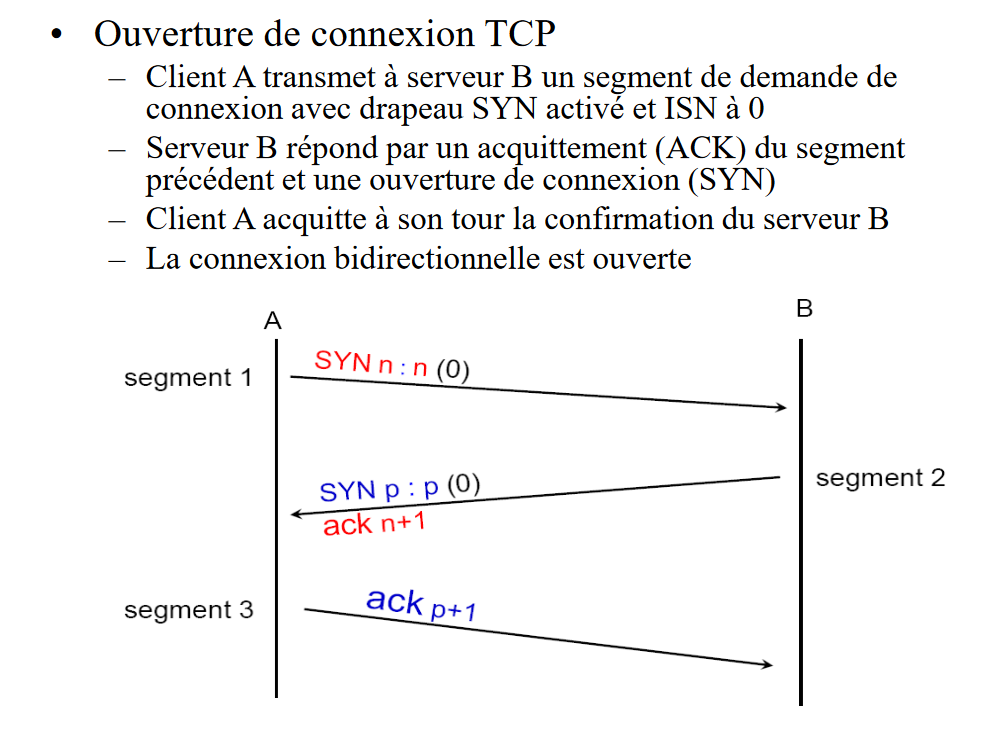
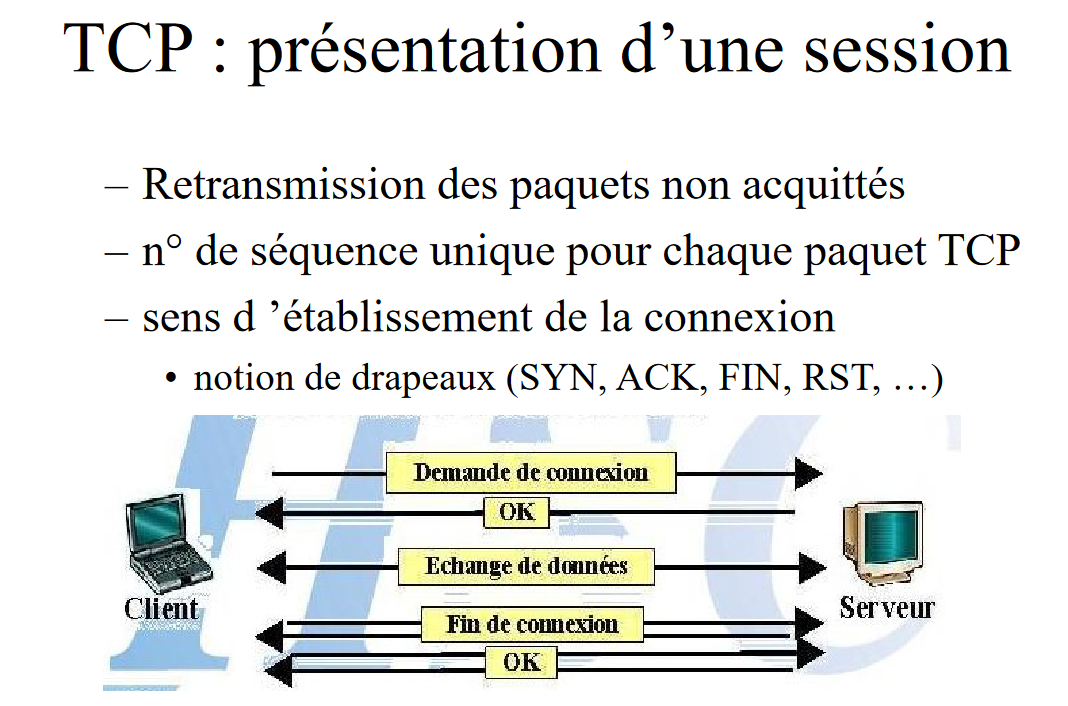
Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

REMARQUE : On préféra utiliser DROP au lieu de REJECT, pour rester discret mais les deux font la même chose.

