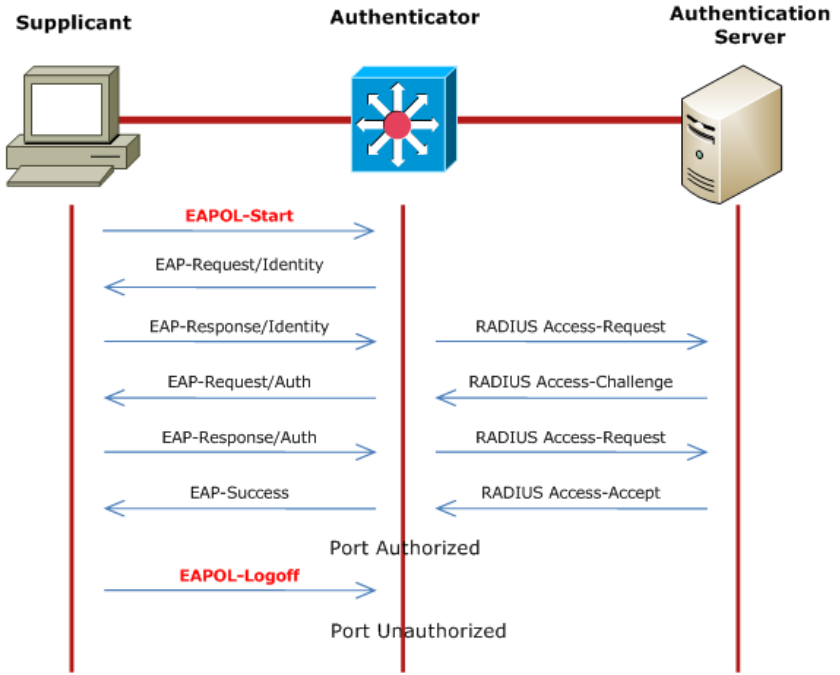
EAP

EAP (Extensible Authentication Protocol)

RFC 3748

- Protocole flexible, capable de transporter différents types d'informations d'authentification.
- Ne dépend pas de la couche de liaison et peut être utilisé en réseau filaire / sans fil.
- Il fonctionne directement au-dessus de la couche de liaison.
- Dans l'Ethernet, EAP est transporté par EAPoL (EAP over LAN).

Procédure 802.1X



La session d'authentification se déroule entre le client et le serveur d'authentification. L'authenticator (commutateur/AP) peut observer qu'un processus d'authentification est en cours, mais il ne fait qu'agir comme un relai.

L'authenticator doit pouvoir joindre le serveur d'authentification.

Résumé (Cours 9)

- ACL
  Standard
  Extended
- ACL & NAT
- BPDU Guard
- DHCP Snooping
- IEEE 802.1X