Compléments d'informations pour IPTABLES/NETFILTER : Mars 2002.

Table of Contents

Complément d'informations pour

configurer
un firewall (NetFilter 1.2.5+string).

But :

Procurer un complément d'informations pour tout ceux qui désirent implémenter un firewall performant à base de produits totalement libre (Cf GPL) . Et comme j'ai été confronté à de nombreux problèmes, ce document rassemble des « conseils » mais aussi des solutions. Toutefois ce document n'est en aucun un remplacement des How-To pour NetFilter, NAT/DNAT... Je conseille à ce titre la lecture des documents joints à coté du mien, et la visite de certains liens internet.

Une fois finie, cette technologie déployée, il vous sera possible de masquer un réseau, partager une connection par modem ou cable et/ou de sécuriser 2 connections issue de milieu différents. Et bien plus aussi, ceci ne dépend que de vous.

Ce document est très probablement perfectible, en ce qui concerne les données « techniques » tout aussi probable, ne pas hésiter à modifier, corriger ou signaler les erreurs (Cf licence LDP)

Pour me contacter : archi_l@club-internet.fr .

Matériel nécessaires :

1. Un PC du 386 à ... , je conseille pour une question de temps l'emploi d'un pentium, lors de la compilation du noyau on comprend mieux pourquoi. Ce système permet d'utiliser un ordinateur qui n'interesse plus personne ... Mais si !!! Gratuitement, il va se transformer en puissant firewall.

2. Temporairement une souris et un lecteur de cédérom.

3. Un disque dur de petite capacité (<=500 Méga est faisable sans problème).

volumineux. De plus il peut être utile de les garder durant X jours/semaines/mois...

4. Un modem ADSL (éventuellement).
5. 2 cartes ethernet (1 en 10 Mhz vers modem possible)
6. De la patience pour la mise au point.

OS utilisé :

Basé sur une RedHat 7.2 en version cédérom, c'est lors de l'installation que le lecteur de Cédérom et la souris sont nécessaires. D'autres distribution peuvent bien évidemment être choisie, comme Mandrake, Suze ... Attention de bien installer la version noyau 2.4.x, car iptable(Netfilter) ne fonctionne que sur des noyaux de versions 2.4 ou 2.5(experimentale à la date du document).

En procédant de cette manière, divers package(RPM) seront installer et seront inutiles pour le firewall, même si vous faites une installation personnalisée en sélectionnant les modules un par un. Je traiterais la suppression des packages après.

Principe du firewall :

Dans le style "tout se qui est autorisé peut passer",
le firewall devra à partir de cette politique être en mesure de réagir selon 3 actions :

1. ACCEPTE, le paquet suit sa route.
2. REJETER, le paquet est détruit mais un message est transmis à l'emetteur pour le lui dire.
3. DROP, le paquet est détruit.

Les actions les plus communes utilisées sont ACCEPT et DROP. Entre DROP et REJECT, peut de différence à l'exception qu'un message est générée pour prevenir l'emetteur du refus d'acceptation. Ceci génére un trafic et une ressource CPU(bien que minime) souvent peut utile (voir après).

Netfilter accepte aussi une 4eme possibilité "QUEUE", qui permet un traitement local des données du paquet. (non traité ici)

Firewall utilisé :

Le firewall sera representée shématiquement comme une unité centrale dans

Cette unité sera décomposée en 2 parties bien distinctes comme ci-dessous dans le shéma :
un proxy HTTP et Iptables (NetFilter) jouant le rôle de filtrage.

L'ajout d'un proxy HTTP permet ce que Iptables ne fait pas aussi bien et inversement.

Ce document ne prendra pas en compte sauf de manière rapide le proxy Squid.

Proxy HTTP :

Il s'agit du proxy applicatif Squid dans sa version 2.4-Stable1-5 en RPM. Squid permet tout sorte de configuration; mon fichier de configuration en exemple. Cette configuration est rapide, sans problème ... au moins jusqu'à présent.

Attention au port d'entrée du proxy, dans les exemples suivants le port 8080 à été choisie par simplicité, par défault Squid écoute sur le port 3128.

Le site de Squid : http://www.squid-cache.org/

L'avantage d'un tel produit est de pouvoir interdire des URL mais aussi d'enregistrer toutes les connections et requètes demandées. (Chose que ne peut faire Netfilter/Ipchains ...etc).

Il ne faut pas oublier bande passante, qui est dans ce cas optimizée par le cache

Exemple de configuration :
http://linux.oreillynet.com/pub/a/linux/2001/07/26/squid.htm #

Voilà ma configuration : L'installation ne requiert rien de bien spécial, je conseille l'installation depuis les sources. Les fichiers joints avec les sources permettent une installation théoriquement sans problème (en Anglais ...)

Pour information, la place occupée par le répertoire cache est de 108 mégas (variable en fonction du fichier configuration).

Mon fichier de configuration, ne comprends pas de restriction "style parentale".

```
http_port 8080
icp_port 8080
cache_mgr archi_l@club-internet.fr
```

```
visible_hostname proxy.archi.fr
cache_mem 16 MB
# 100 M de cache – 16 répertoires et 256 s/rép.
cache_dir ufs /var/spool/squid 100 16 256
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log none
acl manager proto cache_objet
acl CLIENTS src 192.168.0.0/255.255.255.0
acl LOCAL src 127.0.0.1
acl all src 0.0.0.0/0.0.0.0
always_direct allow LOCAL
#always_direct allow CLIENTS

hierarchy_stoplist cgi-bin
acl cgi urlpath_regex cgi-bin \

http_access allow manager LOCAL
http_access allow CLIENTS
http_access deny all

icp_access allow CLIENTS
icp_access allow LOCAL
icp_access deny all
# (on interdit toutes les autres)

logfile_rotate 4
fake_user_agent Mozilla/1.0 (linux;32-bits)

minimum_object_size 0 KB
maximum_object_size 1024 KB
maximum_object_size_in_memory 64 KB
connect_timeout 30 seconds
request_timeout 30 seconds
read_timeout 2 minutes

quick_abort_min 64 KB
quick_abort_max 64 KB
quick_abort_pct 90

#dns_children 5
# 5 processus pour requete DNS-Si compile avec ...
positive_dns_ttl 24 hours
negative_dns_ttl 5 minutes

acl SSL_ports port 443 563
acl Safe_ports port 22 21 54 80 123 210 280 443 488 563 777
1025-65535
acl CONNECT method CONNECT
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

# Limites la taille des requetes
request_header_max_size 10 KB
request_body_max_size 128 KB

client_netmask 255.255.255.0
anonymize_headers deny Referer Server From User-Agent
```

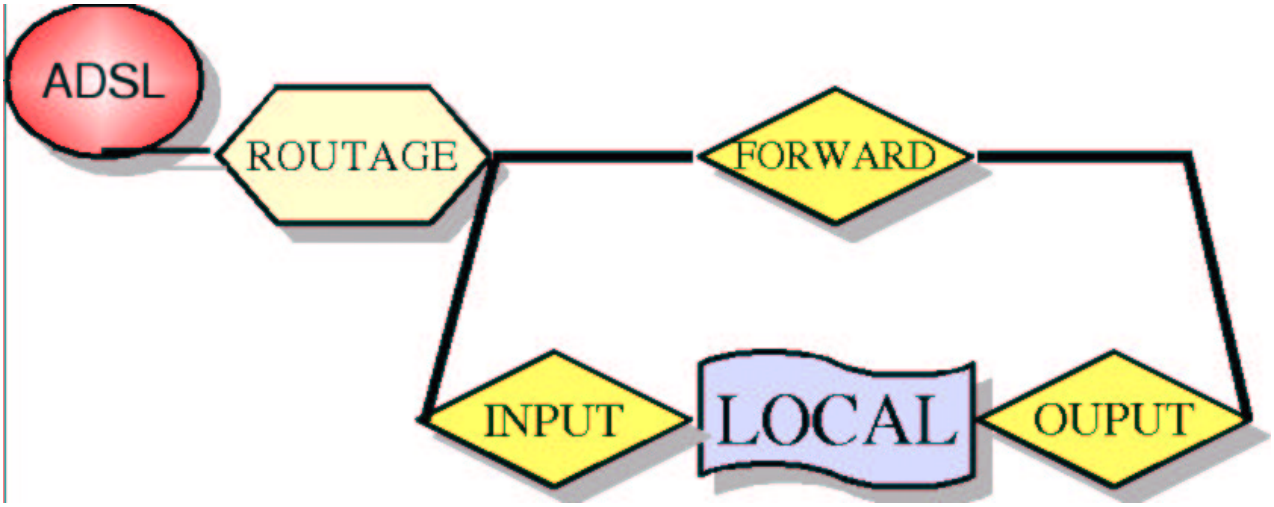
Pourquoi IPTABLE:

Parce qu'il est stable, souple et puissant, de plus je pense que mon investisssement « cérébral » vis à vis d'iptables est viable pour une période relativement longue. Le coté évolutif est a prendre en compte, il est possible de devellopper ses propres programmes (librairies) pour le traitement des paquets (pas pour moi...).

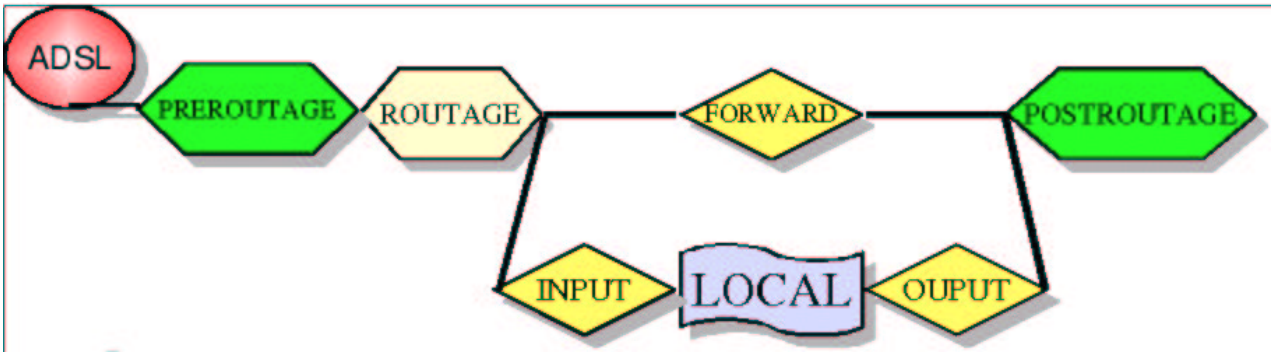
Les grandes nouveautés sont :

- Stateful packet filtering (surveillance des connections).
- NAT avec du SNAT et DNAT (modification d'adresse IP Source et/ou Destination)
- Redirection de connection. (relayage transparent)
- Accès à QoS (Qualité de service)
- Accès aux drapeaux TOS
- Comptage des paquets et des bytes.

Principe du filtrage IPCHAINS (pour mémoire)



Sans rentrer dans le détail, ipchains ne comporte qu'une seule table FILTER avec les chaines INPUT, OUTPUT et FORWARD, le NAT(Network Translation d'Adresse) est possible.



3 tables sont désormais présentes, comme ci-dessous, chaque table contenant des chaines, qui elles mêmes peuvent contenir des règles :

1 - FILTER TABLE

Chain INPUT (policy DROP 0 packets, 0 bytes)
Chain FORWARD (policy DROP 3 packets, 168 bytes)
Chain OUTPUT (policy DROP 149 packets, 204K bytes)

2 - NAT TABLE

Chain PREROUTING (policy ACCEPT 3440 packets, 215K bytes)
Chain POSTROUTING (policy ACCEPT 189 packets, 8228 bytes)
Chain OUTPUT (policy ACCEPT 666 packets, 33279 bytes)

3 - MANGLE TABLE

Chain PREROUTING (policy ACCEPT 110K packets, 17M bytes)
Chain OUTPUT (policy ACCEPT 34268 packets, 17M bytes)

Configuration du noyau :

Configurer le noyau : (2.4.17) (Le fichier complet est joint ci-dessous »Conf_Noyau »), à placer dans le répertoire /usr/sr/linux (lien vers linux-2.14.17 ou autre) en .config

Ne garder que l'essentiel !!! valable pour tout le reste .

Deux possibilités, soit intégrer les modules au noyau ou chargement des modules à volonté en cas de bisoin par le noyau, prenez en compte que le système est plus rapide si les modules sont intégrés dans le noyau, de plus un noyau « type Netfilter » est de taille relativement petite. Ce qui permet d'intégrer les modules dans le noyau, en incluant les modules « mon » noyau fait 690.163 Ko.

Attention au module IRC, si employé bien vérifier la version de Netfilter et utiliser la version qui corrige une bogue important !!!.(Prenez la dernière stable).

Automatically generated by make menuconfig: don't edit
Code maturity level options

CONFIG_EXPERIMENTAL=y
# Loadable module support
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
CONFIG_KMOD=y
# Networking options
CONFIG_PACKET=y
# CONFIG_PACKET_MMAP is not set
CONFIG_NETLINK_DEV=m
CONFIG_NETFILTER=y
# CONFIG_NETFILTER_DEBUG is not set
# CONFIG_FILTER is not set
CONFIG_UNIX=y
CONFIG_INET=y
# CONFIG_IP_MULTICAST is not set
# CONFIG_IP_ADVANCED_ROUTER is not set
# CONFIG_IP_PNP is not set
CONFIG_NET_IPIP=m
CONFIG_NET_IPGRE=m
# CONFIG_ARPD is not set
CONFIG_INET_ECN=y < ------- Conseiller en "is not set".
CONFIG_SYN_COOKIES=y
# IP: Netfilter Configuration
CONFIG_IP_NF_CONNTRACK=y
CONFIG_IP_NF_FTP=y
CONFIG_IP_NF_IRC=y <-------- Conseiller en "is not set".
# CONFIG_IP_NF_QUEUE is not set
CONFIG_IP_NF_IPTABLES=y
CONFIG_IP_NF_MATCH_LIMIT=y
CONFIG_IP_NF_MATCH_MAC=y
CONFIG_IP_NF_MATCH_MARK=y
CONFIG_IP_NF_MATCH_MULTIPORT=y
CONFIG_IP_NF_MATCH_TOS=y
CONFIG_IP_NF_MATCH_LENGTH=y
CONFIG_IP_NF_MATCH_TTL=y
CONFIG_IP_NF_MATCH_TCPMSS=y
CONFIG_IP_NF_MATCH_STATE=y
CONFIG_IP_NF_MATCH_UNCLEAN=y
CONFIG_IP_NF_MATCH_OWNER=y
CONFIG_IP_NF_FILTER=y
CONFIG_IP_NF_TARGET_REJECT=y
CONFIG_IP_NF_TARGET_MIRROR=y
CONFIG_IP_NF_NAT=y
CONFIG_IP_NF_NAT_NEEDED=y
CONFIG_IP_NF_TARGET_MASQUERADE=y
CONFIG_IP_NF_TARGET_REDIRECT=y
CONFIG_IP_NF_NAT_SNMP_BASIC=y
# CONFIG_IP_NF_NAT_IRC is not set **< -- A enlever si pas utile**
CONFIG_IP_NF_NAT_FTP=y
CONFIG_IP_NF_MANGLE=y
CONFIG_IP_NF_TARGET_TOS=y
CONFIG_IP_NF_TARGET_MARK=y <-- Pour accès au QoS
CONFIG_IP_NF_TARGET_LOG=y
CONFIG_IP_NF_TARGET_TCPMSS=y
#
QoS and/or fair queueing<< --- FACULTATIF (QoS)
#
CONFIG_NET_SCHED=y
CONFIG_NET_SCH_CBQ=y
# CONFIG_NET_SCH_CSZ is not set
# CONFIG_NET_SCH_PRIO is not set
# CONFIG_NET_SCH_RED is not set
# CONFIG_NET_SCH_SFQ is not set
# CONFIG_NET_SCH_TEQL is not set
# CONFIG_NET_SCH_TBF is not set
# CONFIG_NET_SCH_GRED is not set
# CONFIG_NET_SCH_DSMARK is not set
# CONFIG_NET_SCH_INGRESS is not set
# CONFIG_NET_QOS is not set
# CONFIG_NET_CLS is not set