Exemple de commandes :

***iptables -A FORWARD -s 192.168.10.97 -j DROP***

regle **d’interdiction de transmettre** le packet (pour un routeur) pour tout traffic venant de 192.168.10.97

L’argument **a** signifie qu’on met cette règle tout en bas de la table (car elle se lit de haut en bas)

***iptables -I FORWARD -s 192.168.20.98 -d 192.168.30.115 -j ACCEPT***

regle d’autorisation de transmettre le packet (pour un routeur) pour tout traffic venant de 192.168.20.98 et allant vers 192.168.30.115.

l’argument **i** signifie qu’on met cette règle tout en haut de la table (car elle se lit de haut en bas)

\*On suppose donc que le reste du traffic est interdit\*

***iptables -D FORWARD -s 192.168.20.98 -d 192.168.30.115 -j ACCEPT***

On supprime la règle d’autorisation juste au-dessus de la table (si elle existe)

-D pour delete

Une image contenant texte

Description générée automatiquement  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Une image contenant texte

Description générée automatiquement  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Une image contenant texte

Description générée automatiquement

Autres arguments :

-p : Quel protocole on utilise TCP/UDP (default les deux)

-i : l’interface si on veut seulement gérer une seule interface (default toutes)

--dport : port(s) de source (default tous)

--sport : port(s) de sortie (default tous)

-s : l’IP source du packet (default 0/0 , donc tout le monde)

-d : L’IP de destination du packet (default 0/0 , donc tout le monde)

NE PAS OUBLIER : Si on bloque toute communication, puis on fait une exception pour deux machines, il faut excepter les deux sens :

Donc deux commandes par exceptions

***iptables -I FORWARD -s 192.168.20.98 -d 192.168.30.115 -j ACCEPT***

***iptables -I FORWARD -s 192.168.30.115 -d 192.168.20.98 -j ACCEPT***

Les filtre :

(INPUT-OUTPUT) : bloquera entièrement toute communication (plus utilisé sur les machines [bloque tout, même le ping, la destination n’existe plus logiquement])

FORWARD : bloquera juste la transmission de packets par le routeur (utilisé sur les routeurs [uniquement les redirections, on peut toujours parler au routeur mais il ne transmettra pas mon message])

Une image contenant table

Description générée automatiquement

Le SSH :

Le SSH, pour Secure Shell, désigne à la fois un protocole de communication et un programme informatique. Il permet la connexion d'une machine distante (serveur) via une liaison sécurisée dans le but de transférer des fichiers ou des commandes en toute sécurité.

Comment se connecter : login@serveur -> ndargazan001@10.3.224.215

Les différents Arguments :

-p : le port (default 22)

-L : ouvrir un tunnel SSH

exemple : ssh ndargazan001@10.3.224.215 -p 2222 -L 0.0.0.0 :80 :localhost :80

* + 1. : interface de « sortie » du tunnel (default : 127.0.0.1, mettre 0.0.0.0 permet aux autres machines du réseaux locale de la machine locale d’y avoir aussi accès)

80 : Port sur la machine locale sur lequel sera redistribué les packets

localhost : Cible qui sera redistribué dans le tunnel (Doit être spécifié, peut-être n’importe quelle cible que la machine distante peut contacté, ce n’est pas seulement la machine distante)

Les clés :

Chaque partie à une clé publique et une clé privée, pour passer par identifications par clé :

Imaginons la machine A qui souhaite s’identifier par clé au serveur B :

1. La machine A génère une clé RSA (ssh-keygen)
2. On transmet la clé PUBLIQUE (fini par .pub) au serveur B
3. On met cette clé sur une seule ligne dans un dossier ~.ssh/authorized\_keys du serveur B
4. Attention chaque utilisateurs sur le la machine A a une clé privée/publique différente de même pour le serveur donc il faut bien penser à mettre la bonne clé dans le .ssh du bon utilisateur.
5. Après avoir ajouté la clé sur le serveur B, l’authentification devrait fonctionner sans avoir a redémarrer le service ssh

Modification possible dans le fichier /etc/ssh/sshd\_config :  
● PermitRootLogin (permet d'empêcher la connexion sur l’utilisateur root)  
● Port 2222 (permet de changer le port en 2222 par exemple)  
● AllowUsers toto/DenyUsers toto (Autoriser, interdire utilisateur)  
● Match Adress 192.168.1.5  
 PermitRootLogin yes  
 (Pour autoriser accès root sur la machine 192.168.1.5)

● Sybsystem sftp internal-sftp #Pour le ftp  
 Match group sftp  
 ChrootDirectory /home/%u

X11Forwarding no  
AllowTcpForwarding no

Activer le ssh sur une machine : /etc/init.d/ssh start (ou restart)

Rappels :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement