Polycopié de TPs réseaux

v.1.2

Auteur : Adlen Ksentini
Maître de Conférences à l'Université de Rennes 1.

Préface :

Ce polycopié de travaux pratiques (TP) regroupe un certain nombre d'énoncés au tour des réseaux. Ces TPs ont été effectués à UFR informatique et télécommunication (ISTIC) de l'Université de Rennes 1 (UR1) durant les six dernières années. Le polycopié est organisé par thème, et chaque thème contient un certain nombre de sujets. La première partie « introduction aux réseaux » est destinée à des personnes n'ayant pas une grande connaissance dans les réseaux et voulant acquérir des notions générales. Les parties suivantes : routage, administration des réseaux, sécurité des réseaux et VoIP, sont destiné à un public spécialisé dans les réseaux. Ces TPs sont dans le programme du master 2 professionnels Ingénierie des réseaux (IR) de UR1.
Les TPs des parties avancées sont à effectuer dans les conditions de la salle i207 (salle réseau de l'ISTIC), où chaque binôme dispose de : 1 PC linux (Debian) avec 5 interfaces Ethernet, 1 PC WinXP avec 1 interface Ethernet, un routeur *Cisco 1941 Series* et un commutateur *Cisco 2950*. Chaque banc de la salle (noté NB dans les TP) peut accueillir deux binômes.
Je tiens à remercier :
- Gilles Guette, pour sa participation à la création des TPs Service de messagerie et Firewall , et la mise à disposition de son TP authentification 802.1X.
- Michel Le Tohic pour sa participation à la création du TP Samba.
- Frédéric Tronel pour sa participation à la création du TP Serveur web et sécurité.

1ère partie : Introduction au réseau

- Analyse du protocole HTTP
- Analyse des protocoles TCP et IP avec Wireshark
- Interrogations d'une infrastructure de réseaux Internet
- Programmation réseau (Socket) en UDP
- Programmation réseau (Socket) en TCP
- Client/Serveur TCP : HTTP
- Socket RAW : ICMP

Analyse du protocole HTTP

Durée : 2h

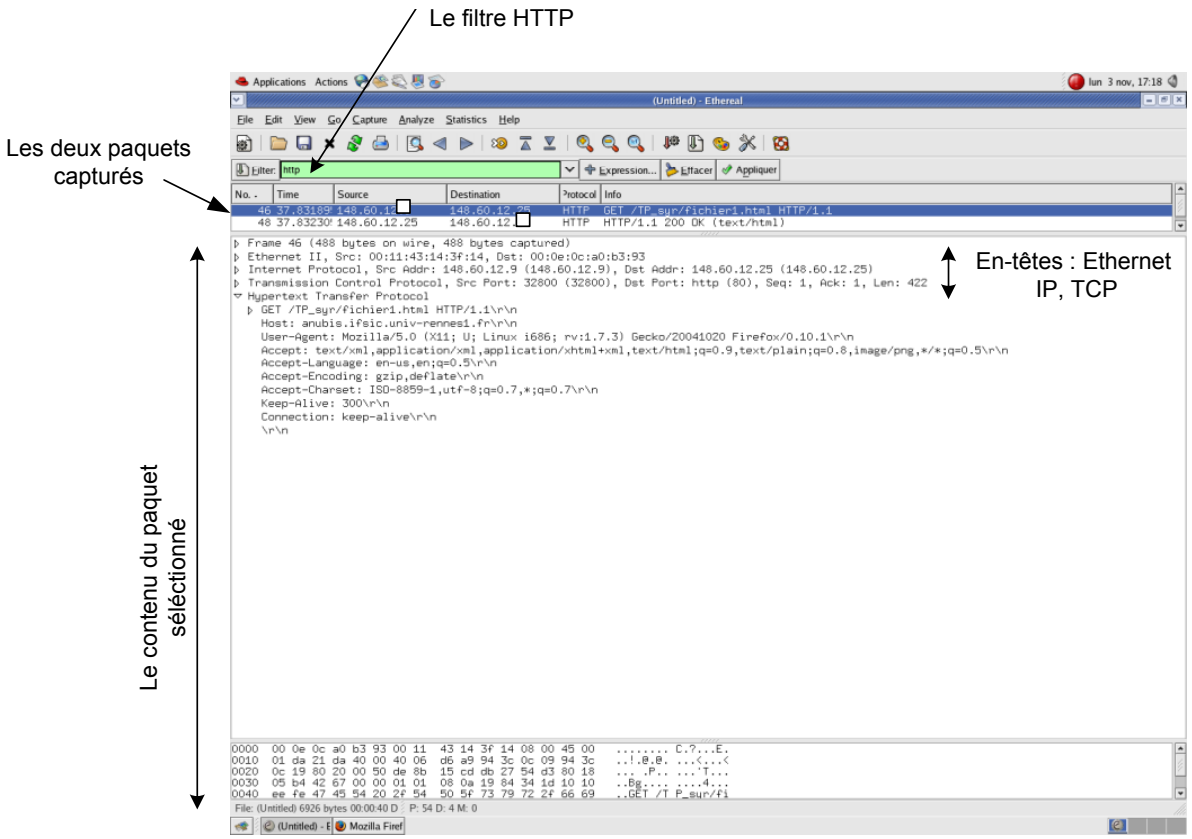
N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) qui doit être envoyé avant le début de séance suivante à l'enseignant responsable de ce TP.

But: Dans ce TP nous allons explorer différents aspects du protocole HTTP : GET/réponse, format des messages HTTP, les fichiers HTTP volumineux, récupérer un fichier HTML avec des objets référencés.

1. Le GET et les réponses HTTP

Nous allons commencer l'exploration du protocole HTTP en téléchargeant un court fichier HTML. Ce fichier ne contient pas de références vers des objets.

a) Lancez votre navigateur web
b) Lancez une capture avec Wireshark. Entrez « http »dans l'onglet *filter*, pour filtrer le trafic HTTP.
c) Attendez au moins une minute, et démarrez la capture avec Wireshark
d) Entrez l'adresse suivante dans votre navigateur web : http://anubis.istic.univ-rennes1.fr/TP_syr/fichier1.html. Votre navigateur doit-être capable d'afficher la page HTML
e) Arrêtez la capture avec Wireshark



Votre fenêtre Wireshark doit ressembler à la figure du dessus, où deux messages HTTP sont capturés (de votre navigateur vers le serveur *anubis*, et du serveur *anubis* vers votre navigateur). Pour rappel, le message HTTP est encapsulé dans un segment TCP, qui est encapsulé dans un paquet IP, qui est encapsulé dans une trame Ethernet.

Note : Il faudrait ignorer la requête sur l'icône faite par votre navigateur, car ce dernier demande au serveur s'il a une icône à afficher.
En se basant sur les informations transportées dans le GET et sa réponse, répondez aux questions suivantes :

a) Quelle est la version du protocole HTTP, 1.0 ou 1.1, que votre navigateur utilise ? Quelle version est utilisée par le serveur ?
b) Quels langages (si plusieurs) votre navigateur supporte ?
c) Quelle est l'adresse IP de votre machine ?
d) Quel est le statut (code) retourné par le serveur web à la requête du navigateur ?
e) Quelle est la date de la dernière modification du fichier envoyé par le serveur ?
f) Quelle est la taille en octets du contenu retourné vers votre navigateur ?
g) Quelle la version du serveur web ?

a) Le GET conditionnel et les réponses HTTP

Il est important de rappeler que les navigateurs web utilisent un cache pour les objets, et ainsi des GET conditionnels sont envoyés pour récupérer un objet manquant. Assurez-vous que le cache de votre navigateur est vide (*Firefox : Outils-> options->avancé->réseau* cache et vider le cache. IE : *tools->Internet option -> delete File*). A présent :
Lancez votre navigateur, et videz votre cache

- Démarrez une capture avec Wireshark
- Entrez l'adresse suivante dans votre navigateur web : http://anubis.ifsic.univ-rennes1.fr/TP_syr/fichier2.html. Votre navigateur affichera une page HTML simple.
- Entrez encore une fois l'adresse ci-dessus (ou rafraichir la page)
- Arrêtez la capture, et filtrez le trafic « http »

Répondre aux questions suivantes :

a) Inspectez le contenu de la première requête GET de votre navigateur vers le serveur. Voyez-vous le la ligne « IF-MODIFIED-SINCE: » dans la requête ?
b) Inspectez le contenu de la réponse du serveur. Est-ce que le serveur renvoie implicitement le contenu du fichier ? Comment pouvez-vous l'affirmer ?
c) Inspectez le contenu de la deuxième requête GET de votre navigateur vers le serveur. Voyez-vous le la ligne « IF-MODIFIED-SINCE: » dans la requête ? Si oui, quelle est l'information qui suit l'en-tête « IF-MODIFIED-SINCE: »
d) Quel est le statut code et la phrase retournée par le serveur en réponse du second GET ? Est-ce que le serveur renvoie implicitement le contenu du fichier ? Expliquez.

3. Retrouver des documents HTML volumineux

Dans nos précédents exemples, les documents demandés sont simples et assez courts. Maintenant, nous allons nous intéresser aux documents HTML volumineux.

- Lancez votre navigateur web, et assurez-vous que le cache du navigateur est vide
- Lancez une capture avec Wireshark
- Entrez l'adresse suivante dans votre navigateur web : http://anubis.istic.univ-rennes1.fr/TP_syr/fichier3.html.
- Arrêtez la capture, et filtrez le trafic « http »

Dans la fenêtre principale de Wireshark, vous devez apercevoir votre message GET, suivi par plusieurs paquets réponse à cette requête. Ceci nécessite une petite explication. Dans le cours nous avons vu qu'une réponse HTTP contient un statut, un en-tête suivi par une ligne vide et le corps du paquet (*entity body*). Dans le cas de notre GET, le corps du message réponse est la totalité de la page HTML. Du coup, ce fichier HTML est volumineux (4500 octets), et il ne peut pas être transmis dans un seul segment TCP. Ainsi la réponse HTTP est divisée en plusieurs pièces par TCP, chaque pièce est encapsulée dans un seul segment TCP. Chaque segment TCP est enregistré séparément par

Wireshark, et le fait qu'une réponse HTTP a été divisée en plusieurs segments TCP est indiqué par la phrase « Continuation ». Il est important de noter qu'il n'y pas de message « Continuation » en HTTP !.

Répondre aux questions suivantes :

- Combien de requêtes GET ont été envoyées par votre navigateur ?
- Combien de segments TCP (transportant les données) sont nécessaires pour transmettre le message réponse HTTP ?
- Quel est le statut (code) et la phrase associée à la réponse du GET ?
- Y'a-t-il des lignes dans les données transmis en « continuation » qui reflètes le statut (code) du message HTTP ?

a) Documents HTML avec des objets référencés

A présent nous allons nous intéresser aux fichiers HTML qui référencent des objets (dans notre exemple des images) stockés sur d'autres serveurs web.

- Lancez votre navigateur web, et assurez-vous que le cache du navigateur est vide
- Lancez une capture avec Wireshark
- Entrez l'adresse suivante dans votre navigateur web : http://anubis.istic.univ-rennes1.fr/TP_syr/fichier4.html. Votre navigateur doit afficher une courte page HTML avec deux images. Ces deux images sont référencées dans le code HTML. Ces deux images ne sont pas contenues dans le fichier HTML; mais uniquement leur URL est définis dans le code. Le logo de l'ISTIC est sur le serveur www.ifsic.univ-rennes1.fr et le logo ENT est sur le serveur www.univ-rennes1.fr
- Arrêtez la capture, et filtrez le trafic « http »

Répondre aux questions suivantes :

- Combien de requêtes GET ont été envoyées par votre navigateur ? À quelles adresses IP sont-elles envoyées ?

Pouvez-vous indiquer si votre navigateur a téléchargé les deux photos en parallèle ou l'une après l'autre (série) ? Expliquez.

Analyse des protocoles
TCP et IP avec Wireshark

Durée : 2h.

N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) qui doit être envoyé avant le début de séance suivante à l'enseignant responsable de ce TP. Ce CR fera état des réponses aux questions 1, 4, 5, 11, 12, 17, 18, 20, 21, 22 et 23. Vous êtes libre d'ajouter tout élément d'information qu'il vous semblera judicieux de donner.

But

L'objectif de ce TP est de faire une analyse de l'empilement protocolaire en utilisant le cas d'une application Web (http). Nous examinerons la structure des en-têtes des PDU (Protocol Data Unit) au niveau de la couche liaison, IP et de la couche transport TCP.

Les protocoles traités

- l'adressage IP et Ethernet
- Le Three-Way Handshake de TCP ainsi que la numérotation des ACK

Procédures

- Lancer Wireshark (Démarrer -> Programmes ->Emulateurs-> Wireshark).
- Lancer Internet Explorer ou Firefox
- Dans le menu capture de Wireshark cliquer sur start.
- Entrer l'adresse suivante : http://www.istic.univ-rennes1.fr dans le navigateur, dès que la page est chargée, arrêter la capture dans Wireshark.
- Dans le champ filtre, tapez TCP (les paquets concernant cette connexion sont affichés en vert)
- Dans l'onglet (Edit → preferences → protocols → tcp), décochez l'ensemble des paramËtres en laissant uniquement "Show TCP Summary in Protocol Tree"

A. Ethernet et IP

Examiner la première trame émise par votre machine

1. Identifier l'adresse Ethernet et l'adresse IP de votre machine, vérifier ces adresses en tapant la commande ***ipconfig /all*** dans une fenêtre MS-DOS.
2. Quel est le contenu du champ ***type*** du paquet Ethernet ?
3. Identifier les adresses de destination Ethernet et IP ?
4. Quel est l'identifiant du constructeur de la carte réseau ?
5. Quelle est la classe des adresses IP source et destination ?
6. Quelle est la taille de l'en-tête du paquet IP ? Quelle est la taille totale du paquet ?
7. Identifier le champ du type du protocole transporté par le protocole IP ?
8. Quelle est la valeur du champ TTL ? Donner une explication

B. TCP Three-way handshake

Identifier le premier segment TCP qui ouvre la connexion HTTP en utilisant le mécanisme three-way handshake.

9. Quel est le numéro de port TCP de destination utilisé par le client, est ce qu'il correspond à un port well-known TCP.
10. Quelle est la taille de l'en-tête du segment TCP ?
11. Quelle est la longueur du segment TCP ? pourquoi ?
12. Quel est le numéro de séquence TCP du premier paquet (du client vers le serveur) ? Quelle est la taille de la fenêtre TCP annoncée par le client ? quelle est sa signification ?

Identifier le deuxième segment TCP appartenant au three-way handshake

13. Donner la valeur des champs suivants :
    a. Adresse Source et Destination ainsi que le champ type de la trame Ethernet
    b. Adresse Source et Destination ainsi que le numéro (en héxa) du protocole contenu dans le paquet IP
    c. Le numéro de l'acquittement contenu dans le segment TCP
14. Quelle est la taille du segment TCP (le segment que vous avez identifié) ?
15. Quel est le numéro de séquence initial (serveur vers le client) ?
16. Quelle est la valeur de la fenêtre TCP annoncée par le serveur ?
17. Quel est l'impact de la différence entre la fenêtre TCP annoncée par le client et la fenêtre TCP annoncée par le serveur ?
18. A partir des deux paquets SYN et SYN-ACK, déterminer la taille du MSS (Maximum Segment Size) accepté par le serveur et le client ? Déduisez la taille du MTU du lien physique utilisé par le serveur et le client ?

Identifier le dernier segment TCP appartenant au three-way handshake

19. Identifier dans ce segment TCP
    a. Le numéro d'acquittement ainsi que le numéro de séquence
    b. La taille de la fenêtre TCP

Pour une meilleure lisibilité des numéros de séquences, on va demander à Wireshark d'utiliser des numéros de séquences relatifs (à partir de 0). Pour cela, cochez les options "Analyse TCP sequence numbers" et "Relative Sequence number and window scaling" dans (Preferences → Protcols → TCP).

20. En vous basant sur les segments qui suivent le three handshak, relevez le contenu des champs numéro de séquence et numéro d'acquittement et déduisez comment ces valeurs évoluent avec le temps ? aidez vous d'un schéma.

Le paquet HTTP 200 OK est une réponse à la requête GET du client. Il contient la taille de la page demandée, c'est à dire les données applicative qui vont suivre du serveur vers le client.

21. En analysant l'en-tête HTTP du 200 OK, relevez la taille de ces données applicative qui vont suivre ?
22. Selon vous à quel niveau la fragmentation de ces données sera effectuée ? vérifiez dans la capture Wireshark ? Qu'est-ce que vous constatez ?

Dans une fenêtre MS-DOS, tapez ***ping l'@IP du serveur***, relevez le maximum du temps aller retour entre le client et le serveur.

23. En vous basant sur les fenêtres annoncées par le client et le serveur dans le three-handshak, donnez une estimation du débit maximum entre le client et le serveur et entre le serveur et le client, sachant que le client et le serveur sont sur des réseaux Ethernet 100baseTX. Les résultats obtenus vous semblent t'ils logique ?.

Interrogations d'une infrastructure de réseaux Internet

Durée : 2h

N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) qui doit être envoyé le à l'enseignant responsable de ce TP. Ce CR fera état des réponses aux questions **2, 3, 6, 10, 14, 15, 17 et 19**. Vous êtes libre d'ajouter tout élément d'information qu'il vous semblera judicieux de donner. On prendra soin de donner les commandes utilisées (avec les paramètres et options) et on expliquera les résultats obtenus.

But

Ce TP a pour objectif de vous faire découvrir quelques composants, mécanismes et protocoles intervenant dans le fonctionnement de l'Internet, via l'utilisation de quelques commandes Unix/Linux en mode super-utilisateur ("root").

Procédures

On emploiera tout au long du TP divers commandes et outils afin de consulter et modifier les configurations d'une machine Linux (Debian). Voici une liste non exhaustive des commandes utilisables (***hostname***, ***ifconfig***, ***route***, ***netstat***, ***nslookup***, ***ping***). Pour ces commandes, vous devez respecter la syntaxe d'appel précise et fournir les paramètres et les éventuelles options nécessaires. Vous avez deux moyens pour obtenir la syntaxe, les paramètres et options d'une commande:

- le manuel en ligne via la commande ***man***,
- l'option *-h* ou *-?* qui affiche la liste des paramètres et options d'une commande.

Pour ce TP, on travaillera en environnement Linux, mais sachez que des commandes analogues existent pour les autres systèmes d'exploitation tels que MacOS, Windows, Unix, FreeBSD, etc.
Utilisez le compte *admin* de votre machine : {*Login : admin* ; *Mot de passe : 2+en+dur*}. Le compte *admin* possède les droits *root,* sur cette machine Linux.

Manip 1 : Environnement IP et Ethernet

En vous aidant des commandes ***hostname*** et ***ifconfig*** :

1. Indiquez le nom et l'adresse IP de votre machine.
2. Indiquez le nombre d'interfaces Ethernet utilisées par votre machine ? Donnez le MTU de chaque interface.

En vous aidant de la commande ***ifconfig*** :

3. Désactivez l'ensemble des interfaces de la machine.
4. Attribuez à l'interface *eth0* l'adresse IP 148.60.12.X/24 (X représente le numéro de machine).

En vous aidant de la commande ***route*** :

5. Ajoutez une route par défaut vers le routeur (passerelle) 148.60.12.254. Quelle est l'utilité d'avoir une route par défaut dans un réseau local ?
6. Affichez la table de routage de la machine. Quelles informations pouvez-vous en tirer ?

Manip 2 : Analyse ARP et ICMP

En vous aidant des commandes ***arp*** et ***ping*** :

7. Consultez et videz la table (cache) ARP de votre machine.

8. Mettez en route une capture de trames avec Wireshark (*Menu KDE->Internet->Wireshark*) et lancez un ***ping*** vers la machine *anubis.istic.univ-rennes1.fr* en ne générant qu'un seul paquet de demande d'écho.

Pour avoir une vision plus claire des résultats de la capture, il est préférable de créer un filtre sur les protocoles ARP et ICMP.

9. Analysez le format d'un message ARP et retrouvez les différents champs en précisant leur rôle respectif.
10. Retracez un échange ARP en précisant les adresses Ethernet et IP utilisées à chaque étape.
11. Analysez le format d'un message ICMP et retrouvez les différents champs en précisant leur rôle respectif.
12. Comparez les données des messages *ICMP Echo Request* et *ICMP Echo Reply* : que constatez-vous ? Quel est le contenu de ces champs de données ?
13. Les protocoles ARP et ICMP sont chacun des protocoles indispensables à la couche 3, cependant ils utilisent un empilement protocolaire différent : expliquez cette différence.

Manip 3 : IP fragmentation

Lancez Wireshark, et grâce à la commande ***ping***, envoyez un ping vers la machine 148.60.4.10 avec une taille de 2000 octets (*ping -c 1 –s 2000 148.60.4.10*).

14. Trouvez le premier fragment du paquet IP. Quelle information dans l'en-tête IP indique que ce paquet IP est fragmenté ? Quelle information dans l'en-tête IP indique que c'est le premier fragment ou le dernier fragment ? Quelle est la taille de ce paquet IP ?
15. Trouvez le deuxième fragment du paquet IP. Quelle information dans l'en-tête IP indique que ce n'est pas le premier fragment ? Y'a-t-il d'autres fragments qui suivent ?
16. Quels champs dans l'en-tête IP diffèrent entre les deux fragments IP ?

Manip 4 : Analyse du DHCP

Grâce à la commande ***ifdown*** libérez l'adresse IP de la carte *eth4* (***ifdown eth4***). Lancez Wireshark, et redemandez une adresse IP pour la carte *eth4* avec la commande ***ifup eth4***. Arrêtez la capture.

17. Est-ce que les messages DHCP utilisent UDP ou TCP ?
18. Dessinez un diagramme temporel pour les quatre premiers messages ((Discover/Offer/Request/ACK) échangés entre le client et serveur DHCP ? Pour chaque paquet indiquez le port source et destination ?
19. Quelles sont les adresses IP source et destination utilisées dans le DHCP Discover et Request ?
20. Quelles valeurs dans le DHCP Discover permettent de le différentier du message DHCP Request ?
21. Quelle est l'adresse du serveur DHCP ?
22. Quelle est l'adresse IP que le serveur DHCP offre à votre machine ? Indiquez quel message DHCP contient l'adresse offerte ?
23. Quel est l'utilité des lignes submask et router dans le DHCP Offer ?

Manip 3 : Services Internet

En vous basant la commande ***netstat*** :

24. Quel est le fichier qui contient les services (les numéros de port) TCP et UDP supportés par votre machine (lisez bien la fin du man) ? Donnez alors les numéros de port pour les principaux services que vous connaissez.
25. Lancez un navigateur web sur un site et donnez les connexions actives sur votre machine.
26. Donnez les ports en écoute (ouverts) sur votre machine.

Manip 4 : DNS

Visualisez le fichier */etc/resolv.conf*.
Donnez le(s) serveur DNS préféré(s) de votre machine.

Programmation réseau
(Socket) en UDP

Durée : 3h.

N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) qui doit être envoyé avant le début de séance suivante à l'enseignant responsable de ce TP. Ce CR fera état d'une description détaillée des programmes développés.

Attention : L'enseignant responsable du TP contrôlera que vos programmes fonctionnent en fin de séance.

