TP Filtrage Réseau avec iptables

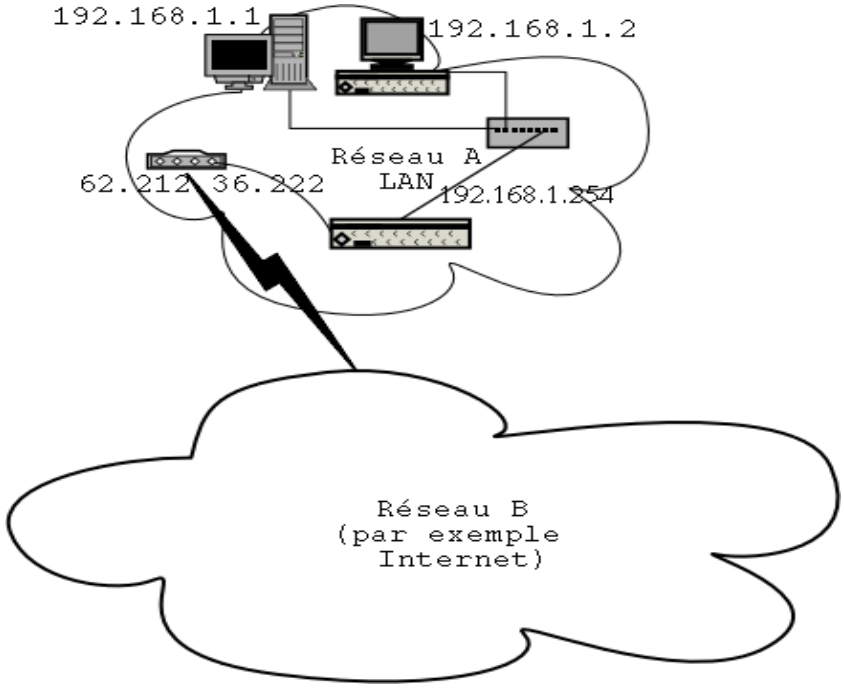
Table des matières

1. COURS

Mise en situation

J'ai 3 machines en réseau(raccordés via un switch) qui forme mon LAN, une de ces machines possède 2 interfaces réseaux : une carte réseau relié au switch qui sera donc mon interface pour joindre mon LAN- une connexion modem relié à un autre réseau, en l'occurrence Internet

Un schéma valant mille explications :

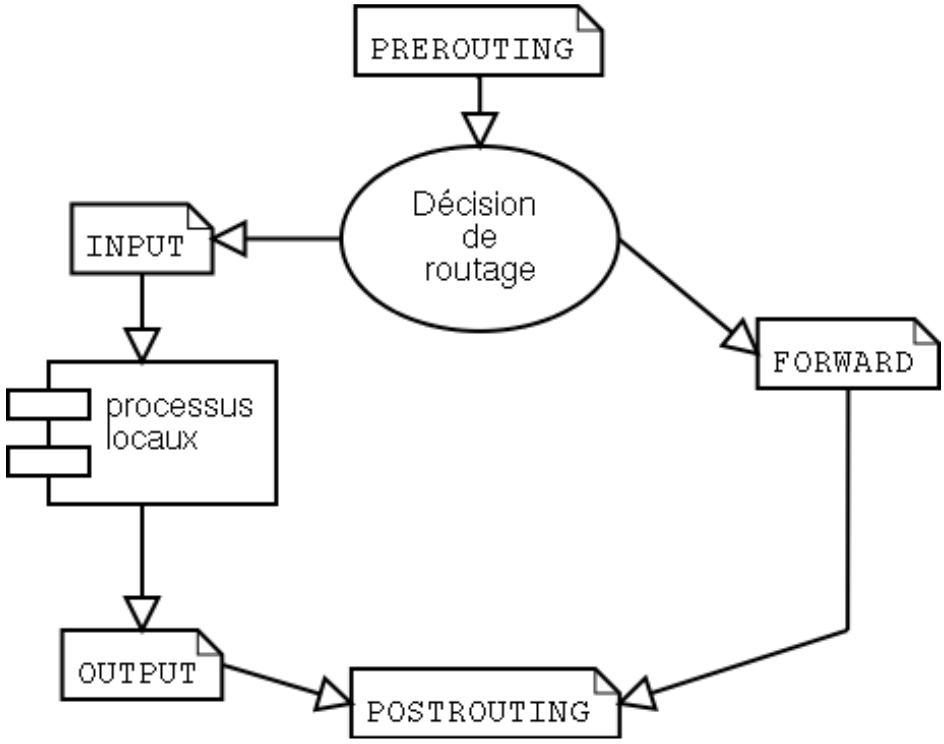


Que puis-je faire avec iptables/netfilter ?

- monter un firewall filtrant basé sur les paquets mais aussi sur le statut des connexions engendrés par les paquets(le suivi de connexion)
- utiliser NAT(Network Address Translation) et le masquerading afin de partager un accès internet à plusieurs machines
- utiliser NAT pour faire du proxy transparent(évite d'avoir à paramétrer le proxy sur les clients/navigateurs web)
- mettre en place(notamment en permettant le marquage des paquets via la table mangle) la possibilité d'utiliser tc+iproute2 dans le but d'obtenir un routeur sophistiqué permettant le QoS(Quality of Service, ie privilégié certains services, mettre en place des limites d'utilisation de bande passante sur un utilisateur, sur un groupe etc)
- manipuler des paquets pour par exemple altéré le champ TOS(2) d'un datagramme IP

Qu'est ce qu'une chaîne ?

Nous allons donc nous positionner comme étant sur la machine qui fera office de firewall/routeur pour tenter de la sécuriser.
Lorsqu'un paquet arrive, il va être orienté(selon un certain nombre de paramètres) dans l'une des différentes chaînes disponibles.



Ainsi que nous le montre le schéma un paquet rentrera toujours dans la machine via la chaîne PREROUTING et sortira toujours de la machine via la chaîne POSTROUTING(chaînes servant notamment à certaines opérations de routage entre les 2 réseaux) raccordés par notre routeur.

Les chaînes INPUT et OUTPUT quand à elle serviront respectivement à placer des règles pour les paquets **destinés à** la machine et ceux **émis par** la machine, pour faire simple si un paquet est destiné à ma machine, "il arrivera dans la chaîne INPUT" [1]

Par exemple si je demande la visualisation d'une page sur le web depuis la machine, j'émets une requête qui sortira par la chaîne OUTPUT et la réponse arrivera sur ma machine par la chaîne INPUT .

[La nuance est dans le "destiné", à savoir que l'on peut considérer qu'un paquet à destination du LAN sera, au moins à certain(s) moment(s) et à certain(s) niveau(x) du modèle OSI, destiné à la machine faisant office de routeur mais sera lui orienté dans la chaîne FORWARD et non pas dans la chaîne INPUT]

Au moment où le paquet rentre dans la chaîne, les règles correspondant à cette chaîne sont appliquées dans l'ordre dans lequel elles sont stockées.

Comment placer une règle dans une chaîne?

SYNOPSIS :

iptables [-t table] -[AIDFP] chain rule-specification [options]

Étant donné que pour la première partie du tp nous n'utiliserons que la table filter, nous pourrons omettre durant celle ci le paramètre -t, car en l'absence de celui-ci la table filter est utilisé par défaut.

Pour manipuler les règles appliquées à une chaîne, on utilisera les paramètres :

-A pour append : rajouter une règle à la suite des autres(la première règle passé par -A se retrouve en première position, les suivantes se retrouve à la suite)
-I pour insert : rajoute une règle qui vient se placer avant toutes les autres(en haut de la pile)
-D pour delete : effacer une règle
    -D numerox : effacer la règle numéro x
    -D règle : efface la règle
-F pour flush : on obtient le même résultat en effaçant toutes les règles une par une
-P : sert à fixer les policy par défaut, c'est à dire à fixer l'action à prendre par défaut(ie quand aucune règle n'est matché).

La premier paramètre à connaître est le paramètre -j, car c'est avec celui-ci que l'on va dire quoi faire lorsqu'un paquet match la règle que l'on écrit. Le paramètre -j s'utilise de la manière suivante :

-j target

Ainsi une règle basique sera de la forme :

iptables -A chaîne -j target

Nous utiliserons dans un premier temps les 2 cibles suivantes(ensuite nous verrons que nous pourrons créer nos propre cibles) :

ACCEPT: laisser passer le paquet DROP : refuser le paquet sans réponse
REJECT : refuser la paquet avec une réponse
[par défaut il existe une cible QUEUE et RETURN que nous n'utiliserons pas dans ce tp]

N.B: avant de commencer le tp vous prendrez soin de couper tout service qui pourrait modifier les règles iptables(/etc/init.d/service stop, où service peut être iptables,shorewall etc).

Vous vérifierez avec la commande iptables -L (liste les règles qui sont en ce moment appliquée) qui devra vous donner ceci :

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Le "policy ACCEPT" n'est pas bénin, en effet ceci précise l'action à prendre par défaut(i.e quand aucune règle ne match)

2. TP IPTABLES

Opérations sur une seule chaîne et sur la table filter:

Créer les règles suivantes : (vous noterez sur cette feuille chacune des règles demandées, ainsi que le test de la règle, à savoir un copié/collé du terminal, et/ou du résultat d'un sniff(ethereal,ngrep etc)

- Première règle

  Interdire tout paquet entrant
  Effacer la règle

- paramètre protocole

  Interdire le protocole ICMP entrant
  Effacer la règle

- paramètre source

  Interdire le protocole ICMP provenant de localhost
  Effacer la règle

- chaîne OUTPUT paramètre destination

  Interdire tout paquet sortant à destination de localhost
  Effacer la règle

- paramètre inversion

  Interdire un paquet s'il ne provient pas de localhost
  Effacer la règle

- paramètre interface d'entrée

  Interdire tout paquet entrant s'il provient de lo (à ne surtout jamais faire sur une machine si l'on ne sait pas EXACTEMENT ce que l'on fait)
  Effacer la règle

- paramètre interface de sortie

  Interdire tout paquet sortant par eth0
  Effacer la règle

- paramètre destination port

  Interdire tout paquet sortant à destination du port ftp
  Effacer la règle

- paramètre source port

  Interdire tout paquet sortant par eth0 dont le numéro de port destination est inférieur à 1025
  Effacer la règle et retester une connexion ftp

Tester une connexion ftp(ou n'importe quelle serveur qui tourne sur un port «privilégié») depuis une machine distante sur votre machine

Les services qui tournent sur des ports < 1025 sont des services qui tournent sous le compte root et sont donc à protéger prioritairement

- paramètre flag TCP

  Interdire toute tentative d'initialisation de connexion TCP provenant de eth0
  Effacer la règle

- paramètre flag icmp
  Interdire tout paquet entrant correspondant à un ping
  Effacer la règle
  Interdire toute réponse à un ping
  Effacer la règle

- paramètre extension:
  - extension mac

Attention on ne peut utiliser l'extension mac que sur les tables INPUT, PREROUTING et FORWARD ;

Interdire tout paquet entrant par eth0 dont l'adresse mac n'est pas celle du voisin

Effacer la règle

- extension limit

Positionner la police par défaut à DROP pour la chaîne INPUT

Écrire une règle qui laisse entrer 5 tentatives de connexion TCP puis qui n'en laisse passer plus que 2 par minute

Faire de même avec les pings

On suppose que le burst a été consommé, combien de temps(sans tentative de connexion ou d'echo-request)faudra t'il pour qu'on puisse à nouveau avoir 5 des ces paquets qui puissent passer à la suite ?

Effacer la règle

- le suivi de connexion(ip_conntrack)

Positionnez les règles par défaut à DROP pour les chaînes INPUT, OUTPUT, FORWARD

Autoriser tout paquet relatif à une connexion déjà établi ou en rapport avec une connexion déjà établi en entrée

Interdire tout paquet relatif à une connexion de type INVALID

Autoriser tout paquet créant une nouvelle connexion en sortie à destination du port 80 Que faut il modifier ici pour que l'on puisse naviguer sur le net ?Effacer la règle

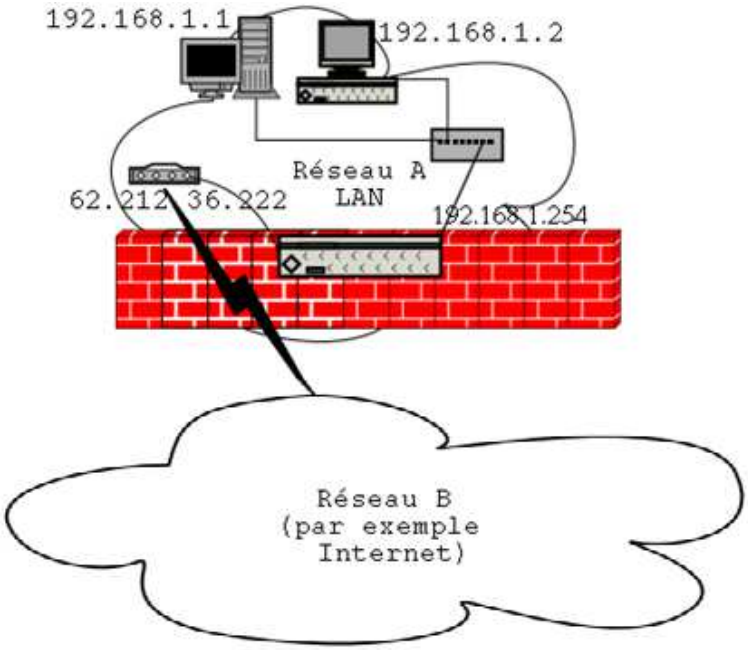
Opérations sur plusieurs chaînes et sur la table filter:

- création d'un nouvelle chaîne

Créer une nouvelle chaîne qui log les paquets entrants en ajoutant le préfixe [INPUT DROP] et qui les drop

Renvoyer sur cette nouvelle chaîne tout paquet engendrant une nouvelle connexion en entrée

Désormais nous sommes en mesure d'obtenir la configuration suivante :



Opérations sur plusieurs chaînes et sur plusieurs tables :

[Pour cette partie nous travaillerons sur des machines ayant au minimum 2 interfaces réseau]

- modification de champ TCP/IP ; table nat et chaînes PREROUTING,POSTROUTING ; cibles SNAT, DNAT, MASQUERADE

Positionnez les règles par défaut à DROP pour les chaînes INPUT, OUTPUT, FORWARD

Créer une règle qui modifie tout paquet qui arrive via l'interface WAN à destination du port 2222 afin que ce paquet ai dans son champ IP DST l'adresse 192.168.0.1 et dans son champ TCP DPORT 22

Que se passe t il si on tente une connexion sur WAN sur le port 2222 ?

Que faut il faire pour que la translation fonctionne effectivement ? (dans un sens comme dans l'autre)

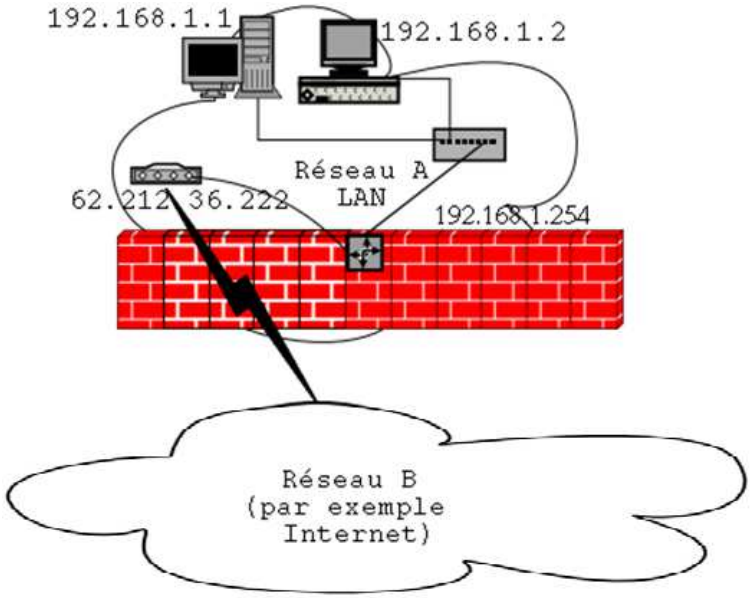
Pour vous aider mettez ces règles dans un script se terminant par une règle qui log et drop tout et ensuite regarder attentivement les logs

Effacer ces règles(sauf les polices par défaut)

Créer une règle qui altère le champ IP SRC de tout paquet sortant via l'interface WAN, en remplaçant la valeur de ce champ par l'adresse IP de cette interface(WAN)

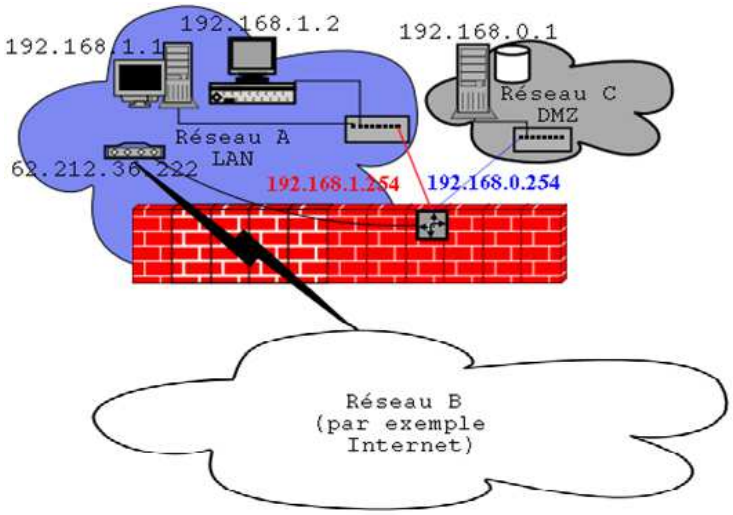
Autoriser tout trafic provenant de LAN à être forwardé par notre machine
Autoriser tout trafic de statuts ESTABLISHED,RELATED à être forwardé par notre machine

Désormais nous sommes en mesure d'obtenir la configuration suivante :



Le but sera ici de se placer dans un cas concret, et de répondre au mieux aux besoins de filtrage, d'accès aux services et de qualité de service.

Le cas concret :



Nous considérerons qu' iptables est installé sur la machine servant de routeur/firewall et nous allons donc nous attacher à écrire le script pour cette machine.
Les machines, le routeur et le serveur placé dans la DMZ, doivent être protégées au mieux. Le routeur a 3 interfaces réseau :

eth0(192.168.0.254) relié à la DMZ
eth1(192.168.1.254) relié au LAN
ppp0(62.212.36.222) relié à internet

La machine doit pouvoir être joignable via SSH depuis le LAN, et depuis Internet. Les machines du LAN doivent pouvoir aller sur Internet(HTTP et FTP).
Les machines du LAN doivent pouvoir pinger une machine sur Internet.
Sur la DMZ, la machine 192.168.0.1 héberge le site web de l'entreprise, un relay mail et un serveur imap-ssl qui doivent être joignable, depuis le LAN, et depuis Internet. Cette machine doit aussi être joignable par SSH depuis le LAN et depuis Internet.
La machine est une debian et est maintenu à jour via apt, aussi la machine devra pour pouvoir aller télécharger via FTP et HTTP les mises à jours sur par exemple ftp.fr.debian.org.

On veut mettre en place un proxy transparent sur un serveur dédié (192.168.0.2) situé dans la DMZ, que faudrait il changer dans notre script ?

Un firewall bloque tout par défaut

Un script firewall commence toujours par effacer toutes les règles(par défaut ou crées par un utilisateur) qui pourraient être actives

La génération de log d'iptables est un atout majeur si les logs générés sont clairs,précis et

lisibles Le renvoi d'un TCP RST peut être utile parfois pour améliorer les temps de réponses

Un certain nombre de restrictions sont accessibles via /proc/sys/net/ voir dans les sources du kernel Documentation/networking/ip-sysctl.cfg

L'utilisation du suivi de connexion sur un système sécurisé est recommandé car même s'il induit une charge supérieur pour le routeur il est très pratique

Attention à la génération de log qui peuvent occuper un volume qui pourrait saturer le

disque La création de ses propres chaînes est avantage dont on peut difficilement se passé.

Un script firewall doit être un maximum modulable(il est tout à fait possible et même conseillé d'y déclarer des variables, par exemple LAN=192.168.0.0/24), facilitant ainsi grandement d'éventuels futures modifications

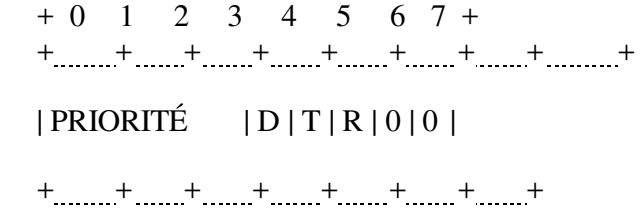
Un script non testé est un mauvais script

Il est courant d'attaquer un serveur dans une DMZ en se servant d'une machine vérolé du LAN, il est donc très important de sécuriser les échanges entre le LAN et la DMZ, tout autant que ceux entre Internet et nos 2 réseaux.

[1] : en vérité le paquet n'arrive pas, physiquement parlant, dans la chaîne mais dans un souci de facilité de compréhension je me suis permis cette légère vulgarisation.

[2] : Type de Service : 8 bits
Le Type de Service donne une indication sur la qualité de service souhaitée, qui reste cependant un paramètre "abstrait". Ce paramètre est utilisé pour "guider" le choix des paramètres des services actuels lorsqu'un datagramme transite dans un réseau particulier. Certains réseaux offrent un mécanisme de priorité, traitant préférentiellement un tel trafic par rapport à un trafic moins prioritaire (en général en acceptant seulement de véhiculer des paquets d'un niveau de priorité au dessus d'un certain seuil lors d'une surcharge momentanée). Principalement, le choix offert est une négociation entre les trois contraintes suivantes : faible retard, faible taux d'erreur, et haut débit.

Bits 0-2 : Priorité.
Bit 3 : 0 = Retard standard, 1 = Retard
faible. Bits 4 : 0 = Débit standard, 1 =
Haut débit.
Bits 5 : 0 = Taux d'erreur standard 1 = Taux d'erreur
faible. Bit 6-7 : Réservé.



+ Aide :

man iptables : INCONTOURNABLE

iptables --help, iptables -p TCP --help, iptables -p icmp --help etc

iptables -L : liste les règles iptables actives

un certain nombre de paramètre sont activable via le système de fichier virtuel /proc, par exemple pour activer le port forwarding on s'y prends comme cela :

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

/etc/services(voir aussi /etc/protocols) : liste les services et leur numéro de port

/usr/include/linux/icmp.h : liste des noms des différents type et code icmp

http://www.netfilter.org/documentation/HOWTO/fr/packet-filtering-HOWTO.html

http://christian.caleca.free.fr/netfilter/

D'une manière générale quand on ne comprends pas ce qui ce passe(pourquoi ça marche ? pourquoi ça marche pas ?) on place une règle de log et on regarde les logs ainsi crées