Sécurité

TP 1 : Découverte du Réseau Local

À rédiger : Un texte de *maximum une page* en format `.txt` avec les commandes que vous avez lancées et les réponses aux questions *sans capture d'écran* à `dp.cnam@gmail.com`. La réponse à une question/commande ne doit pas dépasser une ligne.

1 Découvrir les autres machines d'un réseau local

Démarrer la machine en mode TP Linux. D'abord, lancer `firefox` et configurer le proxy en utilisant *l'adresse de configuration automatique* :`http://iut-rt/proxy.pac`. Ouvrir un terminal et aller en mode root/administrateur, i.e. tapez la commande :

```
sudo -s
```

Exercices :

1. 1 Noter l'adresse IP et MAC de votre machine (*ifconfig*). Exécuter un `ping` sur la machine `edison` (taper `ping edison`) pour récupérer l'adresse IP de cette machine. Comment trouver l'adresse IP de `www.google.fr` ? Est-il possible de trouver l'adresse MAC de `www.google.fr` ? (Re-)familiarisez vous aux commandes `ifconfig` et `ping` !

   

   Rappel : pour **rechercher** une option d'une commande (ex., l'option *-b* de la commande *ping*) dans le manuel Linux, utiliser "`man [NOM_COMMANDE]`" (ex., `man arp`) et taper "/", suivi de la chaine de caractères recherchée (ex., *-n*) et d'*Entrer*. Ensuite, taper "n" pour rechercher l'apparition suivante du mot. Pour quitter le manuel, utiliser "q".

1. 2 Chercher dans le manuel l'objectif de la commande `traceroute`. (à installer avec `apt-get install inetutils-traceroute` si besoin). Peut-on utiliser cette commande pour avoir une idée de la position géographique des serveurs traversés pour joindre `www.google.fr` ?
1. 3 Peut-on utiliser les commandes `nslookup`, `dig` et `less` pour avoir l'adresse IP de `www.google.fr` ?
1. 4 Quel est le résultat de la commande `ping -b 172.31.16.0` ?
1. 5 La commande *nmap* ("Network Mapper") est un important outil d'exploration réseau et scanneur de ports. *Nmap* permet par exemple de déterminer les systèmes d'exploitation des autres machines, les types de dispositifs de filtrage/pare-feux utilisés par les serveurs. Pour ces raisons, *nmap* est souvent utilisé au début des attaques pour chercher des cibles ou des failles de sécurité, e.g. voir la première commande lancée par l'attaque qui a compromis 300 000 Neufbox (`http://bluetouff.com/2010/09/24/deep-packet-inspection-analyse-de-la-compromission-d-un-fai/`).
   - Quel est l'objectif de la commande `nmap -sS iut-rt` ?
   - Taper `nmap` dans le terminal et regarder le résultat (la sortie de la commande). Essayer de comprendre les trois exemples affichés à la fin de la sortie. Modifier le deuxième exemple pour trouver *toutes la machines actives sur votre réseau* `172.31.16.0/20`
   - trouver tous les ports ouverts sur la machine `iut-rt`, ainsi que sur `edison` ; notez dans le compte-rendu toutes les informations sur *edison* (e.g. le système d'exploitation)
   - trouver les ports ouverts sur la machine de votre binôme.
1. 6 Comment trouver votre **passerelle par défaut** ? Donner dans le compte-rendu le maximum de solutions (penser aux commandes `netstat` (option r), `route` (option n) et à la commande `ip route show`).
1. 7 Pour trouver l'adresse MAC d'une adresse IP *A.B.C.D*, il faut exécuter `ping` *A.B.C.D* et ensuite chercher l'adresse mac dans le résultat du `arp -n`. Donner l'adresse MAC de la passerelle.