But: Le but de ce TP est de vous initier à la programmation des sockets, en mode UDP, en langage Java. Ce TP s'étends jusqu'à la moitié de la séance prochaine.

1. Une application Client/Serveur simple

Dans un premier temps, vous programmerez une application client-serveur simple en mode UDP. La spécification de ce programme est la suivante :

- le client envoie en mode UDP une chaîne de caractères au serveur via le réseau,
- le serveur affiche la chaîne qu'il reçoit et la renvoie immédiatement au client, et reste en écoute pour d'autres requêtes
- le client affiche la chaîne qu'il reçoit du serveur et ferme la socket.

Cette application est connue sous le nom d'***écho***, où le serveur renvoie au client le texte reçu. Pour cet exercice, consultez la `javaDoc` des classes `DatagramSocket` et `DatagramPacket` du package `java.net` qui proposent des facilités pour la programmation de socket en mode UDP. Aussi, vous pouvez vous inspirez de l'exemple donné dans le cours. N'oubliez pas de choisir un port d'écoute pour le serveur supérieur à 1024.

En développant votre application, vous lancez votre serveur sur votre machine, et testez votre client en envoyant du texte au serveur local (« `localhost` »). Par la suite, vous pouvez lancer votre client avec un serveur d'un de vos camarades (le nom d'une machine est affiché sur l'unité centrale).

Répondre aux questions suivantes :

- Pourquoi le serveur doit être en écoute sur un port supérieur à 1024 ?
- Quel est le résultat obtenu,
    - si vous lancez le client avant le serveur ?
    - si le client et le serveur n'utilisent pas le même port UDP ?
    - comment le serveur reconnaît-il l'adresse IP de retour du client, sachant qu'il n'y a pas d'établissement de connexion ?

2. L'application PING

a. Introduction

L'application (outil) ***ping***, est utilisée sur Internet pour diagnostiquer le fonctionnement d'une machine sur le réseau. Le ***ping*** consiste en l'envoie d'un texte vers une machine, si elle existe sur le réseau, elle renvoie le texte reçu. Pour le tester, sélectionnez le nom d'une machine dans la salle, et tapez dans une console : ***ping –c 10 nom_machine***

Le résultat de cette commande vous donne des statistiques sur le nombre de paquets envoyés, reçus et perdus, et aussi le RTT entre votre machine et l'autre machine.

Pour cet exercice, vous allez implémenter la partie client du ***ping***. Pour cela vous disposez du code (`serverPING.java`) du serveur dans le répertoire (/share/l3info/syr_r/tp_udp). Vous devez compilez et étudier en détails ce code, qui vous aidera pour la réalisation du client.

Le serveur utilisé permet de simuler l'effet d'une perte de paquets sur UDP, en injectant des erreurs grâce à la variable `LOSS_RATE`. Par exemple, une valeur de 0.3, introduit une perte de paquets de 30% (1 paquet sur trois sera perdu).

b. Le client Ping

Vous devez implémenter un client ping qui envoie 10 requêtes au serveur. Chaque message contiendra le mot clé PING, un numéro de séquence et le temps d'émission. La fin du message est déterminée par un espace et retour ligne. Le message doit contenir la chaîne de caractères suivante :

```
PING numero_sequence temps '\n'
```

Le numéro de séquence représente le numéro de la requête, il est entre 0 et 9. Le temps représente le moment de création du paquet, il peut être obtenu grâce à la classe `Date()` du package `java.util`.
Après une seconde, et si le serveur ne renvoie aucune réponse, vous supposerez que le paquet est perdu dans le réseau. Pour mettre en place ce comportement, vous pouvez utiliser la méthode `setSoTimeout` disponible avec la classe `DatagramSocket` et l'Exception qui lui est associée.
Le résultat de votre client, doit ressembler à celui aperçu dans la section précédente avec la commande ***ping***, c.-à-d., vous devez afficher pour chaque paquet reçu une sortie de la forme :

```
Paquet reçu de : « @IP serveur » , pour la Req. numéro :
       « num_séquence », RTT : , « rtt claculé ».
```

Pour pouvoir afficher ces résultats, il vous suffira de parser le paquet reçu en utilisant la méthode `splite( )` et comme séparateur un espace.

Aussi, vous devez afficher des statistiques sur le nombre de paquets envoyés, reçus et perdus.

Comme pour la première partie, vous lancez votre serveur sur votre machine, et testez votre client en envoyant du texte au serveur local (« `localhost` »). Par la suite, vous pouvez lancer votre client avec un serveur d'un de vos camarades (le nom d'une machine est affiché sur l'Unité).

Programmation réseau
(Socket) en TCP

Durée : 1h

N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) qui doit être envoyé avant le début de séance suivante à l'enseignant responsable de ce TP. Ce CR fera état d'une description détaillée des programmes développés.

Attention : L'enseignant responsable du TP contrôlera que vos programmes fonctionnent en fin de séance.

But: Le but de ce TP est de vous initier à la programmation des sockets, en mode TCP, en langage Java.

3. Programmation socket TCP

Lors du TP précédent vous avez programmez l'application ***écho*** (le serveur renvoie au client le texte reçu) en utilisant UDP. Pour cet exercice, vous utiliserez le service TCP au lieu d'UDP.
N'oubliez pas de consulter la `javaDoc` des classes `ServerSocket` et `Socket` du package `java.net` qui proposent des facilités pour la programmation de socket en mode TCP. Aussi, vous pouvez vous inspirer de l'exemple donné dans le cours. N'oubliez pas de choisir un port d'écoute pour le serveur supérieur à 1024.

En développant votre application, vous lancez votre serveur sur votre machine, et testez votre client en envoyant du texte au serveur local (« `localhost` »). Par la suite, vous pouvez lancer votre client avec un serveur d'un de vos camarades (le nom d'une machine est affiché sur l'unité centrale).
Répondre aux questions suivantes :

- Quel est le résultat obtenu, si vous lancez le client avant le serveur ?
- Quelle est la différence par rapport au cas UDP ?
- Donnez les avantages et inconvénients d'implémenter l'application ***écho*** en utilisant :
    - UDP,
    - TCP

Client/Serveur TCP :
HTTP

Durée : 2h.

N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) à l'enseignant responsable de ce TP. Ce CR fera état d'une description détaillée des programmes développés.

Attention : L'enseignant responsable du TP contrôlera que vos programmes fonctionnent en fin de séance.

Rappel: « Les enseignements de TP sont obligatoires. Deux absences non justifiées à une séance de TP entraînent la note 0 pour la note de TP de la matière au sein de l'UE. »

But: Lors d'un précédent TP, vous avez analysé le protocole HTTP. Pour ce TP, vous allez réaliser une application Client/Serveur qui utilise ce protocole. C'est-à-dire, mettre en place un Client et un Serveur HTTP basics.

4. Rappel sur le protocole HTTP

Une requête HTTP minimale est un message textuel d'une ligne ayant la forme suivante :

```
GET /test.html HTTP/1.0 \n\n
```

Cette requête demande le fichier HTML « test .html » présent dans la racine du serveur, en utilisant la version 1.0 du protocole HTTP.
La réponse du serveur est constituée d'un en-tête et un corps, séparé par une ligne vide :

- L'en-tête est sous la forme suivante :

```
HTTP/1.0 200 OK \r\n
```

  Elle informe le client que la page existe et elle est envoyée dans le corps du message qui suit après la ligne vide. L'en-tête réponse peut contenir d'autres champs, que vous avez pu voir en cours et analysez lors du premier TP.

5. Le client HTTP simple

Ecrivez un programme Java qui se connecte en TCP à un serveur HTTP (par exemple : anubis.ifsic.univ-rennes1.fr), demande la page HTML (par exemple : /TP_syr/fichier1.html), affiche le texte de la page reçue sur la sortie standard.

- Qu'est ce que vous recevez comme réponse ?

6. Le serveur HTTP simple :

Ecrivez un programme attendant une connexion sur un port donné (n'oubliez pas de choisir un port d'écoute pour le serveur supérieur à 1024), capable de recevoir des requêtes HTTP, et de retourner toujours une réponse HTTP simple avec une page HTML basic (<HTML> Hello world </HTML>). Le serveur ne gérera qu'un seul client.

- Testez votre serveur avec votre client HTTP.
- Testez votre serveur avec un navigateur web (*http://localhost:port*).

Enrichissez le message répondu par votre serveur : affichez par exemple l'heure, le nom de la machine serveur, son système d'exploitation, la version de Java utilisée, ou encore des informations sur le client : numéro IP, navigateur, fichier demandé.

7. Aller plus loin : amélioration du serveur

Faites un serveur capable d'envoyer les fichiers demandés dans la requête. Créez un répertoire qui contiendra les pages Web et passez-le en paramètre au démarrage du serveur (ou codez-le en dur dans le programme). Si le fichier n'existe pas, le serveur doit générer une réponse HTTP adéquate

(`404 Not Found`). Pour mener à bien ces nouvelles fonctionnalités sur le serveur, il faut que ce dernier *parse* la requête du client pour extraire le nom du fichier demandé. Comme pour le *ping* UDP, vous utiliserez la méthode `splite( )` et comme séparateur un espace, pour parser la requête du client.
Pour la lecture à partir d'un fichier, vous utiliserez la classe `FileInputStream`. Aussi ce fichier, doit-être envoyé en octets, en utilisant la méthode `write (fichier_en_bytes, 0, la_taille_du_fichier_bytes)`.

- Testez votre serveur en utilisant votre client.
- Testez votre serveur en utilisant un navigateur web (http://localhost:port/nomfichier.html).

Socket RAW : ICMP

Durée 2h.

N.B. : Ces travaux pratiques font l'objet d'un Compte Rendu (électronique) qui doit être envoyé avant le début de séance suivante à l'enseignant responsable de ce TP. Ce CR fera état d'une description détaillée des programmes développés.

Attention : L'enseignant responsable du TP contrôlera que vos programmes fonctionnent en fin de séance.

But:
Le but de ce TP est de manipuler des sockets de type RAW. A cet effet, nous réaliserons un outil qui effectue un ***ping*** en utilisant le protocole ICMP et particulièrement le type ECHO_REQUEST.

Introduction
Dans ce TP, nous allons créer un paquet ECHO_REQUEST et à recevoir son éventuelle réponse, comme le fait le programme ***ping***. Pour cela, rappelons qu'un paquet ECHO_REQUEST est un paquet IP contenant un paquet ICMP, lui-même contenant d'éventuelles données.
Pour la réalisation de ces travaux, vous disposez du squelette du programme. Les endroits ou le squelette est à compléter sont matérialisés comme suit :

```
//commentaire
/ligne blanche à remplir/
//
```

Manipulation
Utilisez le squelette (`icmp.c`) pour coder une application ressemblant au ping. Il est disponible dans le répertoire TP-Repr sur le serveur web *anubis*. Pour la construction des en-têtes IP et ICMP, regardez les fichiers `/usr/include/netinet/ip.h et /usr/include/netinet/ip_icmp.h`

1. Donnez la signification de la `ligne « packet = (char *) malloc(sizeof(struct ip) + sizeof(struct icmp)); »` ?
2. Pourquoi le numéro de port est à zéro ?
3. Donnez les champs constituant l'en-tête IP ? Comment sont-ils codés dans la structure ip ?
4. Donnez la signification de la ligne `« setsockopt(sock, IPPROTO_IP,IP_HDRINCL, &optval, sizeof(int)) ; »` ?
5. Donnez les champs constituant l'en-tête ICMP ? Comment sont-ils codés dans la structure icmp ?
6. Pourquoi le checksum est il calculé à la fin de la construction des en-têtes ?
7. Pourquoi il n'existe pas de fonction `bind()` dans le programme ?

Lancez et vérifiez que votre programme fonctionne bien, c-à-d que les paquets ICMP ECHO_REQUEST et ICMP ECHO_REPLY sont échangés entre les deux machines. Aidez-vous de l'analyseur de trafic Wireshark.

8. Améliorez le programme afin de calculer le RTT entre les deux machines ?

2ème partie : Routage

- Routage statique
- Routage dynamique : RIP
- Routage dynamique : OSPF
- Routage inter-domaine : BGP

Routage statique

Durée : 2h.

A la fin de ce TP vous devez rendre un rapport répondant aux questions des parties 5.2 et 5.3

1. But du TP

Le but de ce TP est de vous familiariser avec l'adressage d'un réseau IP et la configuration statique du routage. Aussi, vous allez découvrir le routeur *Cisco 1941 Series*.
Nous emploierons tout au long du TP diverses commandes et outils afin de contrôler, modifier et vérifier le routage. Voici une liste non exhaustive des commandes utilisables (***man***, ***ifconfig***, ***route***, ***netstat***, ***wireshark***, ***ping***, ***traceroute***, ...).
Wireshark vous permet de visualiser ce qui passe ou ne passe pas je vous conseille d'en avoir toujours un d'ouvert.
💣 ***N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.***

2. Le déroulement du TP

La mise en place du TP se déroulera en trois temps :
- mise en place de la topologie,
- adressage de votre machine,
- prise en main du routeur.

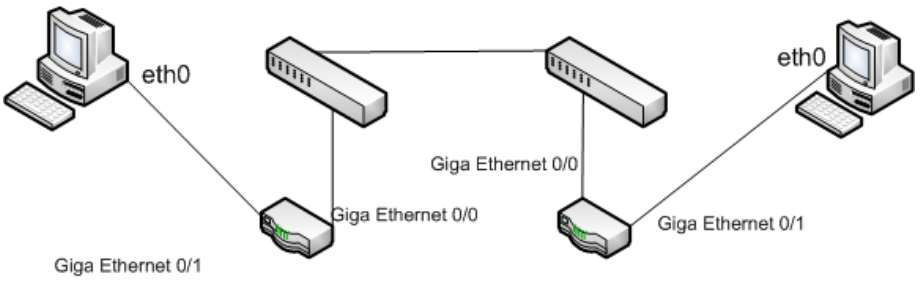
3. Vérifications préliminaires

Utilisez le compte *usertp* de votre machine : {Login: usertp ; Mot de passe : 2+en+dur}. Le compte *usertp* ne possède pas les droits *root* {Login: root ; Mot de passe : 2+en+dur}, donc il faut utiliser le terminal super utilisateur (sur le bureau).
Vérifiez que :
- Le routeur est branché.

Vous disposez maintenant (normalement) de tout ce qu'il faut pour commencer.

4. Mise en place de la topologie

Avant de commencer la partie routage, vous devez au préalable connecter les câbles comme décrits dans la figure ci dessous.

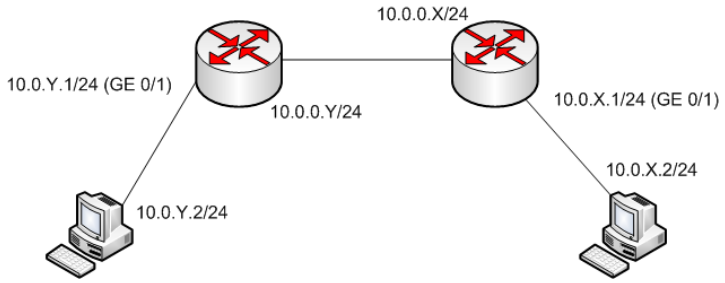


5. Mise en place de l'adressage de votre machine

Nous allons nous servir des différentes interfaces réseau de vos machines pour simuler trois réseaux distincts par banc, vous travaillerez donc en collaboration avec vos voisins.

4.1. L'adressage

Nous allons maintenant simuler deux réseaux différents séparés par Internet comme le montre la figure suivante. Pour cela nous allons utiliser l'interface ***eth1*** du PC.



- Donnez à l'interface ***eth1*** l'adresse 10.0.X.2/24, où X est votre numéro de machine.
- Votre machine ne sait pas sur quelle interface envoyer les paquets. Configurez une route par défaut à destination de l'adresse de l'interface ***Giga Ethernet 0/1*** de votre routeur. Cette adresse aura la forme: 10.0.X.1/24.
- Attendez que vos voisins soient arrivés au même point. Envoyez un ***ping*** sur l'adresse de la machine voisine. Que se passe-t-il ? Pourquoi?

4.2. Configuration des routeurs

Vos routeurs ne sont pas configurés par défaut. Ils disposent de deux interfaces *Giga Ethernet (**giga ethernet 0_0*** et ***giga ethernet1_0*)** et une interface *Fast Ethernet* (100 Mbps). Pour vous connecter au routeur, vous devez au préalable lancer la commande ***minicom***. La commande ***?***, vous donne toutes les options disponibles comme la complétion sous ***bash***.
Branchez vous sur le port console du routeur. Vous devez avoir le prompt *router>*. Vous allez devoir passer en administrateur avec la commande ***enable*** (comme pour le switch).
Il faut commencer par configurer les adresses IP des deux interfaces *Giga ethernet*.

- Accédez au mode configuration (***configure terminal***), et accédez au *context* de l'interface *Giga Ethernet 0/0*. Attribuez l'adresse 10.0.X.1/24 à cette interface (utilisez l'aide ? afin de trouver la commande nécessaire).
- Dans le même *context*, il faut informer le routeur que cette interface doit être activée. Pour cela, il faut taper la commande ***no shutdown.***
- Attribuez l'adresse 10.0.0.X/24 à l'interface *Giga Ethernet 0/0*. N'oubliez pas le ***no shutdown.***
- Attendez que vos voisins soient arrivés au même point. Envoyez un ***ping*** sur l'adresse en 10.0.0.X du routeur voisin, puis sur l'adresse en 10.0.X.2 de la machine voisine. Que se passe-t-il ? Pourquoi ?

La dernière étape consiste à configurer des routes statiques dans votre routeur. Cela se fait dans le *context* racine de « ***configure terminal*** ».

- Configurez votre routeur pour qu'il fasse suivre les paquets à destination de votre voisin sur l'interface *Giga Ethernet 0/0* du routeur voisin (la commande à mettre en place commence par ***ip***).
- Pour vérifier si la route a bien été enregistrée, tapez ***show ip route*** dans le *context* racine du routeur.
- Attendez que vos voisins soient arrivés au même point. Envoyez un ***ping*** sur l'adresse en 10.0.X.2 de la machine voisine. Que se passe-t-il ?

4.3. Observation du protocole ARP

Lorsqu'un routeur fait suivre un *datagramme* IP d'un segment *ethernet* à un autre, il ne modifie pas l'adresse IP destination. En revanche, l'adresse *ethernet* de destination dans l'en-tête *ethernet* est modifiée.

- Effacez la table ARP de votre machine. (***man arp***).
- Attendez que votre voisin ait fait de même.
- Démarrez *wireshark* sur les deux PC et capturez le trafic *ARP* et *ICMP* sur l'interface ***eth1***.
- Les PC côté mur/fenêtre envoient un ***ping*** à leur voisin sur leur adresse en 10.0.X.4.
- Stoppez le ***ping*** et la capture.

- Regardez les adresses source et destination dans les en-têtes *wireshark* sur votre machine et la machine de votre voisin. Expliquez les différences.

4.4. Aller plus loin : observation du proxy ARP

- Sur les routeurs activez la fonctionnalité *proxy-arp* sur les deux interfaces.

Procédure à effectuer uniquement sur un des PCs du banc.

- Effacez la table ARP et la table de routage du PC.
- Modifiez le masque du PC à 255.0.0.0, tel que le PC suppose son appartenance au réseau 10.0.0.0/8 au lieu du 10.0.X.0/24.

Lancez *wireshark* sur les deux PC.

- Lancez un ***ping*** du PC où il y a eu la modification vers l'autre PC du banc. Est-ce que ceci fonctionne ? pourquoi ?

Arrêtez la capture et interprétez le résultat.

- Désactivez la fonctionnalité de *proxy-arp* sur les deux interfaces de un des routeurs.
- Est-il encore possible que les deux PC puissent se pinger ?

6. Remise à zéro de votre matériel

5.1. Remise à zéro du routeur

- Tapez ***reload*** sur le terminal *minicom*. A la demande de sauvegarde, répondez non.

5.2. Remise à zéro de la machine

- Sur le PC Debian, lancez les scripts *init_machine.sh* et *init_reseau.sh* disponibles dans le répertoire /script.

Routage Dynamique
RIP

Durée : 4h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 3.1, 3.2

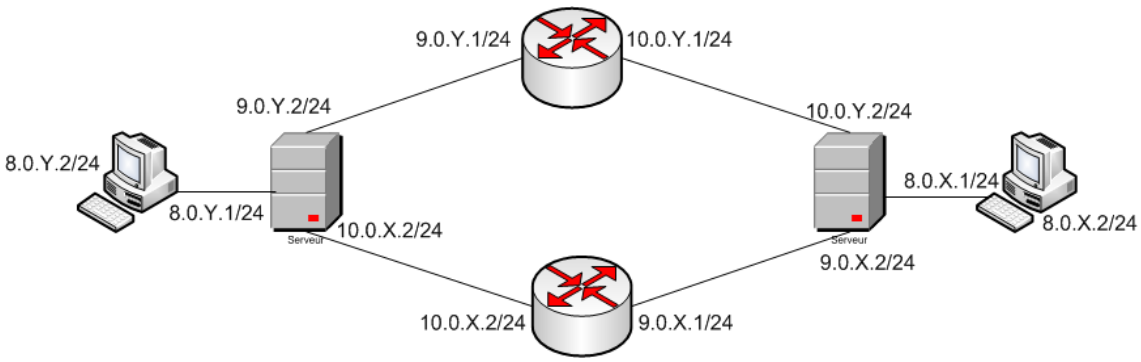
1. But du TP

Ce TP a pour but l'étude de la configuration automatique du routage dans un réseau, grâce au protocole de routage intra-domaine: *RIP* (*Routing Information Protocol*) basé sur les vecteurs de distance.
Nous utiliserons pour cela les routeurs *Cisco* et *Quagga* (anciennement *Zebra*) qui est un routeur logiciel d'interface proche d'un routeur *Cisco*. *Quagga* implémente les protocoles *RIP*, *OSPF* et *BGP*.
Nous utiliserons les quatre PCs du banc (deux sous Debian et deux sous Windows). Les PCs windows serviront comme clients dans la topologie.

N'oubliez pas avant de partir de traiter la partie 4 afin de remettre la salle en état.

2. Mise en place de l'architecture

Ci-dessous l'architecture que vous devriez mettre en place.



2.1. Configuration préliminaire

- Sur la machine Linux, désactivez toutes les interfaces *eth* de la machine. Vérifiez qu'il n'y a plus de routes. Configurez les interfaces *eth1* (reliant le serveur au client), *eth2* (reliant le serveur au routeur de votre banc) et *eth3* (reliant le serveur au routeur de votre voisin) avec les adresses 8.0.X.1/24, 9.0.X.1/24, 10.0.Y.2/24, respectivement. X est votre numéro de machine (serveur) et Y celui de la machine à coté.
- Sur la machine windows (arrêt/défil), configurez l'interface *Broadcom NetXtrem* avec l'adresse IP 8.0.X.2/24. Configurez la route par défaut, en spécifiant comme passerelle l'adresse 8.0.X.1.
- Vérifiez que les deux machines peuvent se *pinger*. N'avancez pas tant que le ***ping*** ne fonctionne pas.

3. Le protocole RIP

Le protocole RIP est l'un des protocoles de routage dynamique le plus utilisé par les routeurs. Il permet à un routeur d'échanger des informations de routage avec un autre routeur, afin de mieux déterminer les chemins à suivre sur le réseau. Il est basé sur un jeu d'algorithmes qui emploient des vecteurs de distances pour comparer mathématiquement des itinéraires.

3.1. Configuration de *RIP* sur la machine hôte Debian *(Quagga)*

Quagga est un routeur logiciel implémentant les principaux protocoles de routage. Il utilise un système de commande similaire à un Cisco. Vous pouvez trouver toutes les informations nécessaires sur le site: *http://www.quagga.net*.

3.1.1. Mise en place

Quagga est déjà installé sur le système dans */etc/quagga*. Pour le lancer ou le relancer, on utilise le script */etc/init.d/quagga*. Le mot de passe à utiliser pour se connecter aux démons *zebra* et *ripd* est « zebra ».

- Editez le fichier */etc/quagga/daemons*, et indiquez *yes* pour que les démons *zebra* et *ripd* soit lancer avec les scripts *init.d*.
- Essayez de démarrer *Quagga*, selon le message d'erreur, copiez le modèle de fichier de configuration minimale de *zebra et ripd* (***zebra.conf.sample et ripd.conf.sample***) (dispo. dans le répertoire */usr/share/doc/quagga/examples*) au bon endroit (*/etc/quagga*).

Il reste à positionner quelques droits. Pour cela tapez dans un terminal, ***chown quagga.quaggavty /etc/quagga/*.conf*** et ***chmod 640 /etc/quagga/*.conf***. Donnez rapidement le sens de chaque commande.

- Lancez *Quagga*
- Sur quel port tourne *zebra et ripd* ?
- Comment se connecte-t-on au démon *zebra* et *ripd* pour les configurer ?
- Lancer la commande ***<<?>>*** pour visualiser toutes les commandes utilisables.
- Quelles commandes permettent de configurer les interfaces du routeur dans le démon *zebra* ?
- Dans *zebra*, configurez les interfaces *eth1*, *eth2* et *eth3* de votre machine comme précédemment.
- Sauvegarder votre configuration avec la commande ***write file***.

Dans un terminal tapez la commande ***echo "1" > /proc/sys/net/ipv4/ip_forward.*** Quel est le but de cette commande ?

3.1.2. RIP

Nous allons maintenant configurer le routage RIP sur *Quagga*. Connectez-vous au démon *ripd*.

- Tapez ***configure terminal***, ***router rip***, ***version*** *2*, ***network 8.0.X.0/24, network 9.0.X.0/24***, ***network 10.0.X.0/24***, ***redistribute connected***, puis ***write file*** pour sauver votre configuration. Expliquez brièvement le rôle de ces commandes ?
- *Zebra* possède une commande pour visualiser les routes *RIP*. Tapez ***show ip rip*** à la racine du démon ***ripd***. Dans un terminal, tapez ***netstat -rn***. Quelles différences voyez-vous entre les deux tables ? Existe-t-il une route par défaut ? pourquoi ?
- Lancez *Wireshark.* Quels sont les paquets *rip* qui circulent ? quel est leur rôle de ces paquets ?

3.2. Configuration de votre routeur *Cisco*

Au préalable vous devez configurer les interfaces *Giga Ethernet 0/0* (reliant le routeur au serveur) et *Giga Ethernet 0/1* (reliant le routeur au serveur voisin) avec les adresses suivantes : 10.0.X.1/24 et 9.0.X.1/24. N'oubliez pas d'activer ces interfaces.

La configuration RIP du Cisco est similaire à celle du logiciel *Quagga*. Il vous faut déclarer les deux routes connues du routeur à savoir 10.0.X.0 et 9.0.X.0.

- Relevez l'état de vos tables de routage (*Quagga* et routeur). Sur le *Cisco* la commande est ***show ip route*** à la racine.
- Vérifiez que du client vous pouvez envoyer un ***ping*** sur les deux interfaces du routeurs ?

- Observez la capture *Wireshark* sans la stopper.
- Quels paquets voyez-vous passez ? quels sont les paquets *rip* qui diffèrent de la section 3.1.2.
- Relevez à nouveau les tables de routages de votre routeur et de votre *quagga*.
- Quelles différences notables y voyez-vous ?

4. Convergence du protocole RIP lors de la panne d'un lien

Avant de casser un lien, il faudrait vérifier avec ***tracert*** sous windows le chemin choisi entre les deux clients.

- Lancez un ***ping*** entre les clients. Faite en sorte que ce ***ping*** envoie une centaine de paquet ICMP.
- Déconnectez un câble constituant le chemin choisi par RIP.
- Attendre jusqu'à ce que le ***ping*** fonctionne, ça peut durer un certain temps.
- A partir des traces affichées par le ***ping*** compter le nombre de paquets perdus. Par la suite, déduire le temps mis par RIP pour trouver un chemin alternatif suite à la perte d'un lien (note : généralement la commande ***ping*** envoie approximativement un *ICMP echo_request* toutes les secondes).

5. Remise à zéro de votre matériel

5.1. Remise à zéro du routeur

- Tapez ***reload*** sur le terminal *minicom*. A la demande de sauvegarde, répondez non.

5.2. Remise à zéro des machines

- Lancez les scripts *init_machine.sh* et *init_reseau.sh* disponibles dans le répertoire /script.

Routage Dynamique
OSPF

Durée : 2h.

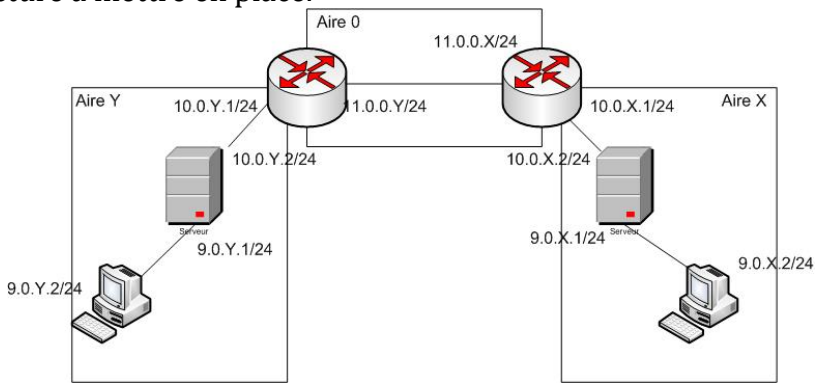
A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 3.1, 3.2

1. But du TP

Ce TP a pour but l'étude de la configuration automatique du routage dans un réseau, grâce au protocole de routage intra-domaine: *OSPF* (*Open Shortest Path First*) basé sur l'état de lien.
Comme pour le TP sur RIP, nous utiliserons les routeurs *cisco* et le logiciel *Quagga* (anciennement *Zebra*) qui est un routeur logiciel d'interface proche d'un routeur *Cisco*. *Quagga* implémente les protocoles *RIP*, *OSPF* et *BGP*. Nous utiliserons les quatre PCs du banc (deux sous Debian et deux sous Windows). Les PCs windows serviront comme clients dans la topologie.

N'oubliez pas avant de partir de traiter la partie 4 afin de remettre la salle en état.

Ci dessous l'architecture à mettre en place.



2. Mise en place de l'architecture

2.1. Configuration préliminaire

- Sur la machine Linux, désactivez toutes les interfaces *eth* de la machine. Vérifiez qu'il n'y a plus de routes. Configurez les interfaces *eth1* (reliant le serveur au client), *eth2* (reliant le serveur au routeur) avec les adresses 9.0.X.1/24, 10.0.X.2/24, respectivement. X est votre numéro de machine (serveur).
- Sur la machine windows (arrêt/défil), configurez l'interface *Broadcom NetXtrem* avec l'adresse IP 9.0.X.2/24. Configurez la route par défaut, en spécifiant comme passerelle l'adresse 9.0.X.1.

Vérifiez que les deux machines peuvent se *pinger*. N'avancez pas tant que le ***ping*** ne fonctionne pas.

3. OSPF

OSPF est un protocole de routage dynamique défini par l'IETF à la fin des années 80. Ce protocole a deux principales caractéristiques : *(1)* Il est ouvert, le sens du terme Open de OSPF ; *(2)* Il utilise l'algorithme du plus court chemin ou Djikstra.
La configuration qu'on voudra mettre en place pour cette partie est illustrée dans la figure ci-dessus.

3.1. Configuration de *OSPF* sur un PC *Linux*

Comme pour *RIP*, *Quagga* implémente le protocole *OSPF*. Le PC *Debain* sera configuré comme routeur ce qui nous permettra de voir les informations *OSPF* échangées entre les routeurs.

- Configurez les interfaces ***eth1*** et ***eth2*** dans *zebra.*

- Pour que le noyau Linux prenne en compte le relais des paquets entre ces interfaces, il faudrait activer l'option *ip_forward* comme suit***: echo "1" > /proc/sys/net/ipv4/ip_forward***
- Faites en sorte que *Quagga* lance *zebra* et *ospfd* avec *init.d.*
- Lancez *Quagga* (n'oubliez pas de positionner les droits sur les fichiers *.conf)
- Connectez-vous à *ospfd*
- Tapez ***configure terminal***, ***router ospf, network 9.0.x.0/24 area x, network 10.0.x.0/24 area x, reditribute connected, exit, write file.*** Expliquez brièvement le rôle de ces commandes ?.
- Démarrez une capture de trafic sur l'interface ***eth2*** de la machine Debian.
- Quels sont les paquets OSPF transmis ? y'a-t-il des réponses ? quelle est l'adresse de destination et pourquoi ?
- *Zebra* possède des commandes pour visualiser la base de donnée topologie OSPF (***show ip ospf database***) et les voisins (***show ip osfp neighbor***). Ces deux commandes doivent être lancées à la racine.

3.2. OSPF sur le routeur *Cisco*

Nous allons rapidement configurer le routeur pour qu'il implémente le protocole OSPF. Il est important ici de noter que ce routeur appartient au ***backbon***e (*area 0.0.0.0*). Donc chaque interface doit appartenir à une *area* différente.

- Configurez les interfaces GigaEthernet 0/0 (reliant le routeur au switch, donc au routeur voisin) et GigaEthernet 0/1 avec les adresses 11.0.0.X/24 et 10.0.X.1/24.
- Lancez *Wireshark* sur l'interface ***eth2*** du pc *Debian*.

La configuration est fait grâce aux commandes suivantes:

Configure terminal, router ospf x, network 11.0.0.0 255.255.255.0 area 0.0.0.0, network 10.0.x.0 255.255.255.0 area x, redistribute connected

- Quels sont les paquets OSPF transmis ? y'a-t-il des réponses ?

Attendez que votre binôme voisin en soit au même point que vous.

- Arrêtez *Wireshark* et donnez la signification des paquets OSPF qui circulent entre le routeur et le PC.
- Quelle est la signification des paquets DB, Desc, LSU, et LSR ?
- En interprétant ces paquets, déduisez les étapes de la création de la table de routage en OSPF ? Relevez l'état de la base de donnée topologie OSPF sur le routeur PC (Zebra) et le routeur. Sur le *cisco* la commande est ***show ip ospf database.***
- Que représentent les différentes tables affichées ? (indication. Une table contient les LSA diffusés par chaque routeur. Une table contient les LSA diffusés par le DR. Une table contient un résumé des routes diffusées par le ABR)

On peut obtenir plus de détails en utilisant la commande ***show ip ospf.*** On peut préciser par exemple les informations pour chaque interface Ethernet déclarée dans la configuration OSPF.

- Quelles sont les informations que les routeurs du *backbone* ont sur la topologie de *l'area x et l'area y.*
- Quelles sont les informations que les routeurs de *l'area y* ont de *l'area x* et *l'area 0.*
- Qui est le routeur désigné de votre area ?
- Affichez dans *Zebra* le (s) routeur(s) de bordure de l'*area x ou y*.

Branchez le câble « salle » au *switch*.

- Affichez la nouvelle table de routage. Voyez-vous l'ensemble des aires ? Quels sont les routeurs de bordures que vous détectez par le biais de la table de routage ?

4. Remise à zéro de votre matériel

4.1. Remise à zéro du routeur

- Tapez ***reload*** sur le terminal *minicom*. A la demande de sauvegarde, répondez non.

4.2. Remise à zéro des machines

- Sur le PC Debian, lancez les scripts *init_machine.sh* et *init_reseau.sh* disponibles dans le répertoire /script.

Sur le PC Windows, re-cochez l'attribution automatique des adresses IP.

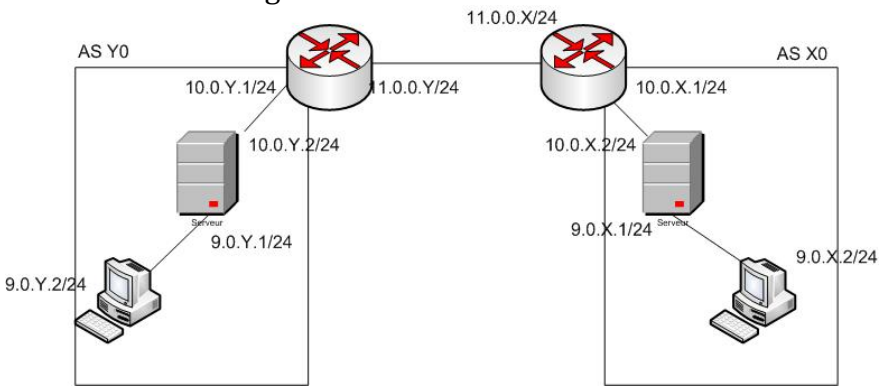
Routage Dynamique
Inter-domaine : BGP

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 4 et 5

1. But du TP

Internet est composé de multiples systèmes autonomes aussi nommés *Autonomous System* (AS), chaque AS est repéré par un identifiant. Derrière un système autonome se trouvent de multiples réseaux différents. A l'intérieur de chaque système autonome il existe des politiques de routage, or ces politiques diffèrent d'un AS à un autre. Pour communiquer entre différents systèmes autonomes il faut donc réussir à gérer ces différences de politiques de routages. C'est le rôle du protocole BGP.
Nous utiliserons pour cela les routeurs *cisco* et le logiciel *Quagga* (anciennement *Zebra*) qui est un routeur logiciel d'interface proche d'un routeur *Cisco*. *Quagga* implémentera le protocole *RIP* pour ce TP. Chaque PC contient une machine hôte Debian faisant office de routeur logiciel, et une machine virtuelle Debian faisant office de client.
Chaque binôme va simuler un domaine Autonome (AS) représenté par un routeur de bordure et un réseau interne comme le montre la figure suivante :



💣 ***N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.***

2. Configuration préliminaire

- Sur le PC Debian, désactiver toutes les interfaces. Vérifiez qu'il n'y a plus de route. Configurez les interfaces ***eth1*** (reliant le PC Debian avec le PC Windows) et ***eth2*** (reliant le PC debian avec le routeur) avec les adresses 9.0.X.1/24 et 10.0.X.2/24, respectivement. X est votre numéro de machine.
- Sur la machine Windows, donnez l'adresse 9.0.X.2/24 à l'interface *ethernet* active. Configurez la route par défaut, en spécifiant comme passerelle l'adresse 9.0.X.1.
- Vérifiez que les deux PC se *ping*, n'avancez pas tant que le ***ping*** ne fonctionne pas.
- Branchez le câble salle au *switch*.
- Activez l'interface ***eth3*** du PC linux, et branchez la au *switch*.
- Sur le *switch* activez la fonctionnalité *Port-Mirroring* pour rediriger le trafic du port salle au port ***eth3***.

3. Déroulement du TP

3.1 Le routage interne à l'AS

Pour le routage interne nous allons utiliser le protocole RIP, inspirez vous du TP sur ce dernier pour réaliser les taches suivantes :

- Configurez *Quagga* sur le PC pour faire du RIP.
- Configurez l'interface ***GigaEthernet 0/1*** pour faire du RIP.

4. Le routage externe *BGP*

BGP est le protocole de routage dynamique inter-domaine (un domaine étant un ensemble de routeurs formant une communauté gérée par une entité administrative unique. Le domaine correspond à la notion de système autonome). *BGP* permet d'échanger des informations entre des réseaux ayant des politiques de routage différentes et notamment d'assurer, par l'utilisation de vecteurs de chemin, une protection contre les boucles de routage.

4.1 Configuration de *BGP* dans le *Cisco*

Pour configurer votre routeur *BGP* il y a trois étapes:

- Lui dire, de quel système autonome il est la bordure.
- Lui dire, qui sont ces voisins.
- Lui dire, quel préfixe de réseau il doit annoncer.

BGP étant un protocole de routage dynamique, toutes la configuration du routeur se fait dans le contexte ***router bgp***.

- Configurez l'interface ***GigaEthernet 0/0*** de votre routeur avec l'adresse 11.0.0.X/24
- Dans le contexte ***router bgp***, configurer votre identifiant d'*AS*.
- Puis, dans le même contexte déclarez vos voisins, grâce à la commande (***neighbor***).
- Enfin, déclarez les préfixes de réseau que vous annoncez : 9.0.X.0/24, 10.0.X.0/24 et 11.0.0.0/24. Ici on a déclaré statiquement les réseaux pouvant être routés par BGP. Il existe aussi la solution de laisser BGP annoncer les routes apprises par RIP.
- Sur le *Cisco*, vous pouvez contrôler votre configuration. A la racine du routeur, tapez ***show ip bgp neighbors***
- Commentez ce que vous obtenez.
- Relevez la table de routage du routeur. Commentez ce que vous voyez.
- Démarrez une capture de trafic sur l'interface ***eth3*** du *PC*.
- Envoyez un ***ping*** vers la machine d'un binôme qui en est au même point que vous.
- Quels types de messages *BGP* voyez vous passer ?
- Quel protocole de transport utilise BGP ? Pourquoi ?
- Quelle est l'adresse IP de l'attribut *next hop* pour l'AS 00(x-1) ?
- Pourquoi, vous n'arrivez pas à pinger une machine à l'intérieur d'un AS voisin ? aidez vous des tables de routages des *routeur*.

5. Convergence du protocole BGP

Nous allons maintenant visualiser les informations échangées par le protocole *BGP* lorsqu'un lien tombe en panne.

- Dans le contexte racine tapez la commande ***ip bgp neighbors***, et relevez les valeurs ***keepalive timers, holdown timer ?*** A quoi correspondent ces *timers* ?
- Passez dans le *contexte bgp* pour modifier ces deux *timers* comme suit: ***bgp timers 10 20***.
- Démarrez une capture sur l'interface ***eth2*** de *PC*.
- Débranchez le câble salle du *Switch* (On arrête le lien qui vous connecte au reste de la salle)
- À partir d'*Wireshark*,
    - Expliquez comment BGP sait qu'un lien est tombé en panne ?
    - Quels messages indiquent qu'il y a un problème sur le lien ?

6. Remise à zéro de votre matériel

6.1. Remise à zéro du routeur

- Tapez ***reload*** sur le terminal *minicom*. A la demande de sauvegarde, répondez non.

6.2. Remise à zéro des machines

- Sur le PC Debian, lancez les scripts *init_machine.sh* et *init_reseau.sh* disponibles dans le répertoire /script.

Sur le PC Windows, re-cochez l'attribution automatique des adresses IP.

3ème partie : Administration des réseaux

- Domain Name Server (DNS) BIND 9
- Service de messagerie : Postfix
- Annuaire LDAP (Lightweight Directory Access Protocol)
- Samba

Domain Name Server (DNS)
BIND 9

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 2 et 4

1. But du TP

Le but de ce TP est de vous familiariser avec le protocole *DNS* et l'architecture qui le supporte. Pour cela, nous rappelons les principes essentiels du *DNS* et nous les mettons en oeuvre grâce au logiciel *BIND* (*Berkeley Internet Name Domain*).

💣 ***N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.***

2. Mise en route

Pour cette partie vous êtes invité à lire les pages du manuel de *dig*
A l'aide de la commande *dig* :

- Déterminez : l'adresse IP de la machine anubis.istic.univ-rennes1.fr l'adresse IP du serveur de nom de la zone istic.univ-rennes1.fr
  - l'adresse IP d'un des serveurs de nom de la zone racine
  - le nom correspondant à l'adresse IP 193.51.24.1
  - le relais de courrier associé à l'ISTIC.
- Comment peut-on visualiser l'utilisation du cache du service DNS à l'aide de *dig* ?
- En utilisant l'option *norecurse,* trouver l'adresse d'une machine lointaine de votre choix. Quel rapport avec ***dig +trace*** ?

3. Installation du logiciel BIND

Le serveur Bind9 est déjà installé sur votre machine. Les fichiers de configuration se trouvent dans le répertoire */etc/bind*. Pour lancer ou arrêter le serveur, il faut utiliser le script *init.d* par la commande ***/etc/init.d/bind9 start/stop/restart.***
Il est important de regarder le fichier de log (*/var/log/syslog*) du *Bind* lorsque vous tentez de le démarrer.

4. Création de votre zone

Lors de ce TP vous allez créer votre propre zone. Pour cela vous serez tous une zone fille de la zone *i207* qui est gérée par la machine 148.60.12.25. Votre nom de zone sera *m0X* où *X* est votre numéro de machine. Ainsi, le nom complet de la zone du binôme situé sur la machine 1 sera *m01.i207*.

- Quel est le mécanisme mis en place au niveau de la machine 148.60.12.25, qui permettent de savoir que vous êtes responsable de la zone fille *m0X.i207*.

Le fichier principal de configuration est le *named.conf*. Parcourez ce fichier, vous remarquez qu'il consiste en une énumération d'options et de zone. A l'intérieur de chaque bloc vous trouvez le chemin vers le fichier qui décrit la zone concernée, ainsi que le rôle de votre serveur pour cette zone: *master* ou *slave*.

- Quelle est la différence entre un serveur primaire et un serveur secondaire ? Quelles sont les relations qui les unissent ?
- En fin du fichier *named.conf.default-zone*, ajoutez le bloc donnant le nom de votre zone, ainsi que le chemin d'accès au fichier descriptif (on l'appellera *db.m0X .i207*).
- A quoi correspondent les différents paramètres ?
- Que faut il au minimum dans un fichier de zone ?

- Créez le fichier *db.m0X.i207* et renseignez-le pour avoir la configuration minimale de votre zone. Il devra avoir la forme suivante:

```
$ORIGIN m0X.i207.
$TTL 64800
@   IN  SOA   ns.m0X.i207. admin.ns.m0X.i207. (
              AAAAMMJJNN ; numéro de série au format
              10800 ; refresh
              3600  ; retry
              604800; expire
              86400 ; minimum
              )
        IN  NS   ns.m0X.i207.
ns   IN  A    148.60.12.X
serveur  IN CNAME ns
```

- A quoi correspond la variable *TTL* ?
- Pourquoi le « . » terminal de la variable *$ORIGIN* est il important ? Commentez ce fichier.

Votre machine se base sur un fichier de configuration pour connaître le serveur DNS à interroger pour résoudre une requête.

- Quel est le nom de ce fichier de configuration ? modifiez ce fichier afin que votre machine interroge votre serveur.
- Testez en faisant un ***ping ns.m0X.i207***.
- En utilisant *dig*, déterminez l'adresse du serveur gérant la zone i207. Quel est le résultat obtenu ? pourquoi ?
- Editez le fichier *named.conf.options*, regardez si vous pouvez régler le problème précédent ? (si vous n'arrivez pas à le faire me consulter).
- Déterminez l'adresse du serveur gérant une autre zone fille de la salle.

5. Montez un serveur secondaire

Vous allez maintenant monter un service secondaire pour supporter votre zone. Votre voisin sera secondaire pour votre zone et vous vous serez secondaire pour la zone de votre voisin.

- Ajouter un bloc dans *named.conf* pour déclarer votre serveur comme secondaire (*slave*) de votre voisin.

```
zone "m0Y.i207" {
        type slave"
        file "slave/m0Y.i207"; //ici il faut créer un répertoire slave
        masters {148.60.12.Y;};
};
```

- Dans le *named.conf* du principal il faut que vous autoriser le transfert de la zone du primaire vers le secondaire. Ajoutez *allow-transfer {148.60.12.X}*
- Rajoutez dans le fichier zone du serveur principal, le serveur secondaire de votre zone.

Lancez *Wireshrak* et relancez les serveurs (si ça ne fonctionne par, la solution se trouve dans les fichiers log)

- Quel est le type de la requête du serveur secondaire qui demande le transfert des enregistrements ?
- Quel est le protocole de transport utilisé ? pourquoi ?
- Peut-on avoir les mêmes résultats avec *dig*

6. Remise à zéro de la machine

Lancez les scripts *init_machine.sh* et *init_reseau.sh* disponibles dans le répertoire /script.

Services de Messagerie
Postfix

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 2 et 3

1. But du TP

Le but de ce TP est de vous présenter la mise en place et l'administration d'un service de messagerie, ainsi que les traitements et restrictions complémentaires devenus indispensables de nos jours.
Pour les besoins du TP, vous administrerez un domaine dont le nom sera *m0X.i207* où x est le dernier chiffre de l'adresse IP de votre machine.
Les *logs* de votre serveur *SMTP* se trouve normalement dans le fichier */var/log/syslog*.
Vous trouvez les mails conservés sur votre machine dans le répertoire */var/spool/mail*.

2. Installation d'un serveur *SMTP*

2.1. Installation de *Postfix*

Si *Postfix* n'est pas installé sur votre machine, tapez dans une console : ***aptitude install postfix.***
Répondre oui pour l'ensemble des questions, et à la fenêtre vous demandant de configurer le serveur, il faut entrer : *ok -> pas de configuration ->ok*.

- Modifiez votre DNS *(/etc/bind/db.m0X.i207* à récupérer sur http://anubis/TP_messagerie*)* pour dire que le relai de messagerie de votre domaine est votre machine (*mail.m0X.i207*).
- Modifiez votre fichier /etc/resolv.conf, à fin d'interroger votre serveur DNS.
- Ajouter un utilisateur au système avec *useradd –r tpmail*

2.2. Vérification de l'installation

Avant de commencer, copiez les fichiers *main.cf* et *master.cf (*disponibles sur le http://148.60.12.25/TP_messagerie) dans */etc/postfix.*
Postfix possède deux fichiers de configuration: *main.cf* et *master.cf*.

- Regardez à quoi ressemblent ces deux fichiers qui se trouvent dans le répertoire */etc/postfix*. Quelle est la syntaxe générale du fichier *main.cf* ?
- Utilisez la commande ***postconf -n*** puis ***postconf -d***. Qu'obtenez vous ? essayez ***postconf*** *un_paramètre*, qu'obtenez vous ?
- Démarrez le serveur, si ça ne marche pas utilisez le résultat de la commande ***postconf –d nom_parametre_manquant*** pour régler ces problèmes.

Vous allez maintenant envoyer un mail à l'utilisateur *tpmail@m0X.i207*. Pour cela, il faut spécifier à votre *Postfix* de délivrer les mails pour le domaine *m0X.i207.*

- Localisez la ligne *#myhostname=*.. qui spécifie le nom de machine hébergeant *Postfix*. Enlever le # et faite en sorte que la ligne ressemble ***myhostname=****mail.m0X.i207*
- Localisez une ligne contenant ***mydestination*** dans votre fichier et assurez-vous qu'elle contient à la fin les lignes : *$mydomain*, *mail.$mydomain*

Tapez ***newaliases*** et redémarrez *Postfix.*
Pour envoyer un mail, utilisez la commande ***mail*** comme suit:
mail tpmail@m0X.i207
sujet : coucou
bonjour
.
Vous allez vérifier que cet email est bien arrivé.

- Connectez-vous sur votre machine en tant que *tpmail* et vérifiez que vous avez bien reçu un mail (toujours grâce à la commande ***mail***).

2.3. Manipulations

Editez le fichier et vérifiez que la ligne *inet_interfaces = all*, car elle spécifie que votre *Postfix* accepte les connexions de l'extérieur. C'est grâce à cette ligne que la commande ***telnet mail.m0X.i207 25*** fonctionne par exemple. *Il faudrait impérativement la mettre à **all** pour la suite du TP.*

Généralement, il est agréable de disposer d'une adresse sous la forme *Prénom.nom@nom_domaine*, plutôt qu'un login. Pour cela on définit des alias pour un login d'une machine. Vous allez donc créer ces alias pour le compte *root*.

Les alias se trouvent dans le fichier */etc/aliases*

- Editez ce fichier pour créer deux alias pour *root* du type *Prénom.Nom* et *Nom.Prénom*.
- Editez le fichier *main.cf* et décommentez les lignes *: alias_maps = hash…..*
- Lancez la commande ***newaliases*** pour mettre à jour la base des alias. Sinon, ça ne fonctionnera pas.
- Recharger la configuration grâce : ***postfix reload***.
- Testez que l'utilisation de vos alias fonctionne en envoyant comme précédemment un mail à *Prénom.Nom@m0X.i207* et un mail à *Nom.Prénom@m0X.i207*.
- Envoyez un mail avec une erreur dans le nom (par exemple: *userto@m0X.i207*). Votre mail est il arrivé ? Où se trouve-t-il ?

A présent vous allez essayer d'envoyer un message vers un compte *root* d'une autre machine.
Vider le cache DNS avec la commande ***rndc flush*** (il se peut que cette commande ne soit pas reconnue par les nouvelles distributions Linux. Ces dernières n'ont pas de cache DNS local). Utiliser la commande ***mail*** tout en ayant *Wireshark* en tache de fond.

- Quelle est la suite de commandes échangées entre les deux machines ?
- Peut-on connaître à partir de la réponse du serveur MTA, le status de l'email envoyé ?

3. Relais SMTP

Pour cette manipulation, vous allez travailler en collaboration avec le poste voisin. Faites d'abord la manipulation d'un binôme vers le second, puis une fois que cela fonctionne, inverser les rôles.
Nous voulons faire transiter du courrier d'un serveur *SMTP* à l'autre sans passer par la résolution de nom. Pour cela, il faut configurer une machine pour recevoir du courrier et l'autre machine pour le relayer. Cela se fait via les fichiers */etc/postfix/transport* (un exemple est disponible dans le répertoire TP-mail à l'adresse http://148.60.12.25) et *main.cf*.

- Editez le fichier */etc/postfix/transport* et regardez les différents commentaires. Ils expliquent les différentes manières de spécifier un transport de mail.
- Sur votre machine, décommettez la ligne *transport_maps =hash:/etc/postfix/transport* de votre **main.cf**. (ajoutez la ligne si la ligne n'existe pas).
- Editez le fichier **transport**, trouvez la ligne de commentaire expliquant:
- **example.com smtp:bar.example.com** et lisez le commentaire. En fin de fichier, ajoutez la ligne *m0Y.i207 smtp:[mail.m0Y.i207]*. La notation *[nom]* permet d'éviter une requête DNS sur ce nom pour demander un enregistrement MX (Mail eXchanger). Votre machine relaie désormais le mail pour le domaine *m0Y.i207*.
- Tapez la commande ***postmap /etc/postfix/transport*** pour créer la base *transport.db*.
- Faites un test en envoyant un mail à *root@m0Y.i207*, avec *Wireshark* comme tache de fond (n'oubliez pas de vider le cache DNS avant)
- En vous basant sur Wireshark et l'email reçu retracez le chemin de ce message. Quelle est la différence avec la résolution DNS.

Ce que vous venez de mettre en place devrait être interdit sur Internet ! cela s'apparente à un relais ouvert et les spammeurs en sont très friands.

4. Restriction sur le relayage

- Regardez la section *example* du fichier ***access*** (un exemple est disponible dans le répertoire TP-mail à l'adresse http://148.60.12.25). Il vous explique ce qu'il faut modifier pour restreindre l'accès à votre serveur *SMTP*.
- Interdisez à la machine *mail.m0X.i207* de se connecter au serveur *SMTP* du *mail.m0Y.i207*. Editez le **main.cf** du *mail.m0Y.i207* et ajouter une ligne ***smtpd_client_restrictions = ...*** comme présenté dans la section **example**. Vous remarquez que la politique d'accès est située dans le fichier ***access***.
- Editez le fichier ***access*** sur la machine *mail.m0y.i207* et ajoutez une ligne interdisant au *mail.m0x.i207* d'envoyer du courrier.
- Il ne faut pas oublier de créer la base, ***postmap access*** et de recharger *postfix*
- Cette fois-ci vous allez envoyer un mail en se connectant au serveur *SMTP* par *telnet*. Que se passe-t-il ?
- Commentez vos deux lignes de restrictions, et refaite le test suivant:

  telnet i207m0y 25
  ...
  hello serveur
  ...
  mail from: root@m0X.i207
  250 OK
  rcpt to: root@m0Y.i207
  250 OK
  Data
  to: a@b.c
  subject: yo !
  .
- Que se passe-t-il ?

Postfix permet aussi de faire du filtrage sur le contenu.

5. Filtrage du contenu

- Regardez dans le fichier *main.cf* les commentaires liés à la ligne ***header_checks = ...*** Vous remarquez que la vérification est faite grâce au fichier ***header_checks***.
- Editez le fichier sur votre machine et ajouter une ligne filtrant les mails contenant: Sujet d'examen.

Envoyez un mail avec cette phrase en sujet grâce à Telnet, que se passe-t-il ?

Annuaire LDAP *(Lightweight Directory Access Protocol)*

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 3 et 4

1. But du TP

Le but de ce TP est de vous familiariser avec l'utilisation des annuaires LDAP comme base d'authentification d'utilisateurs de système *Unix*, ou pour d'autres types d'application tel que : *Apache*, *Samba*, *Postfix*, etc. Pour cela, nous nous baserons sur le serveur open source *OpenLDAP*.

💣 ***N'oubliez pas avant de partir de reinstaller le système.***

2. Service d'annuaire LDAP

Un service d'annuaire est une base de données spécialement optimisée pour la recherche, le survol et la lecture rapide d'informations. Elle stocke des données légèrement typées, organisées selon des classes particulières et présentées dans un arbre. L'exemple le plus commun dont il tire son nom est l'annuaire de personnes. Mais il peut stocker bien d'autres choses : des comptes Unix, des données personnelles (carnet d'adresses, photos, numéros de téléphone, etc), un parc matériels (imprimantes, pc..). Pour cela LDAP dispose de plusieurs schémas (Ex. *nis.schema* pour l'authentification unix) en plus des schémas standard (*core.schema*).

3. Configuration d'OpenLDAP

OpenLDAP est un annuaire libre mettant en œuvre le protocole LDAP, sous une licence qui est équivalente à la licence BSD révisée. Il est dérivé du serveur LDAP de l'Université du Michigan, et a largement évolué depuis. Historiquement, la configuration de ldap reposait uniquement sur le fichier *slapd.conf*. A partir de la version 2.23, ce fichier a été éclaté en sous fichiers disponibles dans le répertoire *slapd.d*. Pour plus de détails voir le lien suivant http://www.openldap.org/doc/admin24/slapdconf2.html).
Utiliser *apt-get* pour installer les paquets *slpad*, *ldap-utils* ainsi que toute dépendance. Si l'installation de *slapd* demande un mot de passe entrez *secret*.
Durant le TP vous avez la responsabilité du domaine *m0X.i207* (X le numéro de votre machine). Pour reconfigurer ldap, on utilisera l'outil disponible avec Debian pour la reconfiguration des packages. Pour cela, tapez ***dpkg-reconfigure slapd***. Les réponses à fournir sont :

Omit LDAP server configuration : non
Domain DNS : m0X.i207
Organization : m2pro
Mot de passe de l'admin : secret
Type de base de données : HDB
Do you want the database to be removed when slapd is purged : Non
Purger la vielle BD : Non
Utiliser le protocole LDAPv2 : Non

Normalement, *Slapd* se lance comme un service grâce à *inet.d*.

- Quel est le port TCP utilisé par votre serveur ldap ?
- Vérifiez à l'aide de la commande netstat que votre serveur est bien à l'écoute sur le port TCP ?
- On peut vérifier aussi si le serveur est bien démarré en l'interrogent. Pour cela on utilisera la commande *ldapsearch* (tapez ***man ldapserach*** pour en savoir plus). La requête sera comme suit :

ldapsearch –x –b "dc=m0X,dc=i207"

- Quel est le résultat retourné ? Expliquez brièvement.

On va maintenant créer une entrée dans l'annuaire, elle concernera la personne qui est par exemple, le responsable de cette organisation.

- Editez un fichier que vous nommerez *base.ldif* (toutes les entrées à rajouter à l'annuaire doivent passer par un fichier .ldif). Ce fichier est structuré comme ceci :

```
#déclaration du responsable
dn: cn=nom, dc=m0X, dc=i207
objectClass : top
objectClass : person
userPassword : secret
cn: nom
sn: responsable
```

- L'utilitaire pour l'ajout dans l'annuaire est *ldapadd* (man *ldapadd*). Pour ajouter les deux enregistrements tapez :
  ldapadd –f base.ldif –x –D "cn=admin,dc=m0X,dc=i207" –W
- Détaillez les paramètres utilisés avec cette commande ?

Pour modifier un enregistrement on utilise ***ldapmodify***. Comme pour ***ldapadd***, éditez un fichier avec comme extension *ldif*, il doit contenir :

```
dn: dc=m0X,dc=i207
changtype: modify
add: telephoneNumber
telephoneNumber: 01 23 45 67 89
```

- Donnez une explication à ces lignes ? utilisez ***ldapmodify*** pour metre à jour cette entrée.
- Utilisez la commande *ldapsearch* pour rechercher l'ensemble des enregistrements rajoutés à la racine *dc=m0X,dc=i207*. Tapez ***ldapsearch –x –b 'dc=m0X,dc=i207' '(objectclass=*)'***
- Raffinez la recherche pour ne récupérer que les objets de type *person.* Trouvez le numéro de téléphone de l'organisation *m2pro*.

4. Authentification basée sur LDAP

A présent, on va utiliser le serveur *LDAP* pour faire de l'authentification d'utilisateurs au lieu des fichiers */etc/group* et */etc/passwd*. Il faudrait passer par deux étapes :

- Ajoutez les utilisateurs et les groupes dans l'annuaire,
- Dire à la machine d'utiliser un serveur LDAP pour authentifier les utilisateurs.

4.1. Ajout des utilisateurs et des groupes dans l'annuaire

Pour faire la différence avec les utilisateurs installés sur la machine, on va créer un nouveau compte utilisateur qui n'est pas déclaré dans le fichier */etc/passwd*.

On va commencer par déclarer les groupes et les utilisateurs comme une *organizationalUnit* de *m0X.i207*.

- Editez un fichier que vous nommerez *o_unit.ldif*. Il doit contenir les lignes suivantes :

```
dn: ou=People,dc=m0X,dc=i207
objectClass: organizationalUnit
ou: People
description: People

dn: ou=Group,dc=m0X,dc=i207
objectClass: organizationalUnit
ou: Group
description: Groupes
```

- Utilisez ***ldapadd*** pour ajouter ces deux enregistrements

On va créer un groupe pour les utilisateurs.

- Editez un fichier que vous nommerez *group.ldif*. Il doit contenir les lignes suivantes :

dn: cn=utils,ou=Group,dc=m0X,dc=i207
cn: utils
objectClass: posixGroup
objectClass: top
gidNumber: 9000
description: utilisateurs sans droits

- Ajouter ce fichier à l'annuaire

Finalement l'utilisateur *toto*.

- Editez un fichier que vous nommerez *utils.ldif.* Il doit contenir les lignes suivantes :
  dn: cn=toto,ou=People,dc=m0X,dc=i207
  cn: toto
  objectClass: account
  objectClass: posixAccount
  objectClass: shadowAccount
  objectClass: top
  uid: toto
  uidNumber: 1025
  gidNumber: 9000
  homeDirectory: /home/toto
  loginshell: /bin/sh
  userpassword:#à chiffrer avec slappasswd en MD5, il faut inclure avec le modt de passe {md5}

- Ajoutez ce fichier à l'annuaire
- Expliquez les différents champs des fichiers *group.ldif* et *utils.ldif* ? retrouvez les différents champs en relation avec une authentification *unix*.
- Lancez une recherche pour afficher les informations concernant toto

4.2. Configuration du PC en client LDAP

La dernière opération consiste à configurer le PC en client LDAP, pour utiliser ce dernier pour l'authentification des utilisateurs. Pour cela, modifiez le fichier */etc/ldap/ldap.conf* (référence pour le clien LDAP). Ce fichier doit contenir les deux lignes suivantes :

BASE dc=m0X, dc=fr
URI ldap://127.0.0.1/

- Installez les packages suivants *libnss-ldap libpam-ldap nscd*
- Répondre aux questions comme suit :
  URI : ldap://127.0.0.1/ (attention ldap et non ldapi)
  Distinguish name : dc=m0X,dc=i207
  Ldap version :3
  LDAP account for root : cn=admin,dc=m0X,dc=i207
  Root passwd : secret
  Allow root local admin : yes
  LDAP database requires login : no
  LDAP account for root : cn=admin,dc=m0X,dc=i207
  Root passwd : secret

- Editez le fichier */etc/nsswitch.conf pour rajouter le support d'authentification LDAP*
  passwd : compat ldap
  shadow : compat ldap
  group : compat ldap
- Copiez le fichier */usr/share/doc/libpam_ldap/example/pam.d/ssh* à l'emplacement */etc/pam.d/sshd*
- Arrêtez le service nscd *(/etc/init.d/nscd stop*).

- Redémarrez le service LDAP

L'utilisateur toto n'a pas de compte de système. Il existe que dans l'annuaire LDAP.

- Testez que toto est visible : ***getent passwd***
- Créez un répertoire et affectez-le au compte toto.
- Ouvrez une session dans votre terminal avec toto (par exemple : ***su toto***), ou lancez un ssh sur votre machine avec le compte toto.

Samba

Durée : 4h.

But : le TP se propose d'investiguer les problématiques de partage de fichiers en réseau hétérogène à travers la mise en œuvre d'un service SMB sur Unix ; chaque binôme utilisera le PC Windows XP le PC Linux Debian. Le but du TP est, en premier lieu, d'installer et configurer Samba sous Debian et interagir avec la machine Windows. Par la suite, accrocher la machine Debian au domaine Active Directory (ADS) de la salle i207 pour assurer une authentification unique basée sur ce dernier (ADS).

Comptes et mot de passe associés :

Compte d'utilisateur	Mot de passe	Accès depuis
admin	2+en+dur	Tout poste Windows XP
admineasi	2+en+dur	Administrateur du domaine Active Directory
usertp	2+en+dur	Utilisateur Debian
root	2+en+dur	Administrateur de la machine Debian

Rappels Linux :

- L'aide sur les différentes commandes s'obtient par **man** <commande> ou (plus succinctement) par <commande> **--help**

Rappels Windows XP:

- L' aide sur les commandes standards de Windows XP s'obtient via **Démarrer ► Aide et support ► Performances et maintenance ► Résolution des problèmes de performances et de maintenance ► Référence de A à Z de la ligne de commande** ; également en tapant le nom de la commande suivie de /?

Windows XP :

---

1. Analyse succincte de la configuration réseau TCP/IP

- Commande **ipconfig**
  b) Affichez toutes les informations de votre configuration IP. Donnez les informations essentielles pour la carte Ethernet « Connexion au réseau local » (nom d'hôte, adresse IP et masque de sous-réseau pour le moins).
  c) Affichez le contenu du cache DNS et tentez d'interpréter brièvement la sortie obtenue.
- Commande **nslookup**
  d) Forcez le serveur DNS à 148.60.12.23 dans les **Propriétés** de la **Connexion au réseau local ► Protocole Internet (TCP/IP) ►Avancé… ►** onglet **DNS ► Ajouter…**Testez quelques résolutions de nom ou d'adresse. Au vu du résultat, qu'en déduisez-vous ?
- Commande **netstat**
  e) Affichez toutes les connexions et les ports en écoute sur votre machine. Y'a-t-il des connexions établies et si oui lesquelles ?
- Commande **ping** puis **ipconfig**
  f) Envoyez des requêtes « ping » sur les autres machines de la salle (i207m10 à i207M20). Au vu du résultat, qu'en déduisez-vous ?
  g) Affichez de nouveau le contenu du cache DNS et justifiez le résultat.

- Commande **`netdiag`**
  h) Comment pouvez-vous facilement générer un rapport détaillé de votre configuration réseau ?

2. Analyse succinte de la configuration réseau NetBT

- Commande **`nbtstat`**
  a) Listez la table de nom NetBIOS locaux et identifiez quelques ressources actives sur votre machine.
  b) Déterminez combien de noms sont résolus par diffusion et via WINS. Au vu du résultat comment votre machine opère-t-elle la résolution nom NetBios ↔ adresse IP ?
- Commande **`net config`**
  c) Affichez les paramètres du service Serveur. Quelles informations pouvez-vous en tirer ?
- Commande **`netdiag`** (Support Tools)
  d) Arrivez-vous à localiser un PDC pour le domaine « Salle i207 » ? S'agit-il vraiment d'un domaine (au sens Windows du terme) ? Même questions pour le domaine « Easi ».
- Utilitaire **browstat** (Kit de Ressources)
  e) Quel est le Maître Explorateur pour votre « domaine » ?

Linux Debian :

---

3. Analyse succincte de la configuration réseau TCP/IP

- Commandes **`hostname`**, **`domainname`**, **`ifconfig`**
  a) Affichez toutes les informations de votre configuration IP. Donnez les informations essentielles (nom d'hôte, adresse IP et masque de sous-réseau pour le moins).
- Editeur **`vi`** (de préférence)
  b) Quelle commande devez-vous utiliser de façon à changer votre nom de machine en *`sambaX`* (*X* devant prendre la valeur 1, 2, ...10 suivant votre emplacement) ?

4. Packages Samba

- Grâce à **`apt-get`** installer les package suivants : samba smbclient winbind
- Commande **`locate`** (précédé d'un **`updatedb`** au besoin)
  b) Où se trouve le fichier de configuration de Samba (*`smb.conf`*) ?
  c) Modifiez ce fichier pour inclure votre samba au workgourp Worgroup.

5. Configuration bancale de Samba ;-)

- Appuyez-vous sur le document *`diagnosis.html`* (/usr/share/doc/samba)
  Dans ce qui suit vous effectuerez les tests un par un (et dans l'ordre). Pour vous BIGSERVER correspondra à votre machine Debian hébergeant le serveur Samba ; ACLIENT correspondra à votre système hôte Windows XP ; TESTGROUP correspondra à votre groupe de travail WORKGROUP.
- Editeur **`vi`** (de préférence)
  Après avoir sauvegardé le fichier *`smb.conf`* éditez-le afin de décommenter/modifier le partage *`[tmp]`* comme indiqué.
  b) Expliquez succinctement ces modifications.
- Commande **`service smb (scripts init.d)`**
  Démarrez (manuellement) le service samba.
  c) Quels services ont été lancés ?
- Commandes diverses (**`nmblookup`**, ...etc)
  Effectuez les tests un par un (et dans l'ordre). Dans l'hypothèse où vous rencontrez des problèmes de mot de passe au point **3**… insistez un peu ☺. Deux conseils : ne modifiez pas dans tous les sens votre *`smb.conf`* et pensez à jeter un œil sur les logs !
  c) **Test 1** : commentez brièvement
  d) **Test 2** : commentez brièvement

e) **Test 3** : commentez la sortie de la commande **smbclient**
f) **Tests 4, 5 et 6** : commentez brièvement
g) **Test 7** : justifiez les problèmes (probables) que vous rencontrez à accéder au partage *[tmp]*
h) **Tests 8 et 9** : commentez brièvement
i) **Test 10** : commentez brièvement
j) **Test 11** : commentez brièvement

6. Configuration de Samba en mode *user*

- Commande **smbpasswd**
  b) Expliquez brièvement l'usage et l'intérêt de ce programme.
  Utilisez **smbpasswd** pour ajouter un utilisateur *admin* dans la base *smbpasswd* (il faut que ce compte existe aussi dans le fichier /etc/passwd, ajoutez ce compte s'il n'existe pas)
- Programme **pdbedit**
  c) Expliquez brièvement l'usage et l'intérêt de ce programme.
  Vérifiez les informations relatives à l'utilisateur *admin*.
- Commande **service smb**
  Redémarrez (manuellement) le service smb.

7. Test de la configuration en mode *user*

- Commande **net** et **Favoris réseau**
  Côté client (Windows XP) reprenez les points 8 et 9 de *diagnosis.html*.
  a) L'accès est-il concluant ? En écriture également ? Justifiez.
  b) L'accès à votre homedir également ? Justifiez.

8. Configuration personnalisée d'un partage (si le temps le permet)

- Commande **groupadd** et **useradd**
  Côté serveur (Samba/Linux) créez un groupe *master. C*réez aussi un compte *stagiaire* ayant le même mot de passe que côté Windows XP, (rajoutez ce compte sous Windows XP si non existant) et intégrez-le au groupe *master*. Ajoutez ce compte dans la base de compte Samba.
  - Donnez l'enchaînement des commandes.

- Editeur **vi** (de préférence)
  En vous inspirant de la section *[myshare]* du fichier *smb.conf*, créez une ressource partagée (ou service) de nom *[master]* pointant sur le répertoire /usr/master (créé pour l'occasion) avec comme commentaire « Partage du groupe master IR », qui soit accessible en lecture/écriture aux seuls membres du groupe *master* ainsi qu'au compte *stagiaire* ; faites-en sorte que l'accès à ce service soit limité à votre poste client Windows XP et soit visible dans les **Favoris Réseau**
  c) Détaillez et validez votre configuration

9. Configuration de Samba en mode ADS et jonction au domaine EASI du serveur Samba

- Commande **ifconfig, vi**
  A présent on va configurer le serveur samba pour s'accrocher au domaine ADS EASI Pour cela, il faut :
  1. Modifiez le fichier */etc/resolv.conf* afin d'inclure la machine Debian au domaine easi.istic.univ-rennes1.fr, et la machine 148.60.12.23 comme serveur DNS préféré.

2. N'avancez pas tant que vous n'arrivez pas à pinger la machine *psyche* (le contrôleur du domaine ADS EASI).

- Commande **`apt-get`**

  Récupérez et installez les packages *krb5-doc krb5-user krb5-config* du serveur de kerberos5 sur votre machine.

- Editeur **`vi`** (de préférence)

  Modifiez votre fichier `/etc/krb5.conf` pour pouvoir vous authentifier via Kerberos sur un contrôleur de domaine Active Directory. Votre fichier doit inclure dans le [realms] les lignes suivantes (Kerberos est sensible à la casse) :

  EASI.IFSIC.UNIV-RENNES1.FR = {
  kdc = psyche.easi.ifsic.univ-rennes1.fr
  admin_server = psyche.easi.ifsic.univ-rennes1.fr
  }

- Commandes **`kinit`** et **`klist`**

  Testez votre configuration Kerberos avec **`kinit`** pour récupérer un ticket Kerberos pour l'utilisateur `admineasi`. Si le système vous renvoie une erreur indiquant que le temps est très large. Il faudra alors installer le package *ntpdate* afin d'utiliser un serveur NTP. Par la suite il faut tapez ntpdate 148.60.12.25 afin de synchroniser le temps de la debian avec la machine 148.60.12.25. Relancez **`kinit,`** et utilisez **`klist`** pour voir le ticket.

  b) Détaillez l'utilisation et les résultats de **`kinit`**/**`klist`**.

- Editeur vi (de préférence)

  Il faut à présent modifier le fichier `smb.conf` afin que le serveur Samba rejoigne le domaine EASI (modifier aussi le workgroup en EASI). Il faut que ce fichier inclus les lignes suivante (si le paramètre existe et pointe vers une autre valeur, le modifier) :

  security = ads
  realm = EASI.IFSIC.UNIV-RENNES1.FR
  netbios name = sambaX
  idmap uid = 10000-20000
  idmap gid = 10000-20000
  winbind separator = /
  winbind enum users = yes
  winbind enum groups = yes
  winbind use default domain = yes
  template homedir = /home/EASI/%U
  template shell = /bin/bash

  Redémarrez le service samba.

- Commandes **`net ads join`**

  Pour rejoindre le domaine il faut utiliser la commande **`net ads join`** avec comme utilisateur `admineasi` (car il a le droit administrateur sur le domaine, et donc permet de rajouter la machine au domaine).

  Redémarrez le service winbind.

- Commande **`wbinfo`**

  Pour vérifier que la machine Linux a bien rejoint le domaine EASI, utilisez la commande **`wbinfo`**.

10. Utilisation du domaine ADS pour authentifier des utilisateurs de la machine Linux

- Editeur **`vi`** (de préférence)

Afin de configurer l'authentification basée par ADS, il faut modifier le fichier *nsswitch* et le système PAM.

Le fichier `/etc/nsswitch.conf` est le fichier de configuration des bases de données systèmes et des services de noms. C'est à lui que l'on va dire d'utiliser winbind pour trouver les noms d'utilisateurs et groupes rattachés au domaine. Il faut ajouter winbind à passwd, group, et hosts.

Modification de PAM
Editez le fichier `/etc/pam.d/common-auth`

Replacez la ligne;
auth required pam_unix.so nullok_secure
Par
auth sufficient pam_winbind.so
auth required pam_unix.so nullok_secure use_first_pass

Cette nouvelle ligne permet l'authentification ave winbind.

Editez le fichier `/etc/pam.d/common-account`

Insérez avant la ligne;
account requiredpam_unix.so nullok_secure
Cette ligne;
account sufficient pam_winbind.so

Editez le fichier `/etc/pam.d/common-session`

Insérez avant la ligne;
session requiredpam_unix.so
Cette ligne;
session requiredpam_mkhomedir.so skel=/etc/skel/ umask=0022

- Expliquez brièvement ces modifications.

11. Test de la configuration en mode ADS (si le temps le permet)

- Commande **`getent`**
  Lancez la commande **`getent`**, qu'est-ce que vous constatez ?
- Commande **`su`**
  Essayer de vous connecter à un terminal avec le compte `admineasi`.

4ème partie : Sécurité des réseaux

- Serveur web et sécurité
- Détection d'intrusion : Snort
- VPN IPSec
- Architecture PKI
- Architecture 802.1X
- Firewall et cache web

Serveur web et sécurité

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 3 et 4

1. But du TP

Le but de ce TP est de mettre en œuvre un serveur web Apache et de le sécuriser grâce à l'authentification Basic, Digest et SSL (HTTPS).

💣 ***N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.***

2. Introduction

Apache est le logiciel le plus utilisé pour la mise en place de serveur Web. Il est inclus dans toutes les distributions de Linux et existe également pour Windows. Il est totalement gratuit et représente une grosse part des serveurs Internet à travers le monde.
SSL (Secure Socket Layer) est un protocole de sécurisation des échanges de données. Il a été conçu pour assurer la sécurité des transactions sur Internet et est intégré depuis 1994 dans les navigateurs. Plusieurs versions de ce protocole existent :

- la version 2.0 développée par Netscape (inventeur de ce protocole);
- la version 3.0, la plus répandue,
- et la version 3.1 rebaptisée TLS (Transport Layer Security, RFC 2246) et standardisé par l'IETF**.**

SSL est un protocole de niveau 4 du modèle OSI et est conçu pour fonctionner de manière transparente pour les applications qui l'utilisent. De par sa simplicité de mise en œuvre et le niveau de sécurité qu'il offre, SSL est aujourd'hui un protocole de sécurité incontournable pour les applications sécurisées du monde Internet.

3. Configuration matériel

Pour les besoins du TP, vous utiliserez la machine Debian comme serveur et la machine Windows comme client.

3.1. Configuration des machines

Vérifiez que Apache2 est bien installé, sinon ***apt-get install apache2***

3.2. Configuration d'Apache

3.2.1. Emplacement des pages web

Pour le besoin du TP vous allez définir l'accès au serveur en utilisant le répertoire associé à un utilisateur. Donc vous pouvez y accéder avec l'url *http://serveur/~utilisateur*.

- Créez le répertoire */home/usertp/public_html*
- Créez un fichier index.html dans ce répertoire, il doit contenir les lignes suivantes :
  <HTML>
  <BODY>
  <P> C'est de plus en plus dur :-(</P>
  </BODY>
  </HTML>

- Placez-vous dans le répertoire */etc/apache2/mods-enbaled* et tapez « ***ln –s ../mods-available/userdir.* .*** » (n'oubliez pas le point à la fin) pour charger les modules Apache qui gèrent les accès aux répertoires utilisateurs.

3.2.1. Démarrage du démon

- Vérifiez que le serveur web Apache est en écoute sur le port 80, sinon démarrez-le par la commande ***/etc/init.d/apache2*** *start*.
- Vérifiez dans le navigateur du poste *<<Client>>* que vous obtenez bien la page de test d'Apache. N'oubliez pas que chaque modification du fichier */etc/apache2/sites-available/default*, nécessite le redémarrage du démon par: ***/etc/init.d/apache2 restart***

Maintenant essayer de vous connecter au répertoire de l'utilisateur *usertp*

- Qu'est ce que vous constatez ?

Pour résoudre ce problème il faudrait taper : ***chmod 755*** */home/usertp*, ***chmod 755*** */home/usertp/public_html*, ***chmod*** *644 /home/usertp/public_html/index.html.*

- Que signifie ces commandes ?

Re-testez l'accès aux pages de l'utilisateur *usertp*

4. Sécurisation par authentification

Il est parfois impératif de protéger certains répertoires du serveur pour éviter que n'importe qui puisse y accéder. Que ce soit par Internet ou en Intranet, il y a différentes méthodes de procéder: utiliser des scripts serveurs (PHP, ASP, CFM...) pour interdire la lecture de manière sélective et dynamique ou utiliser plus simplement les fonctions de protection de répertoires offertes par Apache. Le module nécessaire à l'authentification est *mod_auth*. Il est chargé par défaut dans la configuration de *Apache2*.

.htaccess et *.htpasswd*

La protection d'un répertoire particulier se fait de deux manières différentes: soit on la configure dans le fichier de configuration d'Apache, soit on crée un fichier nommé *.htaccess* (le point fait partie du nom de fichier) qui sera placé dans le répertoire à protéger. La protection des répertoires par le fichier */etc/apache2/sites-available/default* est du seul ressort du webmaster.

Cette commande devra être écrite pour chaque répertoire. Ce système est assez lourd à gérer au niveau des droits et nécessite un redémarrage d'Apache pour prendre en compte la nouvelle configuration. Les fichiers *.htaccess* (point htaccess) sont plus faciles à gérer. Ils s'appliquent à un répertoire et à tous ses sous-répertoires, mais peuvent être modifiés par un autre fichier *.htaccess* dans un sous-répertoire. Si vous souhaitez que le dossier */home/usertp/public_html* soit protégé via ce fichier *.htaccess*, il vous faudra vérifier que les directives *AllowOverride AuthConfig* sont présentes entre les tags *<Directory / > et < /Directory>.* Cela implique que tous ses sous-dossiers vont également accepter cette directive.

- Que signifie *AllowOverride AuthConfig*?

4.1. Authentification *Basic*

Pour cette première étape de sécurisation on va demander à Apache de faire une authentification *Basic*. Pour cela vous allez éditer (créez) le fichier *.htaccess*, qui est un fichier texte simple contenant des commandes Apache comme celles-ci:

```
AuthUserFile /etc/apache2/password/.pwdbasic
AuthGroupFile /dev/null
AuthName "Identification"
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

Quelques explications:

- *AuthUserFile*: c'est le nom et le chemin d'accès du fichier qui contiendra les mots de passe. S'il ne commence pas par un slash (/), ce sera un sous-répertoire du serveur web. Avec l'exemple ci-dessus, les mots de passe seront dans */etc/apache2/password/.pwdbasic*. Si le répertoire n'existe pas, il faudrait le créer. Dans la pratique il vaudra mieux placer ce fichier en dehors du site web pour que personne ne puisse y accéder.

- *AuthGroupFile*: pointe toujours vers */dev/null*. Il faut que cette ligne soit présente.
- *AuthName* : c'est le texte qui apparaîtra dans la fenêtre demandant les mots de passe. Il représente aussi le *realm* paramètre dans l'authentification *Digest*.
- *AuthType* : L'authentification est en générale <<basic>>. Les mots de passe sont alors envoyés en clair sur le réseau. Pour sécuriser davantage l'accès, on peut utiliser la méthode d'authentification « *Digest* » qui crypte les mots de passe en *MD5* (clef de cryptage). Mais ce système n'est supporté que par certains navigateurs (Opera 4+, Internet Explorer 5+, Amaya).
- *Limit*: C'est ici qu'on va indiquer ce qui est autorisé et interdit dans le répertoire.
  Les commandes GET et POST indiquent la récupération de pages web et la réponse à certains formulaires.
- *Require*: On peut entrer ici <<valid-user>>, ce qui accepte tous les utilisateurs qui ont un (login : mot de passe) dans *.pwdbasic*. Mais on peut aussi limiter à un ou plusieurs utilisateurs précis: *<<require user pierre paul techno>>*. Dans ce cas les utilisateurs sont séparés par des espaces.

Testez votre configuration

4.1.1. Générer des mots de passe *Basic*

Pour créer le fichier *.pwdbasic* sous Linux, vous pouvez utiliser la commande ***htpasswd*** sous la forme suivante : ***htpasswd -c .pwdbasic usertp***
On vous demandera alors un mot de passe qu'il faudra répéter deux fois et cela ajoutera au fichier *.pwdbasic* de ce répertoire une ligne de ce type: username:v3l0KWx6v8mQM
Si vous ne disposez pas d'une machine Unix/Linux, vous pouvez utiliser des générateurs de mots de passe sur le web:

Raptor.golden.net: http://home.golden.net/generator/
htpasswd file generator: http://www.wells.org.uk/htpasswdgen.html

4.1.2. Analyse de l'authentification *Basic*

Relancez Apache par ***/etc/init.d/apache2 restart***, vérifiez que le serveur web demande une authentification pour la consultation de la page web. Refaites le test en capturant le trafic avec *Wireshark.* Pour une meilleure visibilité des résultats dans *Wireshark*, ajoutez un filtre http dans l'onglet *Filter*.

- Dans le paquet *401 authorization required,* qu'elle est l'information véhiculée dans le champ *www-Authenticate* de l'en-tête HTTP ?
- Dans la réponse fournie par le client au précédent paquet, qu'elle est l'information véhiculée dans le champ *Authorization* de l'en-tête HTTP ?
- Qu'est ce que vous constatez ?

4.2. Authentification *Digest*

Pour cette deuxième étape de sécurisation on va demander à Apache de faire une authentification *Digest*. C'est-à-dire, chiffrer le mot de passe avec un mécanisme beaucoup plus efficace (MD5). Pour cela, il faudrait modifier le fichier *.htaccess* de façon à ce qu'il ressemble à :

AuthUserFile /etc/apache2/password/.pwddigest
AuthGroupFile /dev/null
AuthName "Identification"
AuthType Digest
<Limit GET POST>
require valid-user
</Limit>

Il faudrait aussi dire à *Apache* de charger ce module d'authentification. Pour cela placez vous dans le répertoire */etc/apache2/mod-enbaled* et tapez « ***ln –s ../mods-available/auth_digest.* . »***.

4.2.1. Générer des mots de passe *Digest*

Pour créer le fichier *.pwddigest,* vous pouvez utiliser la commande ***htdigest*** sous la forme suivante : ***htdigest -c .pwddigest Identification usertp***

- Donnez la signification des paramètres utilisés ?

4.2.2. Analyse de l'authentification *Digest*

Relancez Apache par ***/etc/init.d/apache2 restart,*** vérifiez que le serveur web demande une authentification *Digest* pour la consultation de la page web. Refaites le test en capturant le trafic avec *Wireshark* et notamment les trames concernant l'authentification:

- Dans le paquet *401 authorization required*, qu'elle est l'information véhiculée dans le champ *www-Authenticate* de l'en-tête HTTP ?
- Dans la réponse fournie par le client au précédent paquet, qu'elle est l'information véhiculée dans le champ *Authorization* de l'en-tête HTTP ?
- Qu'elle est la différence par rapport à l'authentification *Basic*

5. Sécurisation par SSL ou le protocole HTTPS

Pour cette partie vous allez mettre à jour le serveur Apache afin de prendre en charge SSL.

- Chargez les modules SSL en vous plaçant dans répertoire */etc/apache2/mod-enbaled* et en tapant ***ln –s*** *../mods-available/ssl.**
- Editez le fichier */etc/apache2/sites-available/default* afin de rajouter la ligne ***SSLEngine on*** après le Tag <*VirtualHost **>.
- Créez deux répertoires, un pour les clés du certificat (*/etc/apache2/ssl/keys*) et un autre pour le fichier contenant le certificat (*/etc/apache2/ssl/certificats*).
- Tapez les commandes suivantes :

  cd /home/usertp/web

  openssl req –new > new.cert.csr

  openssl rsa –in privkey.pem –out new.cert.key

  openssl x509 –in new.cert.csr –out new.cert.cert –req –signkey new.cert.key –days 999

  cp new.cert.key /etc/apache2/ssl/keys/server.key

  cp new.cert.cert /etc/apache2/ssl/certificats/server.crt
- Que signifient toutes ces commandes ? (***man openssl***, ***openssl –help***, etc)
- Modifiez le fichier */etc/apache2/mod-enabled/ssl.conf*, pour dire à Apache où trouver les certificats et les clés associées. Pour cela rajoutez ou modifiez les lignes :

  SSLCertificateFile /etc/apache2/ssl/certificats/server.crt
  SSLCertificatKeyFile /etc/apache2/ssl/keys/server.key
- Modifiez le fichier *.htaccess* afin de refaire une authentification de Base.

5.1. Analyse du trafic

Lancez *Wireshark* sur le poste <<serveur>>, se mettre en capture sur l'interface utilisée par votre réseau local (@IP) et demandez depuis le navigateur du client, les pages suivantes : https://@ip/~usertp. Arrêtez ensuite la capture et isolez les différents paquets dans ce trafic : pour le serveur web sans SSL.
Pour le serveur web avec SSL. Décrivez les paquets que vous pouvez isoler, notamment la phase de connexion (handshake) et l'établissement du mode SSL.

- Pouvez-vous toujours isoler le login utilisateur ?

6. Remise à zéro de la machine

Sur le PC Debian, lancez les scripts *init_machine.sh* et *init_reseau.sh* disponibles dans le répertoire /script.

Détection d'intrusions : Snort

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 3 et 4

1. But du TP

Le but de ce TP est de mettre en œuvre l'application *Snort* comme outil de détection d'intrusions.

💣 ***N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.***

2. Introduction

Snort est un Système de Détection d'Intrusion, dont les qualités et les capacités ne sont plus à démontrer. Il est gratuit, et son moteur d'analyse est basé sur la technique de recherche de signatures dans les paquets. Il a donc un langage qui permet d'écrire les filtres correspondant à des signatures d'attaques connues.
Chaque binôme dispose une machine Windows XP avec le sensor *Snort* pour détecter les intrusions venant du réseau externe et interne, et la machine Linux (Debian) qui agit comme client malveillant.

3. Installation et configuration de *Snort*

3.1. Configuration et lancement

Sur la machine Windows,

- Décompressez et installez le package *Snort* présent dans le répertoire *soft* sur le bureau. Normalement, le répertoire *Snort* est crée dans *C:*.
- Décompressez les règles qui permettent à *Snort* de détecter les intrusions dans le répertoire *C:\snort.*

Etant donné que *Snort* écoute le trafic et analyse le contenu des paquets reçus, il a besoin de la Bibliothèque *Wincap*. Le package contenant cette bibliothèque est présent dans le répertoire *soft*, il suffit de cliquer dessus pour l'installer.
A présent on va commencer la configuration de *Snort*, éditez le fichier *c:\snort\etc\snort.conf*
Faite en sorte que votre fichier de configuration contient les modifications suivantes :

#Vous devez modifier la variable contenant les adresses IP de votre réseau interne à vérifier.
var HOME_NET 192.168.0.0/24

#Définir les adresses extérieurs à votre réseau, on utilise souvent la notation *any*
var EXTERNAL_NET any

#Spécifier le chemin aux règles utilisées par *Snort* pour la détection d'intrusions.
var RULE_PATH c:\snort\rules

#Permettre à *Snort* de remonter les alertes dans le journal d'application géré par windows
output alert_syslog: LOG_AUTH LOG_ALERT

Redéfinir le chemin sous *Windows* des deux librairies utilisable par Snort
dynamicpreprocessor directory c:\snort\lib\snort_dynamicpreprocessor
dynamicengine c:\snort\lib\snort_dynamicengine\sf_engine.dll

Redéfinir le chemin sous *Windows* de ces deux fichiers de configuration
include c:\snort\etc\classification.config

include c:\snort\etc\reference.config

Pour tester l'installation de *Snort* sur votre système

- Lancez une fenêtre MS-DOS, et naviguez jusqu'au répertoire c:\snort\bin
- Pour détecter laquelle des interfaces est active sur la machine Windows tapez ***snort –W***, relevez son numéro.

Snort peut être utilisé comme un analyseur de trafic (comme *ethereal*), pour cela tapez ***snort –v –ix*** (*x* représente le numéro de l'interface qui est active)

Maintenant on va lancer Snort comme un service windwos.

- Dans la fenêtre *MS-DOS* tapez la commande suivante : ***snort /SERVICE /INSTALL -de -c c:/snort/etc/snort.conf -l c:/snort/log –ix***
- Donnez une explication à la commande précédente ?
- Lancez *Snort* avec la commande ***net start snort***, si tout va bien *Snort* est bien lancé sur votre système.
- Pour vérifier ouvrez la *console services (Démarrer -> Panneau de Configuration – Performances et maintenance> Outils d'administration -> Services)* et constatez que le service *Snort* est bien lancé.

3.2. Les règles Snort

Le but principal de *Snort* est d'informer l'administrateur quand une action correspond à une règle déjà définie. L'administrateur peut utiliser une liste non limitée de règles comme le cas d'un firewall.

Généralement les règles suivent le même format, elles tiennent sur une seule ligne. Par exemple:

log tcp any any -> 148.60.12.25 23 (msg : "telnet to anubis";)

Cette règle enregistre une alerte, ainsi que le paquet, si *Snort* détecte une connexion *telnet* vers la machine 148.60.12.25, peu importe l'adresse et le port source.

Le format des règles est comme suit :

action protocol address port direction address port (rule option)

Dans l'exemple précédent, on a uniquement enregistré l'alerte, néanmoins on aurait pu remonter une alerte en utilisant ***alert*** au lieu de ***log***.

L'opérateur de direction est soit "->" ou "<-" ou "<>" dans le cas de trafics bidirectionnels. La partie *rule option* affine la recherche dans un paquet. Voici un exemple d'une règle *Snort* qui permet de détecter un message ICMP *echo* :

alert tcp any any -> 148.60.12.25 any (itype : 8 ; msg : "ping detected";)

- Pourquoi utilise-t-on le numéro 8 avec itype ?

Il est à noter l'existence de beaucoup d'options à placer entre parenthèses, on peut citer "content", "flags", "ipoption".

- A quoi servent ces options ? (regarder dans le répertoire documentation de snort)
- Donnez la signification de la règle suivante ?

alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg : "Telnet login incorrect"; content: "login incorrect" ; flags : A+)

- Ecrire une règle permettant de générer une alerte à chaque demande d'ouverture d'une connexion HTTP vers votre machine windows. Cette règle devra entre-autre afficher le message « connexion entrante HTTP » et avoir un SID valide.

Pour définir une règle Snort, il faut la rajouter à la fin du fichier *local.rules* disponible dans le répertoire *c:/snort/rules.*

4. Nmap sur Linux et vérification des règles (rajoutées)

Nmap permet d'obtenir des informations sur le système distant. Installez le dans le cas où il n'est pas présent dans le système (*apt-get*).

- Donnez un exemple d'utilisation de *nmap* sur un réseau de l'un de vos voisins ?
- Détectez les ports TCP et UDP ouverts (de 1 à 5000) sur la machine Windows de votre banc ?

- Déterminez grâce à *nmap*, quel est l'OS d'une machine ?
- Observez le journal d'application de la machine *Snort*, qu'est-ce-que vous constatez ?

Demandez à votre voisin de se connecter en HTTP à votre machine Windows, afin de vérifier la règle que vous avez rajoutée précédemment.

- Ecrire une règle simple permettant de générer une alerte à chaque requête http accédant au fichier *index.php* à la racine du site web (mot-clé « content »).
- Cette règle comporte plusieurs problèmes au niveau réseau pouvant engendrer des faux-négatifs. Comment ?

5. Utilisation d'un ACID avec Snort

Dans cette partie on va utiliser la base de données *MySql* pour sauvegarder les alertes et les visualiser à travers une interface graphique basée sur *APACHE/PHP*.

- Sur la machine Windows, lancez un navigateur et tapez *http://localhost/,* cliquez ensuite sur le lien *phpmyadmin.*
- Dans le menu Base de données, il faut créer deux nouvelles bases de données, qu'on nommera ***archive*** et ***snort.***
- Rajoutez un utilisateur (login : *snorty* ; mot_passe : *snorty*) ayant tous les droits sur ces BDs.
- Dans chaque BD (*snort* et *archive*), importez les tables à partir du fichier *create_mysql* présent dans le répertoire *c:\snort\schemas*

Maintenant il faut dire à *Snort* de sauvegardez les alertes dans les deux BDs. Pour cela, vérifiez la présence des lignes suivantes :

output database: log, mysql, user=snorty password=snorty dbname=snort host=localhost sensor_name= lenom_de_machine
output database: alert, mysql, user=snorty password=snorty dbname=snort host=localhost sensor_name= lenom_de_machine

Redémarrez *Snort* à partir de la *console services*

Pour visualiser les différentes alertes de *Snort* on va utiliser l'ACID. Pour cela, modifiez le fichier présent *c:\wamp\www\acid\acid_conf.php*, afin que ce dernier prend en compte les BDs *snort* et *archive*. Vérifiez que les champs suivant sont bien modifiés :

```
$DBlib_path = "c:\adodb";

$DBtype = "mysql";

$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "snorty";
$alert_password = "snorty";

$archive_dbname   = "archive";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "snorty";
$archive_password = "snorty";
```

- Connectez-vous à votre serveur web local, et cliquez sur projet local ACID
- Suivez les instructions, et vérifier que tous les champs sont à *DONE*

Maintenant que l'interface graphique est bien en place, vous pouvez l'utiliser pour visualiser les alertes de *Snort.*

Demandez à l'un de vos voisins de lancer des scans sur la machine Windows

- Que remarquez-vous dans les alertes *Snort* ?
- Donnez brièvement les informations qu'on peut récupérer de l'interface ACID ?

VPN IPSec

Durée : 4h

Ce TP est une adaptation d'un TP de l'IUT de Lannion.

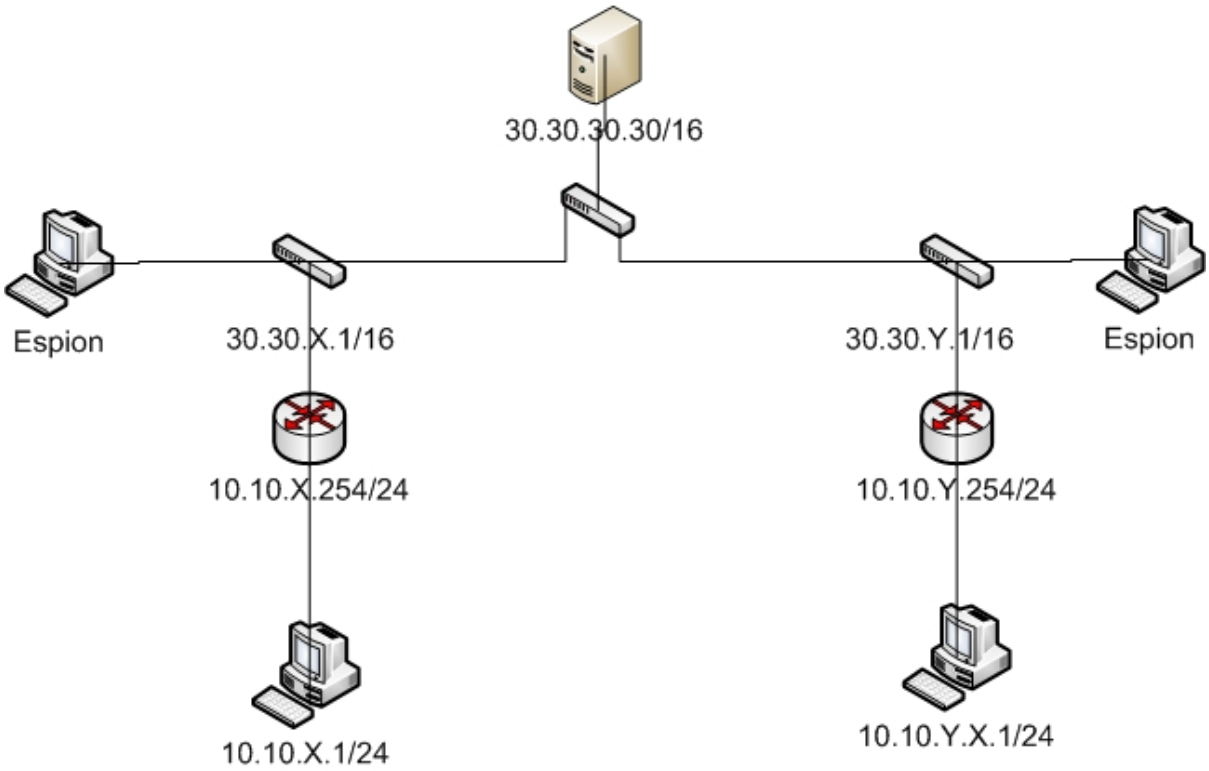
---

1. Objectif du TP

Relier deux sites d'une même entreprise dans une première étape par un tunnel sécurisé par IPSec. Dans une première configuration, vous utiliserez le protocole AH (pas de chiffrement des données) puis vous utiliserez le protocole ESP (chiffrement des données) à clef manuelle.
Vous mettrez ensuite en oeuvre le protocole IKE pour échanger des clefs. Le protocole IKE sera utilisé dans un premier temps avec des clefs pré-partagées et dans un deuxième temps avec une clef publique RSA.

2. Mise place des deux sites

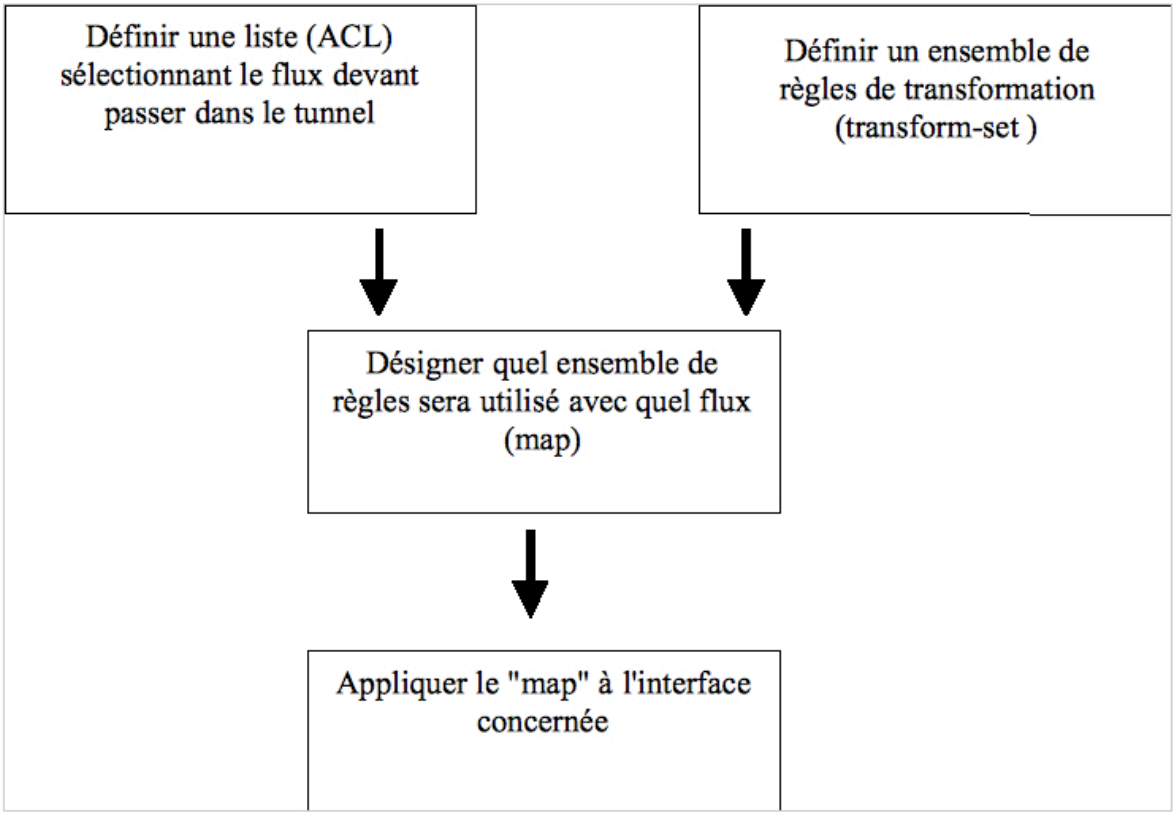
Pour ce TP vous travaillerez avec le binôme voisin.
Câblez votre poste selon le schéma ci-dessous. N'oubliez pas de rajouter les routes et vérifiez que les ***ping*** fonctionnent entre les pc.



- Vérifiez la connectivité.

3. Mise en place d'un tunnel IPSec en AH

Vous allez configurer le tunnel IPSec avec le protocole AH. Les différentes opérations sont à réaliser de chaque coté du tunnel de manière symétrique.
Puis, vous relèverez les trames ICMP (***ping***) passant par le tunnel et les comparerez avec celles qui ne passent pas par le tunnel. Vous noterez les différences et l'intérêt du tunnel ainsi configuré.
On peut schématiser les étapes de configuration à réaliser par le schéma ci-contre :



3.1. Opérations à effectuer pour configurer un tunnel IPSec sans utiliser IKE (clefs manuelles)

1) Spécifier par une liste d'accès les paquets IP qui doivent être protégés, comme ci dessous : ***router(config)# access-list 120 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255*** (autoriser tout trafic du réseau 10.0.1.0 au réseau 10.0.2.0).

Command	Purpose
access-list *access-list-number* {deny | permit} *protocol source source-wildcard destination destination-wildcard* [precedence precedence] [tos tos] [log]	Specify conditions to determine which IP packets will be protected.1 (Enable or disable crypto for traffic that matches these conditions.)

2) Spécifier l'ensemble *transform set* des protocoles de sécurité qui seront utilisés ainsi que les algorithmes correspondants de chiffrement.

Step	Command	Purpose
1	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]	Define a transform set. There are complex rules defining which entries you can use for the transform arguments. These rules are explained in the command description for the crypto ipsec transform-set command, and Table 22 provides a list of allowed transform combinations. This command puts you into the crypto transform configuration mode.
2	initialization-vector size [4 | 8] .	(Optional) If you specified the esp-rfc1829 transform in the transform set, you can change the initialization vector size to be used with the esp-rfc1829 transform
3	mode [tunnel | transport]	(Optional) Change the mode associated with the transform set. The mode setting is only applicable to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode)
4	**exit**	Exit the crypto transform configuration mode.
5	clear crypto sa or clear crypto sa peer {*ip-address* | *peer-name*} or clear crypto sa map *map-name* or clear crypto sa entry *destination-address* protocol spi	This step clears existing IPSec security associations so that any changes to a transform set will take effect on subsequently established security associations. (Manually established SAs are reestablished immediately.) Note Using the clear crypto sa command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the peer, map, or entry keywords to clear out only a subset of the SA database. For more information, see the clear crypto sa command.

Combinaisons possibles de protocoles et algorithmes utilisables dans les *transform set*.

AH Transform **pick up to one**		ESP Encryption Transform **pick up to one**		ESP Authentication Transform **Pick up to one, only if you also selected the esp-des transform** (not esp-rfc1829)	
ah-md5-hmac	AH with the MD5 (HMAC variant) authentication algorithm	esp-des	ESP with the 56-bit DES encryption algorithm	esp-md5-hmac algorithm	ESP with the MD5 (HMAC variant) authentication
ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm	esp-rfc1829	older version of the ESP protocol (per RFC 1829); does not allow an accompanying ESP authentication transform	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
ah-rfc1828	older version of the AH protocol (per 1828)RFC				

3) Création des tunnels IPSec avec clefs manuelles (identiques des 2 côtés). La clé du *inbound* doit correspondre à la clé *outbound* routeur distant. Idem pour le *outband* qui doit être équivalent au *inbound* du routeur distant. Il est à noter aussi que le *spi* prend une valeur arbitraire entre 256 et 4294967295 (FFFFFFFF).

	Command	Purpose
1	**crypto map** *map-name seq-num* **ipsec-manual**	Specify the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
2	**match address** *access-list-id*	Name an IPSec access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. (The access list can specify only one permit entry when IKE is not used.)
3	**set peer** {*hostname* | *ip-address*}	Specify the remote IPSec peer. This is the peer to which IPSec protected traffic should be forwarded. (Only one peer can be specified when IKE is not used.)
4	**set transform-set** *transform-set-name*	Specify which transform set should be used. This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. (Only one transform set can be specified when IKE is not used.)
5	**set session-key inbound ah** *spi hex-key-data** and **set session-key outbound ah** *spi hex-key-data**	If the specified transform set includes the AH protocol, set the AH Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic. (This manually specifies the AH security association to be used with protected traffic.)

7	**set session-key inbound esp** *spi cipher hex-key-data** [authenticator *hex-key-data*] and **set session-key outbound** *esp spi cipher hex-key-data** [authenticator *hex-key-data*]	If the specified transform set includes the ESP protocol, set the ESP Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys. (This manually specifies the ESP security association to be used with protected traffic.)
8	exit	Exit crypto-map configuration mode and return to global configuration mode.

ATTENTION : la clé doit être au minimum de 20 octets.

4) Application à l'interface *Ethernet* sur laquelle IPSec est utilisé.

crypto map map-name *Apply a crypto map set to an interface.*

3.2. Travail à réaliser

Après avoir configuré le tunnel (aux deux extrémités) et avoir configuré un monitoring de port sur le switch :

- Faites un ***ping*** sur le réseau passant par le tunnel et un autre sur 30.30.30.30 (ne passant pas par le tunnel IPSec).
- Relevez les trames de ***ping*** dans les deux cas et comparez les.
- Notez les différences et l'intérêt du tunnel ainsi configuré.

4. Mise en place d'un tunnel IPSec en ESP

Modifiez la configuration précédente pour pouvoir fonctionner en ESP et répondez aux mêmes questions.

5. Utilisation du protocole IKE avec "pre-share key" pour la création des SA IPSec et des clefs de sessions IPSec

La *pre-share key* connue des deux *pairs* permet à IKE d'effectuer le transfert des clefs de manière sécurisée.

5.1. Opérations à effectuer pour mettre en oeuvre IKE

Les opérations suivantes sont soit à rajouter à la configuration précédente soit modifient la configuration précédente.

- Validation de ISAKMP (validé par défaut)
  crypto isakmp enable

- Création de la Politique IKE (à faire en plus de la configuration précédente).

Il s'agit de mettre en oeuvre une *politique* pour assurer la sécurité des échanges de clefs.
Opérations pour mettre en place la *politique* :

Step	Command	Purpose
1	**crypto isakmp policy** *priority*	Identify the policy to create. (Each policy is uniquely identified by the priority number you assign.) (This command puts you into the config-isakmp command mode.)
2	**encryption des**	Specify the encryption algorithm.
3	**hash {sha | md5}**	Specify the hash algorithm
4	**authentication {rsa-sig | rsa-encr | pre-share}**	Specify the authentication method
5	**group** {*1* | *2* | *5*}	Specify the Diffie-Hellman group identifier
6	**lifetime** *seconds*	Specifiy the security association's lifetime
7	**exit**	Exit the config-isakmp command mode

Les paramètres par défaut sont :

Parameter	Accepted Values	Keyword	Default Value
encryption algorithm	56-bit DES-CBC	des	56-bit DES-CBC
hash algorithm	SHA-1 (HMAC variant) MD5 (HMAC variant)	sha md5	SHA-1
authentication method	RSA signatures RSA encrypted nonces pre-shared keys	rsa-sig rsa-encr pre-share	RSA signatures
Diffie-Hellman group *identifier*	768-bit Diffie-Hellman 1024-bit Diffie-Hellman	1 2	768-bit Diffie-Hellman
security association's lifetime *1*	can specify any number of seconds		86400s (1 jour)

Vous utiliserez les paramètres par défaut sauf pour l'authentification que vous définirez en *pre-share*. Au minimum les commandes 1 et 4 suffisent.
Priority peut être défini de façon arbitraire car il n'y aura qu'une *politique*.

- Association d'une clef (*pre-share*) avec l'adresse de l'extrémité opposée du tunnel (à faire en plus de la configuration précédente) :

  crypto isakmp key keystring address peer-address

IMPORTANT : La *pre-share key* doit être la même aux 2 extrémités du tunnel !

- Suppression de la *map* de la configuration précédente

no crypto map map-name seq-num IPSec-manual

NB : La suppression de la map précédente demande au préalable de supprimer l'association de la map précédente avec l'interface sur laquelle elle était appliquée.

no crypto map map-name

- puis création d'une nouvelle *map*

crypto map map-name seq-num IPSec-isakmp

- Association de la nouvelle *map* avec l'interface sur laquelle elle est appliquée

crypto map map-name

5.2. Travail à réaliser

Après avoir configurez IKE et modifiez le paramétrage d'IPSec :

- Faites un ***ping*** sur le réseau passant par le tunnel
- Relevez le dialogue "IKE" et les trames de ***ping***
- Expliquez les trames du dialogue IKE.

6. Utilisation du protocole IKE avec des clefs publiques RSA pour l'échange des clefs de sessions IPSec

Toutes les clefs utilisées par IKE puis IPSec dérivent des clefs créées par l'algorithme RSA et l'échange des clefs publiques. Cet échange sera réalisé ici "manuellement" et de façon non sécurisée mais il pourrait être fait par l'utilisation de certificats.

6.1. Opérations à effectuer pour mettre en oeuvre IKE avec des clefs RSA

- Modifier la "politique" de la configuration précédente en spécifiant comme paramètre d'authentification "*rsa-sig*".
- Création des clefs et visualisation de la clef publique locale (à enregistrer manuellement sur le routeur distant). Vous utilisez anonymous ftp/HTTP/SSH/clé USB pour transmettre la clef publique d'une table à l'autre (Attention, le serveur ftp fait une résolution inverse, modifiez le fichier */etc/hosts* en conséquence).

Step	Command	Purpose
1	**ip domain name** *name*	Nom quelconque, p.ex celui du routeur
2	**crypto key generate rsa** [usage-keys]	Generate RSA keys
3	**show crypto key mypubkey rsa**	View the generated RSA public key (in EXEC mode)*.

*C'est la clef publique qu'il faut transmettre

- Enregistrement de la clef publique sur le routeur distant

Step	Command	Purpose
1	**crypto key pubkey-chain rsa**	Enter public key configuration mode
2	**addressed-key** *key-address*	Indicate which remote peer's RSA public key you are going to specify
3	**key-string**	Specify the remote peer's RSA public key. This is the key viewed by the remote peer's administrator previously when he generated his router's RSA keys.
4	**quit**	

6.2. Travail à réaliser

Après avoir modifié la configuration, refaites les mêmes tests et relevés que pour les manipulations précédentes.

NB : La transmission manuelle de la clef publique du routeur distant est fastidieuse. Il existe une méthode utilisant une Autorité de Certification pour assurer un transfert sécurisé de la clef publique. Cette méthode n'est pas utilisable ici.

7. Nettoyage

Pour remettre la salle en état.
Si vous avez utilisez le routeur :

Router# copy flash:/<la conf de base> startup-config

Eteindre le routeur.

Si vous avez utilisé le switch :

Switch# delete flash:/ vlan.dat (si vous avez fait des vlan)
Switch# erase startup-config

Eteindre le switch

Recâbler correctement. Les 4 câbles de couleur sur le switch et sur la carte 4 ports. Le câble "salle" reste libre. Les deux derniers câbles numérotés ou sans numéro dans les interfaces réseau intégrées des PC.

Sur le PC XP, repasser l'interface réseau en DHCP et éteindre la machine.
Sur le PC Linux, lancer le script ***/script/init_machine.sh*** puis le script ***/script/init_reseau.sh***.

Architecture PKI

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions de la partie 2

1. But du TP

Le but de ce TP est de mettre en place une architecture PKI basée sur *OpenSSL*. Pour ce TP vous allez désigner pour chaque banc une machine qui va contenir l'autorité de certification racine (autosigné), et une machine qui va contenir une autorité de certification fille, cette dernière sera responsable de la signature des certificats serveurs et utilisateurs. Ceci va permettre une délégation dans l'établissement des certificats.

N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.

2. OpenSSL

OpenSSL est un ensemble d'utilitaires de cryptographie implémentant SSL v2/v3 (Secure Socket Layer) et TLS v1 (Transport Layer Security) et les standard qu'ils requièrent (en particulier ceux d'une PKI). Pour le TP n'oubliez pas la commande ***man*** !!

2.1. Création de l'autorité racine

Sur la machine faisant office d'autorité racine

- Créez le jeu de clés privée/publique (RSA) protégés par un chiffrement symétrique de type DES3 de 1024 bits, dans un fichier (*ca-bancx.key)*
- Que signifie les ….+++++ dans la génération des clés ? Afficher le contenu de la clé privée
- Pourquoi le *PublicExponent* est il toujours à 65537 (0x10001) ?
- Auto signer le CA racine (X509) de votre banc, le DN choisi aura la notation suivante C=FR, O=Istic, OU=M2Pro, CN=bancx et le temps de validité de ce certificat est de 365 jours.
- Affichez le détail du certificat généré
- Rangez à présent la clé et le certificat dans le répertoire */home/admin/CA-bancx*

2.2. Création de l'autorité fille du banc

La création d'un certificat signé par une autorité de confiance se fait en deux étapes :

- Préparation du certificat (création de requête de signature CSR)
- Signature de la requête

2.2.1. Préparation du certificat

Sur la machine faisant office d'autorité fille

- Créez le jeu de clé privée/publique dans un fichier *ca-bancx-fille.key*
- Créez la requête de certificat pour la signature (le fichier en sortie doit avoir la forme *ca-bancx-fille.csr*), le DN choisi C=FR, O=Istic, OU=M2pro, CN=bancx-fille
- Transférez les deux fichiers sur la machine CA racine.

2.2.2. Signature de la requête par l'autorité racine

Pour effectuer une signature, *openssl* a besoin d'une configuration spécifique. Cette configuration est typiquement fournie dans un fichier de paramètre et non dans la ligne de commande, d'où la nécessité d'utiliser un script disponible dans le serveur *anubis* dans le répertoire *TP-pki*.

- Editez le ficher *sign.sh*, ce fichier est adapté du fichier fourni avec openssl. Il est à noter que :
  CA_DIR représente le dossier du CA (pour le TP chaque Certificat aura son propre répertoire)
  CA_CRT représente le Certificat signataire
  CA_KEY la clé privée du CA signataire

Sur la machine racine,

- Modifiez le fichier afin que CA_DIR aura pour valeur */home/admin/CA-bancx*, et *CA_CRT* et *CA_KEY* les fichiers du CA racine.
- Signez la requête CSR du CA fille en tapant ***sh sign.sh ca-bancx-fille.csr***
- Affichez les détails du certificat
- Peut-on voir si un certificat est auto signé ou non ?
- A quoi sert le numéro de série ?
- Comment peut-on modifier la date d'expiration des certificats ?
- Quelle est la signification du champ *X509v3 Basic Constraints* ?
- Transférez le certificat signé (.crt) et la clé utilisée (.key) vers la machine CA fille

2.3. Création de certificat utilisateur ou serveur

Sur la machine CA fille

- Créez une paire de clé pour l'utilisateur *admin* de la machine (admin.key)
- Générez la requête avec un DN à C=FR, O=Istic, OU=M2pro, CN=admin

Maintenant il faut signer le certificat d'admin, pour cela nous allons réutiliser le fichier *sign.sh.*

- Modifiez les variables *CA_DIR*, *CA_CRT* et *CA_KEY* suivant la configuration du CA fille.
- Faites en sorte que la ligne *basicConstraints = CA:FALSE*
- Signez la requête du certificat généré pour *admin*

2.4. Vérification d'un certificat

- Vérifiez le certificat de l'utilisateur *admin*
- Que constatez-vous ?
- Est-ce que la validité du certificat *bancx-fille* peut être vérifiée ?

Pour résoudre ce problème il faudrait disposer de la totalité des éléments de la chaine de certification, c.à.d. disposer de l'ensemble des certificats des CA dans un même dossier avec un jeu de liens symboliques pour permettre un accès rapide à ceux-ci via un hash value.

Sur la machine CA fille :

- Créez un répertoire CA-crt
- Copiez les certificats du CA racine (*bancx.crt*) et du CA fille (*bancx-fille.crt*) dans le répertoire CA-crt
- Placez vous dans ce répertoire, créez le fichier hash avec la commande ***openssl x509 –hash –in bancx.crt –noout***
- Créez un lien symbolique avec la valeur hash obtenu et le fichier *bancx.crt* ***(ln –s bancx.crt xxxx.0)*** où *xxxx* est la valeur du hash obtenu
- Vérifiez avec un ***ls –l***
- Faite pareil pour le fichier *bancx-fille.crt*
- Relancez la vérification du certificat.

Authentification 802.1X

Durée 4h

TP crée par Gilles Guette

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 3, 4 et 5.

1. Objectifs du TP

- Configurer un serveur Radius (implémentation OpenSource Freeradius), un commutateur (Cisco 2960) ainsi que les clients (Windows et Linux) pour autoriser l'accès à un réseau filaire par une méthode 802.1X.
- Tester plusieurs méthodes d'authentification 802.1X (login/password MD5 challenge, login/password PEAP, certificat EAP/TLS)

Votre travail consiste à :

- Mettre en place un réseau filaire contrôlé par une authentification 802.1X sur Freeradius.
- Analyser le fonctionnement de 802.1X au niveau trame.

1.1. Composition des postes de travail

La configuration du poste de travail est la suivante :

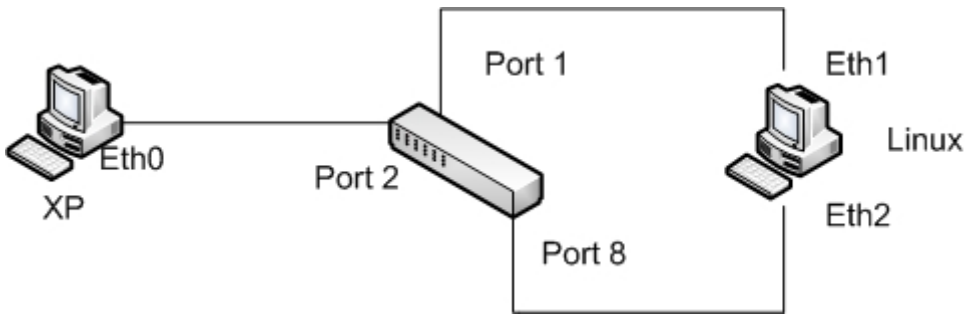
- Un serveur Radius : Démon Freeradius sur votre machine Linux i207m0X.
- Un NAS (Network Access Server) : Switch Cisco 2960 qui sera en fait client pour Freeradius.
- Un Supplicant : le PC XP.

2. Câblage du poste de travail

Effacez la configuration de votre switch en tapant :

Switch# erase startup-config
Switch# reload

Le schéma logique du réseau pour la suite du TP est le suivant :



2.1. Travail à effectuer

- Téléchargez et installez *Freeradius* sur votre poste : ***apt-get install freeradius***. Il démarre automatiquement, donc il faut arrêter le processus : ***killall freeradius***.
- Effectuez la configuration de votre poste de travail.
- Mettez les adresses suivantes sur les différentes machines ***ifconfig ethX @IP netmask le-masque*** (pour désactiver une interface ***ifconfig ethX down***. Sous windows, avec l'interface graphique ou commande ***ipconfig***.

Poste	Interface	Adresse IP	Ports du switch
PC Linux	eth0	Désactivée	-
PC Linux	eth1	10.X.1.1/24	1
PC Linux	eth2	10.X.2.1/24	8
PC Linux	eth3	Désactivée	-
PC Linux	eth4	Désactivée	-
PC XP	eth0	10.X.1.2/24	2

2.2. Configuration du switch

L'objectif de cette manipulation est de configurer le commutateur 2960 en tant que NAS, client du serveur *Freeradius*. Dans un premier temps il va s'agir de configurer deux VLANS : un VLAN "radius" où seront connectés le serveur *Freeradius* et le NAS et un VLAN "utilisateurs" où seront connectés les postes des utilisateurs du réseau d'entreprise (se référer au schéma logique). Ensuite il va falloir apprendre au commutateur à communiquer avec le serveur *Freeradius* et à contrôler ses ports afin d'utiliser l'authentification 802.1X.

- Créez les deux VLANS et assigner l'adresse 10.X.2.2/24 au vlan "radius" (ne pas oublier le *no shutdown* !!).
- Passez les ports concernés dans les VLANS correspondants (voir schéma) en *mode access*.

Configurez l'authentification Radius :

- Activez AAA : ***aaa new-model***
- Créez une méthode d'authentification 802.1X : ***aaa authentication dot1x default group radius***
- Autorisez l'authentification radius pour tous les services : ***aaa authorization network default group radius***
- Activez l'authentification 802.1X sur le switch : ***dot1x system-auth-control***
- Indiquez l'adresse du serveur radius : ***radius-server host 10.X.2.1 auth-port 1812 acct-port 1813 key votre-cle***

Configurez les ports "utilisateurs" :

- Activez le spanning-tree rapide : ***spanning-tree portfast default***
- Activez 802.1X sur le port : ***authentication port-control auto***, remarquez le changement de couleur de la diode du port concerné
- ***dot1x pae authenticator*** (commande nécessaire avec les version 12.2 des IOS, inutile avec une 12.1)

Vérifiez votre configuration ***show authentication***, ***show running-config***, ***show dot1x interface fastethernet***...) et enregistrez la configuration si elle convient : ***copy running-config startup-config***.

- Pourquoi assigner une adresse IP au Vlan "radius" ?
- Pour quelle raison n'est-ce pas nécessaire sur le VLAN "utilisateurs" ?

3. Configuration du serveur Freeradius

L'objectif de cette manipulation est de configurer *Freeradius* pour une authentification basique, c'est-‡-dire par couple login/password haché MD5.
Tous les fichiers de configuration de *Freeradius* se trouvent dans le répertoire */etc/freeradius*. *Freeradius* se comporte en tant que service et se manipule par conséquent par */etc/init.d/freeradius*. Cependant dans un but d'étude et de débogage vous le démarrerez tout au long de ce TP en ligne de commande : ***freeradius -X***, ce qui correspond au mode Debug verbeux et vous évitera d'ouvrir fréquemment les fichiers de log. En premier lieu vous configurerez le client ou NAS, puis un compte utilisateur qui permettra le login d'accès réseau.

3.1. Déclarer le commutateur comme client

Editez le fichier */etc/freeradius/clients.conf*, effacer ce qui s'y trouve et y ajouter une section :

```
client 10.X.2.2 {
shortname=cisco
secret=votre_cle
nastype=cisco
}
```

3.2. Créer un utilisateur de test

Editez le fichier */etc/freeradius/users* et y ajouter une ligne (impérativement en début de fichier)

```
votre_login_de_test Cleartext-Password := "votre_passwd_de_test"
```

3.3. Démarrage de *Freeradius*

Exécutez dans un terminal : ***freeradius -X***. Le service démarre alors en mode Debug verbeux. Pour le stopper il suffit de taper ***Ctrl+C*** .

- Repérez les ports réseau de fonctionnement de *Freeradius*, décrivez-les et donnez leurs fonctions.
- Observez la sortie texte lorsque le démon est lancé en debug et décrivez-en les différentes sections.

4. Authentification login/password EAP/MD5 et PEAP

L'objectif de cette partie est de configurer les postes utilisateurs grâce à des *supplicants* (*wpa_supplicant* pour Linux et supplicant intégré pour Windows XP) pour des authentifications simples.

4.1. Machine sous Linux, eth1 de i207m0X

Sur votre machine créez un fichier */etc/wpa_supplicant/monradius.conf* contenant ces lignes :

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
ap_scan=0
network={
key_mgmt=IEEE8021X
eap=MD5
identity="Votre_login_de_test"
password="Votre_passwd_de_test"
eapol_flags=0
}
```

- Lancez simultanément des captures de trame sur *eth1* et *eth2* de votre machine
- Lancez le client : ***wpa_supplicant -D wired -c /etc/wpa_supplicant/monradius.conf -i eth1***
- Vérifiez la connectivité au réseau.
- Commentez la sortie verbeuse du démon *Freeradius*.
- Dressez le double diagramme d'échange EAP Eth0|Cisco2960|Eth1 en vous aidant de vos captures.
- Peut-on parler ici d'authentification mutuelle ? Pourquoi ?

4.2. Machine sous Windows XP

Sous Windows XP, l'authentification 802.1X filaire nécessite le démarrage d'un service système qui fait office de *supplicant*. Le but de la manipulation est de démarrer ce service et de le configurer pour le type d'authentification souhaité.

4.3. Authentification MD5-Challenge

- Sous SP3, démarrez le service « Configuration Automatique de réseau câblé » (Clic droit sur le poste de Travail), puis « Gérer », puis « Services », clic droit sur le service souhaité puis « Démarrer ».
- Sous SP2, démarrez le service « Configuration Automatique de réseau sans fil » (Clic droit sur le poste de Travail), puis « Gérer », puis « Services » , clic droit sur le service souhaité puis «Démarrer ».
- Ouvrir les Connexions Réseau, puis un clic droit sur *eth0* et onglet « Authentification ». Cocher «activer l'authentification IEEE 802.1X », décocher l'option de mise en mémoire cache des informations utilisateur et choisir « MD5-challenge » comme méthode d'authentification.

Une fenêtre pop-up apparait et vous demande le couple login/mdp. Utilisez les paramètres de votre fichier *users* créé précédemment sur votre serveur *Freeradius*. Ne rien renseigner pour l'onglet « Domaine ».

- Vérifiez la connectivité au réseau.

4.4. Authentification PEAP

- Retournez sur l'onglet « Authentification » de l'interface *eth0*, choisir « EAP protégé (PEAP) » comme méthode d'authentification.
- Cliquez sur « Paramètres », décocher « Valider le certificat du serveur » et sélectionner « Mot de passe sécurisé » en tant que méthode d'authentification.
- Décochez « Activer la reconnexion rapide », puis cliquer sur « Configurer » et décocher le paramètre qui concerne l'utilisation du nom d'ouverture de session Windows.
- Commentez la sortie verbeuse du démon *Freeradius*.

Faites des captures de trame des deux cités du NAS.

- Comparez la méthode d'authentification avec EAP/MD5. La solution vous semble-t-elle plus sécurisée ? Peut-on parler ici d'authentification mutuelle ? ?

5. Authentification par certificat EAP/TLS

L'objectif de cette partie est d'obtenir une authentification Radius par EAP/TLS. Ce type d'authentification requiert l'utilisation de certificats de confiance. Pour le TP vous serez votre propre autorité de certification (ce qui est évidemment à proscrire pour une utilisation en production, mais très pratique pour une plate-forme de test). Vous utiliserez *Openssl* (à travers de quelques scripts pratiques inclus dans la distribution de *Freeradius*) pour créer et signer tous les certificats nécessaires à la manipulation.

5.1. Création des certificats

Attention : la syntaxe doit impérativement être respectée pour les fichiers de configuration des certificats.}

- En premier lieu, stoppez le service *Freeradius*. Allez ensuite dans le répertoire */etc/freeradius/certs/* et videz le ***rm -Rf***
- Il faut commencer par aller chercher les scripts : ***cp /usr/share/doc/freeradius/examples/certs/****
- Exécuter les commandes ***make clean*** et ***make destroycerts*** pour supprimer les certificats existants.

Il y a trois certificats à créer/signer, celui de l'autorité de certification, celui du serveur et celui du client. Nous créerons le certificat client dans un deuxième temps.

Tout d'abord la CA et le serveur : éditez l'un après l'autre les fichiers suivants : *ca.cnf* et *server.cnf*. Chacun d'eux contient les sections suivantes (à modifier) :

[req]
prompt = no
distinguished_name = certificate_authority
default_bits = 2048
input_password = whatever
output_password = whatever
x509_extensions = v3_ca

[certificate_authority]
countryName = FR
stateOrProvinceName = Bretagne
localityName = Rennes
organizationName = ISTIC
emailAddress = admin@example.com
commonName = "TPRADIUS"

Remplacez les champs *whatever* par les valeurs de votre choix en respectant les mêmes choix dans chacun des fichiers. Attention ! Si vous changez les *input/output_password*, il faudra également éditer le fichier */etc/freeradius/eap.conf* et remplacer « *private_key_password = whatever* » (section EAP/TLS) par le mot de passe choisi ici.
Relancez ensuite *Freeradius* en mode Debug : ***freeradius -X***; les certificats CA et Server se créent. Générez de l'entropie en secouant la souris. Arrêtez le service et passer en commentaire la ligne :
make_cert_command = "${certdir}/bootstrap" du fichier */etc/freeradius/eap.conf*.

Pour le certificat client, allez dans le répertoire */etc/freeradius/certs/* et modifiez les sections suivantes en fonction de votre configuration précédente:

[req]
prompt = no
distinguished_name = client
default_bits = 2048
input_password = whatever
output_password = whatever

[client]
countryName = FR
stateOrProvinceName = Bretagne
localityName = Rennes
organizationName = ISTIC
emailAddress = test@test.com
commonName = test

Tapez la commande ***make client***, le certificat client est créé.

Il nous reste encore à déclarer que l'utilisateur créé (ici ***test***) se connectera selon la méthode EAP. Pour ceci éditez le fichier */etc/freeradius/users* et remplacez la ligne ajoutée précédemment par ceci :
votre_nom_utilisateur Auth-Type := EAP

Lancez ensuite Freeradius : ***freeradius -X***

5.2. Importation des certificats

Rapatriez les fichiers */etc/freeradius/certs/ca.** et *client.** sur le poste Windows XP.
Il nous faut d'abord indiquer à Windows que vous êtes une autorité de certification digne de confiance. Pour cela, exécutez la commande « mmc » puis dans le menu « Fichier », choisir « Ajouter un composant enfichable », puis cliquez sur ajouter, choisir « Certificats » et validez.
Se placer sur le magasin de certificats « Autorités de certification racines de confiance », clic droit puis « toutes les tâches » et importer. Importez le fichier *ca.der* copié précédemment.
Ensuite il nous faut importer le certificat client. Pour ceci, naviguez dans le répertoire des certificats et double-cliquer sur *client.p12*. Suivez les options par défaut.

Important : Assurez-vous que les horloges des machines sont à la même heure.

5.3. Test de la connexion

Ouvrez les propriétés des connexions réseau. Dans l'onglet « Authentification » de l'interface eth0 choisissez « carte à puce ou autre certificat » comme méthode d'authentification. Dans les propriétés cocher « Utiliser un certificat sur cet ordinateur » puis « Utiliser la sélection de certificat simple », « Valider la connexion à ces serveurs », votre autorité de certification dans la liste et enfin « Utiliser un nom d'utilisateur différent pour la connexion ».

- Désactivez et réactivez l'interface, un pop-up vous présentant le certificat client attend votre validation.
- Testez la connectivité réseau
- Effectuez vos captures comme précédemment.
- Décrivez le mécanisme d'authentification.
- Cette méthode d'authentification est-elle mutualisée ?
- Quelles contraintes vous semblent gênantes dans cette méthode, particulièrement dans le cas ou vous auriez beaucoup d'utilisateurs à gérer ?
- Décrivez brièvement la fonction du fichier */etc/freeradius/eap.conf*
- Pourquoi les horloges de S_n et D_n doivent-elles être synchronisées ?

5.4. Pour aller plus loin

Vous avez testé ici quelques fonctionnalités de base d'une authentification Radius, mais beaucoup de fonctionnalités restent à "creuser". Ces mécanismes d'authentification sont fréquemment utilisés pour gérer les accès Wifi, le NAS est alors un point d'accès Wifi, il serait intéressant de tester cette utilisation.
De plus, en ce qui concerne le commutateur en tant que NAS, vous ne l'avez utilisé que dans sa configuration la plus simple, mais il est possible d'y apporter beaucoup d'améliorations (période de réauthentification, timeout d'authentification, blocage des ports en cas d'Èchec, assignation automatique de VLANS,…).
Dans le domaine de gestion des utilisateurs également, la manipulation est restée simpliste. Il est possible d'exploiter beaucoup plus de fonctionnalités de l'utilisation de LDAP et également de se servir d'Active Directory en tant qu'annuaire d'utilisateurs.
Aujourd'hui, la succession de Radius est assurée par *Diameter* qui propose plus de fonctionnalités natives notamment au niveau sécurité. Il permet également des échanges au dessus de TCP alors que Radius travaille sur UDP.

7. Nettoyage

Pour remettre la salle en état.
Si vous avez utilisez le routeur :

Router# copy flash:/<la conf de base> startup-config

Eteindre le routeur.

Si vous avez utilisé le switch :

Switch# delete flash:/ vlan.dat (si vous avez fait des vlan)
Switch# erase startup-config
Eteindre le switch

Recâbler correctement. Les 4 câbles de couleur sur le switch et sur la carte 4 ports. Le câble "salle" reste libre. Les deux derniers câbles numérotés ou sans numéro dans les interfaces réseau intégrées des PC.

Sur le PC XP, repasser l'interface réseau en DHCP et éteindre la machine.
Sur le PC Linux, lancer le script ***/script/init_machine.sh*** puis le script ***/script/init_reseau.sh***.

Firewall et Cache Web

Durée : 4h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions des parties 5,6,7,8 et 9

7. But du TP

Le but de ce TP et de mettre en œuvre une politique de sécurité grâce au firewall Linux *Netfilter* et le cache web basé sur le proxy *Squid*.

1. Introduction

Linux propose en standard de nombreuses fonctionnalités permettant de manipuler les paquets qui transitent sur le réseau. *Netfilter* est un composant intégré dans Linux depuis la version 2.4. Sa vocation consiste à filtrer ou modifier les paquets Internet reçus par l'ordinateur sur lequel il est activé, et à les retransmettre éventuellement vers d'autres machines sur le réseau. Il se voit principalement exploité pour partager des connexions Internet ou pour créer un firewall, mais peut également être utilisé à d'autres fins telles que la répartition de charge ou la protection d'un réseau contre les attaques de déni de service.

2. Configuration matérielle

Pour les besoins du TP, vous utiliserez les deux machines. La machine Windows qu'on notera <<Client>>, vous servira à tester la fiabilité de votre *firewall* depuis votre réseau interne en 172.16.0.0/24. La machine hôte, notée <<Netfilter>>, sera votre passerelle pour votre réseau interne vers le monde extérieur. Ce PC dispose donc de 2 interfaces réseaux actives, les interfaces ***eth0*** et ***eth1***. Il sera par conséquent chargé du routage entre les 2 sous-réseaux.
Sur la machine *Windows* :

- Donnez l'adresse 172.16.0.128 avec le masque 255.255.255.0
- Rajoutez une route par défaut, avec comme Gateway l'adresse de ***eth1*** (à savoir 172.16.0.1).
- Modifiez le DNS à interroger pour pointer vers le 148.60.4.1
- Vérifiez que vous arrivez à pinger l'adresse 172.16.0.1

3. Configuration du routage

Pour activer le routage, il faut modifier la valeur contenue par le fichier */proc/sys/net/ipv4/ip_forward*. En effet, le fichier *ip_forward* permet aux interfaces du système de réacheminer des paquets aux autres interfaces. Par défaut, ce fichier est paramétré sur 0 pour désactiver le réacheminement, mais si vous paramétrez ce fichier sur 1, le réacheminement sera activé.
Sur la machine <<Netfilter>>, tapez la commande suivante:

echo "1" > /proc/sys/net/ipv4/ip_forward

4. Première approche de *Netfilter* avec *Iptables*

On va utiliser la ligne de commande de *Netfilter* (*Iptables*) pour débuter. Ces premiers tests serviront à prendre en main cet outil, notamment pour vérifier l'état des différentes chaînes de règles et pour vérifier le trafic traversant (ou non !) le firewall.
Sur le poste <<Netfilter>>:

- Tapez ***iptables -L*** et commentez les résultats.
- Tapez ***iptables -L -v*** et relevez l'état de la chaine FORWARD

Sur le poste <<Client>>: faites un ***ping*** vers le poste Serveur

Sur le poste <<Netfilter>>: Tapez ***iptables -L -v*** et commentez le résultat de la chaine FORWARD.

- Comment est-il possible de résoudre notre problème de routage ?

5. Définition d'une politique de sécurité

Notre architecture est une simulation du monde réel : on dispose d'un firewall (poste <<Netfilter>>) qui protège le réseau interne (poste <<Client>>) du monde extérieur (machine anubis qu'on notera poste <<Serveur>>). On veut que:

- Le réseau interne puisse surfer sur internet librement, qu'il puisse utiliser la commande ***ping*** pour tester le réseau externe et qu'il puisse faire ce qu'il veut en interne.
- Le monde extérieur dispose du moins d'informations possibles sur le réseau interne.
- Seul le SSH soit disponible depuis l'extérieur pour la maintenance à distance.

Sur le poste Netfilter de l'un de vos voisins :

- Donnez le résultat de la commande ***nmap 148.60.12.X*** (l'adresse de votre machine Netfilter) et commentez.

Sur le poste <<Netfilter>>: dans un premier temps, on ferme tout.

- Donnez les commandes nécessaires (il est préférable d'utiliser l'option –P)

Sur le poste <<Client>>:

- Donnez le résultat de ***ping*** en direction des postes FW Netfilter.

Sur le poste <<Netfilter>>:

- Donnez le résultat du ***ping*** vers les trois interfaces (***lo***, ***eth0*** et ***eth1***).

Sur une des machines Netfilter voisines

- Donnez le résultat de la commande ***nmap 148.60.12.X*** et commentez

Sur le poste <<Netfilter>> :

- Donnez les commandes permettant d'autoriser tout le trafic local (***lo***, ***eth0*** et ***eth1***), testez les 3 interfaces ? utilisez la commande vue en TD pour mettre en place ce mécanisme.

Relevez la table des règles grâce à la commande ***iptables –L –v***,

- Quel est l'ordre d'application des règles ? quelle est la règle la plus prioritaire ?
- Quelle est la règle par défaut ?

A ce stade du TP, on ne peut toujours pas <<sortir>> depuis le réseau interne. A l'aide de vos (récentes) recherches sur *Netfilter*, on va partager l'accès au monde extérieur du poste <<Netfilter>> à tout le réseau interne.

Sur le poste FW <<Netfilter>> :

- Donnez le résultat de la commande ***iptables -t nat -L***.
- Donnez les commandes que vous utiliserez pour initialiser la table *nat*.
- Citez les deux méthodes possibles pour faire ce partage de connexion et donnez la ou les commandes nécessaires.

Sur le poste <<Client>>:

- Faites un ***ping*** vers le poste Serveur (148.60.12.25), un ***ping*** vers l'interface de sortie du poste <<Netfilter>> (148.60.12.X). Donnez et expliquez vos résultats.
- On ne peut toujours pas sortir vers l'extérieur, pourquoi ?
- Quelle est la chaine à modifier étant donné qu'on a un NAT active ?
- Testez, qu'est ce que vous constatez ?

6. Règles par protocole et la règle REJECT

Notre politique n'étant pas très sécurisée du point de vue réseau interne, on va maintenant tenter de l'améliorer. On va par exemple décider que les utilisateurs du réseau interne ne peuvent que surfer sur le réseau externe par leur navigateur. Le reste leur est interdit.

- Supprimez les anciennes chaines FORWARD
- Donnez les changements à appliquer à *Netfilter* ainsi que les nouvelles lignes qui autorisent la sortie (seulement pour http).
- Lancez un navigateur web, vous est il possible de vous connectez ? aidez vous de *Wireshark* sur la machine *Netfilter* pour trouver la cause.

A présent, on va utiliser le REJECT au lieu du DROP. On va bloquer toute demande de connexion *ftp* en provenance du réseau local vers le réseau extérieur.

- Donnez la différence entre les deux techniques ?

On utilisera l'option *--reject- with* pour associer une réponse à donner avec ce REJECT

- Quels sont les paramètres possibles avec cette option ?
- Ecrivez la règle *IPtables*.

En utilisant *Wireshark*

- Vérifiez la réponse de la machine Netfilter à une demande de connexion *ftp* provenant du client vers le serveur 148.60.12.25 ?

Si ça ne fonctionne pas, jetez un coup d'œil aux résultats de ***iptables –L –v.*** Si vous n'arrivez pas connaître la source d'erreur faite moi signe.

7. Connexion avec SSH

On veut maintenant pouvoir se connecter au firewall depuis l'extérieur du réseau afin de permettre les opérations de maintenance à distance. (si *sshd* n'est pas installé, ***aptitude install openssh-server***)

- Donnez les commandes pour laisser passer le trafic *SSH*.

Sur un poste voisin :

- Donnez le résultat de la commande *nmap* 148.60.12.X et testez la connexion en *SSH* (commande : *ssh 148.60.12.X*).

8. Cache Web

Dans cette partie vous allez mettre en place un cache web afin que seule la machine *Netfilter* puisse accéder au réseau public et que cette dernière puisse faire office de serveur mandataire pour les services *ftp*,*http*,*https*.

Sur la machine *Netfilter*

- Interdisez à la machine interne l'accès au réseau public (en modifiant Iptables).
- Installez le proxy *squid*.

Le proxy *squid* est basé sur un seul fichier de configuration *squid.conf* disponible dans */etc/squid/.* Un exemple de ce fichier de configuration est disponible sur le serveur web *anubis* dans le répertoire *TP_Squid*. La plupart des options par défaut du fichier *squid.conf* ne sont pas à changer, néanmoins certaines lignes doivent être adaptées à votre configuration (surtout ceux concernant votre réseau interne).

Squid doit être lancé avec un utilisateur différent de *root*, pour ce TP nous utiliserons l'utilisateur *proxy*.

Arrêtez *Squid* s'il est démarré. Avant de (re)démarrer *Squid*, il est nécessaire de créer les répertoires de swap avec la commande ***squid –z***. Aussi, faite en sorte d'attribuer les fichiers *'/var/spool/squid/swap.state'* et *'/var/spool/squid/swap.state.last-clean'* à l'utilisateur *proxy*.

- Démarrez Squid. Vérifiez qu'il n'y pas de problème de configuration en regardant le fichier de log *'/var/log/squid/cache.log'*
- Repérez dans le fichier de configuration les lignes « acl » (ligne 2419) et « http_allow_access ». Décrivez le processus mis en place pour configurer la connexion des clients au proxy ?
- Adaptez cette configuration à votre réseau ?
- Sur quel port, est-il en écoute ?

Faite en sorte que la machine client (navigateur web) de votre réseau puisse utiliser le proxy *Squid* pour les accès *ftp*, *http*, *https*.

En vous aidant de *Wireshark* si nécessaire

- Décrivez les étapes de connexions d'un client vers un serveur web à l'extérieur de votre réseau ?
- Quelles sont les informations contenues dans l'entête http ayant un lien avec le cache web ?

Sur votre proxy, faites en sorte d'interdire l'accès aux sites web contenant le mot SEXE. Pensez à utiliser les « acl ».

- Comment peut-on suivre les connexions passant par votre proxy *Squid*.
- En utilisant *IPtables*, faite en sorte que votre proxy devient transparent aux clients (plus besoin d'indiquer aux navigateurs d'utiliser un proxy web). Note, il faut aussi modifier le fichier *squid.conf* à fin de dire à Squid qu'il fonctionne en mode transparent. Assurez-vous que la ligne comme suit existe : *http_port :numero transparent*

4ème partie : VoIP

- Introduction PABX
- Asterisk: SIP et IAX2

Introduction au PABXIP : Asterisk

Durée : 2h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions de la partie 3

1. But du TP

Le but de ce TP est de mettre en œuvre une configuration de base d'un serveur PABX IP open source. Ce TP comprendra l'installation et la configuration du serveur *Asterisk* et du client SIP (*zoiper*). Pour ce TP, on utilisera les deux machines Debian et Windows, et vous travaillerez par banc. C'est à dire que chaque banc aura un seul serveur *Asterisk* (sur Debian), et deux clients VoIP.

N'oubliez pas avant de partir de traiter la partie 6 afin de remettre la salle en état.

2. Serveur PABXIP : *Astersik*

Asterisk est un logiciel vous permettant d'émuler complètement un système téléphonique (PBX) et ce de façon logicielle avec les téléphones internet (VoIP) ainsi que l'intégration possible avec des PBX matériels si vous désirez ajouter des fonctionnalités à votre existant. Pour plus d'informations visitez le site web www.asterisk.org.

Pour ce TP vous allez travailler en binôme pour la configuration du serveur, c'est-à-dire que chaque banc contiendra un serveur et deux clients. Vous utiliserez les machines Windows comme client (avec ***Zoiper***) et une machine Debian comme serveur Asterisk.

2.1. Installation et fichiers de configuration du serveur *Asterisk sur la machine Debian*

A présent vous allez procéder à l'installation du serveur *Asterisk*. Tapez ***apt-get install asterisk***.
Le serveur *Asterisk* est maintenant installé, et tous les fichiers de configuration se trouvent dans le répertoire */etc/asterisk*. Pour ce TP, on va s'intéresser à trois fichiers :

- *sip.conf* : contient la déclaration des clients SIP
- *extensions.conf* : c'est le fichier principal de la configuration du serveur, il contient l'ensemble des plans de communications (les actions à exécuter par *Asterisk*, par exemple l'établissement de communication entre clients)
- *voicemail.conf* : contient les informations pour envoyer les messages laissés dans la boite vocale par email.

3. Etablissement d'une communication vocale

On va commencer par déclarer les clients (*pcX* et *pcX+1* du banc) qui utiliseront le serveur *Asterisk*. On va créer un contexte de communication locale qu'on nommera *interne*, et déclarer les deux clients appartenant à ce contexte comme clients SIP.

- Editez le fichier *sip.conf* et rajoutez les lignes suivantes à la fin du fichier :

  [pcX]
  username=pcX; login client
  secret=pcXi207; mot de passé en claire
  type= friend; type du client
  host=dynamic; adresse ip dynamique
  context=interne; le contexte à qui appartient ce client
  callerid="pcx_m2pro" <011>; l'identifiant du client

  [pcX+1]
  username=pcX+1

secret=pcX+1i207
type= friend
host=dynamic
context=interne
callerid="pcX+1_m2pro" <012>

- Editez le fichier *extensions.conf* et rajoutez les lignes suivantes à la fin du fichier :

[interne]
exten=>011,1,Dial(SIP/pcX) ; lorsque le client tape 011 on appel pcX établir une connexion avec pcX
exten=>012,1,Dial(SIP/pcX+1) ; lorsque un client compose le 012 établir une connexion avec pcX+1

Il est à noter qu'après chaque modification du fichier, il faut relancer le serveur. Pour cela on va se connecter à *Asterisk* à travers un CLI en tapant ***asterisk –r***. A l'invite de commande tapez ***reload*** à chaque modification de fichier.
On pourra vérifier s'il a bien pris en compte les clients SIP. Pour cela vous disposez de la commande *sip*. Trouver la bonne commande pour afficher les noms de connexions SIP.
A présent on est prêt à établir une communication entre *pcX* et *pcX+1*. Vérifiez votre audio et micro au niveau du système.
A présent il faut configurer le client ***zoiper***. Cliquez sur l'icône en bas à droite. Dans le menu configurations, allez dans SIP account et ajoutez un compte SIP avec les informations définies pour le client, et dans le champ domaine mettre comme domain = i207m0X.istic.univ-rennes1.fr (m0X la machine hébergeant le serveur *Asterisk*).

- Lancez *Wireshark* et enregistrez votre client SIP au serveur *Asterisk* (à travers le menu).

Attendez que votre client soit connecté au serveur, et arrêtez la capture.

- Quels sont les paquets SIP que vous avez capturés ?
- Comment s'effectue l'authentification des clients ?

Maintenant vous pouvez tester en appelant de *pcX* vers *pcX+1* et inversement, n'oubliez pas de lancer *Ethereal* avant. Si vous avez des problèmes penser à vérifier vos paramètres son sur Linux.

- Retracez l'échange SIP entre les deux machines ?
- Quelles sont les adresses SIP utilisées ?
- Quel est le contenu d'un paquet SIP ?
- Après l'établissement de la communication, quels sont les paquets échangés entre les clients ?
- Donnez le contenu de ses paquets ? pourquoi contiennent-ils des numéros de séquences ?
- Quel est le type du codeur audio utilisé ?

Un des deux participants décroche

- Quels sont les messages SIP échangés ?

4. Configuration de Voicemail

Maintenant on va permettre à notre contexte *interne* l'utilisation de *Voicemail* lorsque le client *SIP* appelé ne décroche pas ou il est n'est pas connecté.

- Editez le fichier *voicemail.conf* et rajouter les lignes suivantes :

[interne]
011 =>5555,pcX,pcX,root@i207m0y.ifsic.univ-rennes1.fr | attach=yes
012=>5555,pcX+1,pcX+1,usertp@i207m0y.ifisic.univ-rennes1.fr | attach=yes

Il est à noter que *Y* est le numéro de machine hébergeant le serveur du banc (X ou X+1). De plus on va utiliser des comptes locaux à la machine serveur étant donné qu'il n'y a pas de serveur SMTP actif.

Il reste à modifier le fichier *extensions.conf* pour que le serveur d'utilise *Voicemail* avec les comptes *pcX* et *pcX+1*

- Editez le fichier *extensions.conf* et modifiez le contexte interne comme suit :

  [interne]
  exten=>011,1,Dial(SIP/pcX,10) ; 10 seconde avant le timeout
  exten=>011,2,VoiceMail(u011@interne)
  exten=>011,3,VoiceMail(u011@interne)
  exten=>012,1,Dial(SIP/pcX+1,10)
  exten=>012,2,VoiceMail(u012@interne)
  exten=>012,3,VoiceMail(u012@interne)

- Testez

5. Plus loin

D'autres fonctionnalités dans *Asterisk* permettent entre autres de jouer du son, ou décrocher et faire attendre l'interlocuteur. Pour cette partie on va créer un IVR (Interactive Virtual Response) ou un menu virtuel. Grâce à ***apt-get***, installez les sons nécessaires pour ce IVR ***apt-get install asterisk-sounds-extra***

- Editez le fichier *extension.conf*
- Dans le contexte *intene* rajoutez les lignes suivantes :
  exten=>800,1,Answer()
  exten=>800,n,Goto(menu,s,1)
- Rajoutez un autre contexte juste à la suite du contexte interne en rajoutant les lignes suivantes:
  [menu]
  exten=>s,1,Background(enter-ext-of-person)
  exten=>s,2,Wait(2)
  exten=>1,1,Background(digits/1)
  exten=>1,2,Goto(menu,s,1)
  exten=>2,1,Background(digits/2)
  exten=>2,2,Goto(menu,s,1)
- Dans quel format les tonalités DTMF sont-elles transportées ?
- Quel est le payload utilisé ?
- Sur le client zoiper et Asterisk (fichier sip.conf), faite en sorte que ces tonalités soient transportées par SIP ?

6. Remise à zéro de la machine

Redémarrez votre machine et à l'écran *Rembo*, cliquez sur *Reinstallation du système*.

Asterisk: SIP et IAX2

Durée : 4h.

A la fin de ce TP vous devez me rendre un rapport répondant aux questions de la partie 3 et 4

1. But du TP

Durant les cours de TFM, vous avez découvert les protocoles de la VoIP : SIP, H.323, MGCP. Néanmoins, il existe un autre protocole propriétaire (en cours de standardisation) qui est IAX2, développé par les concepteurs d'*Asterisk*. Il est particulièrement utilisé pour l'interconnexion des PABXIP *Asterisk*. Pour ce TP, on vous propose de découvrir ce protocole et de le comparer à SIP, en mettant en place une interconnexion SIP/IAX2.

2. Serveur PABX-IP : *Astersik*

Pour ce TP vous allez travailler en binôme pour la configuration du serveur, c'est-à-dire que chaque banc contiendra un serveur et deux clients.

3. SIP (Session Initiation Protocol)

3.1. Connexion simple

Inspirez vous du TP précédent pour installer et configurer votre serveur de banc *Asterisk* et ajouter deux clients SIP ayant comme identificateur (numéro de téléphone) NB00 et NB01.
Créez le contexte *interne* pour qu'une communication vocale puisse être établie entre les deux clients.
A présent on est prêt à établir une communication entre *pcX (NB00)* et *pcX+1(NB01)*. Mais avant, il faudrait configurer les clients SIP sur *pcX* et *pcX+1*. Sur la machine Windows configurez ***Zoiper*** comme pour le TP précédent.

- Lancez *Wireshark*

Maintenant vous pouvez tester en appelant de *pcX* vers *pcX+1* et inversement.

- Quels sont les messages SIP contenant la description de session SDP ?
- Donnez la signification des champs *o*, *c*, *m* de la description SDP capturée ?
- En se basant sur les paquets (capturés) SIP, comment peut-on connaître les ports qui seront ouvert pour le transfert des medias ?
- Quelle est l'utilité des messages STUN envoyés par votre client SIP ? y'a-t-il des réponses à ces requêtes ?
- Est-ce qu'on peut lire le contenu des réponses ? selon vous pourquoi ?

3.2. Interconnexion des PABX-IP de la salle

Pour permettre l'établissement des communications vocales entre l'ensemble des PC de la salle, on va interconnecter les PABX-IP de la salle entre eux. Cette interconnexion sera réalisée grâce à SIP. En premier lieu, vous allez interconnecter le PABX-IP de votre banc avec le banc voisin, et par la suite l'interconnecter à l'ensemble de la salle.
On notera le PABX-IP de votre banc par *i207m0Y*, et celui avec lequel l'interconnexion va se faire par *i207m0Z*.
Sur le PABX-IP de votre banc éditez le fichier *sip.conf.*

- Rajoutez dans la partie générale *([general]*), la ligne suivante : *register=> i207m0Y :welcome@148.60.12.Z/i207m0Z*. Cette ligne indiquera à votre PABX-IP de s'enregistrer auprès de votre voisin.
- Rajoutez le bloc suivant, qui déclarera le PABX-IP voisin :
  [i207m0Z]

type=friend
secret=welcome
context=interne
host=dynamic

Editez le fichier *extensions.conf*, et dans le contexte interne rajoutez la ligne suivante: *exten=>_NBzXX,1, Dial(SIP/i207m0Z/${EXTEN}). NBz* représente le numéro de banc du PABX-IP *i207m0Z*.

- Donnez une explication à cette ligne
- Vérifiez grâce à la commande *sip show registery* que votre PABX-IP s'est bien enregistré auprès de l'autre PABX-IP

Etablissez une communication entre le PC client de votre banc avec le client du banc voisin.

- Retracez l'échange SIP ? quelle est la différence par rapport à la partie 3.1.
- Quel est le champ du message SIP qui permet de connaître le chemin traversé ?
- Selon ces traces, *Asterisk* fait office d'un proxy SIP stateful ou stateless ?

Généralisez la configuration afin que chaque PC de la salle soit joignable.

A la fin de cette manipulation commentez les lignes d'enregistrements (general) de votre PABX-IP auprès des autres PABX-IP.

4. IAX2 (InterAsterisk eXchange)

IAX2 est le protocole de VoIP développé avec *Asterisk*. Il permet de mettre en place la signalisation nécessaire pour établir une communication VoIP en toute simplicité. En premier lieu, il faut rajouter un compte IAX sur le client VoIP *zoiper*.

4.1. Connexion simple

La configuration IAX2 est pratiquement identique à SIP sur le serveur. Il faut déclarer les clients dans le fichier *iax.conf*. Voici un exemple de la déclaration d'un client IAX :

[pcX]
type=friend
secret=pcXi207
context=interne_iax
host=dynamic
auth=md5,plaintext,rsa

- Ajoutez les deux clients de votre banc.

Pour IAX, la mise en place de la communication est identique à SIP, il faut mettre à jour le fichier *extensions.conf*. Voici un exemple de redirection d'appels dans le contexte *interne_iax* : *exten=>NB00,1,Dial(IAX2/pcX)*. Ici il faudrait que vous commentiez l'ancien contexte SIP.

- Lancez *Wireshark*
- Testez un appel IAX entre les clients de votre banc
- Retracez les échanges entre les deux terminaux pour l'établissement de l'appel ? .
- Quel est le type des paquets transmis après la signalisation ? utilisent-ils RTP ? .
- Quels sont les ports utilisés par les messages IAX2 (signalisation et médias) ? Qu'est-ce que vous pouvez en conclure ?

Un de vous arrête la communication.

- Quel est le paquet annonçant cette interruption ?

4.2. Interconnexion des PABX-IP avec IAX2

Pour cette dernière partie en va essayer d'interconnecter les PABX-IP de la salle avec IAX2, toute en utilisant des clients SIP. Dans le fichier *extensions.conf* mettez en commentaire l'ensemble des extensions que vous avez rajoutées, mais aussi commentez la ligne qui permet à votre serveur *Asterisk* de se connecter aux autres serveurs de la salle.

En premier lieu on va modifier le fichier *iax.conf*, afin de permettre à votre PABX-IP de s'enregistrer auprès de votre voisin. Editez le fichier, et rajoutez la ligne suivante dans la partie [general] :

register=> i207m0Y :welcome@148.60.12.Z.

- Rajoutez le bloc définissant votre PABX-IP voisin comme suit :

[i207m0Z]
type=friend
secret=welcome
context=mZ_incoming
host=dynamic
trunk=yes

Enfin, on va modifier le fichier *extensions.conf* afin d'organiser plus finement le dialplan :

[interconnexion]; dans le contexte [default] rajouter la ligne include => interconnexion
include=>interne
include=>remote
include=>phones

[remote]
exten=>_NBzXX,1,Dial(IAX2/i207m0Z/${EXTEN})

[interne]
exten=>NB00,1,Dial(SIP/pcX)
exten=>NB01,1,Dial(SIP/pcX+1)

[phones];dans le fichier sip.conf modifiez le contexte interne en phones
include=>interne
include=>remote

[mZ_incoming]
include=>interne

- Donnez une explication à ce plan de numérotation.
- Sur le serveur PABX-IP, lancez une capture avec *Wireshark*

Testez une communication entre le client de votre banc et celui du banc voisin.

- Quels sont les messages IAX2 qui correspondent à INVITE et BYE de SIP ?
- Quels sont les messages IAX2 qui contiennent la description de la session média ?
- Les flux média passent-t-ils par le PABX-IP ?
- Avec un schéma, retracez l'échange des messages, de la signalisation jusqu'à la mise en place de la communication.

Généralisez la configuration afin que chaque client VoIP de la salle soit joignable.