Toutefois en se basant sur les souces d'IPTABLES, il est possible d'installer beaucoup d'autres modules. Dans la version 1.2.5 + kernel 2.4.17 testé et qui fonctionne contrairement à : iptable 1.2.6 et alpha +kernel 2.4.18 ou de gros problèmes arrivent ... Il est donc possible d'installer les modules suivants :

/iptables-1.2.5/patch-o-matic/base/psd.patch
/iptables-1.2.5/patch-o-matic/base/NETMAP.patch
/iptables-1.2.5/patch-o-matic/base/mport.patch
/iptables-1.2.5/patch-o-matic/base/realm.patch
/iptables-1.2.5/patch-o-matic/base/iplimit.patch
/iptables-1.2.5/patch-o-matic/base/ulog.patch
/iptables-1.2.5/patch-o-matic/base/ah-esp.patch
/iptables-1.2.5/patch-o-matic/base/IPV4OPTSSTRIP.patch
/iptables-1.2.5/patch-o-matic/base/pkttype.patch
/iptables-1.2.5/patch-o-matic/base/pool.patch
/iptables-1.2.5/patch-o-matic/base/random.patch
/iptables-1.2.5/patch-o-matic/base/quota.patch
/iptables-1.2.5/patch-o-matic/base/SAME.patch
/iptables-1.2.5/patch-o-matic/base/NETLINK.patch
/iptables-1.2.5/patch-o-matic/base/TTL.patch
/iptables-1.2.5/patch-o-matic/base/time.patch
/iptables-1.2.5/patch-o-matic/base/ftos.patch
/iptables-1.2.5/patch-o-matic/base/ipv4options.patch
/iptables-1.2.5/patch-o-matic/base/nth.patch
/iptables-1.2.5/patch-o-matic/extra/helper.patch
/iptables-1.2.5/patch-o-matic/extra/record-rpc.patch
/iptables-1.2.5/patch-o-matic/extra/tcp-window-tracking.patch
/iptables-1.2.5/patch-o-matic/extra/talk-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/extra/eggdrop-conntrack.patch
/iptables-1.2.5/patch-o-matic/extra/pptp-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/extra/ctnetlink.patch
/iptables-1.2.5/patch-o-matic/extra/recent.patch
/iptables-1.2.5/patch-o-matic/extra/dropped-table.patch
/iptables-1.2.5/patch-o-matic/extra/string.patch
/iptables-1.2.5/patch-o-matic/extra/tftp.patch
/iptables-1.2.5/patch-o-matic/extra/ftp-fxp.patch
/iptables-1.2.5/patch-o-matic/submitted/ip6tables-export-symbols.patch
/iptables-1.2.5/patch-o-matic/submitted/conntrack-errormsg.patch
/iptables-1.2.5/patch-o-matic/submitted/tcp-MSS.patch
/iptables-1.2.5/patch-o-matic/submitted/ip_nat_irc-srcaddr-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/skb_clone_copy.patch
/iptables-1.2.5/patch-o-matic/submitted/sackperm.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_LOG.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_MIRROR-ttl.patch
/iptables-1.2.5/patch-o-matic/submitted/module-license.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_REJECT-checkentry.patch
/iptables-1.2.5/patch-o-matic/submitted/TOS-oops-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_mac-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_unclean-ecn.patch

/iptables-1.2.5/patch-o-matic/submitted/ipchains-redirect-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/2.4.4.patch
/iptables-1.2.5/patch-o-matic/newnat/h323-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/newnat/talk-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/newnat/0-newnat5.patch
/iptables-1.2.5/patch-o-matic/newnat/irc-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/pending/mangle5hooks.patch

A chaque patch, est présent un fichier d'aide, le stricte minimum ressemble à se style : (Pour ce patch un "exemple" est donnée dans ma configuration) : En tout cas, il ne faut pas avoir peur de lire chaque module.

Exemple « /iptables-1.2.5/patch-o-matic/extra/string.patch.help» :

- *Author: Emmanuel Roger <winfield@freegates.be>*

Status: Working, not with kernel 2.4.9

This patch adds CONFIG_IP_NF_MATCH_STRING which allows you to

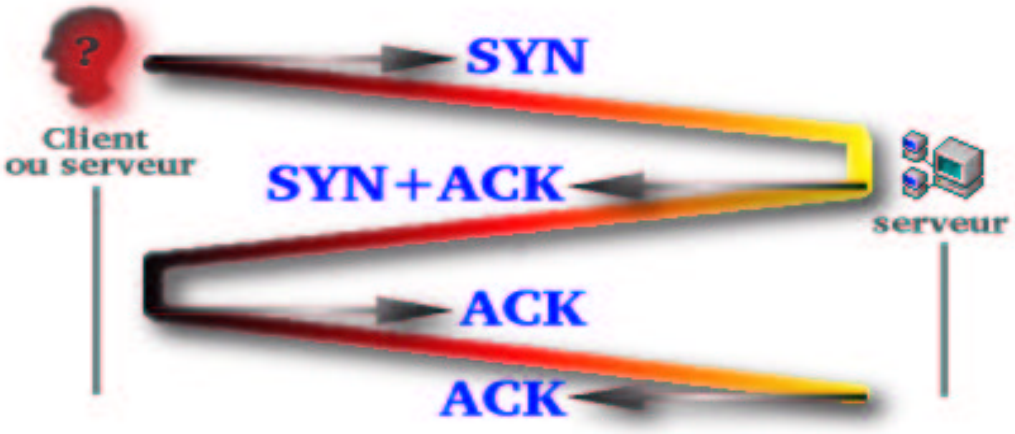
match a string in a whole packet.

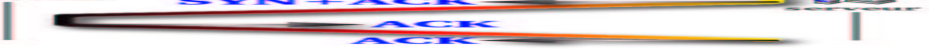
THIS PATCH DOES NOT WORK WITH KERNEL 2.4.9 !!!

(C'est géant ... et après au boulot ...)

Mode « StateFull » ou «IP conntrack » !

Voici une demande de connection (simplifiée), de type classique qui s'établit, et sans incident de transmissions ...





Le client ou serveur demande une connection au serveur par l'envoi d'un SYN, le serveur répond par un SYN+ACK. Le client répond par ACK, la connection est établie, à chaque transmission de paquet le serveur renvoit un ACK (aquittement) ... etc ... Donc une connection commence par SYN, de là il suffit de refuser tous les SYN en provenance d'une interface/adresse pour interdire une connection nouvelle(NEW). Par contre vous pouvez autorisez que votre réseau envoit des SYN(NEW) et recoivent des ACK(ESTABLISHED) (RELATED pour une connection FTP). C'est déjà un filtrage efficace. Pour compléter le sytème de filtrage, il est impératif que celui qui demande et reponde soit mémorisée afin que personne ne s'interpose entre les deux il suffit de mémoriser l'adresse IP du demandeur, le type (TCP/UDP..), le port, l'IP du destinataire, l'état de la connection, et voilà. Le serveur donnera le port pour la connection théoriquement souvent supérieur à 1024 et son IP. C'est la définition simplifiée du mode StateFull, il est possible de rencontrer le drapeau RST, qui permet de couper de façon brute une connection. Par défault, mais cette valeur est paramétrable, une connection reste dans un « TIME_WAIT » DURANT 3 minutes. Et vous pouvez aussi rencontrez sur le port 110, votre FAI vous envoit un ACK+RST et là le firewall jete automatiquement le paquet, ce qui ne porte pas préjudice, mais ajoute une ligne dans le fichier de logs...

Un état « INVALID » existe dans Iptables, il correspond à un paquet de type inclassable, donc détruit.

De plus une limitation du nombre de paquets peut être appliquer, ce qui evite en particulier certaines attaques (DoS).

