10/10/2024

Bloc2\_sem3-4\_atelier-02\_Les ACLs

**Document de l’atelier :**

* [Bloc2\_sem3-4\_atelier-02\_activite-1.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-1.pkt)
* [Bloc2\_sem3-4\_atelier-02\_activite-2.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-2.pkt)
* [Bloc2\_sem3-4\_atelier-02\_activite-3-Decouverte\_AccessListsCisco-01.pka](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-3-Decouverte_AccessListsCisco-01.pka)
* [Bloc2\_sem3-4\_atelier-02\_activite-4-decouverte\_acl-fin.pka](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-4-decouverte_acl-fin.pka)
* [Bloc2\_sem3-4\_atelier-02\_activite-5\_access\_list\_reflexives.gns3](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-5_access_list_reflexives/Bloc2_sem3-4_atelier-02_activite-5_access_list_reflexives.gns3)
* [Bloc2\_sem3-4\_atelier-02\_activite-6-ACL-DMZ.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-6-ACL-DMZ.pkt)
* [Bloc2\_sem3-4\_atelier-02\_activite-7-acl-vlan.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-7-acl-vlan.pkt)
* [Bloc2\_sem3-4\_atelier-02\_activite-8-zpf.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-8-zpf.pkt)
* [Bloc2\_sem3-4\_atelier-02\_Standard\_ACL\_physique.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_Standard_ACL_physique.pkt)

Table des matières

[1 Introduction : 5](#_Toc179750561)

[2 ACL sur les fichiers 6](#_Toc179750562)

[2.1 Sous UNIX 6](#_Toc179750563)

[2.2 Windows 6](#_Toc179750564)

[3 En réseau 6](#_Toc179750565)

[3.1 Introduction 6](#_Toc179750566)

[3.2 Iptables 6](#_Toc179750567)

[3.2.1 iptables s’utilise en « root » 6](#_Toc179750568)

[3.2.2 iptables -L : afficher les règles 7](#_Toc179750569)

[3.2.3 Le principe des règles 7](#_Toc179750570)

[3.2.4 Ajouter et supprimer des règles 7](#_Toc179750571)

[3.2.5 Autoriser les pings 8](#_Toc179750572)

[3.2.6 Autoriser les connexions locales et déjà ouvertes 8](#_Toc179750573)

[3.2.7 Refuser toutes les autres connexions par défaut 8](#_Toc179750574)

[3.2.8 Appliquer les règles au démarrage 9](#_Toc179750575)

[3.3 Les ACLs et le matériel Cisco 9](#_Toc179750576)

[3.4 Positionnement des access-lists 9](#_Toc179750577)

[3.4.1 Utilité 10](#_Toc179750578)

[3.4.2 Logique d’application 10](#_Toc179750579)

[3.4.3 Caractéristiques 11](#_Toc179750580)

[3.4.4 Traitement 11](#_Toc179750581)

[3.4.5 Différence entre liste d’accès standard et liste d’accès étendue 11](#_Toc179750582)

[3.4.6 Désignation d’une liste d’accès 11](#_Toc179750583)

[3.4.7 Le masque générique (Wildcard Mask) 11](#_Toc179750584)

[3.4.8 Règles d’application 12](#_Toc179750585)

[3.4.9 Syntaxe des commandes 12](#_Toc179750586)

[3.4.10 Liste d’accès standard 12](#_Toc179750587)

[3.4.11 Liste d’accès étendue 13](#_Toc179750588)

[3.4.12 Liste d’accès nommée 13](#_Toc179750589)

[3.4.13 Activation d’une liste d’accès sur une interface 13](#_Toc179750590)

[3.4.14 Diagnostic 13](#_Toc179750591)

[4 Énoncé 1 : Comment configurer une Acl standard ? 13](#_Toc179750592)

[5 Énoncé 2 : ACL étendue (Extended ACL) 17](#_Toc179750593)

[6 Énoncé 3 : exercice récapitulatif et progressif afin de bien comprendre 21](#_Toc179750594)

[6.1 Présentation du scénario 21](#_Toc179750595)

[6.2 Remarques 22](#_Toc179750596)

[6.3 Organisation des instructions (dans le fichier Packet Tracer) 22](#_Toc179750597)

[6.3.1 Scénario 1 : isolation de l’entreprise BLUE 22](#_Toc179750598)

[6.3.2 Mise en œuvre du filtrage (solution 1) 22](#_Toc179750599)

[6.3.3 Mise en œuvre du filtrage (solution 2) 25](#_Toc179750600)

[6.4 Scénario 2 : accès distinctif entre les réseaux Cobalt et CIEL 27](#_Toc179750601)

[6.4.1 Cahier des charges 27](#_Toc179750602)

[6.4.2 Mise en œuvre du filtrage 27](#_Toc179750603)

[6.4.3 Résultat obtenu/explication 28](#_Toc179750604)

[6.5 Scénario 3 : accès restreint pour le réseau COBALT 32](#_Toc179750605)

[6.5.1 Cahier des charges 32](#_Toc179750606)

[6.5.2 Mise en œuvre du filtrage 32](#_Toc179750607)

[6.6 Scénario 4 : blacklister un PC sur www.emeraude.com 35](#_Toc179750608)

[6.6.1 Cahier des charges 35](#_Toc179750609)

[6.6.2 Mise en œuvre du filtrage 35](#_Toc179750610)

[6.6.3 Mise en œuvre effective 35](#_Toc179750611)

[6.7 Scénario 5 : accès restrictif à certains services sur le réseau ORANGE 37](#_Toc179750612)

[6.7.1 Cahier des charges 37](#_Toc179750613)

[6.7.2 Vérifications avant mise en place du filtrage 37](#_Toc179750614)

[6.7.3 Mise en œuvre du filtrage 39](#_Toc179750615)

[6.7.4 Vérifications après mise en place du filtrage 40](#_Toc179750616)

[6.8 Scénario 6 : modification d’acces-list pour ajouter de nouvelles règles 42](#_Toc179750617)

[6.8.1 Cahier des charges 42](#_Toc179750618)

[6.8.2 Mise en œuvre du filtrage 42](#_Toc179750619)

[6.8.3 Test des nouvelles règles 45](#_Toc179750620)

[6.9 Scénario 7 : autoriser les Pings sortants, mais refuser les pings entrants 47](#_Toc179750621)

[6.9.1 Cahier des charges 47](#_Toc179750622)

[6.9.2 Mise en œuvre du filtrage 47](#_Toc179750623)

[6.9.3 Vérification du résultat obtenu 48](#_Toc179750624)

[7 Énoncé 4 : découverte approfondie des ACLs 49](#_Toc179750625)

[7.1 Objectifs 50](#_Toc179750626)

[7.2 Contexte/Préparation 50](#_Toc179750627)

[7.3 Étape 1 : vérification de la connectivité du réseau 50](#_Toc179750628)

[7.4 Étape 2 : examen des listes de contrôle d’accès configurées sur les routeurs 51](#_Toc179750629)

[7.5 Étape 4 : examen des effets d’une liste de contrôle d’accès 52](#_Toc179750630)

[7.6 Etape 5 : Protocol-discovery et filtrage du traffic Peer-to-Peer 54](#_Toc179750631)

[7.6.1 Activer la découverte de protocol sur l’interface où l’on veut appliquer le filtrage 54](#_Toc179750632)

[7.6.2 Créer une « class-map » de sorte à identifier le traffic généré par les applications P2P 54](#_Toc179750633)

[7.6.3 Définition de la police à appliquer sur l’interface 55](#_Toc179750634)

[7.6.4 Appliquer la police à l’interface 55](#_Toc179750635)

[7.6.5 Remarques 55](#_Toc179750636)

[8 Les access-list réflexives 56](#_Toc179750637)

[8.1 La topologie 56](#_Toc179750638)

[8.2 Objectif 57](#_Toc179750639)

[8.3 Méthode générale 57](#_Toc179750640)

[8.4 Considération dans le cas présent 57](#_Toc179750641)

[8.5 Configuration 57](#_Toc179750642)

[8.6 Vérification 58](#_Toc179750643)

[8.7 Quelques remarques 60](#_Toc179750644)

[9 Énoncé 9 : ACL et DMZ 61](#_Toc179750645)

[9.1 But 61](#_Toc179750646)

[9.2 Travail à réaliser 61](#_Toc179750647)

[9.2.1 Le réseau 61](#_Toc179750648)

[9.2.2 Routage et DNS 61](#_Toc179750649)

[9.3 Translation d’adresses 63](#_Toc179750650)

[9.4 Protection du réseau de l’entreprise. 64](#_Toc179750651)

[9.5 Protection de la DMZ 66](#_Toc179750652)

[9.6 Conclusion 67](#_Toc179750653)

[10 Énoncé 10 : VLAN Access-List (VACL) 68](#_Toc179750654)

[10.1 La topologie 68](#_Toc179750655)

[10.2 Configuration initiale du switch 69](#_Toc179750656)

[10.3 Objectif de la configuration 70](#_Toc179750657)

[10.4 Configuration de la VACL 70](#_Toc179750658)

[10.4.1 Étape n° 1 : Créer une ACL (classique) pour identifier le trafic à traiter 71](#_Toc179750659)

[10.4.2 Étape n° 2 : Créer la VACL pour associer des « match » à des « action » 71](#_Toc179750660)

[10.4.3 Étape n° 3 : Appliquer la VACL au(x) VLAN(s) désiré(s) 72](#_Toc179750661)

[10.5 Test de la configuration 72](#_Toc179750662)

[11 Enoncé 11 : Mise en place d'une Zone-based Policy Firewall (ZPF) - défi 73](#_Toc179750663)

[11.1 Qu’est-ce que ZPF ? 73](#_Toc179750664)

[11.1.1 La notion de "ZONE" 73](#_Toc179750665)

[11.1.2 Les actions de ZPF 74](#_Toc179750666)

[11.2 Configurer ZPF 74](#_Toc179750667)

[11.2.1 Création des zones 75](#_Toc179750668)

[11.2.2 Création de la Class-Map 75](#_Toc179750669)

[11.2.3 Création de la policy-map 76](#_Toc179750670)

[11.2.4 Création de la zone paire 76](#_Toc179750671)

[11.2.5 Affectation des interfaces à une zone 76](#_Toc179750672)

[12 Enoncé 12 Bonus : Mise en place Standard\_ACL\_physique 77](#_Toc179750673)

[13 Conclusion : 79](#_Toc179750674)

[14 Webographie : 80](#_Toc179750675)

# Introduction :

Les listes de contrôle d'accès (ACL, pour Access Control Lists) jouent un rôle essentiel dans la sécurité informatique. Elles sont utilisées dans deux domaines principaux :

* **Gestion des droits d'accès aux fichiers** : Une ACL permet une gestion fine des permissions d'accès aux fichiers, offrant une flexibilité supérieure aux méthodes d'attribution des droits d'accès traditionnellement utilisées par les systèmes UNIX.
* **Sécurité réseau** : Dans un contexte réseau, les ACL servent à définir les règles permettant de contrôler l'accès aux ressources en fonction d'adresses IP ou de ports spécifiques, souvent utilisées dans la configuration de pare-feu.

Les ACL se composent d'entrées appelées ACE (Access Control Entries), qui attribuent ou révoquent des permissions à des utilisateurs ou des groupes. Ce document abordera, dans un premier temps, l'utilisation des ACL dans la gestion des fichiers, en présentant les implémentations sous UNIX et Windows. Ensuite, nous explorerons leur rôle dans la gestion de la sécurité des réseaux, en passant en revue des outils tels que **iptables** et les équipements Cisco.

# ACL sur les fichiers

## Sous UNIX

Sous UNIX, les ACL complètent la méthode traditionnelle de gestion des droits, conforme à la norme POSIX 1, sans la remplacer. Traditionnellement, UNIX offre trois types de droits (lecture, écriture, exécution) pour trois types d'utilisateurs : le propriétaire, le groupe associé, et les autres utilisateurs. Cependant, cette méthode est limitée, notamment dans les environnements d'entreprise où il est souvent nécessaire d'attribuer des droits spécifiques à plusieurs groupes ou utilisateurs. Les ACL permettent de contourner cette limite en offrant une gestion plus fine des permissions sans être restreinte au nombre d'utilisateurs. Mac OS X prend en charge les ACL depuis la version 10.4. Les commandes principales pour gérer les ACL sous UNIX sont getfacl pour visualiser et setfacl pour modifier les permissions, après activation de l'option "acl" dans le fichier /etc/fstab.

## Windows

Sous Windows, les ACL sont gérées par le système de fichiers NTFS. Elles peuvent être appliquées aux fichiers et répertoires avec divers types de droits, tels que : parcours de dossier, lecture des métadonnées, ajout de fichiers ou de données, modification des droits, suppression, lecture, appropriation, et exécution. Ces droits peuvent être attribués à des utilisateurs (réels, virtuels ou systèmes) ou à des groupes d'utilisateurs, via des ACE (Access Control Entries). Il est possible de configurer l'héritage des ACL entre répertoires, ainsi que le type d'héritage (sur les dossiers, fichiers, etc.).

# En réseau

## Introduction

Une ACL sur un pare-feu ou un routeur est une liste d'adresses ou de ports autorisés ou bloqués par le dispositif de filtrage. Il existe trois types principaux d'ACL : l'ACL standard qui contrôle l'adresse IP source et partiellement celle de destination via un masque générique ; l'ACL étendue qui permet de contrôler des éléments supplémentaires tels que l'adresse IP de destination, les protocoles (TCP, UDP, etc.), les ports source et destination, ainsi que les flux et priorités IP ; et l'ACL nommée-étendue, qui est une ACL étendue à laquelle un nom est attribué. Sous Linux, Netfilter gère les ACL, et la commande iptables est utilisée pour les configurer.

## Iptables

Pour sécuriser un réseau, il est possible d'utiliser iptables afin de bloquer des ports spécifiques, comme le port 21 utilisé par FTP, si ce protocole est jugé non sécurisé. Cependant, la méthode recommandée consiste généralement à bloquer par défaut tous les ports, puis à n'autoriser que ceux qui sont nécessaires, afin de minimiser les risques d'accès non autorisé.

### iptables s’utilise en « root »

Pour utiliser iptables, il est nécessaire d'avoir les privilèges du superutilisateur (« root »). Il est donc recommandé de passer en mode superutilisateur en exécutant la commande suivante :

$ sudo su

### iptables -L : afficher les règles

Avec la commande iptables -L (avec un "L" majuscule), vous pouvez afficher les règles actuelles du pare-feu. Cette commande montre trois chaînes principales :

* **Chain INPUT** : gère le trafic entrant.
* **Chain FORWARD** : gère la redirection du trafic (non abordée ici).
* **Chain OUTPUT** : gère le trafic sortant.

Dans l'exemple donné, les règles sont vides et la politique par défaut est "ACCEPT", ce qui signifie que tout le trafic est autorisé. Le pare-feu est donc inactif dans cet état. Pour réinitialiser toutes les règles, on utilise la commande iptables -F, mais il faut faire attention, car cela efface toutes les règles existantes.

### Le principe des règles

Lorsqu'on définit des règles avec iptables, l'ordre des règles est crucial, car iptables les lit de haut en bas. Par exemple, dans la section INPUT, la commande iptables -L peut afficher des règles telles que :

Chain INPUT (policy DROP)

ACCEPT tcp -- anywhere anywhere tcp dpt:www

ACCEPT tcp -- anywhere anywhere tcp dpt:ssh

ACCEPT tcp -- anywhere anywhere tcp dpt:imap2

Chaque ligne représente une règle qui filtre un port ou une IP spécifique. L'option --line-numbers permet d'afficher les numéros des règles. Voici quelques informations importantes sur les colonnes :

* **target** : l'action de la règle (ici, "ACCEPT" autorise un port ou une IP).
* **prot** : le protocole utilisé (tcp, udp, icmp).
* **source** : l'adresse IP source pour les connexions entrantes.
* **destination** : l'adresse IP de destination pour les connexions sortantes.
* La dernière colonne indique le port correspondant (ici, par exemple, www, ssh, et imap2 sont autorisés).

Dans cet exemple, la politique par défaut est "DROP", ce qui signifie que tout autre trafic est bloqué.

### Ajouter et supprimer des règles

Les principales commandes pour manipuler **iptables** sont les suivantes :

* -A chain : ajoute une règle à la fin de la chaîne spécifiée (ex. INPUT, OUTPUT).
* -D chain rulenum : supprime la règle n° *rulenum* de la chaîne spécifiée.
* -I chain rulenum : insère une règle à la position spécifiée dans la chaîne. Si *rulenum* n’est pas précisé, la règle est insérée en premier.
* -R chain rulenum : remplace la règle n° *rulenum* dans la chaîne spécifiée.
* -L : liste les règles actuelles.
* -F chain : vide toutes les règles de la chaîne spécifiée.
* -P chain règle : modifie la politique par défaut pour la chaîne (ex. policy DROP ou ACCEPT).

Pour ajouter une règle, utilisez cette syntaxe :

iptables -A (chain) -p (protocole) --dport (port) -j (décision)

Exemples :

* iptables -A INPUT -p tcp --dport ssh -j ACCEPT : autorise le trafic entrant via le protocole TCP sur le port SSH (22).
* iptables -A INPUT -p tcp --dport www -j ACCEPT : autorise le trafic entrant sur le port web (80).
* iptables -A INPUT -p tcp --dport imap2 -j ACCEPT : autorise le trafic entrant sur le port IMAP (143).

Si aucun port n’est spécifié, tous les ports seront autorisés.

### Autoriser les pings

En plus d’autoriser le trafic sur ces ports, je peux vous conseiller d’autoriser le protocole ICMP (pour pouvoir faire un ping) sur tous ces derniers :

# iptables -A INPUT -p icmp -j ACCEPT

Comme je n’ai pas indiqué de section --dport, cette règle s’applique à tous les ports, **mais pour les pings (icmp) uniquement** !

Votre ordinateur répondra alors aux « pings » pour indiquer qu’il est bien en vie.

Vos règles iptables pour INPUT devraient maintenant ressembler à ceci :

# iptables -L

Chain INPUT (policy ACCEPT)

target     prot opt source               destination

ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:www

ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh

ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:imap2

ACCEPT     icmp --  anywhere             anywhere

### Autoriser les connexions locales et déjà ouvertes

En plus d'autoriser le trafic sur certains ports, il est recommandé d'autoriser le protocole ICMP pour permettre les pings, avec la commande suivante :

iptables -A INPUT -p icmp -j ACCEPT

Cette règle autorise tous les pings (ICMP) sur tous les ports, car aucune section --dport n'a été spécifiée. Ainsi, votre ordinateur pourra répondre aux pings, signalant qu'il est en ligne. Une fois ces règles appliquées, les règles pour la chaîne INPUT ressembleront à ceci :

Chain INPUT (policy ACCEPT)

ACCEPT tcp -- anywhere anywhere tcp dpt:www

ACCEPT tcp -- anywhere anywhere tcp dpt:ssh

ACCEPT tcp -- anywhere anywhere tcp dpt:imap2

ACCEPT icmp -- anywhere anywhere

### 

### Refuser toutes les autres connexions par défaut

Il est essentiel de définir une politique par défaut pour que toutes les connexions non autorisées soient bloquées. Utilisez la commande suivante pour changer la règle par défaut en DROP :

iptables -P INPUT DROP

Cela signifie que tout le trafic sera bloqué, sauf ce qui est explicitement autorisé dans les règles. Une fois cette modification effectuée, la commande iptables -L affichera :

Chain INPUT (policy DROP)

ACCEPT tcp -- anywhere anywhere tcp dpt:www

ACCEPT tcp -- anywhere anywhere tcp dpt:ssh

ACCEPT tcp -- anywhere anywhere tcp dpt:imap2

ACCEPT icmp -- anywhere anywhere

Ce filtrage est strict, et vous pourriez constater que certaines applications ne fonctionnent plus parce que leurs ports sont bloqués. Il faudra identifier les ports utilisés par ces applications et ajuster les règles en conséquence. Pensez également à configurer les règles pour le trafic sortant (OUTPUT) si nécessaire.

### Appliquer les règles au démarrage

Les règles définies avec iptables sont volatiles et disparaissent après chaque redémarrage. Pour les appliquer au démarrage, il est nécessaire de créer un script qui sera exécuté automatiquement au démarrage du système. En attendant, il est recommandé de consulter la documentation d'Ubuntu pour obtenir des instructions détaillées sur la mise en place de ces règles au démarrage.

Pour simplifier l'utilisation d'iptables, un programme appelé ufw (Uncomplicated Firewall) a été développé. Il est disponible sur les versions récentes d'Ubuntu et rend la gestion des règles plus accessible.

Enfin, les ACL sont particulièrement adaptées aux protocoles utilisant des ports statiques (comme SMTP), mais elles ne suffisent pas pour les logiciels qui utilisent des ports dynamiques, comme BitTorrent.

## Les ACLs et le matériel Cisco

Une liste d'accès (ACL) est un ensemble de règles basées sur des protocoles de couche 3 et supérieures, utilisées pour filtrer le trafic. Ces règles sont appliquées sur les interfaces réseau pour bloquer ou autoriser certaines parties du trafic. Les ACL font partie des fonctionnalités de type "firewall" dans les IOS Cisco. On se concentre ici sur les ACL standard et étendues, qui peuvent être identifiées par un numéro ou un nom.

Il existe deux principaux types d'ACL :

* **ACL standard** : filtre uniquement sur l'adresse IP source.
* **ACL étendue** : permet de filtrer en fonction de l'adresse IP de destination, du type de protocole (TCP, UDP, ICMP, HTTP, etc.), du port source, et du port de destination.

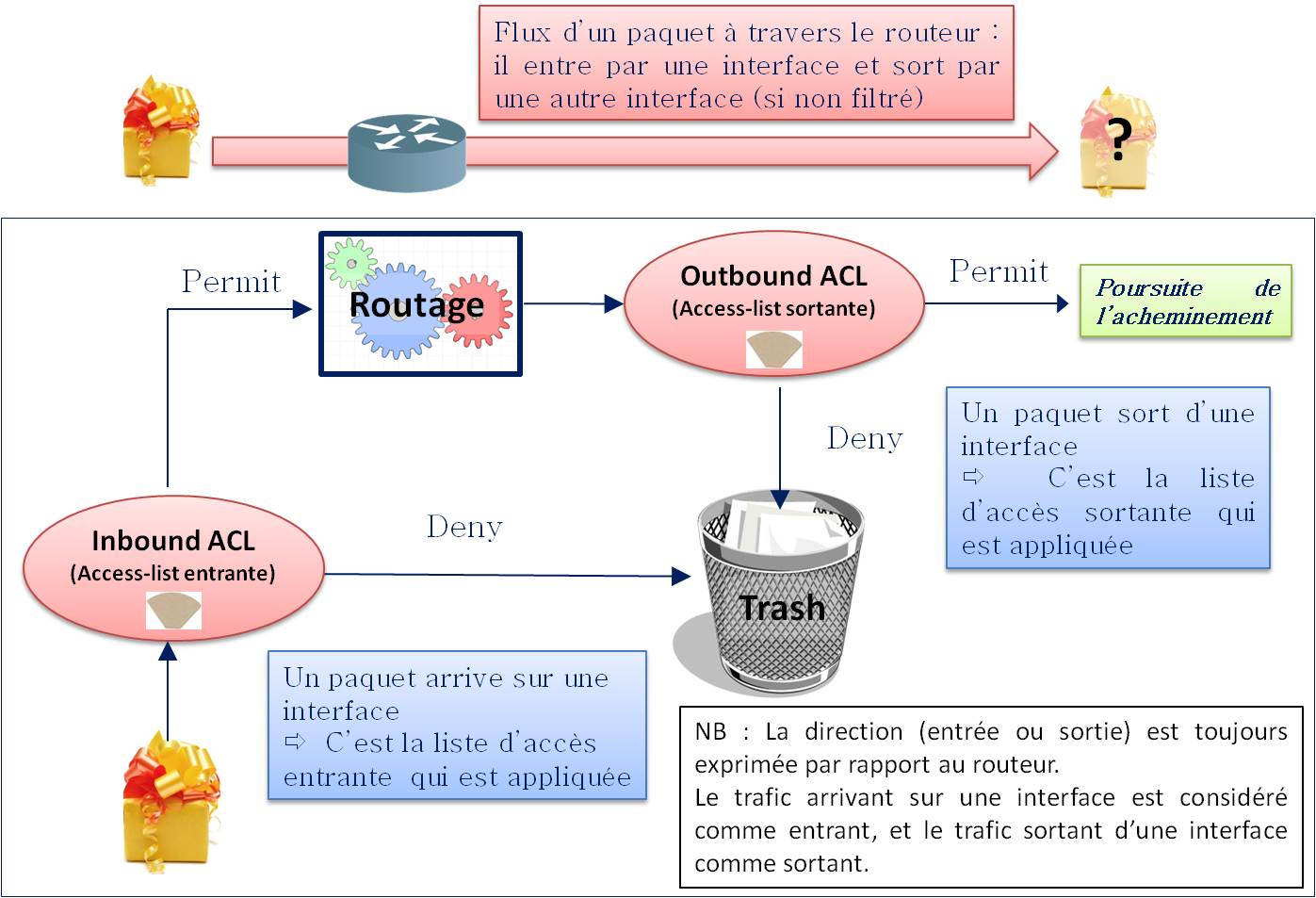
## Positionnement-Access-list.jpgPositionnement des access-lists

### Utilité

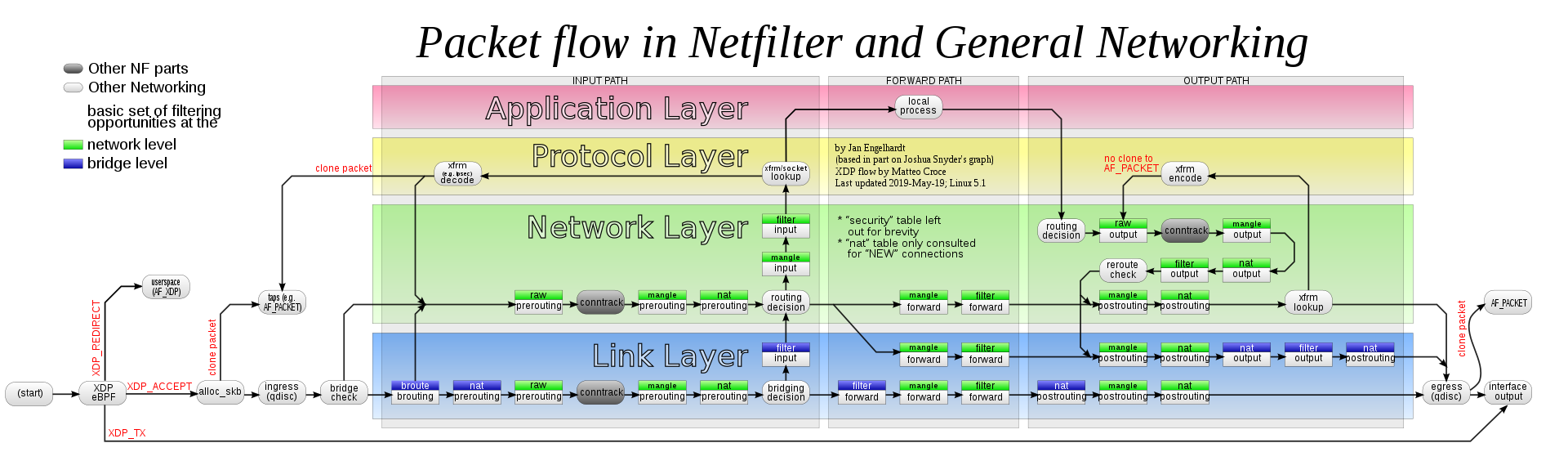
Une liste d’accès va servir :

* À supprimer des paquets pour des raisons de sécurité (pour du trafic de données ou des accès VTY)
* À filtrer des mises à jour de routage
* À filtrer des paquets en fonction de leur priorité (QoS)
* À définir du trafic intéressant pour des configurations spécifiques (NAT, ISDN, etc.)

### Logique d’application

Une liste d’accès, comportant une suite d’instructions de filtrage, va être appliquée sur une interface du routeur, pour le trafic entrant ou pour le trafic sortant. Il va falloir appliquer une logique sur les interfaces en sortie ou en entrée.

Si on associe aux ACL, le routage et le NAT, on aura une succession d’interventions sur les flux s’organisant de la façon suivante :



### Caractéristiques

Les principales caractéristiques des ACL sont les suivantes :

* Les paquets peuvent être filtrés en entrée (quand ils entrent sur une interface) **avant la décision de routage**
* Les paquets peuvent être filtrés en sortie (avant de quitter une interface) **après la décision de routage.**
* Le mot-clé IOS est "deny" pour signifier que les paquets doivent être filtrés ; précisément les paquets seront refusés selon les critères définis.
* Le mot-clé IOS est "permit" pour signifier que les paquets ne doivent pas être filtrés ; précisément les paquets seront permis selon les critères définis.
* La logique de filtrage est configurée dans les listes d’accès.
* **Une instruction implicite rejette tout le trafic à la fin de chaque liste d’accès**

### Traitement

Les paquets sont vérifiés en fonction des règles d'une ACL dans l'ordre où elles ont été définies :

* Si un paquet correspond aux critères d'une règle, l'action définie (**permit** ou **deny**) est appliquée, et les règles suivantes ne sont pas prises en compte.
* Si le paquet ne correspond pas à une règle, la règle suivante est vérifiée.
* Si aucune règle ne correspond, l'action **deny** est implicitement appliquée, même si elle n'est pas explicitement définie dans la liste.

### Différence entre liste d’accès standard et liste d’accès étendue

Une liste d'accès **standard** examine uniquement l'adresse IP source.  
Une liste d'accès **étendue** permet d'examiner non seulement les adresses IP source et destination, mais aussi les ports et le type de protocole (IP, ICMP, TCP, UDP).  
De plus, il est possible d'utiliser un **masque générique** (wildcard mask) pour vérifier une partie des adresses IP.

### Désignation d’une liste d’accès

Une liste d'accès peut être identifiée par un numéro ou un nom et appliquée à une interface en entrée ou en sortie. Si un numéro est utilisé, il doit être choisi dans une plage spécifique selon le protocole de couche 3 :

| **Protocole** | **Plage** |
| --- | --- |
| IP | 1 - 99 et 1300 - 1999 |
| IP étendu | 100 - 199 et 2000 - 2699 |
| Apple Talk | 600 - 699 |
| IPX | 800 - 899 |
| IPX étendu | 900 - 999 |
| Protocole IPX Service Advertising | 1000 - 1099 |

Si un nom est utilisé, il est nécessaire de préciser si la liste est **standard** ou **étendue**.

### Le masque générique (Wildcard Mask)

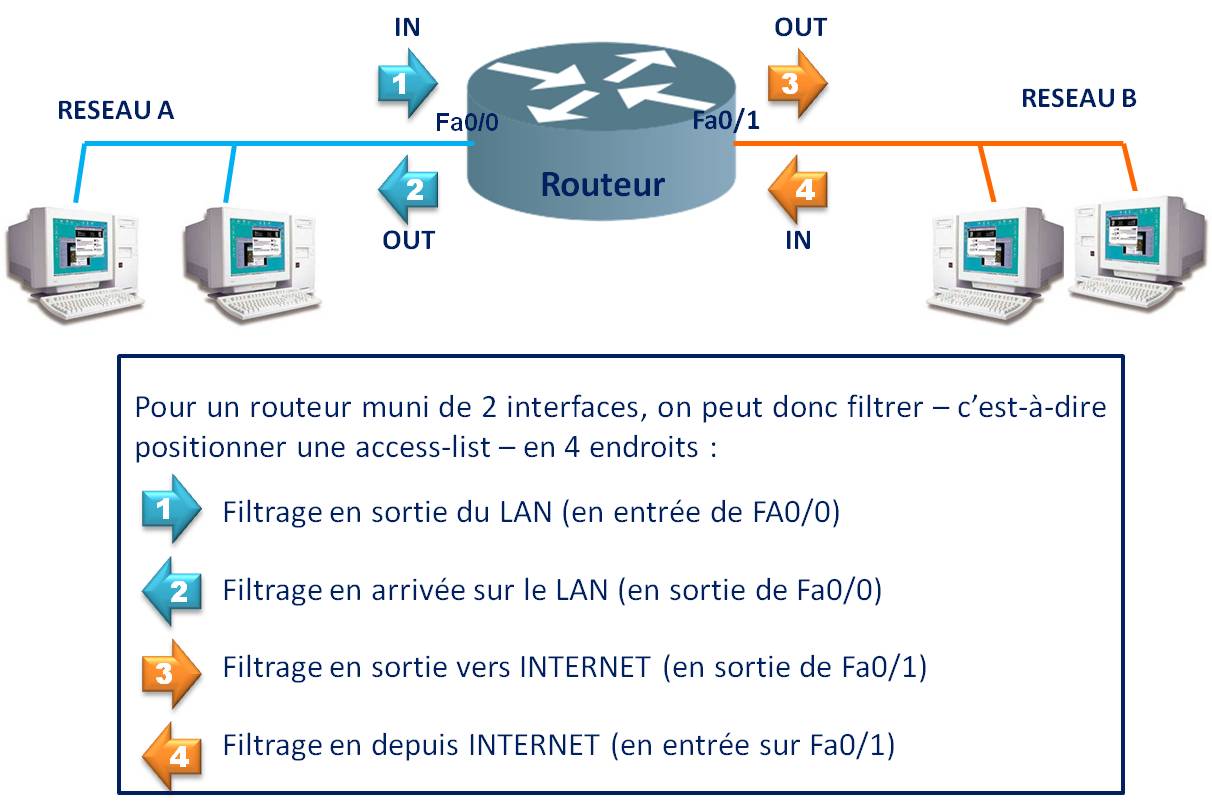
Un **masque générique** est un outil de filtrage utilisé dans les ACLs. Si un bit du masque est à 0, l'adresse IP de référence sera vérifiée pour ce bit. Si le bit est à 1, il ne sera pas vérifié. En général, le masque générique est l'inverse du masque de sous-réseau.

Exemples :

* Pour filtrer le réseau **192.168.1.0/24** (masque 255.255.255.0), on utilise le masque générique **0.0.0.255**.
* Pour filtrer le réseau **192.168.1.0/27** (masque 255.255.255.224), on utilise le masque générique **0.0.0.31**.
* Le mot **"any"** remplace l'adresse **0.0.0.0 255.255.255.255**, c'est-à-dire toutes les adresses IP.
* Le mot **"host"** remplace le masque **0.0.0.0**, par exemple, **10.1.1.1 0.0.0.0** peut être écrit comme **"host 10.1.1.1"**.

### Règles d’application

Les règles d'application des ACL sont les suivantes :

* **Pour une ACL étendue**, placez-la aussi près que possible de la **source** des paquets, c'est-à-dire au niveau de l'interface.
* **Pour une ACL standard**, placez-la au plus près de la **destination**, car elle ne vérifie que l'adresse source.
* Placez en tête de liste les règles les plus spécifiques et les plus générales en fin de liste.
* Ces recommandations doivent être suivies tout en respectant les restrictions d'accès identifiées.

### Syntaxe des commandes

La mise en œuvre d'une ACL se déroule en deux étapes :

1. **Création de la liste** : les instructions sont ajoutées une par une, suivies d'un retour chariot.
2. **Application de la liste** : elle est appliquée à une interface, en entrée ou en sortie.

Il est important de noter qu'il n'est pas possible d'insérer ou de modifier une instruction dans une liste existante. Seules des instructions supplémentaires peuvent être ajoutées à la fin de la liste.

### Liste d’accès standard

La syntaxe pour créer une liste d'accès standard est la suivante :

Router(config)#**access-list** *numéro-liste-accès* {**deny**|**permit**} *adresse-source* [*masque-source*] [**log**]

Cette liste permet de filtrer uniquement en fonction de l'adresse IP source.

### Liste d’accès étendue

La syntaxe pour créer une liste d'accès étendue est la suivante :

Router(config)#**access-list** *numéro-liste-accès* {**deny**|**permit**} *protocole adresse-source* *masque-source* [*opérateur* *port*] *adresse-destination* *masque-destination* [*opérateur* *port*] [**log**]

Les listes d'accès étendues permettent de filtrer en fonction du protocole, des adresses source et destination, et des ports.

Les opérateurs disponibles pour définir les conditions sur les ports sont :

* **lt** : inférieur à,
* **gt** : supérieur à,
* **eq** : égal à,
* **neq** : différent de,
* **range** : intervalle inclusif.

Le paramètre "port" peut être un numéro de port entre 0 et 65535 ou un nom de service comme HTTP, Telnet, FTP, etc.

### Liste d’accès nommée

Pour créer une liste d'accès nommée :

Router(config)#**ip access-list standard** *nom*

Router(config-ext-nacl)#**permit**|**deny**...

**Liste d'accès étendue** :

Router(config)#**ip access-list extended** *nom*

Router(config-ext-nacl)#**permit**|**deny...**

### Activation d’une liste d’accès sur une interface

Pour appliquer une liste d'accès sur une interface :

Router(config-if)#**ip access-group** {*numéro-liste-accès*|*nom* [**in** | **out**]}

Cela applique l'ACL en entrée (**in**) ou en sortie (**out**) de l'interface spécifiée.

### Diagnostic

Les commandes de diagnostic permettent de vérifier l'état des ACL :

* **Router#show ip interface [type numéro]** : affiche les informations sur l'interface IP.
* **Router#show access-lists [numéro-liste-accès|nom-liste-accès]** : affiche les détails de toutes les ACL.
* **Router#show ip access-list [numéro-liste-accès|nom-liste-accès]** : affiche les informations spécifiques aux ACL IP.

# Énoncé 1 : Comment configurer une Acl standard ?

Nous nous appuierons sur le réseau suivant : [Bloc2\_sem3-4\_atelier-02\_activite-1.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-1.pkt) dont la topologie est la suivante.

Une image contenant texte, diagramme, carte, capture d’écran

Description générée automatiquement

1. Utiliser les commandes Cisco pour obtenir la configuration suivante du routeur Router0, mettre en place le routage sur Router1 et Router2

Configuration obtenue avec la commande show run sur le Routeur0 :

Router#sh run

Building configuration...

Current configuration : 673 bytes

!

version 12.2

no service timestamps log datetime msec

no service timestamps debug datetime msec

no service password-encryption

!

hostname R0 (routeur 0)

!

interface FastEthernet0/0

ip address 192.168.1.1 255.255.255.0

ip access-group 1 out

duplex auto

speed auto

!

interface Serial0/0

ip address 192.168.4.1 255.255.255.0

clock rate 56000

!

interface Serial0/1

ip address 192.168.5.1 255.255.255.0

clock rate 56000

!

router rip

network 192.168.1.0

network 192.168.4.0

network 192.168.5.0

!

ip classless

!

!

access-list 1 deny host 192.168.2.2

access-list 1 permit any

!

line con 0!

line aux 0

!

line vty 0 4

login

!

end

1. Présenter ci-dessous votre script de mise en place du réseau et des ACLs :

**Routeur 0 :**

Router0(config)# interface FastEthernet0/0

Router0(config-if)# ip address 192.168.1.1 255.255.255.0

Router0(config-if)# no shutdown

Router0(config-if)# exit

Router0(config)# interface Serial0/0

Router0(config-if)# ip address 192.168.4.1 255.255.255.0

Router0(config-if)# no shutdown

Router0(config-if)# exit

Router0(config)# interface Serial0/1

Router0(config-if)# ip address 192.168.5.1 255.255.255.0

Router0(config-if)# no shutdown

Router0(config-if)# exit

Router0 (config)# access-list 1 deny host 192.168.2.2

Router0 (config)# access-list 1 permit any

Router0 (config)# interface FastEthernet0/0

Router0(config-if)# ip access-group 1 out

Router0(config-if)# exit

**Routeur 1 :**

Router1(config)# interface FastEthernet0/0

Router1(config-if)# ip address 192.168.2.1 255.255.255.0

Router1(config-if)# no shutdown

Router1(config-if)# exit

Router1(config)# interface Serial0/0

Router1(config-if)# ip address 192.168.4.2 255.255.255.0

Router1(config-if)# no shutdown

Router1(config-if)# exit

Router1(config)# router rip

Router1(config-router)# network 192.168.2.0

Router1(config-router)# network 192.168.4.0

Router1(config-router)# version 2

Router1(config-router)# no auto-summary

**Routeur 2 :**

Router2(config)# interface FastEthernet0/0

Router2(config-if)# ip address 192.168.3.254 255.255.255.0

Router2(config-if)# no shutdown

Router2(config-if)# exit

Router2(config)# interface Serial0/0

Router2(config-if)# ip address 192.168.5.2 255.255.255.0

Router2(config-if)# no shutdown

Router2(config-if)# exit

Router2(config)# router rip

Router2(config-router)# network 192.168.3.0

Router2(config-router)# network 192.168.5.0

Router2(config-router)# version 2

Router2(config-router)# no auto-summary

Router2(config-router)# exit

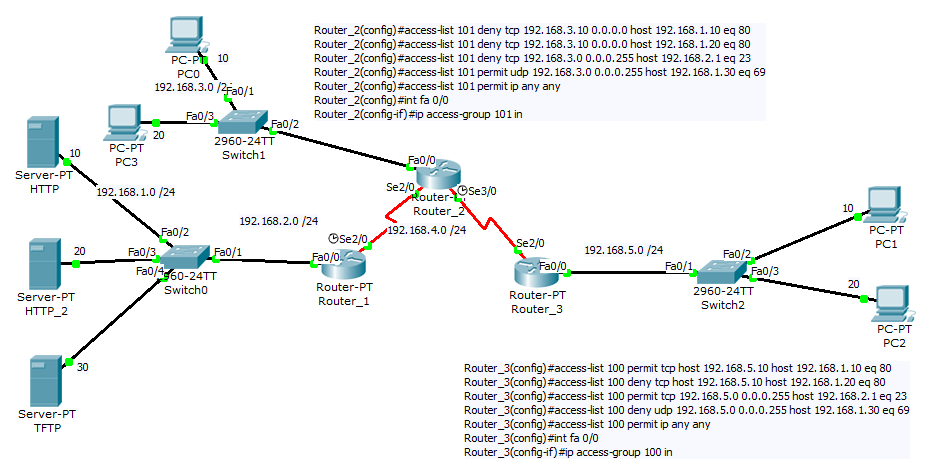
1. Tester le fonctionnement : le ping à partir du PC0 vers l’hôte 192.168.2.2 (PC1) doit échouer, le ping de Pc0 vers Pc2 doit fonctionner ainsi que le ping de Pc3 vers Pc2.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

La capture d'écran montre que trois pings ont réussi et un a échoué. Le ping échoué est entre PC0 et PC1, en raison de l'ACL configurée sur le Routeur 0, qui bloque l'accès à PC1 (adresse 192.168.2.2). Les autres communications sont réussies. Cela prouve que l'ACL appliquée à l'interface Fa0/0 fonctionne correctement et qu'elle n'est active que sur cette interface. Si elle avait été appliquée à une interface série, PC3 n'aurait pas pu communiquer avec PC1.

# Énoncé 2 : [ACL étendue (Extended ACL)](http://www.networking-space.com/2011/05/acl-etendue-extended-acl.html)

Nous utiliserons le fichier suivant : [Bloc2\_sem3-4\_atelier-02\_activite-2.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-2.pkt) dont la topologie est la suivante :

**Configuration :**

**Router\_1 :**

Router (config)#host Router\_1

Router\_1 (config)#int fa 0/0

Router\_1 (config-if)#IP address 192.168.1.1 255.255.255.0

Router\_1 (config-if)#no sh

Router\_1 (config-if)#interface Serial 2/0

Router\_1 (config-if)#IP address 192.168.2.1 255.255.255.0

Router\_1 (config-if)#clock rate 64000

Router\_1 (config-if)#no sh

Router\_1 (config)#router rip

Router\_1 (config-router)#network 192.168.1.0

Router\_1 (config-router)#network 192.168.2.0

Router\_1 (config)#line con 0

Router\_1 (config-line)#line vty 0 4

Router\_1 (config-line)#password cisco

**Router\_2 :**

Router (config)#hostname Router\_2

Router\_2 (config)#int fa 0/0

Router\_2 (config-if)#IP address 192.168.3.1 255.255.255.0

Router\_2 (config-if)#no sh

Router\_2 (config-if)#interface Serial 2/0

Router\_2 (config-if)#IP address 192.168.2.2 255.255.255.0

Router\_2 (config-if)#no sh

Router\_2 (config-if)#interface Serial3/0

Router\_2 (config-if)#IP address 192.168.4.1 255.255.255.0

Router\_2 (config-if)#clock rate 64000

Router\_2 (config-if)#no sh

Router\_2 (config)#router rip

Router\_2 (config-router)#network 192.168.2.0

Router\_2 (config-router)#network 192.168.3.0

Router\_2 (config-router)#network 192.168.4.0

Router\_2 (config-router)#exit

Router\_2 (config)#access-list 101 deny TCP 192.168.3.10 0.0.0.0 host 192.168.1.10 eq 80

Router\_2 (config)#access-list 101 deny TCP 192.168.3.10 0.0.0.0 host 192.168.1.20 eq 80

Router\_2 (config)#access-list 101 deny TCP 192.168.3.0 0.0.0.255 host 192.168.2.1 eq 23

Router\_2 (config)#access-list 101 permit udp 192.168.3.0 0.0.0.255 host 192.168.1.30 eq 69

Router\_2(config)#access-list 101 permit ip any any

Router\_2 (config)#int fa 0/0

Router\_2 (config-if)#IP access-group 101 in

Router\_2 (config)#line con 0

Router\_2 (config-line)#line vty 0 4

Router\_2 (config-line)#password cisco

**Router\_3 :**

Router (config)#hostname Router\_3

Router\_3 (config)#int fa 0/0

Router\_3 (config-if)#IP address 192.168.5.1 255.255.255.0

Router\_3 (config-if)#no sh

Router\_3 (config-if)#interface Serial 2/0

Router\_3 (config-if)#IP address 192.168.4.2 255.255.255.0

Router\_3 (config-if)#no sh

Router\_3 (config)#router rip

Router\_3 (config-router)#network 192.168.4.0

Router\_3 (config-router)#network 192.168.5.0

Router\_3 (config-router)#exit

Router\_3 (config)#access-list 100 permit TCP host 192.168.5.10 host 192.168.1.10 eq 80

Router\_3 (config)#access-list 100 deny TCP host 192.168.5.10 host 192.168.1.20 eq 80

Router\_3 (config)#access-list 100 permit TCP 192.168.5.0 0.0.0.255 host 192.168.2.1 eq 23

Router\_3 (config)#access-list 100 deny udp 192.168.5.0 0.0.0.255 host 192.168.1.30 eq 69

Router\_3(config)#access-list 100 permit ip any any

Router\_3 (config)#int fa 0/0

Router\_3 (config-if)#IP access-group 100 in

Router\_3 (config)#line con 0

Router\_3 (config-line)#line vty 0 4

Router\_3 (config-line)#password cisco

1. Rédigez en français une description du réseau en y précisant les règles de gestion cachées derrière les différentes commandes décrites soient dans le réseau soit dans les instructions de configuration ci-dessus :

Le réseau est composé de trois routeurs interconnectés, chacun gérant des sous-réseaux distincts. Le protocole de routage RIP est utilisé pour partager les informations de routage entre les routeurs. Des listes de contrôle d'accès (ACL) sont mises en place pour filtrer et gérer le trafic au sein du réseau.

Router\_1 est connecté aux réseaux 192.168.1.0 et 192.168.2.0. Il utilise une configuration RIP pour ces réseaux, mais sans ACL complexe.

Router\_2 gère les réseaux 192.168.2.0, 192.168.3.0, et 192.168.4.0. Une ACL 101 est configurée pour bloquer certains accès : les hôtes du réseau 192.168.3.0 ne peuvent pas accéder à des services spécifiques sur le réseau 192.168.1.0, comme HTTP ou Telnet, tandis que certains services UDP, comme TFTP, sont autorisés.

Router\_3 est connecté aux réseaux 192.168.4.0 et 192.168.5.0. Une ACL 100 est mise en place pour autoriser ou bloquer l'accès à certains services, comme l'HTTP et Telnet, entre les réseaux 192.168.5.0 et 192.168.1.0.

Les ACLs jouent un rôle clé dans la sécurisation du réseau en filtrant les accès aux services spécifiques tels que HTTP, Telnet et TFTP, garantissant que seuls certains types de trafic sont autorisés selon les règles définies.

1. Configurer le réseau avec ses ACLs

Router\_2(config)#access-list 101 deny tcp 192.168.3.10 0.0.0.0 host 192.168.1.10 eq 80

Router\_2(config)#access-list 101 deny tcp 192.168.3.10 0.0.0.0 host 192.168.1.20 eq 80

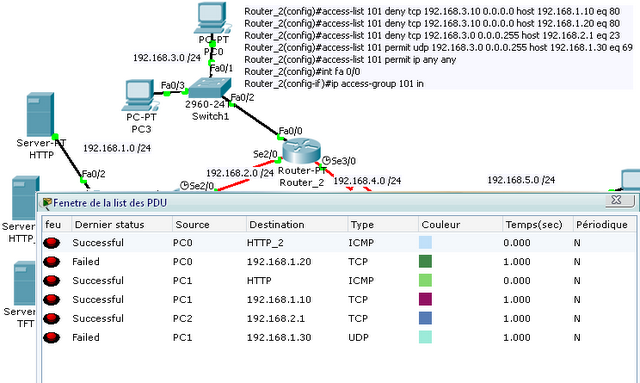
Router\_2(config)#access-list 101 deny tcp 192.168.3.0 0.0.0.255 host 192.168.2.1 eq 23

Router\_2(config)#access-list 101 permit udp 192.168.3.0 0.0.0.255 host 192.168.1.30 eq 69

Router\_2(config)#access-list 101 permit ip any any

Router\_2(config)#interface FastEthernet 0/0

Router\_2(config-if)#ip access-group 101 in

1. Montrer que vous obtenez bien les résultats suivants :

On peut déjà obtenir ce résultat car les commandes avec les adresses IP permettent de le faire, mais en spécifiant les ports, on sait exactement ce que l'on bloque, ce qui peut être mieux. Cependant, dans ce cas, les nouvelles commandes sont un peu moins restrictives car nous perdons le blocage du HTTPS (port 443).

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Avec les nouvelles configurations j’ai bien obtenu les résulta attendu.

Une image contenant texte, logiciel, Icône d’ordinateur, nombre

Description générée automatiquement

# Énoncé 3 : exercice récapitulatif et progressif afin de bien comprendre

On utilisera le fichier : [Bloc2\_sem3-4\_atelier-02\_activite-3-Decouverte\_AccessListsCisco-01.pka](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-3-Decouverte_AccessListsCisco-01.pka)

## Présentation du scénario

L’entreprise MURDEFEU est une société de service informatique qui installe des pare-feux, notamment sur des architectures à base d’équipements CISCO. Dans un objectif de formation, elle vous soumet quelques scénarios que vous devez tester pour vous approprier la notion d’*access-list*.

Le schéma réseau imaginé pour effectuer ces tests est celui d’une entreprise "BLUE" qui accéderait à différents sites, dont ceux hébergés par MULTiCOLOR, une société qui héberge différents sites, dont certains ne sont pas vraiment utiles à la productivité de l’entreprise BLUE.

La maquette suivante est donc fonctionnelle en l’état, sans filtrage. L’entreprise BLUE peut accéder aux 3 sites hébergés par MULTiCOLOR ainsi qu’à www.orange.com :

Une image contenant texte, diagramme, capture d’écran, Plan

Description générée automatiquement

*\* Le Nat/Pat ne fera pas l’objet de ce support. Par simplification d’ailleurs, il n’est pas mis en œuvre ici, puisque la maquette ne comporte que des réseaux supposés publics qui communiquent entre eux.*

Les premiers scénarios utilisent les *access-lists* **standard**, puis des scénarios plus complexes mettront en œuvre des *access-lists* **étendues**. D’autres explications seront données au fur et à mesure des cas pratiques étudiés, de manière à bien comprendre le fonctionnement des *access-lists*.

## Remarques

* Tous les scénarios n’auront pas forcément un objectif très réaliste ; il s’agit bien entendu de scénarios à vertu pédagogique, applicables sur une seule maquette et forcément limités à quelques règles.
* La *philosophie* CISCO ne sera pas toujours respectée dans ce scénario, du fait que la maquette est volontairement simplifiée (chaque entreprise n’a pas son propre routeur, la maquette ne comporte qu’un seul routeur d’interconnexion).
* Rappel : la *philosophie* CISCO consiste (en simplifiant) :
  + À appliquer les *access-lists* standard au plus près de la destination, puisqu’on ne peut filtrer que la source ;
  + À appliquer au contraire les *access-lists* étendues au plus près de la source, autant que possible, pour éviter notamment le traitement (routage) inutile de paquets par les routeurs.

## Organisation des instructions (dans le fichier Packet Tracer)

Chaque scénario est sur une page distincte d’instructions. Il y a 7 scénarios au total, cumulables pour atteindre une configuration de maquette plausible. Pour une progression cohérente, il faut effectuer les scénarios dans l’ordre, et valider chaque scénario par des tests judicieusement choisis.

### Scénario 1 : isolation de l’entreprise BLUE

#### Cahier des charges

On souhaite simplement isoler l’entreprise "BLUE" des autres réseaux, en permettant aux deux réseaux "COBALT" (192.192.1.0) et "CIEL" (192.192.2.0) de communiquer entre eux, mais pas avec le *reste du monde*.

Deux solutions peuvent être envisagées :

1. Empêcher tout trafic sortant sur Se0/0/0.
2. Filtrer les accès sur les deux interfaces Gi0/0 et Gi0/1.

La première solution est bien entendu plus facile à mettre en œuvre, mais l’intérêt pédagogique de la 2e solution nous amènera à l’étudier également dans un 2e temps.

Avant de poursuivre, vérifier que les communications sont possibles avant filtrage :

* Par exemple de PC1-11 vers Serveur CIEL (192.192.2.80).
* Par exemple de PC1-11 vers www.rose.com (vous pouvez utiliser le nom DNS ou l’adresse 201.10.10.80).

### Mise en œuvre du filtrage (solution 1)

#### Mise en œuvre de l’access-list

# Création de l’access-list numéro 1 (une access-list standard doit être numérotée de 1 à 99 - ou entre 1300 et 1999)

Router1(config)#access-list 1 deny 192.192.0.0 0.0.255.255

Router1(config)#access-list 1 permit any

# Application de l’access-list en sortie de s0/0/0

Router1(config)#interface Se0/0/0

Router1(config-if)#ip access-group 1 out

#### Résultat obtenu/explication

Vérifions que l’accès reste possible entre CIEL et COBALT, mais que la communication vers un autre site n’est plus possible :

|  |  |
| --- | --- |
| **Résultat attendu** |  |
|  | 12-CaptureErreurTimeout.JPG |

|  |  |
| --- | --- |
| **Résultat obtenu** |  |
|  |  |

Pour être complet, il faut prendre le soin de tester également une communication du réseau COBALT vers le réseau CIEL et du réseau COBALT vers *le reste du monde*. Vous devez obtenir un résultat similaire : Réussite pour le test vers 192.192.1.80, Échec pour le test vers 201.10.10.80.

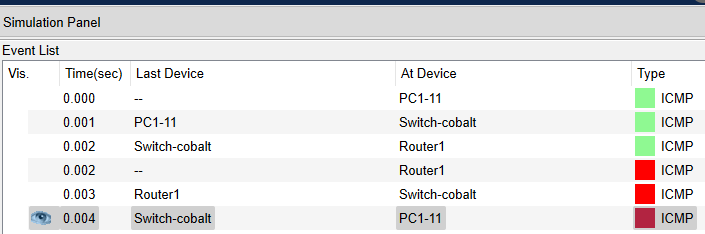
Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

ATTENTION ! On n’a filtré que les communications à destination du *reste du monde* ; mais dans l’état actuel, le *reste du monde* peut tenter de communiquer avec BLUE ; cependant il n’obtiendra pas de réponse, êtes-vous d’accord avec cette affirmation ?

Le filtrage actuel bloque uniquement le trafic sortant du réseau "BLUE". Le reste du monde peut encore tenter de communiquer avec "BLUE", mais ce dernier ne répondra pas car les paquets sortants sont bloqués.

Montrer en mode simulation le filtrage des paquets et les éventuelles trames prouvant l’application de ce dernier.



Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Dans les deux simulations, les pings (ICMP) montrent que les requêtes sortantes de PC1-11 passent par le réseau, atteignent Router1, mais ne reçoivent pas de réponse en raison du blocage du trafic sortant imposé par les access-lists. Cela signifie que la communication de PC1-11 avec l'extérieur est bloquée, empêchant toute réponse vers le reste du réseau.

Dans le cas du ping de [www.orange.com](http://www.orange.com) vers PC1-11, les requêtes ICMP entrent dans le réseau "BLUE", mais les réponses de PC1-11 sont également bloquées au niveau de Router1. Cela confirme que le réseau "BLUE" peut recevoir des requêtes, mais ne peut pas envoyer de réponse, conformément aux access-lists configurées.

### Mise en œuvre du filtrage (solution 2)

#### Mise en œuvre des access-lists

Avec une *access-list* standard, on ne peut filtrer que la source, donc il ne nous est pas possible d’appliquer une *access-list* standard en entrée de Gi0/0 et Gi0/1 qui ne laisserait passer que les paquets à destination de 192.192.2.0 (respectivement 192.192.1.0).

En revanche, si on souhaite absolument n’utiliser que les *access-lists* standard, on peut appliquer une *access-list* en sortie de Gi0/0 et Gi0/1 qui ne laisserait passer que les paquets en provenance de 192.192.1.0 (respectivement 192.192.2.0). Autrement dit, on accepterait que la requête parte, mais on refuserait toute réponse qui ne viendrait pas de l’autre réseau de BLUE.

Il nous faut d’abord annuler la règle précédente, pour éviter tout télescopage entre les deux solutions : On supprime la liste et on supprime son application sur s0/0/0.

Router1(config)#interface Se0/0/0

Router1(config-if)#no ip access-group 1 out

Router1(config-if)#exit

Router1(config)#no access-list 1

# Création des nouvelles *access-lists 1 et 2 respectivement pour le réseau 192.192.1.0 et le réseau 192.192.2.0*  (il en faut 2 puisque ce ne sont pas les mêmes règles sur Go0/0 et Gi0/1)

Router1(config)#access-list 1 deny 192.192.1.0 0.0.0.255

Router1(config)#access-list 1 permit any

Router1(config)#access-list 2 deny 192.192.2.0 0.0.0.255

Router1(config)#access-list 2 permit any

Remarque : inutile d’interdire les autres communications puisqu’elles le sont par défaut. L’*access-list* n° 1 n’autorise que les paquets en provenance du réseau "COBALT". L’*access-list* n° 2 n’autorise que les paquets en provenance du réseau "CIEL".

**Remarque importante sur le masque** : Pour les *access-lists* on parle aussi de **masque inversé**: les bits positionnés (à 1) ne sont pas ceux que l’on vérifie, mais au contraire ceux que l’on ne vérifie pas. Dans les règles ci-dessus, on autorise tout le réseau 192.192.1.0/24 ou 192.192.2.0/24. La valeur du 4e octet n’est pas vérifiée, mais seulement les 3 premiers octets.

# Application de l’access-list en sortie de gi0/0 et gi0/1

Router1(config)#interface Gi0/0

Router1(config-if)#ip access-group 1 in

Router1(config-if)#exit

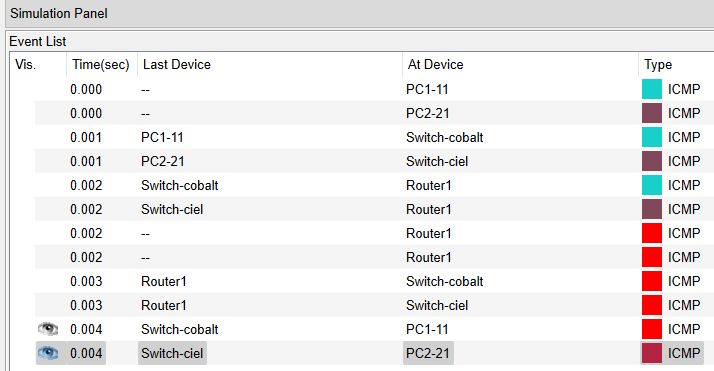
Router1(config)#interface Gi0/1

Router1(config-if)#ip access-group 2 in

> En sortie du routeur par l’interface gi0/0 qui est connectée au réseau "COBALT", on n’autorise que les paquets qui proviennent du réseau "CIEL"

Une image contenant texte, Police, nombre, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement> En sortie du routeur par l’interface gi0/1 qui est connectée au réseau "CIEL", on n’autorise que les paquets qui proviennent du réseau "COBALT"

#### Résultat obtenu/explication

Vous devriez obtenir exactement le même résultat qu’avec la solution 1 si vous ne faites pas d’erreur. On se dispensera donc des copies d’écran. Cette fois ce sont les réponses (*du reste du monde*) qui sont filtrées, et non les envois. On peut le vérifier par une analyse des paquets.

Les communications initiées depuis le *reste du monde* ne sont pas possibles non plus. Petite différence quand même sur l’acheminement des paquets : si vous observez attentivement ce qui se passe en mode simulation, vous constaterez que la requête HTTP ne parvient même pas au serveur - contrairement à l’observation pour la solution 1 - puisqu’elle est filtrée en sortie de l’interface Gi0/0.

## Scénario 2 : accès distinctif entre les réseaux Cobalt et CIEL

Le scénario 1 avait un intérêt limité : pourquoi connecter l’entreprise BLUE au *reste du monde* si aucun poste ne peut communiquer avec elle ? On va donc passer à un cas de figure plus intéressant.

### Cahier des charges

On souhaite maintenant faire une distinction entre les deux réseaux de l’entreprise "BLUE" :

* Le réseau "COBALT" (192.192.1.0) aura accès au *reste du monde.* Et réciproquement, le réseau COBALT (son serveur par exemple) sera accessible depuis *le reste du monde*.
* Le réseau "CIEL" (192.192.20) ne pourra pas communiquer avec le *reste du monde.*

NB : La communication entre les deux réseaux "COBALT" (192.192.1.0) et "CIEL"  (192.192.2.0) devra rester possible.

### Mise en œuvre du filtrage

Rappel : avec une *access-list* standard, on ne peut filtrer que la source. On pourrait envisager les 2 solutions, comme précédemment pour le scénario 1 :

1. Soit autoriser en sortie de s0/0/0 seulement ce qui provient du réseau COBALT.
   * Ce qui permettrait au reste du monde d’atteindre le réseau CIEL, mais sans obtenir de réponse.
2. Soit autoriser en sortie de Gi0/1 uniquement ce qui provient du réseau COBALT et ne mettre aucune règle ailleurs :
   * Ce qui permettrait au réseau CIEL d’envoyer des requêtes vers le reste du monde, mais il ne recevrait aucune réponse, car bloquée au retour.

Dans les deux cas, même si le filtrage est efficace, on autorise quand même des flux non souhaités, et inutiles au final puisque sans suite. Pour isoler complètement le réseau CIEL du reste du monde, on va utiliser deux règles :

* Une *access-list* en sortie de se0/0/0 qui n’autorise que les paquets "COBALT" de sortir vers le *reste du monde*.
* Une *access-list* en entrée de se0/0/0 qui n’autorise que les paquets à destination de "COBALT" de rentrer.

Comme vous l’avez sans doute deviné, la 2e ACL ne pourra pas être une *access-list* standard puisqu’elle filtre sur la destination.

On souhaite toujours laisser les communications possibles entre les réseaux CIEL et COBALT.

On s’aperçoit que de fait il est dangereux et totalement inapproprié de vouloir modifier les access-lists précédentes, on préfèrera tout supprimer avant de réécrire la totalité des ACLs. On commence donc par annuler les access-lists du SCÉNARIO 1 qui ne sont plus nécessaires :

Router1(config)#no access-list 1

Router1(config)#interface GigabitEthernet0/0

Router1(config-if)#no ip access-group 1 in

Router1(config)#no access-list 2

Router1(config)#interface GigabitEthernet0/1

Router1(config-if)#no ip access-group 2 in

* Une **access-list étendue** possède un numéro entre **100 et 199** (ou entre **2000 et 2699**). Elle permet de filtrer sur la source, la destination, le protocole, et éventuellement le numéro de port.
* La source ou la destination dans une ACL peuvent être définies de trois façons :
  1. **any** (pour n'importe quelle source ou destination).
  2. **<adresse> <masque-inversé>** (pour un réseau ou une partie de réseau).
  3. **host <adresse-hote>** (pour un hôte spécifique).

Dans ce scénario :

* Une **ACL standard** numérotée **2** sera utilisée en sortie de l'interface **Serial0/0/0** pour autoriser uniquement les paquets venant du réseau **COBALT** (192.192.1.0).
* Une **ACL étendue** numérotée **102** sera appliquée en entrée de **Serial0/0/0**, permettant à tous les paquets IP (TCP, UDP, ICMP) de passer si la destination est **192.192.1.0**, quelle que soit leur source.

Router1(config-if)#access-list 102 permit ip any 192.192.1.0 0.0.0.255

Router1(config)#interface Serial0/0/0

Router1(config-if)#ip access-group 102 in

Router1(config)#access-list 2 permit 192.192.1.0 0.0.0.255

Router1(config)#interface Serial0/0/0

Router1(config-if)#ip access-group 2 out

### Résultat obtenu/explication

Vérifions que l’accès reste possible entre CIEL et COBALT, mais que la communication vers un autre site n’est possible que depuis le réseau COBALT :

Résultats obtenus :

|  |  |
| --- | --- |
| Une image contenant texte, capture d’écran, logiciel, Page web  Description générée automatiquement |  |

Une image contenant capture d’écran, texte, logiciel, Icône d’ordinateur

Description générée automatiquement

**Surprise !** On n’obtient pas vraiment ce que l’on attendait. Plus aucune communication avec le *reste du monde* n’est possible !

En fait, le routage était assuré par le protocole RIP. Le routeur ne reçoit des informations que si elles sont à destination du réseau 192.192.1.0. Du coup les échanges RIP sont refusés et la table de routage ne contient plus - par conséquent - de route dynamique. Montrer qu’il en est ainsi en utilisant d’une part un tracert et d’autre part en montrant la table de routage du Routeur 1.

Une image contenant texte, logiciel, Icône d’ordinateur, Page web

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Remarque : Il faut un certain délai pour que les routes disparaissent, donc cela peut fonctionner encore quelques instants ; les routes deviennent « possibly down » au bout de 3 min et disparaissent complètement après 9 minutes normalement (Fast Forward Time).

* Le test ci-dessus montre bien que là est le problème. On pouvait penser à un filtrage trop important, mais le résultat du ping est assez parlant : Router1 (192.168.1.1) n’a pas de route vers la destination 201.10.10.80.
* L’absence de routes fournies par le protocole RIP dans la table de routage confirme le soupçon.

Il faut donc s’intéresser à un échange RIP afin de capturer une trame grâce au mode simulation, permettant de repérer le protocole et le port utilisé par RIP pour l’autoriser. Encore une fois, on profite des riches fonctionnalités de Packet Tracer. Montrer la trame obtenue :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Comment se font les échanges avec RIP ? Quel est le port utilisé ?

Les échanges RIP se font en envoyant des mises à jour de routage périodiques entre routeurs. Chaque routeur partage sa table de routage toutes les 30 secondes pour informer ses voisins des routes qu'il connaît. RIP utilise le protocole UDP sur le port 520, à la fois pour l'envoi et la réception des paquets. Ces messages contiennent des informations sur les routes disponibles et le nombre de sauts nécessaires pour y accéder. De plus on peut voir que le protocole RIP est la versions 1.

Cet imprévu va nous permettre d’appréhender un peu plus la puissance des *access-lists* étendues. Nous allons ajouter une autorisation spécifique au protocole RIP.

# On autorise tout trafic udp (on pourrait préciser la source éventuellement, mais ce n’est pas très utile) pourvu que le port (source **et** destination ici) soit 520.

Router1(conf)# **access-list 102 permit udp any eq 520 any eq 520**

Suite à cet ajout, a-t-on bien les constats souhaités/attendus, après quelque 30 secondes de délai :

* À partir de PC11, ouvrez un navigateur et connectez-cous au site web de rose.com, cela devrait fonctionner.
* À partir de PC21, ouvrez un navigateur et faites de même, cela ne devrait pas fonctionner.

Pour être complet, il faut prendre le soin de tester également :

* une communication de COBALT vers CIEL et *vice-versa*. Vous devez obtenir un résultat positif.
* Une communication depuis l’extérieur vers COBALT et CIEL : seule la 1re communication est possible.

Grâce à la nouvelle access-list, le traceroute a pu fonctionner correctement.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## Scénario 3 : accès restreint pour le réseau COBALT

L’accès internet pour le réseau COBALT est "universel". Nous souhaiterions maintenant interdire l’accès à quelques sites.

### Cahier des charges

On souhaite simplement :

* Continuer à autoriser le réseau "COBALT" (192.192.1.0) à avoir accès au *reste du monde.*
* Supprimer l’accès à quelques destinations, peu recommandables : www.rose.com et www.violette.com.

NB : La communication entre les deux réseaux "COBALT" (192.192.1.0) et « CIEL » (192.192.2.0) devra rester possible. Même s’il est peu probable qu’on empêche cette communication, puisque le filtrage va probablement se faire sur l’interface se0/0/0, il est indispensable d’effectuer des tests de "non régression" quand on modifie une configuration.

### Mise en œuvre du filtrage

Les nouvelles restrictions portant sur la destination, on a encore (comme c’est souvent le cas) au moins deux possibilités :

* interdire les accès sortants vers les destinations prohibées ;
* interdire les accès entrants depuis les sites prohibés.

On va choisir la solution de la raison, plutôt que la solution de la facilité : pourquoi autoriser des requêtes sortantes alors que l’on sait très bien que les réponses seront filtrées ?

Mais pour ce faire, il nous faut utiliser une *access-list* étendue en sortie, puisque l’on souhaite filtrer à la fois sur la source et sur la destination, alors que dans le scénario 2 on avait utilisé une standard en sortie.

# Suppression de l’access-list précédente

Router1(conf)# **no access-list 2**

# Remarque : inutile de supprimer l’affectation à l’interface s0/0/0, car elle sera écrasée ultérieurement ; on ne peut associer qu’une seule access-list en in ou out d’une interface.

# Création de l’access-list étendue 103 (il n’est pas nécessaire de modifier l’access-list 102 en entrée à priori)

Router1(conf)# **access-list 103 deny ip 192.192.1.0 0.0.0.255** **201.10.10.0 0.0.0.255**

Router1(conf)# **access-list 103 deny ip 192.192.1.0 0.0.0.255** **202.20.20.0 0.0.0.255**

Router1(conf)# **access-list 103 permit ip 192.192.1.0 0.0.0.255 any**

# Remarque : **Attention ! L’ordre importe**. Si on inverse les règles, puisque l’algorithme de filtrage s’arrête à la première règle qui "matche", l’accès serait toujours autorisé !

Par ailleurs, puisqu’on n’a pas précisé les services utilisables, on reste au niveau le plus "bas" possible (donc IP) pour ne filtrer aucun protocole en particulier.

De manière générale, en autorisation ou en interdiction, il faut toujours commencer par la règle propre à l’exception : ici on autorise tout sauf deux cibles.

# Association à l’interface s0/0/0

Router1(conf)# **interface s0/0/0**

Router1(conf-if)# **ip access-group 103 out**

Si on regarde la *running-config*, on voit bien que l’access-list **103** s’est bien substituée (et non cumulée) à l’access-list **2** pour l’association à l’interface en **out**:

!

interface Serial0/0/0

ip address 200.1.2.1 255.255.255.0

ip access-group 102 in

**ip access-group 103 out**

clock rate 64000

!

Une image contenant texte, Police, reçu, blanc

Description générée automatiquement

Vérifiez que le nouveau scénario est fonctionnel :

* Une image contenant texte, capture d’écran, logiciel, Page web

  Description générée automatiquementUne image contenant texte, fruit, orange, capture d’écran

  Description générée automatiquementL’accès à **www.orange.com** et **www.emeraude.com** doit rester possible depuis **COBALT**.
* Une image contenant texte, logiciel, capture d’écran, ligne

  Description générée automatiquementUne image contenant texte, ligne, logiciel, Police

  Description générée automatiquementL’accès à **www.violette.com** et **www.rose.com** ne soit plus être possible depuis **COBALT.**
* Les communications entre **CIEL** et **COBALT** doivent toujours être possibles.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

* Depuis **CIEL**, aucun accès externe n’est possible, pas plus vers **ORANGE** et **ÉMERAUDE** que vers **ROSE** ou **VIOLETTE.**

Une image contenant texte, logiciel, capture d’écran, Icône d’ordinateur

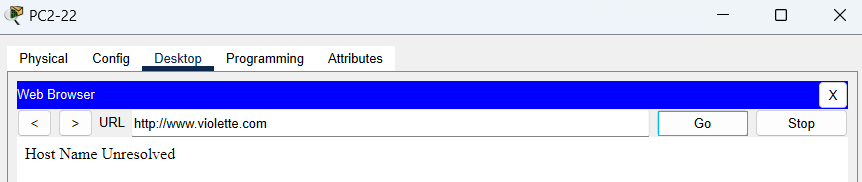
Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Une image contenant texte, logiciel, ligne, capture d’écran

Description générée automatiquement



## Scénario 4 : blacklister un PC sur www.emeraude.com

### Cahier des charges

PC1-11 a été repéré par le système de surveillance automatique du serveur **www.emeraude.com**, ce qui va provoquer le bannissement de cet hôte fauteur de trouble, d’après son IP.

Le scénario 4 va consister à mettre en place l’*access-list*, sur Router3, qui va bannir PC1-11 et seulement PC1-11. L’interdiction se fera seulement vers le serveur www.emeraude.com (on choisit de ne pas interdire tout le réseau 203.30.30.0).

### Mise en œuvre du filtrage

* Il faut choisir si l’*access-list* se met en entrée de Se0/0/1 ou en sortie de Gi0/2 :
  + Si on la met sur S0/0/0, il faut une *access-list* étendue pour filtrer à la fois sur la source (PC1-11) et la destination (www.emeraude.com).
  + Si on la met sur Gi0/2, on pourrait utiliser une *access-list* standard si elle concernait tout le réseau, mais comme elle ne concerne que le serveur, il nous faut également une *access-list* étendue.
* Il faut choisir si l’*access-list* s’applique en *in* ou en *out* :
  + Si on la met sur S0/0/0, il faut la mettre en **entrée**, donc en **in**.
  + Si on la met sur Gi0/2, il faut la mettre en **sortie** (attention on raisonne par rapport au routeur, pas par rapport au réseau ÉMERAUDE), donc en **out**.
* Enfin, il faut que toutes les autres communications restent autorisées, donc il faut :
  + **Premièrement,** interdire l’adresse bannie.
  + **Deuxièmement,** autoriser toute autre communication (avant l’interdiction par défaut implicite finale, à prendre en compte dans toute *access-list*).

Par principe, on bloque généralement autant que possible le trafic le plus tôt possible, donc le plus à l’extérieur possible : au moins le loup n’est pas du tout entré dans la bergerie.

Mais on peut aussi imaginer que l’administrateur d’ÉMERAUDE ait son mot à dire pour l’interface qui le concerne lui uniquement et pas les autres entreprises hébergées.

On va choisir, à titre plutôt exceptionnel cette 2e solution.

### Mise en œuvre effective

# Création de l’access-list étendue 104 sur Router3

Router3(conf)# **access-list 104 deny ip host 192.192.1.11 host 203.30.30.80**

Router3(conf)# **access-list 104 permit ip any any**

# Remarque : **Attention ! L’ordre importe**. Si on inverse les règles, puisque l’algorithme de filtrage s’arrête à la première règle qui "matche", l’accès sera toujours autorisé !

Par ailleurs, puisqu’on n’a pas précisé les services utilisables, on reste au niveau le plus "bas" possible (donc IP) pour ne filtrer aucun protocole en particulier.

Enfin on remarquera l’utilisation du mot-clé "**host**". Rappel : **host 192.192.1.1**  *est équivalent à*  **192.192.1.1  0.0.0.0**  *(on vérifie tous les bits)*

# Association à l’interface gi0/2

Router3(conf)# **interface gi0/2**

Router3(conf-if)# **ip access-group 104 out**

**Vérification du résultat**

PC1-11 ne doit pas pouvoir accéder au site [www.emeraude.com](http://www.emeraude.com)

Les autres PC, tels que PC1-12, peuvent accéder au site.

Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement

Par acquit de conscience, il faudrait normalement vérifier que toutes les autres communications fonctionnent (test de non-régression), mais normalement cette règle étant appliquée sur un nouveau routeur, et sur une interface bien spécifique, les effets de bord sont peu probables.

Vérifiez seulement que le réseau ÉMERAUDE peut communiquer avec les serveurs ORANGE et COBALT.

Une image contenant texte, Appareils électroniques, capture d’écran, Police

Description générée automatiquement

## Scénario 5 : accès restrictif à certains services sur le réseau ORANGE

Le scénario 5 va permettre de découvrir des possibilités d’accès restrictif à certains services seulement, par exemple d’après le n° de port. Il permettra également de voir que des effets de bord sont possibles et qu’ils peuvent être importants si on manque d’expérience et si on ne procède pas aux fameux tests de non-régression.

### Cahier des charges

Le scénario 5 autorisera uniquement l’utilisation des services HTTP et DNS :

* Le protocole DNS utilise le port UDP 53 (Une vérification des échanges DNS nous permet de le vérifier).
* Le service HTTP utilise bien entendu le port TCP 80.
* Le filtrage s’effectuera en entrée, sur l’interface Se0/0/0 de Router4.

*En effet, il est préférable de filtrer à l’entrée du routeur, pour éviter tout traitement (routage) inutile.*

* Les autres accès initiés depuis l’extérieur ne sont pas acceptés : on pourra notamment tester qu’une communication FTP sur www.orange.com échoue.
* Les autres communications doivent rester possibles, et notamment les serveurs d’Orange pourront continuer à accéder aux différents serveurs WEB externes et obtenir une réponse.
* Le routeur devra également accepter les informations concernant le protocole de routage RIP

Pour résumer, les seuls paquets entrants sur le réseau ORANGE qui seront acceptés sont :

* soit des demandes concernant les services HTTP et DNS ;
* soit des informations RIP ;
* soit des réponses à des requêtes sortantes, initiées depuis le réseau Orange.

### Vérifications avant mise en place du filtrage

On vérifie, avant de mettre en place ce filtrage supplémentaire, que les communications suivantes sont possibles :

* Un ping est possible depuis un serveur ORANGE et vers un serveur ORANGE (vers et depuis COBALT par exemple).
* Une communication HTTP est possible depuis un serveur ORANGE et vers www.orange.com (vers et depuis VIOLETTE par exemple).
* Une communication FTP est possible sur le serveur ORANGE (sauf depuis CIEL qui n’a pas d’accès aux autres réseaux que ceux de la BLUE Company).

Indication : par défaut, un utilisateur cisco/cisco est défini sur les serveurs pour les accès FTP.

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, Police

Description générée automatiquementCopies d’écran des tests précédents :

Une image contenant texte, fleur, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, fruit, orange, capture d’écran

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Une image contenant texte, fruit, capture d’écran, orange

Description générée automatiquement

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

### Mise en œuvre du filtrage

La mise en œuvre se fait donc par définition d’une *access-list* (105) et par application de cette *access-list* en **entrée de l’interface Se0/0/0**.

# Création de l’access-list étendue 105 sur Router4

Router4(conf)# **access-list 105 permit  tcp any any eq www**

Router4(conf)# **access-list 105 permit udp any host 200.200.4.53 eq 53**

Router4(conf)# **access-list 105 permit udp any eq 520 any eq 520**

# Remarques :

***# - On autorise les demandes TCP vers n’importe quel serveur (on imagine qu’il pourrait y en avoir plusieurs - d’ailleurs le DNS Orange répond aussi en HTTP)***

*#* ***- On autorise les demandes DNS, mais uniquement si elles sont destinées au serveur DNS***

*#* ***- On autorise la réception des mises à jour RIP***

*#* ***- On peut utiliser les opérateurs eq, qt, lt, neq, range qui correspondent aux opérateurs =, >, <, <> ou bien à un intervalle***

*#* ***- On peut utiliser soit un n° de port soit un mot-clé réservé comme le montre l’extrait ci-dessous.***

Router(config)# **access-list 105 permit tcp any any eq ?**

<0-65535> Port number

ftp File Transfer Protocol (21)

pop3 Post Office Protocol v3 (110)

smtp Simple Mail Transport Protocol (25)

telnet Telnet (23)

www World Wide Web (HTTP, 80)

Router(config)# **access-list 105 permit tcp any any eq www**

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

# Association à l’interface s0/0/0

Router4(conf)# **interface s0/0/0**

Router4(conf-if)# **ip access-group 105 in**

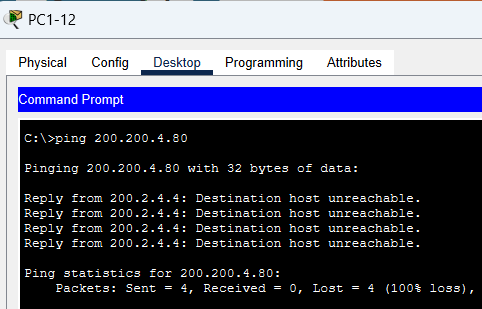
### Vérifications après mise en place du filtrage

La vérification est assez simple à faire :

* Le bon fonctionnement des requêtes HTTP et DNS est facile : il suffit d’effectuer l’accès à **www.orange.com** avec le nom DNS et on fait d’une pierre deux coups ;-)
* L’impossibilité d’accès aux autres services peut se faire avec une tentative de connexion FTP. La 2e copie d’écran montre qu’elle échoue, tout autant d’ailleurs qu’une tentative de PING, également refusée.
* Le test de non-régression est assez simple à effectuer, pour constater que l’on a un souci (cf. 3e copie d’écran) : un accès externe - depuis ORANGE vers un serveur web situé sur un autre réseau – échoue.

Fournissez les preuves des énoncés ci-dessus :

Une image contenant texte, capture d’écran, logiciel, ligne

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Le 3e constat s’explique assez facilement : une demande HTTP vers l’extérieur n’est pas filtrée, mais la réponse, en revanche, est bien filtrée, puisqu’elle sera à destination d’un port aléatoire, qui ne sera ni celui correspondant au service DNS, ni celui correspondant au service HTTP.

La solution passe, pour les requêtes TCP, et plus précisément par **l’état TCP** :

* Une requête sortante établit forcément une connexion TCP, donc si on peut accepter les réponses qui correspondent à un état "***established***", autrement dit à une connexion déjà initiée, cela solutionne notre problème ; et les *access-lists* le permettent justement.
* Une tentative de connexion établie depuis l’extérieur ne sera pas acceptée puisqu’il s’agira d’une demande de synchronisation (SYN) et donc l’état ne sera pas "***established***" pour la connexion TCP.

Ajoutons donc une règle autorisant les paquets réponses TCP, en vérifiant ainsi que la connexion est déjà initiée (sous-entendu depuis l’intérieur d’ORANGE forcément).

Router4(conf)# **access-list 105 permit  tcp  any  any  established**

*# On autorise tout flux TCP, pourvu que l’état soit "established"*

Nouveau test : l’accès vers l’extérieur, depuis ORANGE, devient possible, en tous cas pour les requêtes TCP :

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquementUne image contenant texte, fleur, capture d’écran, violet

Description générée automatiquement

On laissera de côté pour l’instant les requêtes UDP, dont les réponses pourraient être assez facilement autorisées par exemple en acceptant les paquets dont le port source correspond à un service UDP.

## Scénario 6 : modification d’acces-list pour ajouter de nouvelles règles

Le scénario 6 va permettre d’étudier la modification d’une *access-list* déjà en place (l'access-list 103, mise sur Router1), ceci afin d’intégrer de nouvelles règles. Il permettra aussi de mieux comprendre l’ordre des règles.

### Cahier des charges

Certains employés ont une adresse mail chez violette.com. On ne souhaitait pas leur donner accès au site WEB, parce qu’il comportait des parties peu recommandables, mais on est finalement d’accord pour permettre aux employés de récupérer leurs mails depuis ce site :

* Le réseau "COBALT" (192.192.1.0) aura accès au service de messagerie sur violette.com, mais pas aux autres services.
* Le réseau "CIEL" (192.192.2.0) n’aura toujours pas accès aux réseaux du *reste du monde,* donc pas plus au service de messagerie VIOLETTE.
* Les protocoles de messagerie utilisés seront les protocoles POP3 et SMTP (ports 110 et 25).
* La mise en place du serveur de mail sur violette.com n’a pas été effectuée : il faudra donc définir le domaine (***violette.com***) et créer un compte bluette@violette.com (mot de passe : mdp) sur le serveur **www.violette.com**. *(Onglet Services/bouton EMAIL)*
* Pour faire plus "joli", on ajoutera également une résolution DNS sur le serveur DNS Orange, qui sert de serveur DNS pour toute la maquette, qui résoudra le nom **mail.violette.com** en **202.20.20.80**.

Si on veut être rigoureux, il faut vérifier, avant toute modification, qu’aucun accès POP3 ou SMTP n’est possible sur le serveur de mail, mais comme de toute façon on a interdit tout accès au réseau violette.com, il n’y a pas de raison que cela ait changé. *Faites-le si vous n’êtes pas convaincu !*

### Mise en œuvre du filtrage

Rappel de la liste de règles avant modification : La commande **show access-lists** permet de visualiser les *access-lists* déjà présentes sur le routeur.

Résultats attendus :

Router1#**sh access-lists**

Extended IP access list **102**

**10** permit ip any 192.192.1.0 0.0.0.255 (82 match(es)

**20** permit udp any eq 520 any eq 520 (53 match(es))

Extended IP access list **103**

**10** deny ip 192.192.1.0 0.0.0.255 201.10.10.0 0.0.0.255 (12 match(es)

**20** deny ip 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 (45 match(es)

**30** permit ip 192.192.1.0 0.0.0.255 any (94 match(es))

Router1#

Nous en profitons au passage pour commenter cet affichage :

* Le nombre de 'matches' entre parenthèses indique le nombre de fois qu’un flux a validé la règle. C’est très intéressant pour vérifier que les règles ont une utilité. Ici par exemple :
  + La règle qui permet les réponses vers le réseau 192.192.1.0 ainsi que les requêtes sur ce réseau depuis les autres réseaux extérieurs a été validée 82 fois.
  + La règle qui autorise la réception de mises à jour RIP a été validée 53 fois.
  + La règle qui interdit l’accès au réseau 200.10.10.0 (www.rose.com) a "*matché*" 12 fois.
  + La règle qui interdit l’accès au réseau 200.10.10.0 (www.violette.com) a "*matché*" 45 fois.
  + La règle qui autorise les autres flux sortants depuis le réseau 192.192.1.0 (depuis ÉMERAUDE) a "*matché*" 94 fois.
* Bien que nous n’ayons pas créé l’*access-list* en spécifiant des n° de ligne, le système les a automatiquement numérotées de 10 en 10, dans l’ordre d’exécution, et c’est cette numérotation croissante qui définit l’ordre de parcours des règles.

L’analyse du cahier des charges nous amène immédiatement à comprendre que :

1. La règle d’autorisation pour le service de messagerie devra être prioritaire sur la règle d’interdiction actuelle vers le réseau 202.20.20.0, et devra donc se situer en amont dans la liste.
2. La règle devra spécifier la source (192.192.1.0) pour éviter d’autoriser le réseau CIEL. On pourrait aussi définir une règle d’interdiction pour ce réseau, mais à quoi bon faire compliqué quand on peut faire simple ;-)

Pour modifier une *access-list*, on a deux solutions :

* Soit redéfinir complètement la liste, dans l’ordre adéquat après l’avoir supprimée.
* Soit intercaler les nouvelles règles dans la liste en indiquant un n° de ligne correspondant à l’endroit d’insertion.
* NB : Il est possible aussi de redéfinir une liste en renumérotant les lignes automatiquement si on n’a plus d’espaces pour les insertions.

Exemple : **ip access-list resequence 130 10 10** permet de redéfinir les règles de 10 en 10 pour l’access-list 130 *(1er 10 = point de départ - 2e 10 = pas ou incrément)*

NB : Les deux solutions sont présentées, mais vous utiliserez la 2e solution.

Solution 1

Router1(conf)# **no access-list 103**

Router1(conf)# **access-list 103 deny ip 192.192.1.0 0.0.0.255** **201.10.10.0 0.0.0.255**

Router1(conf)# **access-list 103 permit tcp 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 eq pop3**

Router1(conf)# **access-list 103 permit tcp 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 eq smtp**

Router1(conf)# **access-list 103 deny ip 192.192.1.0 0.0.0.255** **202.20.20.0 0.0.0.255**  
Router1(conf)# **access-list 103 permit ip 192.192.1.0 0.0.0.255 any**

# Remarque : il est inutile de réaffecter l’access-list à l’interface, car la suppression de l’access-list ne supprime pas l’affectation. Mais évidemment, si on ne recrée pas l’access-list, l’association entre access-list et interface n’a aucun effet.

Solution 2

On ne peut pas simplement ajouter les règles, puisqu’elles seraient positionnées en dernier. On peut en revanche les insérer en précisant le n°. D’après le contenu de la liste, il faut l’insérer avant la règle 20, et si on veut obtenir un résultat similaire à la solution 1, on peut l’insérer avec les n° 15 et 16, pour qu’elle soit entre les règles 10 et 20.

Router1(config)#**ip access-list extended 103**

Router1(config-ext-nacl)#**15 permit tcp 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 eq pop3**

Router1(config-ext-nacl)#**16 permit tcp 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 eq smtp**

Router1(config-ext-nacl)#**exit**

Router1(config)#

**On utilisera donc cette 2e solution** pour utiliser le nouveau format de commande.

Après modification l’*access-list* devient :

Router1#**sh access-lists**

Extended IP access list 102

10 permit ip any 192.192.1.0 0.0.0.255 (86 match(es))

20 permit udp any eq 520 any eq 520 (139 match(es))

Extended IP access list 103

10 deny ip 192.192.1.0 0.0.0.255 201.10.10.0 0.0.0.255

15 permit tcp 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 eq pop3

16 permit tcp 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 eq smtp

20 deny ip 192.192.1.0 0.0.0.255 202.20.20.0 0.0.0.255 (1 match(es))

30 permit ip 192.192.1.0 0.0.0.255 any (1 match(es))

Router1#

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

On pourrait re-séquencer la liste pour qu’elle soit renumérotée de 10 en 10, **mais cette fonctionnalité n’est pas implémentée sous Packet Tracer**.

Router1(config)#**ip access-list resequence 103 10 10**

*# inutile d’essayer donc, cela ne fonctionne pas*

Remarque : on ne s’occupe pas de l’*access-list* en entrée de s0/0/0 puisqu’elle autorise toute communication à destination du réseau COBALT

### Test des nouvelles règles

Comme indiqué dans le cahier des charges, il faut configurer le serveur de mail (cliquer sur **set** pour définir le domaine **violette.com**), ajouter la résolution DNS sur le serveur DNS Orange, configurer un client de messagerie (par exemple sur PC1-11).

Ensuite, il suffit de tenter l’envoi de mail et la réception de mail. Vous pouvez soit vous envoyer un message à vous-même *(vous, Bluette)*, soit créer un 2e compte mail sur le serveur de mail.

Les 4 copies d’écran ci-dessous montrent les 4 étapes à suivre.

|  |  |
| --- | --- |
| 61-ServeurMailViolette.JPG | 62-AjoutDNS-MailViolette.JPG |
| 63-ConfigurationClientMail.JPG | 64-TestMail.JPG |

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquementUne image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquementUne image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquementCi-dessus, dans la 4e capture, on voit que l’envoi et la réception ont fonctionné, puisqu’on a reçu le message de test que l’on s’est préalablement envoyé.

## Scénario 7 : autoriser les Pings sortants, mais refuser les pings entrants

Le scénario 7 va nous permettre de découvrir encore une autre sorte de règle, qui concerne le protocole ICMP et permet de distinguer les requêtes ICMP des réponses ICMP.

### Cahier des charges

Chez ***COBALT***, on souhaite se protéger des *ping*, qui sont souvent une première tentative de recherche de failles sur les réseaux ciblés par les hackers. En revanche on souhaite s’autoriser à effectuer des *ping* vers l’extérieur, sans que les réponses soient bloquées.

En résumé, on souhaite donc :

* Bloquer les *ping* initiés par  *le reste du monde ;*
* Continuer à effectuer des *ping* sortants, et donc à recevoir les réponses *du reste du monde.*

NB : Ce cahier des charges n’est pas sans nous rappeler, même si ce n’est pas exactement la même chose, les communications TCP pour lesquelles on ne voulait autoriser que les réponses aux requêtes émises.

### Mise en œuvre du filtrage

ICMP est un protocole, au même titre que TCP ou UDP, donc pas de souci de ce côté-là. Par contre il nous faut différencier les demandes et les réponses. Et il existe effectivement des mots-clés pour cela, autorisés par les *access-lists* CISCO.

Il nous reste ensuite à savoir où placer les nouvelles règles éventuellement nécessaires :

* En sortie de l’interface se0/0/0, les ping sont déjà autorisés, sauf vers les réseaux **violette.com** et **rose.com**, mais c’est volontaire.
  + En tous cas, à priori rien à modifier sur l’access-list 103 (on pourrait interdire les réponses, mais si les demandes de ping sont arrêtées, c’est inutile : il n’y aura pas de réponse à donner ;-)
* En entrée sur l’interface se0/0/0, il faut :
  + interdire les demandes de ping.
  + autoriser les réponses de ping.

Il s’agit donc de l’access-list **102** qui contient actuellement 2 règles :

Extended IP access list 102

**10** permit ip any 192.192.1.0 0.0.0.255 (86 match(es))

**20** permit udp any eq 520 any eq 520 (139 match(es))

Si les pings sont autorisés, c’est qu’ils sont inclus dans la 1re règle qui concerne tout IP. Il faut donc ajouter les règles nécessaires avant la règle n° 10. On peut préciser plus ou moins la règle, en fonction ce qui est déjà paramétré, donc il n’y a pas qu’une seule possibilité.

Mais il faut forcément interdire les requêtes ICMP entrantes avant la règle 10, par exemple en 5.

On modifie donc la règle en insérant une nouvelle règle.

Router1(conf)# **ip access-list extended 102**

Router1(conf)# **5 deny icmp any 192.192.1.0 0.0.0.255 ?**

Le ? nous permet de connaître les options icmp.

Router1(config-ext-nacl)#5 deny icmp any 192.192.1.0 0.0.0.255 **?**

<0-256> type-num

echo                                Echo (ping)

echo-reply                      Echo reply

host-unreachable            Host unreachable

net-unreachable              Net unreachable

port-unreachable            Port unreachable

protocol-unreachable     Protocol unreachable

ttl-exceede                     TTL exceeded

unreachable                   All unreachables

<cr>

Router1(config-ext-nacl)# **5 deny icmp any 192.192.1.0 0.0.0.255 echo**

La solution retenue a consisté à interdire uniquement les echo ICMP en entrée de s0/0/0. Les 'echo-reply' sont de fait autorisés par la règle 10. La règle a été insérée en 5, donc avant la règle d’autorisation en 10.

### Vérification du résultat obtenu

Il faut se souvenir que :

* Les ping vers le réseau ORANGE ne sont plus autorisés, puisque l’*access-list* en entrée sur ce réseau a limité les flux aux services autorisés.
* L’accès à ÉMERAUDE n’est pas possible pour PC1-11 puisqu’il est *blacklisté*.

En revanche :

* Un ping depuis PC1-12 vers **www.emeraude.com** devrait rester possible.
* Un ping des routeurs (ex : 200.1.2.2 ou 200.2.3.3) reste possible également.

Enfin :

* Les ping depuis **rose.com** et **violette.com** n’étaient déjà plus possibles puisque la réponse ne peut pas leur parvenir du fait du filtrage sur Router1.

Donc la vérification ne peut se faire que :

* Entre PC1-12 et ÉMERAUDE pour les ping sortants qui devraient continuer à être autorisés.
* Entre COBALT et les routeurs pour les pings sortants également (qui devraient continuer à être autorisés).
* Depuis ÉMERAUDE vers COBALT, pour les ping entrants qui devraient maintenant être bloqués. (ou depuis un routeur).

Montrez qu’il en est ainsi

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Conclusion : Les ping sortants continuent bien à être autorisés (sauf règle contraire déjà définie). En revanche un ping sur un hôte du réseau COBALT, depuis le serveur **www.emeraude.com,** est bien bloqué désormais.



Vérifiez que vous avez atteint les objectifs en lançant l’auto-correction (check result). La vérification prend quelques secondes, ainsi que le passage d’un onglet à l’autre.

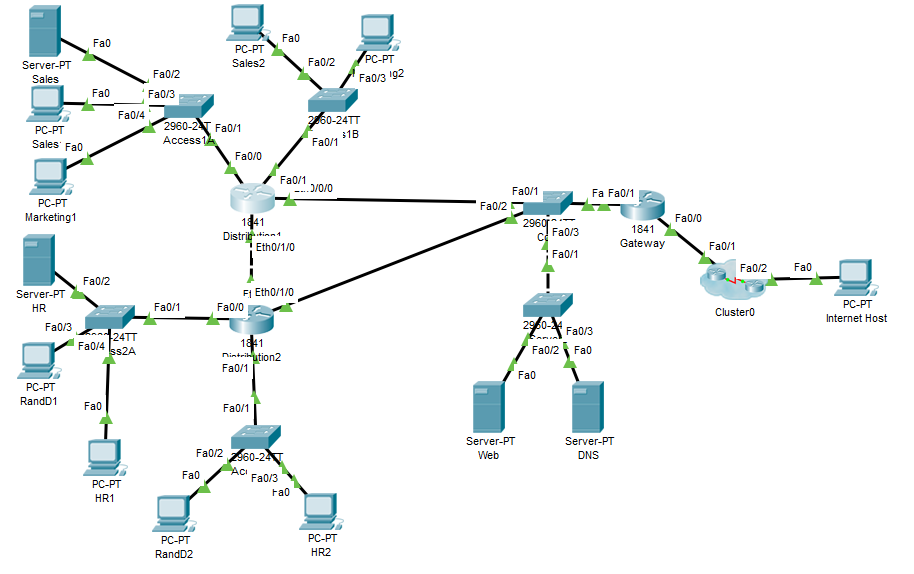
Une image contenant texte, Appareils électroniques, capture d’écran, Police

Description générée automatiquementSoyez patient !

# Énoncé 4 : découverte approfondie des ACLs

On utilisera le fichier suivant : [Bloc2\_sem3-4\_atelier-02\_activite-4-decouverte\_acl-fin.pka](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-4-decouverte_acl-fin.pka)

Soit le réseau suivant



**Emplacement des listes de contrôle d’accès**

## Objectifs

* Vérifier la connectivité du réseau
* Examiner les listes de contrôle d’accès configurées sur les routeurs
* Déterminer l’interface appropriée pour appliquer des listes de contrôle d’accès
* Examiner les effets d’une liste de contrôle d’accès

## Contexte/Préparation

Cet exercice illustre comment l’application d’une liste de contrôle d’accès pour autoriser ou refuser le trafic sur le réseau affecte le flux du trafic réseau. L’administrateur réseau a décidé que tout le trafic web externe accède uniquement au serveur web. En outre, seul le personnel des ressources humaines a accès au serveur HR. La protection des données de salariés est ainsi garantie. C’est pourquoi les listes de contrôle d’accès doivent être implémentées sur le réseau. Un autre technicien réseau a déjà configuré les listes de contrôle d’accès requises sur les routeurs Gateway et Distribution2. Toutefois, ces listes n’ont pas encore été appliquées à une interface. Vous avez pour tâche d’appliquer les listes de contrôle d’accès et de vérifier si le trafic approprié est autorisé ou refusé.

## Étape 1 : vérification de la connectivité du réseau

1. Vérifiez que tous les ordinateurs peuvent communiquer les uns avec les autres, ainsi qu’avec les serveurs.

Une image contenant texte, logiciel, nombre, Icône d’ordinateur

Description générée automatiquement

Une image contenant texte, logiciel, Icône d’ordinateur, nombre

Description générée automatiquement

1. Vérifiez que l’hôte internet peut accéder au serveur web (192.168.0.3), au serveur Sales (192.168.10.2) et au serveur HR (192.168.40.2) à l’aide du navigateur.

Une image contenant texte, capture d’écran, logiciel, ligne

Description générée automatiquement

## Étape 2 : examen des listes de contrôle d’accès configurées sur les routeurs

1. Accédez au routeur Distribution2. Utilisez les commandes suivantes pour afficher la liste de contrôle d’accès configurée sur le routeur Distribution2 :

show running-config

show access-lists 1

Quel est le but de ces listes de contrôle d’accès ?

Les listes de contrôle d'accès (ACL) servent à filtrer le trafic réseau. Dans le cas, l'ACL permet uniquement le trafic provenant du réseau 192.168.40.0/24. Elle est utilisée pour contrôler l'accès aux ressources, comme restreindre les connexions à certaines parties du réseau ou au routeur.

1. Accédez au routeur Gateway. Utilisez les commandes suivantes pour afficher la liste de contrôle d’accès configurée sur le routeur Gateway :

show running-config

show access-lists 100

Quel est le but de ces listes de contrôle d’accès ? Ecrivez en français, la ou les règles de gestion ainsi mises en œuvre.

L'ACL étendue 100 sur le routeur Gateway est conçue pour filtrer le trafic en fonction de critères précis. Elle est utilisée pour contrôler l'accès aux services, en particulier le service HTTP (port 80). Cette ACL contrôle qui peut accéder à un serveur web spécifique et bloque certains types de connexions HTTP.

Autoriser tout le trafic TCP provenant de n'importe quelle adresse vers le serveur 192.168.0.3 sur le port 80 (service HTTP).

Bloquer tout le trafic TCP sur le port 80 provenant de n'importe quelle source vers n'importe quelle destination autre que 192.168.0.3.

Autoriser tout le trafic IP, sans restriction, entre toutes les adresses sources et destinations.

Étape 3 : détermination de l’interface appropriée pour appliquer les listes de contrôle d’accès

Après avoir examiné les listes de contrôle d’accès, déterminez sur quelle interface les appliquer.

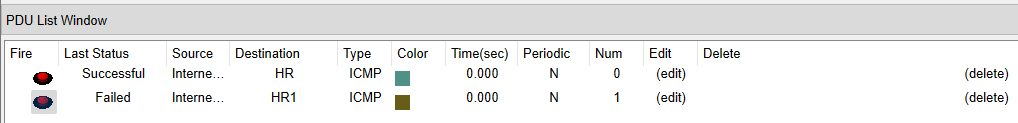
1. Vous devez appliquer une liste de contrôle d’accès à une interface ou à une sous-interface avant qu’elle puisse avoir une incidence quelconque sur le trafic réseau.
2. La liste de contrôle d’accès étendue doit être la plus proche de la source, alors que la liste de contrôle d’accès standard doit être la plus proche de la destination.
3. N’oubliez pas qu’une seule liste de contrôle d’accès est permise par port, par protocole et par direction.
4. Appliquez la liste de contrôle d’accès à l’interface ou à la sous-interface appropriée.

interface FastEthernet0/1

ip access-group 100 out

## Étape 4 : examen des effets d’une liste de contrôle d’accès

1. L’hôte internet doit pouvoir envoyer une requête ping à tout périphérique sur le réseau, à l’exception du serveur HR1 ou HR.



On peut voir qu'un seul des deux a bien été bloqué. Dans ce cas, j'ai bloqué HR1 et testé HR.

1. L’hôte internet doit pouvoir accéder au serveur web (192.168.0.3) à l’aide du navigateur.



1. L’hôte internet ne doit pas pouvoir accéder ni au serveur HR (192.168.40.2) ni au serveur Sales (192.168.10.2) à l’aide du navigateur.

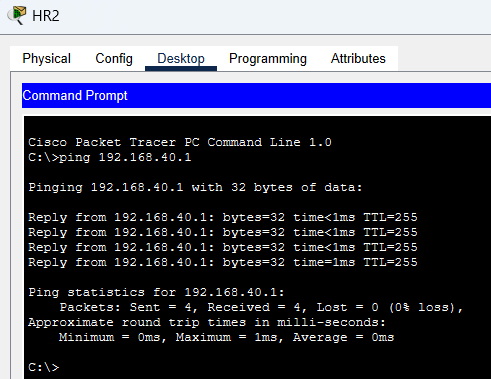
Une image contenant texte, logiciel, Police, ligne

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, ligne

Description générée automatiquement

1. HR2 doit pouvoir accéder au serveur HR (192.168.40.1) à l’aide de la commande ping ou du navigateur.



1. RandD2 ne doit pas pouvoir accéder au serveur HR (192.168.40.1) à l’aide de la commande ping ou du navigateur.

Une image contenant texte, logiciel, ligne, capture d’écran

Description générée automatiquement

1. Comment utiliser les listes de contrôle d’accès pour contrôler le flux du trafic réseau ?

Les listes de contrôle d'accès permettent de filtrer le trafic réseau en définissant des règles basées sur des critères comme les adresses IP, les protocoles, ou les ports. Elles peuvent autoriser ou bloquer des paquets spécifiques à l'entrée ou à la sortie d'une interface réseau. Les ACL étendues filtrent avec plus de précision, tandis que les ACL standard se concentrent sur l'adresse IP source. Cela permet de sécuriser et de gérer efficacement le flux de données dans le réseau.

1. Quelle sera toujours la dernière instruction par défaut dans une liste de contrôle d’accès ?

La dernière instruction qui sera toujours la dernière instruction par défaut dans une liste de contrôle d'accès est "deny any". Cela signifie que tout le trafic qui ne correspond à aucune des règles précédentes est automatiquement bloqué, même si cette instruction n'est pas explicitement écrite dans l'ACL.

## Etape 5 : Protocol-discovery et filtrage du traffic Peer-to-Peer

S’il y a bien une nuisance courante dans un réseau et qui plus  est qui est difficile à endiguer, c’est le trafic généré par les applications basées sur le principe du Peer-to-Peer. Hormis le fait que certaines applications sont souvent utilisées pour du téléchargement illégal, le principe même de leur fonctionnement perturbe aisément une structure, et tout particulièrement en ce qui concerne le point de vue de la connexion internet.

Si l’on veut empêcher le traffic Peer-to-peer, on butte généralement rapidement sur le fait que le panel des ports utilisés par ces applications est énorme, et qu’il est difficilement envisageable de les bloquer tous, sous peine d’empêcher d’autres applications de fonctionner correctement.

Cisco apporte un outil efficace sur ses routeurs (pour autant que l’IOS le permette bien entendu). Il s’agit de NBAR « Network-Based Application Recognition », une fonctionnalité qui va analyser les données au delà de la couche réseau ou de la couche de transport afin d’identifier un flux et de pouvoir le catégoriser.

Lorsqu’on utilise NBAR conjointement avec des « class-map » et « policy-map », on peut alors filtrer le trafic identifié.

Voici un configuration simple mais fonctionnelle (basée sur un routeur Cisco 2801 IOS IPBase 15.0M)…

### Activer la découverte de protocol sur l’interface où l’on veut appliquer le filtrage

Si on ne l’active pas, le filtrage n’aura pas lieu, simplement parce que le routeur n’ira pas inspecter les flux de données.

Router> enable

Router# configure terminal

Router(config)# interface fastethernet 0/1

Router(config-if)# ip nbar protocol-discovery

Router(config-if)# exit

Router(config)#

### Créer une « class-map » de sorte à identifier le traffic généré par les applications P2P

On va ici faire en sorte que le traffic identifié comme provenant d’applications utilisant les protocoles Bittorrent, eDonkey, Gnutella ou encore Fasttrack soient catégorisés.

Router(config)# class-map match-any P2P

Router(config-cmap)# match protocol bittorrent

Router(config-cmap)# match protocol edonkey

Router(config-cmap)# match protocol gnutella

Router(config-cmap)# match protocol fasttrack

Router(config-cmap)# exit

Router(config)#

Attention à bien créer une class-map « match-any » … celà signifie que dès qu’une des conditions (ici un des protocoles) est remplie, le trafic est mis dans la classe « P2P ». Par défaut la commande class-map crée une classe « match-all » qui demande que tous les « match » soient vérifiés pour que le trafic soit associé à la classe.

### Définition de la police à appliquer sur l’interface

Avant de configurer, ayez bien en tête qu’une police s’applique sur une interface, soit en entrée, soit en sortie ou les deux. Mais qu’une interface ne peut avoir qu’une seule police applique par interface et par sens du trafic.

On va ici créer une police qui filtrera tous les trafics de la classe P2P. Le trafic reconnu par la classe « P2P » sera purement et simplement jeté.

Router(config)# policy-map DROP-P2P

Router(config-pmap)# class P2P

Router(config-pmap-c)# drop

Router(config-pmap-c)# exit

Router(config-pmap)# exit

Router(config)#

### Appliquer la police à l’interface

L’interface utilisée ici est l’interface côté LAN, on va donc filtrer le trafic qui entre sur cette interface

Router(config)# interface fastethernet 0/1

Router(config-if)# service-policy input DROP-P2P

Router#

### Remarques

* La commande « show policy-map interface fastethernet 0/1″ afficher les stats de la police active sur l’interface.
* Tout le trafic P2P ne sera pas droppé, après quelques tests certains torrents passent, mais à un taux de transfert proche 0 et généralement il s’agissait de torrents « légaux », par exemple pour un téléchargement d’une ISO Ubuntu alors que des tests de téléchargements de torrents « douteux » quant à eux ne démarraient jamais. Toujours est-il que c’est un grand pas vers la suppression de cette nuisance.
* La commande « show ip nbar protocol-discovery » vous en apprendra beaucoup sur les protocoles utilisés et identifiés sur l’interface ou nbar est activé. (traffic HTTP, Skype, ftp …).

**Travail à faire** : mettre en œuvre cette ACL

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

# Les access-list réflexives

Lorsque l’on veut protéger un réseau du trafic venant de l’extérieur, on est souvent confronté  au problème suivant : « Comment prévoir ce qui peut/devrait pouvoir passer ? ». D’un point de vue général, la politique la plus sécuritaire consiste à bloquer tout trafic par défaut et n’autoriser que le strict nécessaire, mais cela devient vite difficile à gérer si on ne veut pas trop brider les communication provenant de l’intérieur du réseau.