2 Connexions TCP via `netcat`

La commande *netcat* est le principal outil de la console Linux pour manipuler des connexions TCP et UDP. Il est possible de lancer *netcat* en deux modes :

- mode serveur : `netcat -l [NUMEROPORT]`. Cela ouvre une socket serveur qui permet à un seul client de se connecter.
- mode client : `netcat [IPSERVEUR] [NUMEROPORT]`. Cela permet de se connecter comme client au port `[NUMEROPORT]` de la machine `[IPSERVEUR]`. Le client *netcat* est assez similaire à la commande *telnet* (voir la 2ème commande de l'attaque ci-dessus).

De plus, *netcat* permet de chercher des ports ouverts sur une autre machine (port scanner). En fait, son manuel affirme que *netcat* est utile pour "anything under the sun involving TCP or UDP" (voir aussi sa page Wikipedia).

1. Utiliser `netcat` (en mode serveur, utiliser l'option "-l") pour construire une socket serveur à l'écoute sur le port 3128. Ensuite, utiliser `firefox` pour se connecter à `http://localhost:3128`. Observer la requête *http* dans la console `netcat` ;
2. Utiliser `netcat` (mode client) pour se connecter à la machine *iut-rt* sur le port 80 et envoyer une requête `http` (version 1.0). Par exemple, *GET / HTTP/1.0*, suivi d'une ligne vide.
3. trouver les ports ouverts sur la machine *iut-rt* avec `netcat` (indication : ouvrir le manuel "man netcat" et rechercher les options "-z" et "-v")
4. Utiliser la commande netcat pour construire un système de **messagerie instantanée**. Travailler en binôme : un utilisateur lance netcat en mode serveur/écouteur (ex., `netcat -l 3128`, recherche "-l" dans le manuel `netcat`) et l'autre utilisateur se connecte comme client sur le port 3128).
5. Utiliser `netcat` afin de pouvoir envoyer une page `html` aux clients qui se connectent sur le port 3128. Construire un fichier *page.html* qui sera utilisé comme fichier (flux) d'entrée à la commande `netcat` via `netcat -l ... < page.html`. La première ligne de ce fichier est : "HTTP/1.1 200 OK". Cette ligne est suivie d'une ligne vide et du reste des balises `html`. Exemple de fichier `page.html` :

```
HTTP/1.1 200 OK

<html><body><h1>aaaaaaaaaaa</h1></body></html>
```

3 Notions de base `iptables` : manipulations des connexions locales

Familiariser vous avec les notions de filtrage `iptables` présentées en cours (voir les diapos). Ensuite, parcourir le document `iptables` disponible à `http://iut-rt/~porumbel/secu/docsIptables.pdf` pour vous familiariser avec la commande.

3.1 Informations de base `iptables`

Tout paquet capté par la carte réseau est traité par une ou plusieurs *chaînes iptables*. Une chaine *iptables* contient une série de *règles* qui sont appliquées sur le paquet. En principe, une règle a la forme : *si condition alors action*. Une exemple de condition est "paquet en provenance de 10.0.0.7". Un exemple d'action est "ignorer le paquet". Voici un exemple de règle : "`Si (paquet envoyé par 10.0.0.7), alors (ignorer)`". La commande iptables serait :

```
iptables -A INPUT -s 10.0.0.7 -j DROP
```

Dans la commande ci-dessus, l'expression "`-A INPUT`" indique le fait qu'on ajoute une règle dans la chaîne INPUT ("-A" signifie "append rule" ou "ajouter règle"). La chaîne INPUT est une chaîne prédéfinie appliquée sur les paquets destinés à la machine locale. La règle ajoutée serait : si la source du paquet est `10.0.0.7`, alors appliquer la *cible* DROP. Les chaînes de règles sont stockées dans des *tables*. Il y a deux tables importantes : *filter* (la table utilisée par défaut, par exemple dans la commande ci-dessus) et la table *nat*. Dans ce TP, nous travaillerons plus souvent avec la table *nat* (Network Address Translation) est avec ses chaînes : OUTPUT, PREROUTING et POSTROUTING.

PREROUTING (table nat) Les paquets vont entrer dans cette chaîne avant qu'une décision de routage ne soit prise. On peut changer la destination du paquet avec la cible $DNAT$.

INPUT (table filter) Le paquet va être livré à la machine locale.

FORWARD (table filter) Tous les paquets qui ont été acheminés et ne sont pas livrés sur place parcourent la chaîne.

OUTPUT (table nat et filter) Les paquets envoyés à partir de la machine elle-même se rendront à ces chaînes : il y a une chaîne dans la table nat et un autre dans la table filter. La chaîne de la table filter permet de modifier la source du paquet avec la cible $DNAT$.

POSTROUTING (table nat) La décision de routage a été prise. Les paquets entrent dans cette chaîne, juste avant qu'ils ne soient transmis vers le matériel. On peut modifier la source du paquet avec la cible $SNAT$.

Les chaînes prédéfinies ont une politique par défaut (par exemple ACCEPT), qui est appliquée au paquet s'il arrive à la fin de la chaîne. Le flux des paquets à travers les chaînes est présenté dans la Figure 1 ci-dessous.

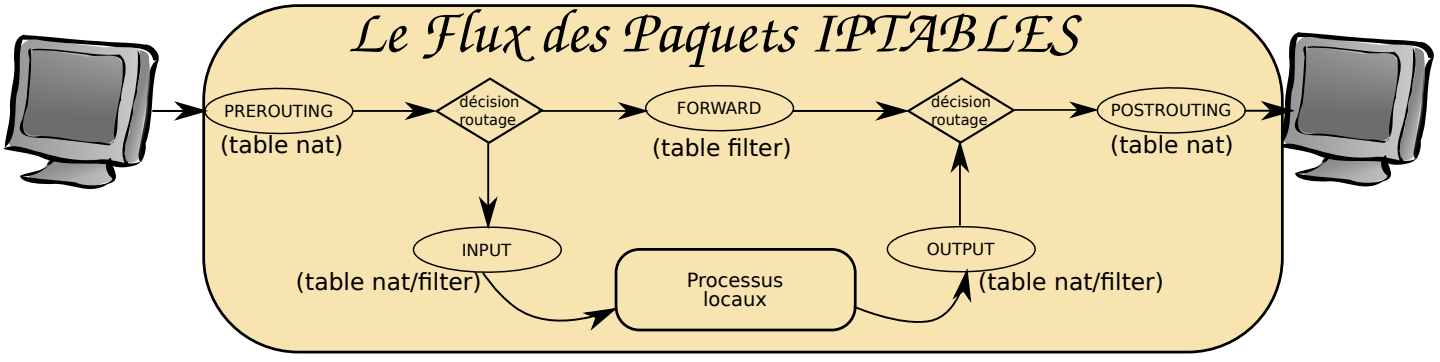

FIGURE 1 – Schéma simplifié du flux des paquets. Tout paquet reçu par une machine (routeur !) arrive d'abord dans la chaîne PREROUTING. Après des décision de routage, la dernière chaîne de traitement est POSTROUTING. Ces deux chaînes appartiennent à la table `nat`

Exercice 3.1 : Indiquer dans le compte-rendu l'objectif de chaque commande ci dessous.

Pour le manuel officiel, taper `man iptables` : pour chercher un mot clé dans ce manuel, tapez "/" suivi par le mot clé et "Entrer". La prochaine apparition du mot clé est disponible avec la touche "n".

```
1  iptables -F
2  iptables -t nat -L
3  iptables -t filter -L
4  iptables -P INPUT ACCEPT
5  iptables -P OUTPUT ACCEPT
6  iptables -P FORWARD ACCEPT
7  iptables -A INPUT -p udp -j REJECT
8  iptables -A INPUT -p udp -s votre_serveur_dns - -sport 53 -j ACCEPT
9  iptables -A INPUT -s 172.34.5.8 -j DROP
10 iptables -A INPUT -p TCP -s iut-rt -j DROP
11 iptables -I INPUT -p TCP -s iut-rt  - -tcp-flags ACK ACK -j ACCEPT
12 iptables -t nat -I OUTPUT -d 193.49.62.52 -j DNAT  - -to-destination 127.0.0.1
```

Exercice 3.2 : Travailler en binôme sur deux machines notées A et B. Donner les commandes *iptables* que A pourrait exécuter afin d'interdire à B de se connecter à la machine A. Attention, il *ne faut pas couper complètement la connexion* entre A et B. Par exemple, A devrait toujours pouvoir se connecter via `netcat` à un serveur ouvert par B. Cependant, B ne devrait pas pouvoir se connecter au serveur qui tourne sur A. **Indications :** analyser les exemples de filtrages présentés en cours.

4 Ssh

4.1 TP Linux : Ssh sans mot de passe

L'objectif de cette section est de se connecter à une machine distante (`iut-rt`) avec `ssh` sans taper le mot de passe. Vous pouvez utiliser votre login d'étudiant ou bien se servir des comptes `guest01`, `guest02`, etc. L'idée de base est de placer une **clé publique** RSA sur le compte `iut-rt` et la clé privé RSA sur la machine locale. [1]

On travaille **d'abord sur la session à distance :**
- ouvrir la session via `ssh [LOGINETUDIANT]@iut-rt`
- taper `ssh-keygen` et accepter les noms de fichiers proposés par défaut : `.ssh/id_rsa.pub` pour la clé publique et `.ssh/id_rsa` pour la clé privée. Il ne faut pas donner de passephrase.
- placez vous dans le dossier `.ssh/`
- copier le fichier `id_rsa.pub` (la **clé publique**) dans le fichier `authorized_keys`

On revient sur une **console de la machine locale**.
- Démarrer un nouveau terminal (pour travailler sur la machine locale sans activer le mode `sudo`)
- Placez vous dans le dossier `$HOME/.ssh/`
- La **clé privée** `id_rsa` générée sur la session distante doit être copiée sur la **machine locale** `scp [LOGINETUDIANT]@iut-rt:.ssh/id_rsa ./`
- Taper `ssh -v [LOGINETUDIANT]@iut-rt` et vérifier si vous pouvez se connecter sans taper le mot de passe.

Question 1 Est-ce qu'il y a des risques si vous oubliez la **clé privée** sur la machine locale ?

1. voir aussi http ://www.generation-linux.fr/index.php ?post/2008/02/26/79-se-connecter-en-ssh-sans-demande-de-mot-de-passe

Question 2 Est-ce qu'il y a des risques si vous oubliez la **clé publique** sur la machine locale ?

À faire valider par l'enseignant : montrez si vous pouvez vous connecter sans taper le mot de passe !

Si la connexion sans mot de passe fonctionne, garder votre clé privée sur une clé USB pour l'utiliser plus tard dans la Section **??**, ou bien sur d'autres machines.

5 TP WINDOWS : se connecter sans mot de passe

Le client *ssh* Windows s'appelle *putty*. Télécharger ce programme et trouver une solution pour se connecter à votre compte *iut-rt* avec *putty* sans taper le mot de passe. **Indications :** Essayer d'abord d'utiliser la clé privée déjà générée en Linux. Si besoin, vous pouvez générer une autre clé publique et privée avec `putty-key-gen` (à télécharger). Enregistrez la clé privée sur le disque local. Concernant la clé publique, utiliser la zone de texte "public key for pasting into OpenSSH authorized_keys". Le texte de cette zone doit être ajouté au fichier `authorized_keys` de votre compte `iut-rt`. Rappel : vous pouvez utiliser `pscp` ou `Winscp` pour transférer des fichiers sur votre compte `iut-rt`.

À faire valider par l'enseignant : montrez si vous pouvez vous connecter sans taper le mot de passe !