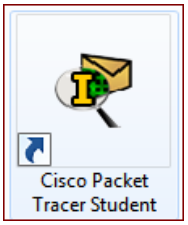
EXOLAB : ACTIVITE PACKET TRACER de DÉCOUVERTE *Access-lists* (règles de filtrage) CISCO



Description du thème

Propriétés	Description
Intitulé long	ACTIVITE PACKET TRACER de découverte des *access-lists* CISCO. *Maquette de base fournie, à compléter en fonction du travail demandé et à valider*
Formation concernée	BTS Services Informatiques aux Organisations
Matière	SISR2 Conception des infrastructures réseaux
Présentation	Cette activité propose plusieurs scénarios de filtrage, permettant de comprendre le fonctionnement des *access-lists* CISCO, appliquées aux interfaces de routeur, en entrée et en sortie. Les 7 scénarios proposés permettent d'aborder successivement : les *access-lists* standard, les *access-lists* étendues, le filtrage en fonction de la source et/ou de la destination IP, le filtrage en fonction du protocole concerné (ICMP), du service demandé, de l'état TCP.
Notions	Maquettage d'infrastructure réseau Configuration d'éléments d'interconnexion Mise en place de règles de filtrage
Transversalité	SI5 Cette activité s'appuie sur les connaissances communes des modules SI2 (pour la maquette de départ) et SI5 (pour le fonctionnement des services HTTP et DNS).
Pré-requis	Une connaissance de base de l'outil Packet Tracer pour modifier des maquettes. Cette activité de découverte fournit les éléments de configuration, mais nécessite la connaissance de la notion de pare-feu
Outils	Packet Tracer **Student** v6.2 (minimale pour utiliser les ressources fournies)
Mots-clés	Packet Tracer, Activité, Maquette, Filtrage, Pare-Feu, Access-List, ACL.
Durée	1h30 environ (temps de lecture / compréhension important)
Niveau de difficulté	Assez facile à exécuter, mais difficulté d'appropriation de la notion d'*access-list* (6/10) avec une maîtrise préalable de Packet Tracer.
Auteur(es)	David Duron avec la relecture d'Apollonie Raffalli et de Gaëlle Castel
Version	v 1.01
Date de publication	Mars 2016
Contenu du package	Document WORD & PDF présentant les instructions Fichier .pka de l'activité (Cisco Packet Tracer version 6.2)

La suite du document comporte les instructions fournies avec la maquette. Ces instructions sont présentes dans l'activité, dans une boite de dialogue associée à l'activité.

Un bouton – au bas de cette boite de dialogue – permet à l'étudiant de vérifier l'atteinte des objectifs (« check result »).

Activité "Mur de feu" : Découverte du filtrage par Access-List Cisco

OBJECTIFS

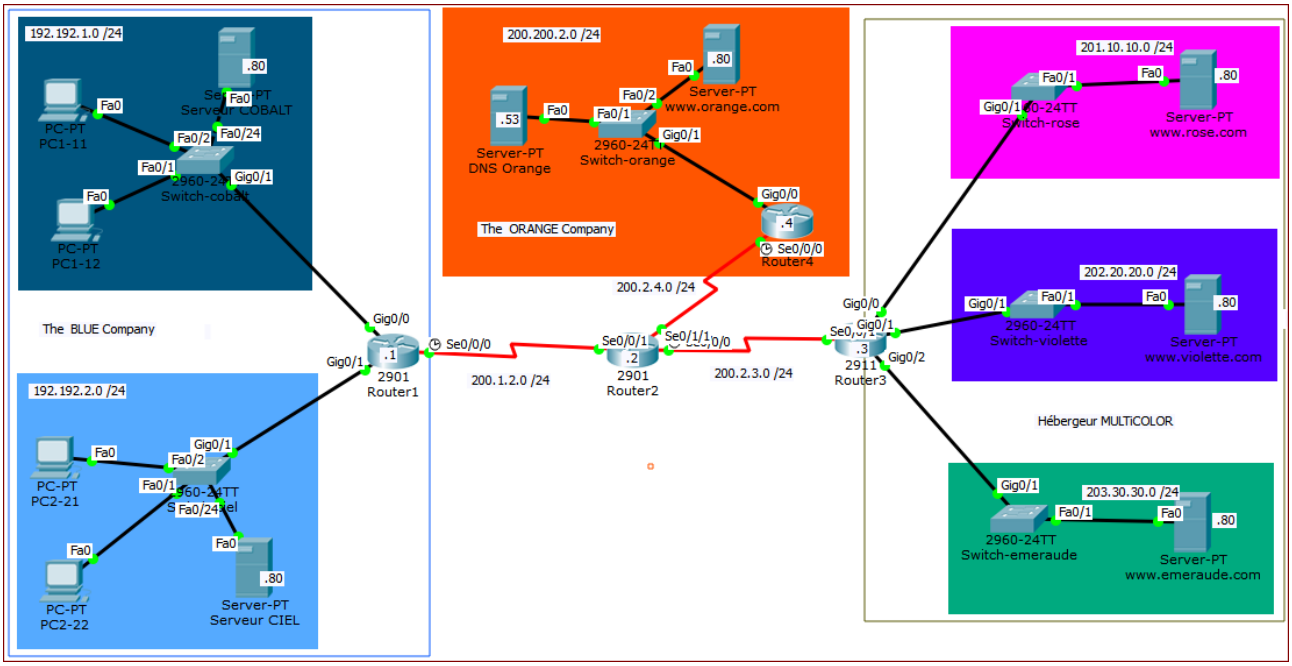
- **Comprendre les *access-lists* standard et étendues.**
- **Mettre en œuvre plusieurs scénarios de filtrage, sur la source et/ou la destination des paquets.**

PRÉSENTATION du SCENARIO

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 :



Les différents scénarios vont permettre de découvrir et comprendre les *access-lists* CISCO, sans être trop technique, ni exhaustif sur les possibilités qui sont quasiment infinies.

Les ACL (*Access Control List*) permettent de filtrer les paquets en fonction de critères définis par l'utilisateur comme l'adresse IP source ou destination, le protocole, le port, etc.

Ces ACL peuvent êtres appliquées en entrée ou en sortie d'une interface de routeur, de manière à filtrer les paquets entrants et/ou sortants.

Il faut savoir que sur les équipements CISCO, il existe 2 types d'ACL :

- Les *Access-lists* Standard : elles permettent de filtrer uniquement sur les IP sources et sont utilisées pour le filtrage, mais aussi pour l'activation du Nat/Pat*.
- Les *Access-lists* Étendues : elles permettent de filtrer sur la plupart des champs des en-têtes IP, TCP et UDP.

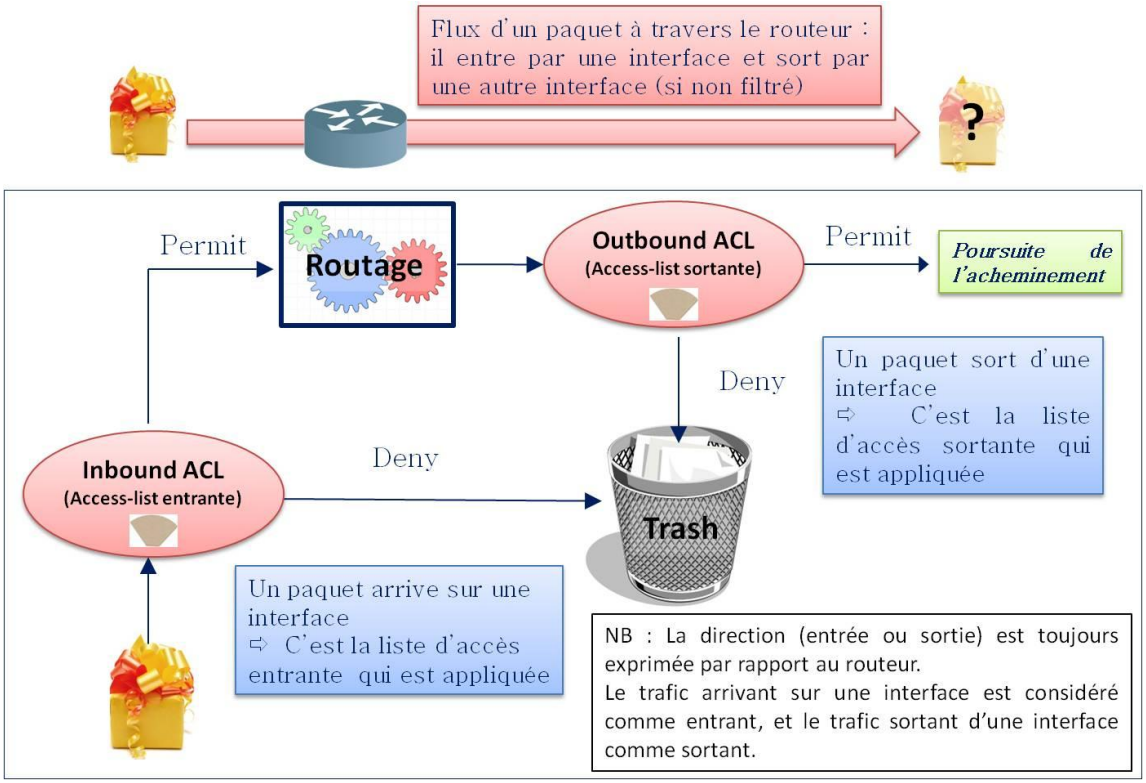
** Le Nat/Pat ne fera pas l'objet de ce support. Par simplification d'ailleurs, il n'est pas mis en œuvre ici, puisque la maquette ne comporte que des réseaux supposés publics qui communiquent entre eux.*

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*.

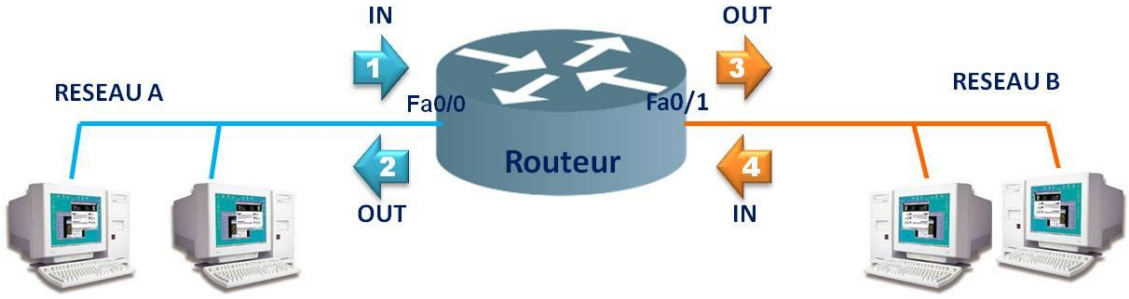
Remarques

- 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).
- Pour information 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.

Principe de filtrage par *access-list*



Positionnement des access-lists



Pour un routeur muni de 2 interfaces, on peut donc filtrer – c'est-à-dire positionner une access-list – en 4 endroits :

1. Filtrage en sortie du LAN (en entrée de FA0/0)
2. Filtrage en arrivée sur le LAN (en sortie de Fa0/0)
3. Filtrage en sortie vers INTERNET (en sortie de Fa0/1)
4. Filtrage en depuis INTERNET (en entrée sur Fa0/1)

Fonctionnement des *access-lists*

Avant d'aller plus loin, il est nécessaire d'expliquer le fonctionnement d'une *access-list*, qui comme son nom l'indique, peut comporter plusieurs règles :

- Les paquets sont vérifiés en testant les règles dans l'ordre où elles ont été définies.
- Si le paquet vérifie le critère énoncé par la règle, l'action définie (permit = autorisation, deny = interdiction) est appliquée et on ne regarde pas les règles suivantes.
  - Sinon, on passe à la règle suivante.
- En fin de liste, si aucune règle ne peut s'appliquer au paquet, c'est l'action **deny** qui est appliquée (même si elle ne figure pas dans la liste, elle est implicite).

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.

SCENARIO 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 2ème solution nous amènera à l'étudier également dans un 2ème 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)

NB : Pour faciliter l'appropriation des commandes à utiliser, elles seront toujours précédées du prompt. La commande est en gras pour la différencier du prompt.

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

Router1(conf)# **access-list 1 deny any**

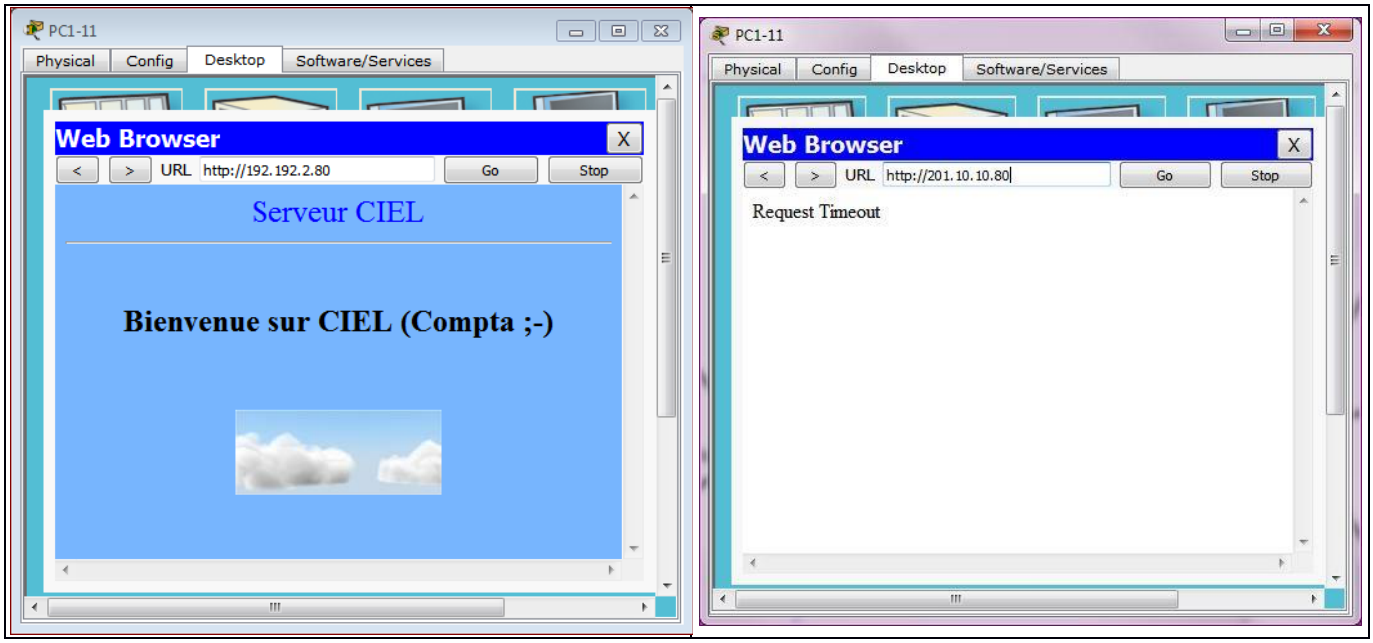
Application de l'access-list en sortie de s0/0/0

Router1(conf)# **interface s0/0/0**
Router1(conf-if)# **ip access-group 1 out**

Remarque : Cette configuration crée une *access-list* qui interdit toute communication, quelle que soit la source, et l'applique en sortie de s0/0/0, donc pour toute communication sortante, depuis BLUE vers *le reste du monde*.

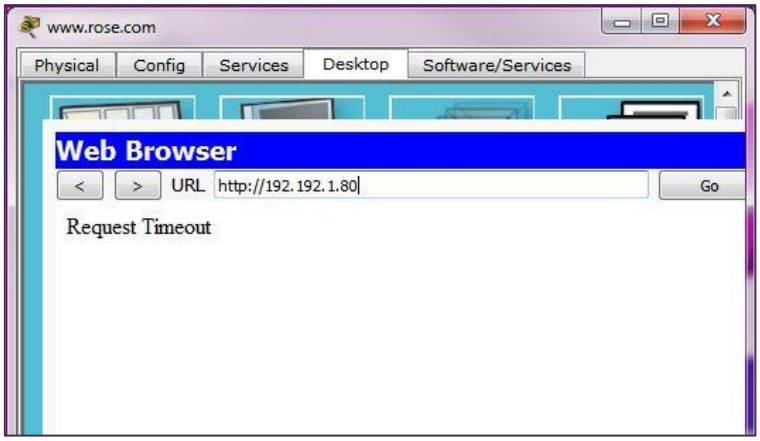
Résultat obtenu / explication

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



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.

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, comme le prouve la copie d'écran ci-dessous :



Explication complémentaire : Si on examine attentivement le trafic, on constate que les paquets (requêtes) HTTP parviennent bien au serveur COBALT, mais que la réponse est filtrée en sortie. Je vous invite d'ailleurs à passer en ***mode simulation*** pour faire ce constat intéressant : Packet Tracer est assez génial de ce point de vue là.

Mise en œuvre du filtrage (solution 2)

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(conf)# **no access-list 1**
Router1(conf)# **interface s0/0/0**
Router1(conf-if)# **no ip access-group 1 out**

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

Router1(conf)# **access-list 1 permit 192.192.1.0 0.0.0.255**
Router1(conf)# **access-list 2 permit 192.192.2.0 0.0.0.255**

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 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 4ème 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(conf)# **interface gi0/0**
Router1(conf-if)# **ip access-group 2 out**

> 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"

Router1 (conf)# **interface gi0/1**
Router1 (conf-if)# **ip access-group 1 out**

> 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.

SCENARIO 2 : ACCÈS DISTINCTIF entre les réseaux COBALT et CIEL

Le scenario 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 scenario 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 2ème 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 commence par annuler les access-lists du SCENARIO 1 qui ne sont plus nécessaires :

```
Router1(conf)# no access-list 1
Router1(conf)# no access-list 2
Router1(conf)# interface gi0/0
Router1(conf-if)# no ip access-group 2 out
Router1(conf-if)# exit
Router1(conf)# interface gi0/1
Router1(conf-if)# no ip access-group 1 out
Router1(conf-if)# exit
```

access-list standard pour s0/0/0 en sortie (on autorise les paquets en provenance du réseau COBALT uniquement)

Router1(conf)# **access-list 2 permit 192.192.1.0 0.0.0.255**

access-list étendue pour s0/0/0 en entrée
Router1(conf)# **access-list 102 permit ip any 192.192.1.0 0.0.0.255**

Cette *access-list* nécessite des explications complémentaires :

- Une *access-list* étendue possède un n° entre 100 et 199 (ou entre 2000 et 2699). *On pourrait aussi les nommer, mais pour cette découverte, on utilisera uniquement des numéros.*
- Une *access-list* permet de définir une source et une destination, ainsi qu'un protocole, voire un n° de port.
- La source ou la destination peuvent être indiquées sous 3 formes :
  - **any** *(n'importe quelle source ou destination)*
  - <***destination***> <***masque-inversé***> *(un réseau ou une partie de réseau y compris d'ailleurs avec des bits non contigües pour le masque - ;-)*
  - **host** <***adresse-hote***> *(une seul hôte désigné par son adresse IP)*

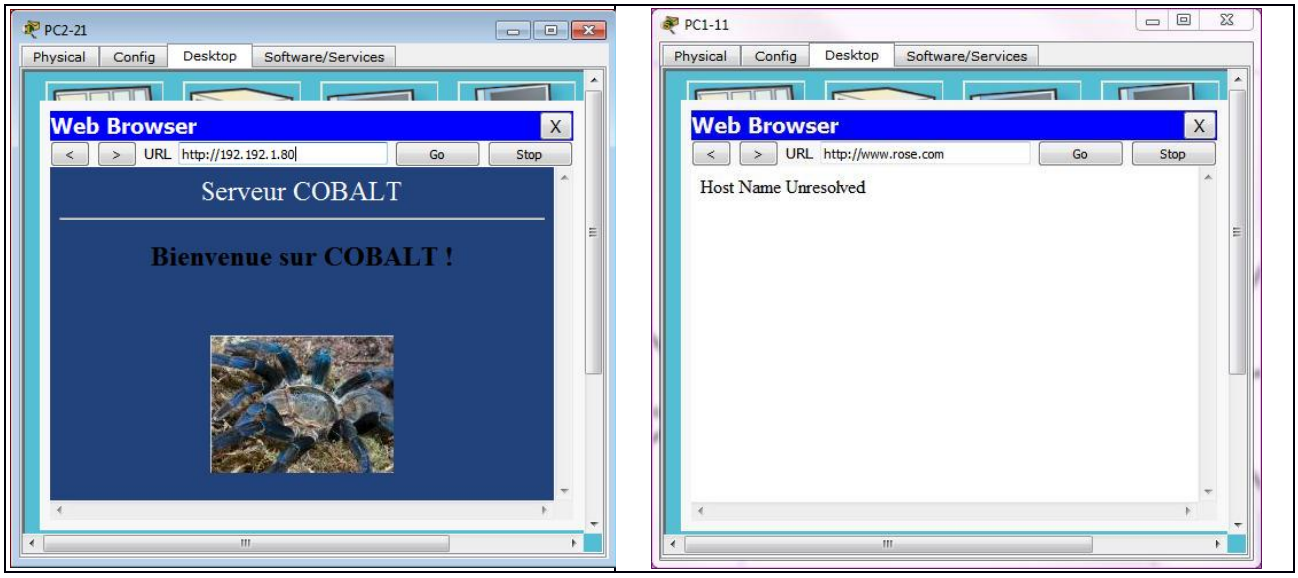
La règle ci-dessus autorise donc tout paquet IP (que celui-ci contienne du TCP ou de l'UDP ou même de l'ICMP) à destination du réseau 192.192.1.0, et ce quelle que soit la source du paquet (*any*).

application des access-list sur s0/0/0

Router1(conf)# **interface s0/0/0**
Router1(conf-if)# **ip access-group 2 out**
Router1(conf-if)# **ip access-group 102 in**

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 :



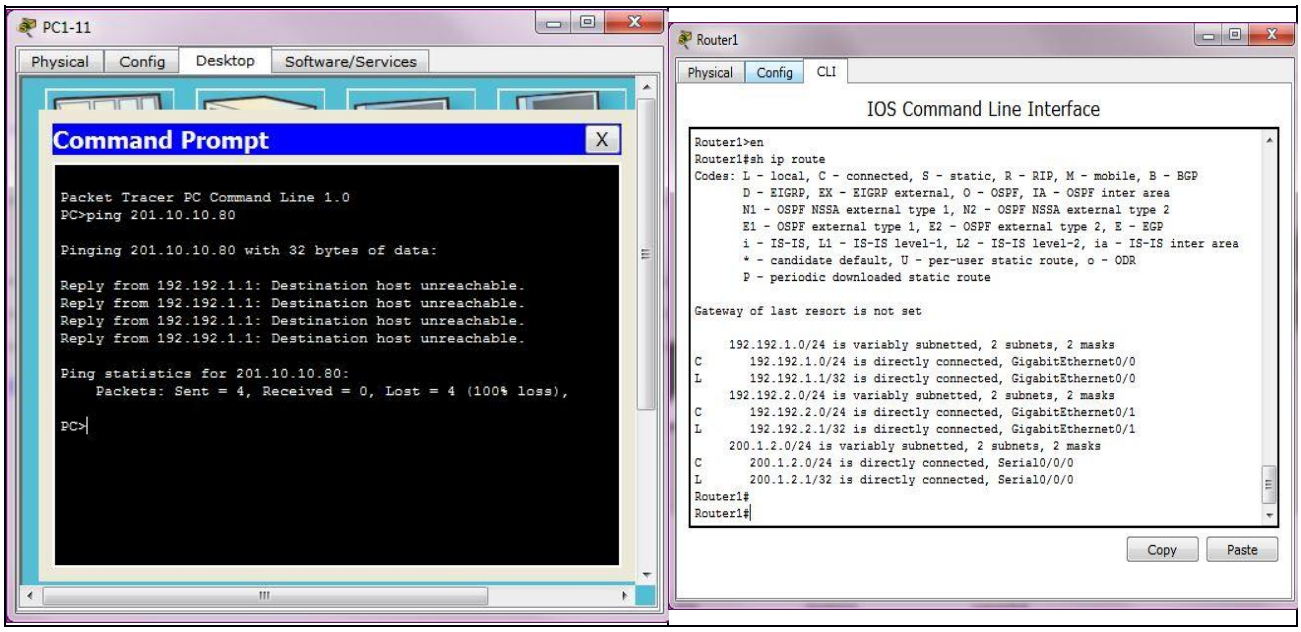
Surprise ! On n'obtient pas vraiment ce que l'on attendait. Plus aucune communication avec le *reste du monde* n'est possible !

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.

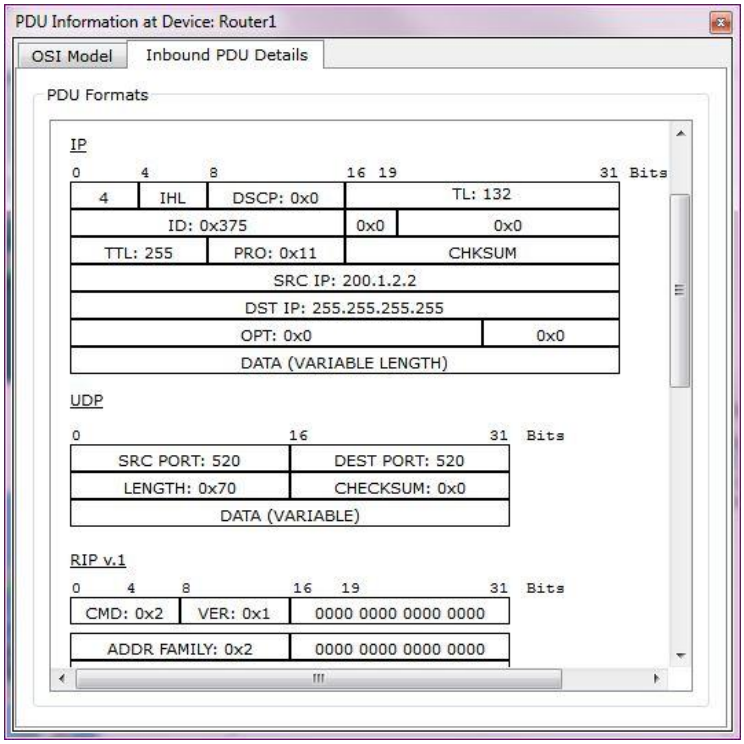
Remarque : Il faut un certain délai pour que les routes disparaissent, donc cela peut fonctionner encore quelques instants ; les routes deviennent « possibly down » au bout de 3 min et disparaissent complètement après 9 minutes normalement.

Les copies d'écran ci-dessous montrent :

- Le test qui a aidé à identifier 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.



Un échange RIP, capturé grâce au mode simulation, permet de repérer le protocole et le port utilisé par RIP pour l'autoriser. Encore une fois, on profite des riches fonctionnalités de Packet Tracer.



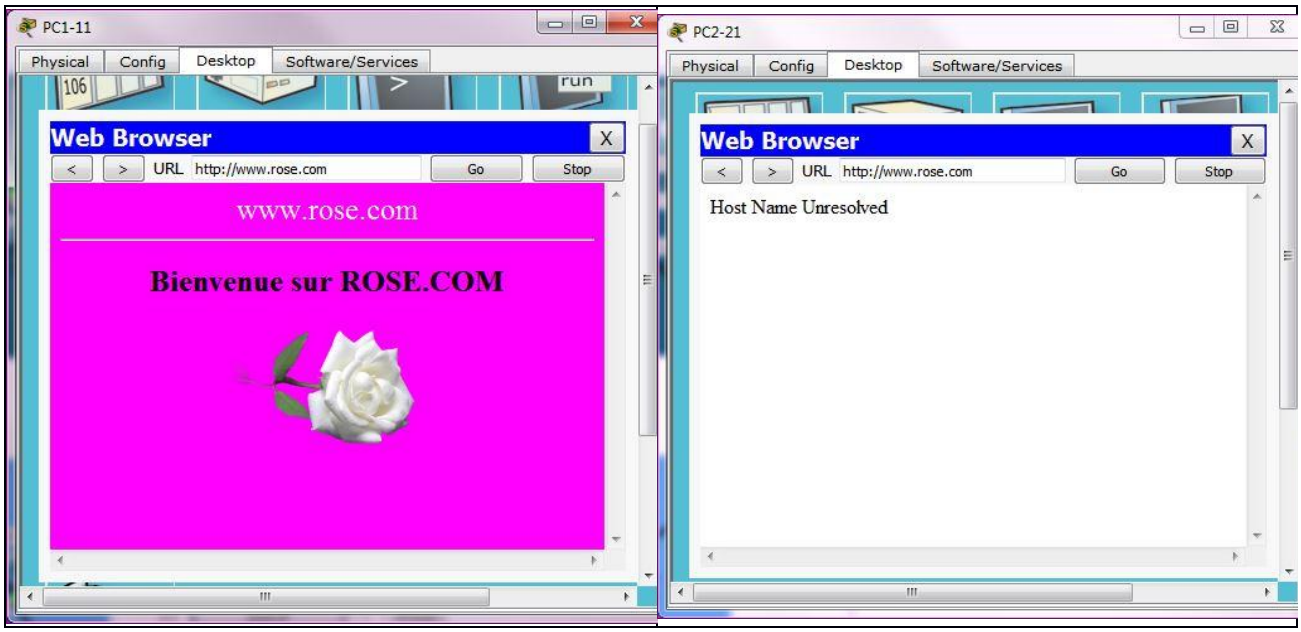
On peut repérer que RIP utilise des *broadcasts* et que les échanges se font en UDP sur le port 520 (aussi bien en source qu'en destination).

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, on a bien les constats souhaités/attendus, après quelques 30 secondes de délai :



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 1ère communication est possible.

Maintenant que nous avons vu les bases, nous allons chercher à mettre en place quelques scénarios complémentaires.

SCENARIO 3 : ACCES 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
!

Vérifiez que le nouveau scénario est fonctionnel :

- L'accès à **www.orange.com** et **www.emeraude.com** doit rester possible depuis **COBALT**.
- L'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 possible.
- Depuis **CIEL**, aucun accès externe n'est possible, pas plus vers **ORANGE** et **EMERAUDE** que vers **ROSE** ou **VIOLETTE.**

SCENARIO 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 scenario 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

Réflexion publique ;-)

- 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 ***Se0/0/1***, 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 ***Se0/0/1***, 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 EMERAUDE), 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'EMERAUDE 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 2ème 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 !

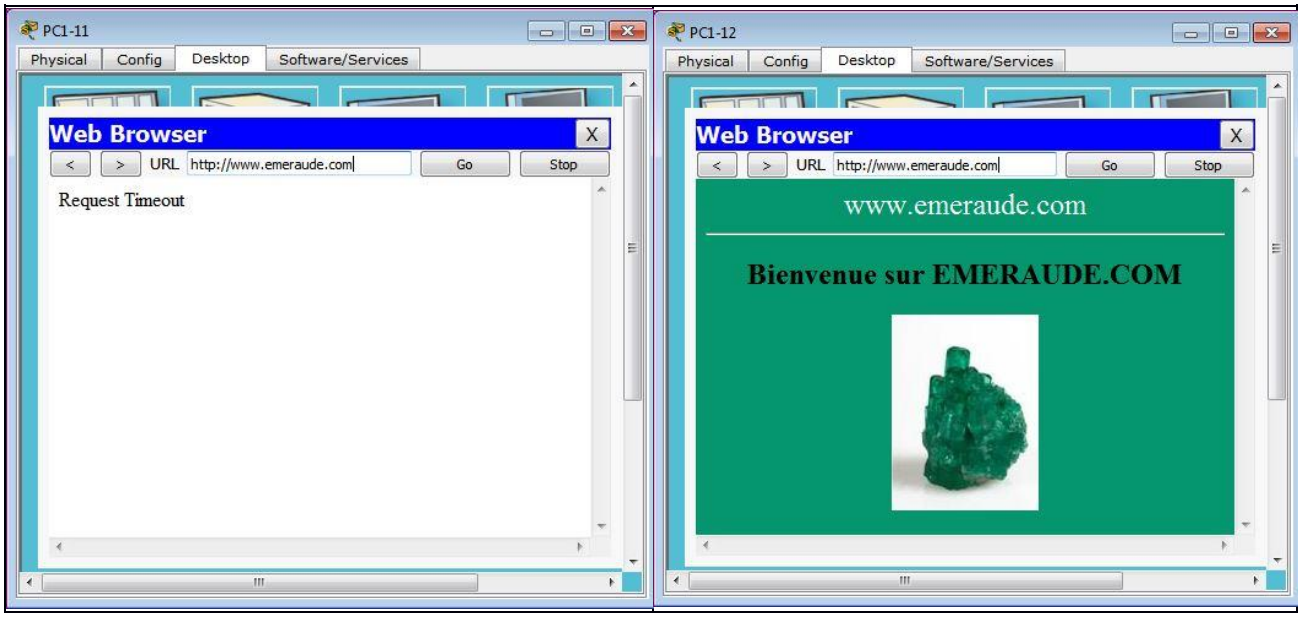
Par ailleurs, puisqu'on n'a pas précisé les services utilisables, on reste au niveau le plus "bas" possible (donc IP) pour ne filtrer aucun protocole en particulier.

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



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 EMERAUDE peut communiquer avec les serveurs ORANGE et COBALT.

SCENARIO 5 : ACCES RESTRICTIF à certains SERVICES sur le réseau ORANGE

Le scenario 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 scenario 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 :

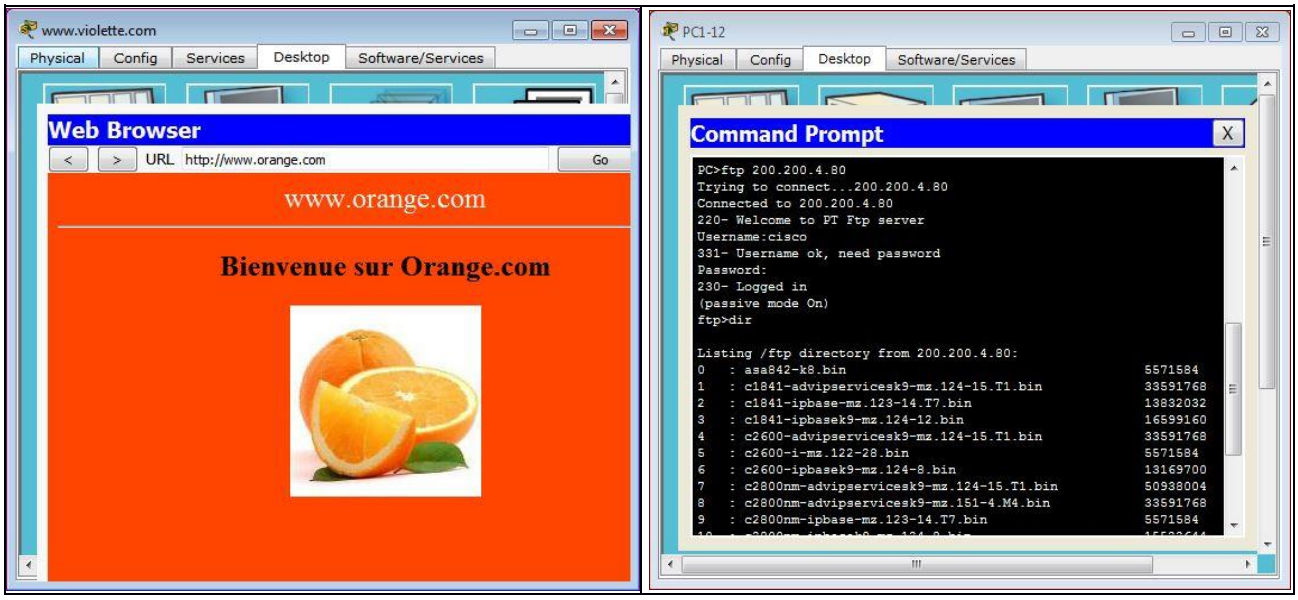
- soit des demandes concernant les services HTTP et DNS ;
- soit des informations RIP ;
- soit des réponses à des requêtes sortantes, initiées depuis le réseau Orange.

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.

Les copies d'écran ci-dessous montrent seulement deux de ces tests, mais tous devraient réussir :



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.

*# Intéressant à savoir : en invite de commande, on peut à tout moment utiliser le **?** pour connaître la liste des commandes possibles ou la liste des mot-clés utilisables pour terminer une commande. Ci-dessous, c'est ce qui nous a permis de connaître les valeurs possibles pour un n° de port derrière l'opérateur eq :*

```
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
```

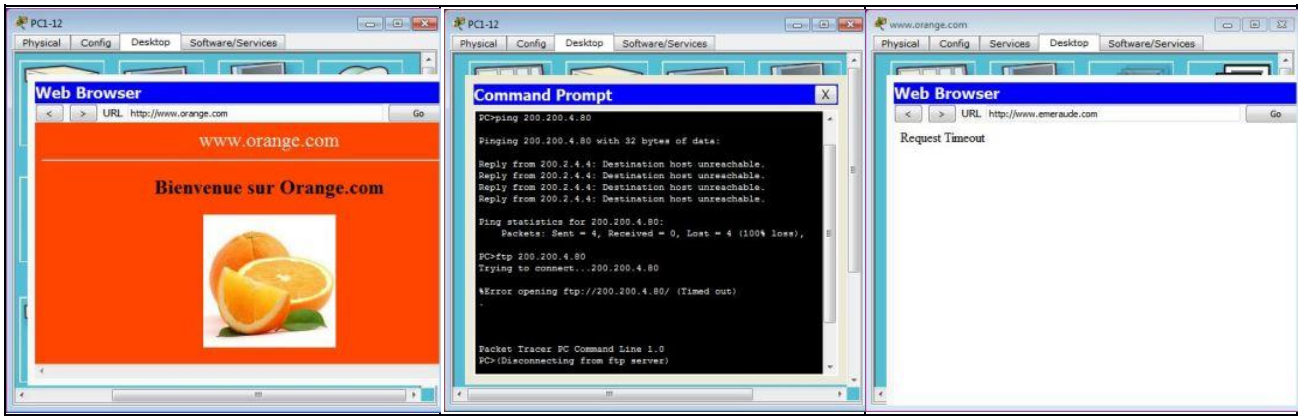
Association à l'interface s0/0/0

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 2ème 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. 3ème copie d'écran) : un accès externe - depuis ORANGE vers un serveur web situé sur un autre réseau – échoue.



Le 3ème 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 soit 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 :

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.

SCENARIO 6 : MODIFICATION D'ACCESS-LIST pour ajouter de nouvelle REGLES

Le scenario 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" (192.192.20) 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.

```
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 EMERAUDE) 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 - 2ème 10 = pas ou incrément)*

NB : Les deux solutions sont présentées, mais vous utiliserez la 2ème 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 2ème 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#

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**.

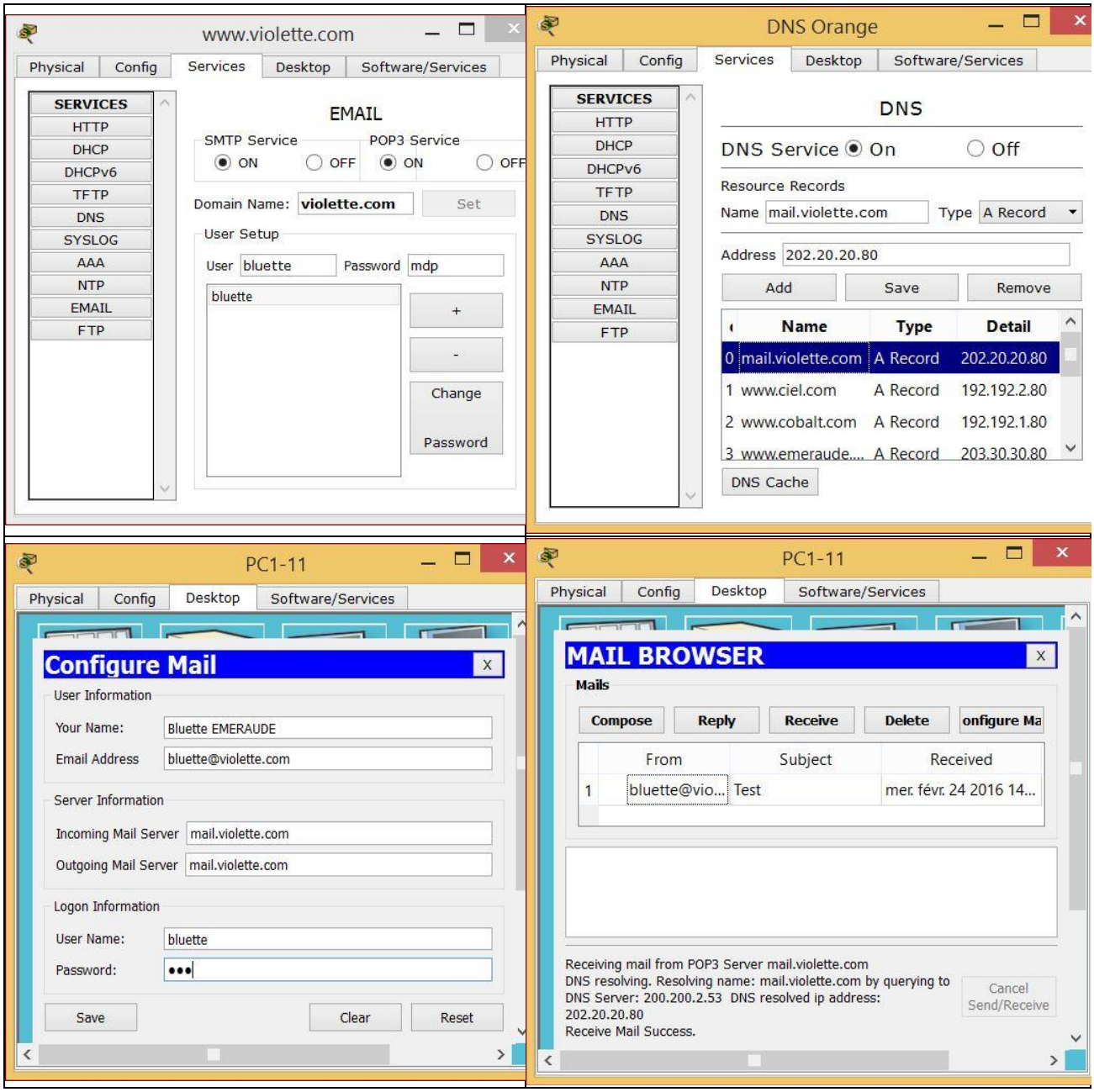
Router1(config)#**ip access-list resequence 103 10 10**
 # inutile d'essayer donc, cela ne fonctionne pas

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 2ème compte mail sur le serveur de mail.

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



Ci-dessus, dans la 4ème 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.

SCENARIO 7 : AUTORISER les *PING* SORTANTS, mais REFUSER les *PING* ENTRANTS

Le scenario 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 ping sont autorisés, c'est qu'ils sont inclus dans la 1ère 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 EMERAUDE n'est pas possible pour PC1-11 puisqu'il est *blacklisté*.

En revanche :

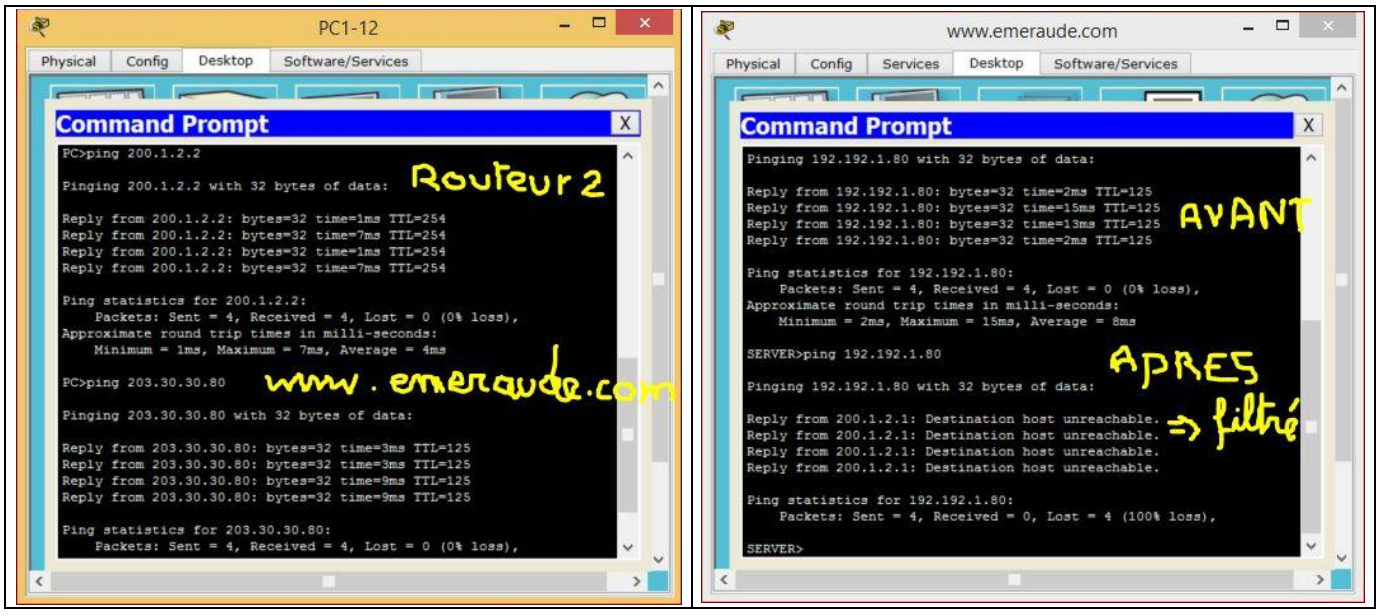
- Un ping depuis PC1-12 vers **www.emeraude.com** devrait rester possible.
- Un ping des routeurs (ex : 200.1.2.2 ou 200.2.3.3) reste possible également.

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



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.

Soyez patient !