Il existe plusieurs méthodes pour gérer au mieux cette situation:

* Le filtrage sur base des flags TCP (un segment tcp entrant n’ayant pas le flag ACK activé ne peut pas être originaire de l’intérieur, …), l’inconvénient majeur, c’est qu’il n’est pas possible d’en faire de même pour le protocole UDP.
* Le filtrage par Access-list réflexive, ce que nous allons voir ici, qui consiste à générer une ACL dynamiquement en fonction du trafic sortant afin d’autoriser temporairement le retour.
* Les « Zone-Based Firewall » qui se basent sur le principe de définition de zones et de création de règles d’une zone à une autre (la méthode la plus moderne).

Voyons ce qu’il en est pour les ACL réflexives …

Une image contenant diagramme, capture d’écran, ligne, texte

Description générée automatiquement

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

## La topologie

Un LAN internet (192.168.0.0/24) est connecté à Internet par l’intermédiaire de R2 qui est configuré pour faire du NAT overload.

Les trois routeurs utilisent Ripv2 comme solution de routage.

## Objectif

Configurer R2 de sorte qu’il n’autorise les paquets provenant d’Internet à l’unique condition qu’ils correspondent à une « conversation » initiée depuis le LAN interne.

## Méthode générale

1. Configurer une ACL définissant le trafic autorisé à sortir (LAN vers Internet) en spécifiant l’ACL dynamique à peupler.
2. Appliquer cette ACL sur une interface.
3. Créer un ACL correspondant au trafic autoriser à entrer où l’on fait évaluer l’ACL dynamique.
4. Appliquer cette ACL en sens inverse de l’ACL pour le trafic sortant.

## Considération dans le cas présent

Le NAT ne nous facilite pas la vie. Il faut garder en mémoire que l’opération du NAT (modification de l’adresse source du paquet) est effectuée avant le filtrage. Dés lors si on n’y prête pas attention on risque fort de ne pas arriver à identifier correctement le trafic.

Dés lors il faut commencer par se fixer une base de travail, à savoir, à quel moment le filtrage aura lieu. Si on le fait sur l’interface de R2 côté LAN, on devra tenir compte des adresses locales, si on filtre à la sortie de l’interface côté publique, il faudra traiter les données en fonction de l’adresse utilisée par le NAT.

Pour la configuration, on filtrera la trafic côté LAN de R2. L’avantage majeur est que l’on peut distinguer les adresses privées, l’inconvénient c’est que le routeur effectuera l’opération de routage même si les paquets termineront à la poubelle juste après.

## Configuration

Commençons par créer l’ACL autorisant le trafic sortant:

R2(config)#ip access-list extended ACL-L2W

R2(config-ext-nacl)#permit ip 192.168.0.0 0.0.0.255 any **reflect** **ACL-REFLECTED**

R2(config-ext-nacl)#exit



ACL-L2W (Lan to Wan) sera l’ACL que l’on appliquera sur l’interface côté LAN de R2 en « input ». On y autorise le trafic ip (n’importe quel protocole) provenant de 192.168.0.0/24 vers n’importe quelle direction.

L’instruction **reflect** indique au routeur qu’il devra créer une règle réflexive (correspondant au trajet inverse) dans l’access-list dynamique **ACL-REFLECTED**.

Créons maintenant l’ACL qui filtrera le trafic entrant et que l’on appliquera de nouveau sur l’interface LAN de R2 mais cette fois en « output ».

R2(config)#ip access-list extended ACL-W2L

R2(config-ext-nacl)#**evaluate ACL-REFLECTED**

R2(config-ext-nacl)#exit



ACL-W2L (Wan to Lan) ne contient ici qu’une seule instruction, **evaluate** **ACL-REFLECTED**, qui indique au routeur qu’il doit utiliser les règles cntenues dans l’ACL ACL-REFLECTED. Pour ceux qui ont des notions de programation, cela reviendrait à faire un « include » en PHP par exemple.

Il reste maintenant à appliquer ces deux ACL sur l’interface LAN (se1/0 dans le cas présent) de R2.

R2(config)#int se1/0

R2(config-if)#ip access-group ACL-L2W **in**

R2(config-if)#ip access-group ACL-W2L **out**

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

## Vérification

Commençons par vérifier si R1 arrive bien à communiquer avec R3, cela devrait fonctionner puisqu’il génèrerait des paquets provenant de 192.168.0.10 à destination de 80.0.0.1, ce qui correspond à la clause « permit » de ACL-L2W.

Une image contenant texte, capture d’écran, Police, jaune

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

A priori tout va bien … vérifions le fonctionnement des ACLs…

Une image contenant texte, capture d’écran, Police, jaune

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

L’access-list 1 est utilisée ici pour le NAT. Ce qui nous intéresse ici, c’est l’ACL-L2W … « 12 matches » … les 12 paquets icmp générés par le ping sont bien passés et ont été traités correctement par l’ACL. Mieux encore … l’ACL-REFLECTED contient une règle qui correspond au trafic correspondant à la réponse de R3 au ping de R1: des paquets ICMP en provenance de 80.0.0.1 à destination de 192.168.0.10.

Nous pouvons avoir une confirmation que si le dialogue est initié par un élément autre que dans le réseau interne, cela ne fonctionne pas en pingant l’adresse de PC1 par PC2.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Une image contenant capture d’écran, texte, Police

Description générée automatiquement

Réaliser la même analyse en utilisant les PC et montrer plus précisément le rejet des trames en provenance de PC2 vers PC1, via l’utilisation de l’ACL.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

L'ACL-L2W permet le trafic sortant du LAN vers le WAN, et l'ACL-REFLECTED gère les réponses entrantes via des règles réflexives créées automatiquement. L'ACL-W2L évalue ces règles réflexives pour autoriser le trafic entrant uniquement si initié depuis le LAN. Le système fonctionne bien, avec des paquets autorisés selon les connexions établies.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Le ping de PC1 vers PC2 est bloqué par l'ACL, indiquant que les communications initiées de l'extérieur ne sont pas autorisées. Cela confirme que l'ACL réflexive fonctionne correctement.

## Quelques remarques

* Comme je l’ai dit un peu plus haut, le NAT est assez gênant dans le sens où il nous impose de travailler en in et out du même côté de R2. Si les ACLs étaient appliquées du côté WAN de R2, il aurait fallu remplacer la règle « permit ip 192.168.0.0 0.0.0.255 ….  » par « permit ip host 80.0.0.2 any … « , puisqu’à la sortie de l’interface WAN de R2, l’adresse source des paquets est remplacée par l’adresse configurée sur son interface WAN.
* Point important … l’utilisation d’access-list réflexive ne peut se faire que dans des access-list étendues.
* Il est tout à fait possible de mélanger des règles statiques avec une ou plusieurs instructions « evaluate », il faut juste garder en tête qu’une ACL s’exécute séquentiellement, règle après règle, par ordre de séquence. Les règles contenues dans l’ACL évaluées seront donc exécutées comme si elles étaient définies à cet endroit dans l’ACL.
* En l’absence de NAT, il vaut clairement mieux appliquer l’ACL autorisant le trafic sortant côté interne en input, et l’ACL filtrant le trafic entrant côté extérieur en « input » également. Cela évite au routeur de traiter des paquets qui ne sont pas autorisés.

# Énoncé 9 : ACL et DMZ

Fichier utiliser pour l’énoncé 9 : [Bloc2\_sem3-4\_atelier-02\_activite-6-ACL-DMZ.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-6-ACL-DMZ.pkt)

## But

Le but de se TP est de se familiariser avec l’utilisation des listes de contrôle d’accès étendues. Pour illustrer leur utilisation, vous allez simuler la mise en place d’une DMZ (Zone Démilitarisée) pour un réseau d’entreprise qui héberge son propre site web.

## Travail à réaliser

### Le réseau

Réalisez la structure de réseau suivante :

|  |
| --- |

### Routage et DNS

**Appliquez à tous les routeurs le protocole RIP.**

**Private :**

Private(config)#router rip

Private(config-router)#network 172.16.0.0

Private(config-router)#network 10.0.0.0

Private(config-router)#version 2

**Public :**

Public(config)#router rip

Public(config-router)#network 10.0.0.0

Public(config-router)#network 216.90.80.0

Public(config-router)#version 2

**Internet :**

Internet(config)#router rip

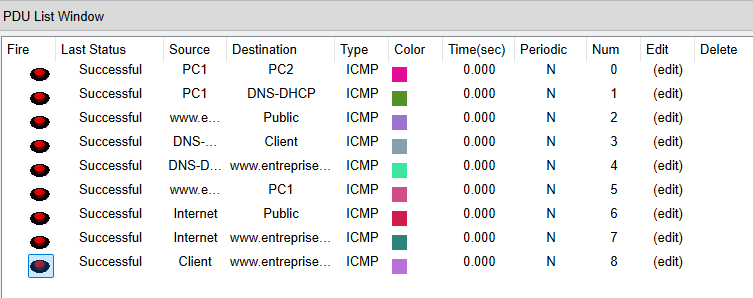
Internet(config-router)#network 216.90.90.0

Internet(config-router)#network 216.90.100.0

Internet(config-router)#network 216.90.80.0

Internet(config-router)#version 2

Vérifiez que tous les postes et serveurs puissent communiquer entre eux.



Pinguez [www.entreprise.com](http://www.entreprise.com/) depuis Client:

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Pour permettre de pinguer à partir des noms de domaine, les DNS doivent être renseignés :

* Client : DNS 216.90.100.1
* Réseau entreprise : DNS 172.16.0.1

En mode simulation (filtrer les protocoles ICMP et DNS), pinguer [ww.entreprise.com](http://www.entreprise.com/) depuis Client puis, cliquer sur l’enveloppe du message final du protocole DNS (le rechercher dans la liste des événements et double-cliquer dessus pour faire apparaître l’enveloppe du message sur le schéma).

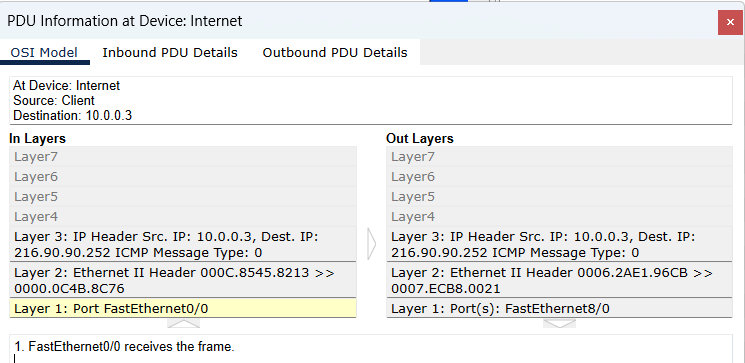
Consultez la partie DNS de la trame (Inbound PDU Details). Quelle est l’adresse renvoyée par le DNS ?

Une image contenant texte, capture d’écran, Parallèle, nombre

Description générée automatiquement

Dans la section de la réponse DNS dans les détails PDU. On peut voir l'adresse IP renvoyée par le serveur DNS pour le nom de domaine [**www.entreprise.com**](http://www.entreprise.com) est **10.0.0.3.**

Cliquez sur l’enveloppe du message final ICMP. Quelles sont les adresses sources et destinations ?



On voit que dans le message final ICMP, l'adresse source est 10.0.0.3 et l'adresse de destination est 216.90.90.252.

## Translation d’adresses

Pour protéger le réseau de l’entreprise et limiter le nombre d’adresses publiques à utiliser, on a recours à la translation d’adresses privées en adresses publiques :

Vu de l’extérieur (internet), le réseau de l’entreprise a pour adresse 216.90.80.254 et son serveur web 216.90.80.253.

* Lors du passage d’une trame par le routeur « public » dans le sens private → public, les adresses privées (176.16.0.0) sont remplacées par l’adresse externe du routeur (216.90.80.254). L’adresse interne est stockée dans une table NAT pour la remettre en place lors du retour de la trame.
* Lors du passage d’une trame par le routeur « public » dans le sens public → private, la seule adresse routable est celle du serveur web de l’entreprise (216.90.80.253). Elle est donc remplacée par son adresse interne (10.0.0. 3) à l’allée et remise en place au retour. Indiquez au routeur quelles sont ses interfaces interne et externe :

| *Interface interne* | *Interface externe* |
| --- | --- |
| Public (config)# interface FastEthernet1/0  Public (config)# ip nat inside | Public (config)# interface FastEthernet0/0  Public (config)# ip nat outside |

Écrivez la règle de translation NAT statique permettant de translater 10.0.0. 3 en 216.90.80.253 :

Public(config)# ip nat inside source static 10.0.0.3 216.90.80.253



Pour permettre la translation des adresses du réseau 172.16.0.0 en l’adresse publique du routeur, il faut créer une règle de translation dynamique. Il faut créer une liste qui contient les adresses réseau à translater.

Créez une access-list standard numérotée 1 et permettant l’accès au réseau 172.16.0.0 :

Public(config)# access-list 1 permit 172.16.0.0 0.0.255.255



Créez la règle de translation NAT dynamique ayant pour source la liste 1 et pour adresse translatée, l’interface publique du routeur

Public(config)# ip nat inside source list 1 interface FastEthernet1/0 overload



Procédez aux tests de connectivité consultez les trames pour observer la translation d’adresses.

| *Source* | *Destination* | *Résultat – Adresses translatées* |
| --- | --- | --- |
| PC1 ou 2 | Client | 216.90.80.253 |
| www.entreprise.com | Client | 216.90.80.253 |
| Client | PC1 ou 2 | 216.90.80.254 |
| Client | www.entreprise.com | 216.90.80.253 |

La translation d’adresses à surtout pour but de cacher le réseau interne de l’extérieur, mais si on connaît le type d’adresses en vigueur dans le réseau, on peut peut toujours y accéder directement. Il faut donc protéger l’accès au réseau en appliquant des règles de filtrage. Le DNS d’internet (DNS-WEB) doit être mis à jour avec l’adresse translatée de [www.entreprise.com](http://www.entreprise.com).

## Protection du réseau de l’entreprise.

Le réseau de l’entreprise doit être parfaitement hermétique à toute intrusion de l’extérieur. Cependant, les postes de ce réseau doivent toujours pouvoir accéder à internet.

On affectera donc à l’interface côté réseau d’entreprise, une liste de contrôle d’accès en entrée et en sortie :

En entrée :

* Autoriser tous les protocoles IP venant 172.16.0.0 vers tous les hôtes.

En sortie :

* Autoriser le protocole TCP en provenance du port 80 de tous les hôtes à destination du réseau 172.16.0.0 s’il s’agit de la réponse à une de ces requêtes.
* Autoriser le protocole ICMP en provenance de la DMZ à destination du réseau d’entreprise
* Autoriser le protocole ICMP en provenance de tous les hôtes et à destination du réseau d’entreprise lorsqu’il s’agit d’une réponse à une de ces requêtes.
* Le protocole ICMP est utile pour tester la connectivité et s’assurer que les problèmes éventuels ne sont pas liés au câblage, à l’alimentation des hôtes ou au routage. Cependant, on autorisera seulement son utilisation depuis le réseau de l’entreprise ou depuis la DMZ.

Écrivez les listes de contrôle d’accès correspondantes :

| *En entrée*  *ACL 101* | *private(config)# access-list 101 permit ip 172.16.0.0 0.0.255.255 any* |
| --- | --- |
| **En sortie**  **ACL 100** | *private(config)# access-list 100 permit tcp any eq 80 172.16.0.0 0.0.255.255 established*  *private(config)# access-list 100 permit icmp 10.0.0.0 0.0.0.7 172.16.0.0 0.0.255.255*  *private(config)# access-list 100 permit icmp any 172.16.0.0 0.0.255.255 echo-reply* |

Activez les listes sur l’interface du routeur :

| private(config)# interface Fa1/0  private(config-if)# ip access-group 101 in  private(config-if)# ip access-group 100 out |
| --- |

Procédez aux tests de connectivité consultez les trames au passage du routeur « private ». Vérifiez l’application des règles de filtrages.

| **Source** | **Destination** | **Résultat – Adresses translatées** |
| --- | --- | --- |
| PC1 ou 2 | Client | 216.90.80.254 |
| www.entreprise.com | Client | 216.90.80.253 |
| Client | PC1 ou 2 | 172.16.0.2 |
| Client | www.entreprise.com | 10.0.0.3 |

En mode simulation, filtrez les protocoles DNS et HTTP.

Depuis le navigateur web de PC1 ou PC2, entrez l’adresse des sites WEB [www.entreprise.com](http://www.entreprise.com/) et [www.siteweb.com](http://www.siteweb.com/). Vérifiez la possibilité d’accéder aux deux sites web.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

## Protection de la DMZ

La DMZ est accessible depuis internet pour tous les protocoles. Or celle-ci n’héberge qu’un serveur web. Le seul protocole qui devrait être autorisé à la pénétrer est le HTTP soit les requêtes TCP à destination du port 80 du serveur web de l’entreprise.

Cependant, il ne faut pas oublier que les hôtes du réseau de l’entreprise doivent pouvoir accéder à internet. Il faut donc laisser passer tous le trafic TCP en provenance d’un port 80 lorsqu’il s’agit d’une réponse.

Notons aussi que les routeurs entretiennent leur table de routage automatiquement grâce au protocole RIP. Le routeur « public » doit pouvoir propager ce protocole vers les autres routeurs et l’accepter des autres routeurs. Comme il n’est pas directement possible de filtrer RIP, on filtrera les protocoles IP en entrée lorsque la source des trames est l’interface de sortie du routeur « Internet ».

Pour assurer l’administration du réseau, on autorisera aussi le protocole ICMP lorsqu’il s’agit d’une réponse et lorsque la destination est l’adresse de substitution du serveur web de l’entreprise.

Vous numéroterez cette liste de contrôle d’accès 111 et vous l’appliquerez à l’interface d’entrée de la DMZ (côté internet) :

public(config)# access-list 111 permit tcp any host 216.90.80.253 eq 80

public(config)# access-list 111 permit tcp any eq 80 172.16.0.0 0.0.255.255 established

public(config)# access-list 111 permit udp any any eq 520

public(config)# access-list 111 permit icmp any host 216.90.80.253 echo-reply

public(config)# interface FastEthernet0/1

public(config-if)# ip access-group 111 in

Effectuez les tests de connectivité et vérifiez les trames au passage du routeur « public ».

Procédez aux tests de connectivité consultez les trames au passage du routeur « private ». Vérifiez l’application des règles de filtrages.

| **Source** | **Destination** | **Résultat – Adresses translatées** |
| --- | --- | --- |
| PC1 ou 2 | Client | 216.90.80.254 |
| www.entreprise.com | Client | |  | | --- | |  |  |  | | --- | | 216.90.80.253 | |
| Client | PC1 ou 2 | 172.16.0.2 |
| Client | www.entreprise.com | 10.0.0.3 |

En mode simulation, filtrez les protocoles DNS et HTTP.

Depuis le navigateur web de PC1 ou PC2, entrez l’adresse des sites WEB [www.entreprise.com](http://www.entreprise.com/) et [www.siteweb.com](http://www.siteweb.com/). Vérifiez la possibilité d’accéder aux deux sites web.

Faites la même chose depuis Client.

Pour vérifier l’hermétisme aux autres protocoles, modifiez la référence au DNS de Client par 216.90.80.253 et refaites depuis le navigateur de client une tentative d’accès à [www.entreprise.com](http://www.entreprise.com/).

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Consultez la trame au passage du routeur « public ». Quelle règle s’est appliquée ?

La règle qui s'est appliquée est celle de l'ACL 111 sur le routeur public, plus précisément la règle qui permet les connexions TCP vers le port 80 (HTTP) à destination de l'adresse 216.90.80.253 (le serveur web dans la DMZ).

## Conclusion

Le serveur web de l’entreprise devra aussi héberger un serveur FTP. Quelles sont les règles à ajouter ?

Autoriser le port 21 (FTP contrôle) pour les commandes FTP vers le serveur :

public(config)# access-list 111 permit tcp any host 216.90.80.253 eq 21

Autoriser le port 20 (FTP données) pour les transferts FTP actifs :

public(config)# access-list 111 permit tcp any host 216.90.80.253 eq 20

# Énoncé 10 : VLAN Access-List (VACL)

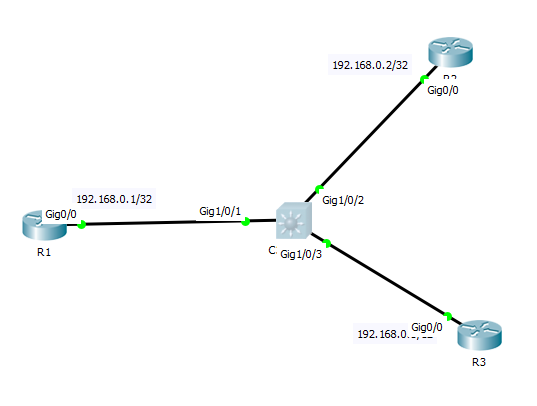
Nous utiliserons le fichier suivant : [Bloc2\_sem3-4\_atelier-02\_activite-7-acl-vlan.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-7-acl-vlan.pkt) dont la topologie est la suivante :

Les ACLs (Access Control Lists) traditionnelles, aussi parfois appelées RACL (Router-based ACL) nous permettent de filtrer du trafic qui passe d’un réseau à un autre, donc dans le cas de VLANs,… le trafic qui est routé d’un VLAN à l’autre. En aucun cas celles-ci ne permettent de filtrer le trafic qui circule au sein d’un VLAN. Pourtant… c’est faisable…

Pour cela on fait appel aux VACLs (Vlan Access Control List), celles-ci, une fois appliquées à un VLAN vont nous permettre d’autoriser ou non la propagation d’une trame et même éventuellement de contrecarrer la logique de commutation du switch en redirigeant celle-ci vers une interface spécifique.

Voici comment configurer ces VACLs…

## La topologie



Nous avons ici trois routeurs, qui serviront ici de simples hôtes dans un VLAN.

* R1 : 192.168.0.1/24
* R2 : 192.168.0.2/24
* R3 : 192.168.0.3/24

Le switch (c3750-1) est ici l’unique élément intéressant. Afin de concentrer le config sur les VACLs, tous les ports de celui-ci sont placés dans le VLAN10, en mode accès.

## Configuration initiale du switch

C3750-1(config)#vlan 10

C3750-1(config-vlan)#exi

C3750-1(config)#int ran gi1/0/1-24

C3750-1(config-if-range)#sw mod acc

C3750-1(config-if-range)#sw acc vlan 10

C3750-1(config-if-range)#span

C3750-1(config-if-range)#spanning-tree portfast

%Warning: portfast should only be enabled on ports connected to a single

host. Connecting hubs, concentrators, switches, bridges, etc... to this

interface when portfast is enabled, can cause temporary bridging loops.

Use with CAUTION

…

C3750-1(config-if-range)#ex

Petits tests de routine, R1, R2 et R3 doivent pouvoir communiquer sans problème…

R1#ping 192.168.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms

R1#ping 192.168.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms

R1#

R2#ping 192.168.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms

R2#

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

## Objectif de la configuration

Alors que sans VACLs, les trois routeurs peuvent communiquer entre eux sans problème, nous allons faire en sorte que R1 puisse communiquer avec R2 et R3… mais que R2 et R3 ne puissent pas communiquer entre eux.

## Configuration de la VACL

De manière générale, une VACL fonctionne sur un principe similaire à celui des route-maps. Il s’agit d’une liste ordonnée de règles, chacune ayant un numéro de séquence. Pour chacune de ces règles, nous devons identifier le trafic correspondant à l’aide d’une clause « match », à laquelle nous ferons correspondre une « action » qui peut être l’une des trois suivantes :

* **forward** : le trafic est traité normalement en suivant la logique de commutation du switch.
* **drop** : le trafic est rejeté.
* **redirect** : le trafic est redirigé vers une interface spécifique, indépendamment de la logique de commutation du switch.

Les clauses « match » utilisent des ACL (soit IP, soit MAC). C’est donc par-là que nous devrons commencer.

Chose importante… À l’instar d’une ACL classique, les VACL rejettent tout ce qui n’est pas permis. Une VACL sans règle « forward » bloquera tout simplement tout le trafic dans le vlan donné.

### Étape n° 1 : Créer une ACL (classique) pour identifier le trafic à traiter

Attention à la confusion, l’ACL sert ici à identifier un trafic spécifique. Il faut donc qu’une règle « permit » corresponde à ce que l’on souhaite rejeter ou autoriser, indépendamment de l’action. Ici, ce que nous voulons identifier, c’est le trafic entre R2 et R3, parce que nous voulons le bloquer. Donc il faut une règle "permit" qui y corresponde.

C3750-1(config)#ip access-list extended vlan10

C3750-1(config-ext-nacl)#per

C3750-1(config-ext-nacl)#permit ip host 192.168.0.2 host 192.168.0.3

C3750-1(config-ext-nacl)#permit ip host 192.168.0.3 host 192.168.0.2

C3750-1(config-ext-nacl)#ex



### Étape n° 2 : Créer la VACL pour associer des « match » à des « action »

**Désolé, on ne peut pas le faire sous packet tracer**, mais cela a permis de mesurer une nouvelle possibilité de faire communiquer plusieurs routeurs entre-eux ?

Nous allons donc créer ici une Vlan Access Map (l’élément qui combine les matches avec les actions), munie de deux règles.

1. Le trafic correspondant à l’ACL VLAN10 doit être « droppé » (**n° de séquence 10**)
2. Le reste du trafic doit être « forwardé » (**n° de séquence 20**)

3750-1(config)#vlan access-map **VMAP-VLAN10 10**

3750-1(config-access-map)#match ip address **VLAN10**

3750-1(config-access-map)#action drop

3750-1(config-access-map)#exit

3750-1(config)#vlan access-map **VMAP-VLAN10 20**

3750-1(config-access-map)#action forward

3750-1(config-access-map)#exit

### Étape n° 3 : Appliquer la VACL au(x) VLAN(s) désiré(s)

Il ne reste plus qu’à appliquer ces règles au **Vlan 10**.

3750-1(config)#vlan filter **VMAP-VLAN10** vlan-list **10**

## Test de la configuration

Si tout se passe comme prévu, un ping entre R1 et R2, ou entre R1 et R3 devrait fonctionner, tandis qu’entre R2 et R3, le trafic devrait être bloqué…

**R1#ping 192.168.0.2**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:

**!!!!!**

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R1#

R1#

**R1#ping 192.168.0.3**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

**!!!!!**

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

R1#

**R2#ping 192.168.0.3**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

**.....**

Success rate is 0 percent (0/5)

R2#

Tout fonctionne comme prévu ! Nous arrivons donc bien ici à filtrer du trafic au sein du VLAN10, chose impossible avec les ACLs classiques.

# Enoncé 11 : Mise en place d'une Zone-based Policy Firewall (ZPF) - défi

## Qu’est-ce que ZPF ?

ZPF permettant de filtrer les communications en se basant sur des zones, en effet lors de l’implémentation de ZPF nous allons segmenter notre réseau en plusieurs zones (typiquement : LAN, WAN et DMZ). Puis, nous allons définir la politique à appliquer entre chaque zone c’est-à-dire qu’on va spécifier le trafic autorisé ou interdit entre une zone A et une zone B, mais aussi le trafic inverse entre zones B et A.

**ZPF présente plusieurs avantages :**

* ZPF ne dépend pas des interfaces.
* ZPF permet de visualiser le réseau facilement grâce aux différentes zones.
* Utilise le **C3PL** (**C**isco **C**ommon **C**lassification **P**olicy **L**anguage) ce qui rend la configuration de notre politique facile à comprendre et à débugger.

**ATTENTION : Le trafic au sein d’une même zone n’est pas filtré par les règles ZPF.**

### La notion de "ZONE"

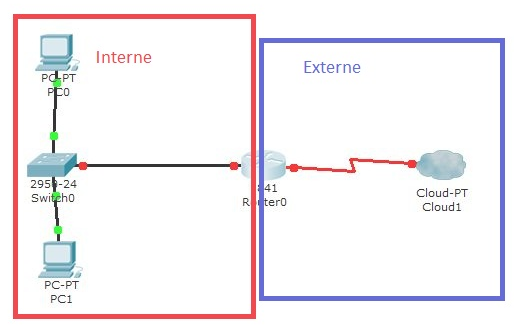


Figure 1 : Exemple 1 à base de routeur bi-pattes

Dans ce cas de figure très classique, la zone INTERNE correspond au LAN et la zone externe au WAN.

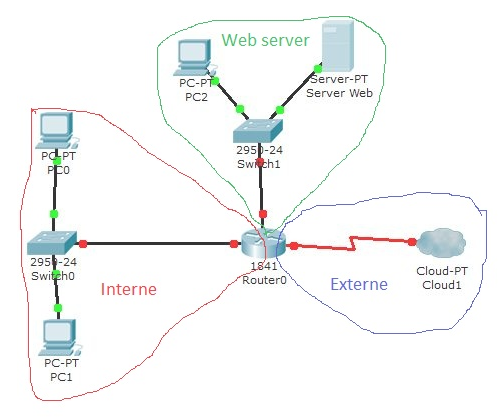


Figure 2 : Exemple 2 à base de routeur tri-pattes

Encore un cas très classique avec ici 3 zones : LAN, WAN et DMZ.

### Les actions de ZPF

Lors de l’analyse du trafic, ZPF peut appliquer 3 actions :

* **Inspect** (inspecter le paquet)
* **Drop** (supprimer le paquet)
* **Pass** (autoriser le paquet)

## Configurer ZPF

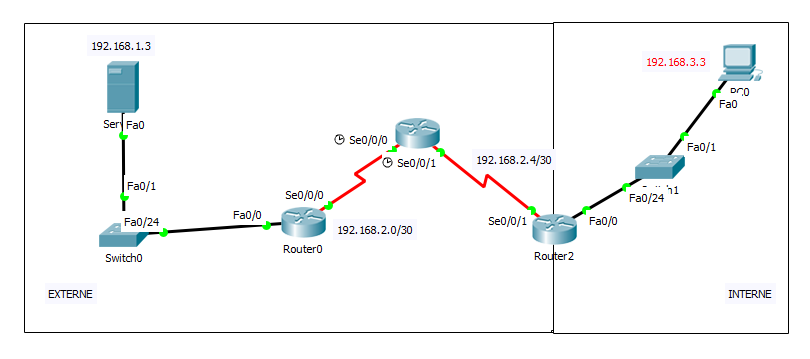
On utilisera le fichier : [Bloc2\_sem3-4\_atelier-02\_activite-8-zpf.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_activite-8-zpf.pkt)

Vous trouverez ci-dessous les éléments pour débuter la solution au problème posé qui est de n’autoriser que le protocole FTP du poste vers le serveur.

La configuration de ZPF se fait en plusieurs étapes :

* Création des zones.
* Création d’une class-map.
* Création d’une policy-map.
* Création des zones paire.
* Affecter les interfaces à une zone.

Nous allons nous baser sur l’exemple suivant :



### Création des zones

Nous allons donc configurer les zones ZPF sur le router R2 :

R2(config)#zone security INTERNE

R2(config-sec-zone)#exit

R2(config)#zone security EXTERNE

R2(config-sec-zone)#exit

Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement

### Création de la Class-Map

La class-map va nous permettre de spécifier quel trafic nous allons analyser.

Nous pouvons analyser le trafic selon le protocole utilisé (TCP, UDP, ICMP…), mais aussi en fonction d’une ACL.

Dans cet exemple, on va créer une ACL étendue qui va spécifier le trafic sortant du réseau interne (192.168.3.0/24) vers tous les autres réseaux.

R2(config)#access-list 101 permit ip 192.168.3.0 0.0.0.255 any



Nous allons maintenant créer une class-map nommée **INTERNE-EXTERNE-CMAP** et spécifier qu’il faut analyser le trafic correspondant à mon ACL précédemment créée correspondant au protocole FTP.

R2(config)#class-map type inspect INTERNE-EXTERNE-CMAP

R2(config-cmap)#match access-group 101

R2(config-cmap)#match protocol FTP

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Nous pouvons bien sûr créer d’autres Class-Map pour spécifier d’autres types de trafic.

### Création de la policy-map

Les commandes suivantes permettent de créer une policy-map nommée **INTERNE-EXTERNE-PMAP** qui va définir l’action **inspect** pour la class-map précédemment créée (nous pouvons bien sûr utiliser une autre des 3 actions possibles).

R2(config)#policy-map type inspect INTERNE-EXTERNE-PMAP

R2(config-pmap)#class INTERNE-EXTERNE-CMAP

R2(config-pmap-c)#inspect



### Création de la zone paire

Nous allons créer une “zone-paire” qui va permettre de définir une zone source et une zone de destination sur lesquelles nous allons appliquer notre policy-map.

R2(config)#zone-pair security INTERNE-TO-EXTERNE source INTERNE destination EXTERNE



R2(config-sec-zone-pair)#service-policy type inspect INTERNE-EXTERNE-PMAP



Ces commandes nous permettent de créer une zone paire nommer INTERNE-TO-EXTERNE qui a pour zone source INTERNE et destination EXTERNE c’est-à-dire que cette zone-paire spécifie la politique à appliquer pour le trafic circulant de la zone INTERNE vers la zone EXTERNE, ce trafic sera alors soumis aux politiques définies dans notre policy-map.

### Affectation des interfaces à une zone

Cette dernière étape consiste à affecter nos interfaces à une zone.

R2(config)#interface fastEthernet 0/1

R2(config-if)#zone-member security INTERNE

R2(config)#interface se0/0/1

R2(config-if)#zone-member security EXTERNE

Une image contenant texte, reçu, Police, blanc

Description générée automatiquement

Ces commandes nous permettent d’affecter l’interface fastEthernet0/1 à la zone INTERNE et l’interface se0/0/1 à la zone EXTERNE.

**Seul le trafic ayant pour source INTERNE et pour destination EXTERNE a été traité. En pratique, il faut donc créer une autre zone-paire pour autoriser le trafic issu de la zone EXTERNE à atteindre la zone INTERNE.**

# Enoncé 12 Bonus : Mise en place Standard\_ACL\_physique

On utilisera pour l’énoncé 12 le fichier : [Bloc2\_sem3-4\_atelier-02\_Standard\_ACL\_physique.pkt](../Packet_tracert/Bloc2_sem3-4_atelier-02_Standard_ACL_physique.pkt)

En premier, il faut analyser les réseaux. Nous voyons qu’il y a trois clusters réalisés par des câbles série.

Une image contenant diagramme, ligne, capture d’écran

Description générée automatiquement

Ensuite, il faut vérifier tous les clusters :

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquementLe premier cluster est composé d’un routeur et d’un PC.  
Je vais vérifier que toutes les adresses IP ont bien été mises.Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

On peut voir que tout a bien été mis, mais que le routeur n’a pas de hostname correct. J’ai donc rajouté un nom au routeur R2.

Router(config)#hostname R2

R2(config)#

J’ai fait pareil pour les autres clusters.

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquementDans le cluster de droite, on peut voir qu’il y a un routeur, un switch et deux PC qui sont correctement configurés.

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquementDans le cluster du haut, on peut voir qu’il y a un routeur et un PC.

Sur Routeur0 j'ai utilisé ces commandes pour configurer une liste de contrôle d'accès (ACL) afin de gérer le trafic réseau sortant sur l'interface FastEthernet 0/0 du routeur.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

Ces commandes ont été mises en place pour restreindre l'accès d’hôte 192.168.2.2 qui est le PC1 spécifique tout en maintenant la fluidité du reste du trafic réseau.

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquementTest est vérification :

Nous le voyons sur ces captures d’écran, l’hôte PC1 a bien été bloqué dans le cluster0, alors que PC2 n’est pas bloqué.

# Conclusion :

Cette nouvelle approche de filtrage des communications offre un avantage significatif, car elle permet de segmenter le réseau non plus en fonction des interfaces physiques, mais selon une logique basée sur des zones. Ce raisonnement logique apporte une plus grande flexibilité dans la gestion des accès et améliore la sécurité globale du réseau en facilitant l'application de politiques adaptées à chaque zone. En réduisant la dépendance aux structures physiques, cette méthode optimise également l'efficacité du filtrage tout en simplifiant la gestion et l'administration des infrastructures réseau.

# Webographie :

Alvinashcraft. « Access Control Lists - Win32 Apps », 7 février 2023. <https://learn.microsoft.com/en-us/windows/win32/secauthz/access-control-lists>.

Newell, Glen. « An Introduction to Linux Access Control Lists (ACLs) ». Red Hat, Inc., 6 février 2020. <https://www.redhat.com/sysadmin/linux-access-control-lists>.

Cisco. « Configure and Filter IP Access Lists ». Consulté le 8 octobre 2024. <https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html>.

Shirey, Robert W. « Internet Security Glossary, Version 2 ». Request for Comments. Internet Engineering Task Force, août 2007. <https://doi.org/10.17487/RFC4949>.

Dancuk, Milica. « Iptables Tutorial: Ultimate Guide to Linux Firewall ». Knowledge Base by phoenixNAP, 30 mai 2024. <https://phoenixnap.com/kb/iptables-linux>.

SHEKH, Nour. « Les listes de contrôle d’accès (ACL) avec Cisco | IT-Connect », 24 février 2022. <https://www.it-connect.fr/les-listes-de-controle-dacces-acl-avec-cisco/>.