Les règles d'iptables:

Extrait du «man iptables »

-t, --table

Cette options specifie dans quelle table le paquet va être trait. Le noyau chargera automatiquement le module approprié si compilé intégré

Les tables sont :

Filter

C'est la table par défault. Elle contien les chaines INPUT-FORWARD et OUTPUT.

Nat

Cette table est consulté lors d'une nouvelle connection. Elle consiste à modifier en PREROUTING l'adresse de destination, OUTPUT pour modifier les paquets

source IP de l'emetteur.

Mangle

Cette table est utilisé pour altéré les paquets en PREROUTING (pour modifier les paquets arrivants avant routage) et OUTPUT (pour altérer les paquets générés localement) avant routage.

Options :

iptables -[ADC] Spécification pour les règles dans les chaines [options]
iptables -[RI] chaine règle règle-spécification [options]
iptables -D chaine règle [options]
iptables -[LFZ] [chaine] [options]
iptables -[NX] chaine
iptables -P chaine cible [options]
iptables -E vieille chaine-Nom nouveau-Chaine Nom

-A, --append

Ajoute une ou plusieurs règles dans une chaine.

-D, --delete

Efface une ou plusieurs règles dans la chaine sélectionnée.

-R, --replace

Remplace une règle dans la chaine sélectionnée

-I, --insert

Insert une ou plusieurs règles dans la chaine sélectionnée comme donné dans la chaine numéroté, si le nombre est 1, insert en début de chaine.

-L, --list

Liste toutes mes règles d'une chaine, Si la chaine n'est pas spécifiée, liste toutes les chaines.

C'est équivalent à l'option -Z qui liste et remet à zéro les compteurs.

-F, --flush

Supprime la chaine sélectionnée, équivalent à efface toutes les règles dans 1 chaine.

Remet les compteurs de paquets + bytes à zéro et liste les chaines.

-N, --new-chain

Créer une nouvelle chaine utilisateur.

-X, --delete-chain

Supprime une chaine utilisateur.

-P, --policy

Applique une politique de traitement de paquets à une chaine

Seuls les chaines non utilisateurs doivent avoir cette politique par défault.

-E, --rename-chain

Renomme une chaine utilisateur.

Paramètres :

-p, --protocol [!] protocol

Choix de : tcp, udp, icmp et all.

Le signe ! Veut dire différent de ... ex : ! tcp

-s, --source [!] addresse[/masque]

Spécification de l'adresse IP source, peut être un nom (résolution par DNS), emplois du masque possible, valeur ! Est utlisable comme ci-dessus.

--src est l'alias de cette option.

-d, --destination [!] addresse[/masque]

Spécification de l'adresse IP destination.

--dst est son alias.

-j, --jump target

Specifie ou va aller le paquet si la régle est correcte

Nom d’une interface en entrée, pour INPUT, FORWARD et PREROUTING.

Le signe "!" peut être utilisé. Le "+" signifie toute interface, type ppp+, cela correspond à ppp0, ppp1 ...

-o, --out-interface [!] [name]

Ide mci-dessous mais en sortie pour FORWARD, OUTPUT et POSTROUTING.

[!] -f, --fragment

Cette option désigne des paquets fragmentés, à savoir que certaines attaques partent de paquets fragmentés, donc destruction grace à cette option.

-c, --set-counters PKTS BYTES

Active le comptage de paquets et bytes dans une chaine durant (INSERT,

APPEND et REPLACE opérations) *Non utilisé me concernant.*

Autres options :

-v, --verbose

Cette option liste les chaines, les règles et les marques TOSVerbose output, elle peut recevoir d’autres options.

-n, --numeric

Sortie numérique des adresses IP(non résolution DNS) et des ports .

-x, --exact

Montre les valeurs exactes des paquets et des bytes au niveau des compteurs.

Valeur en K’s (multiples of 1000) M’s (multiples de 1000K) ou en G’s (multiples of 1000M).

Option valide avec seulement le -L.

--line-numbers

Ajoute une numéro à chaque ligne dans les chaines.

--modprobe=<command>

-A, --append

Ajoute une ou plusieurs règles dans une chaine spécifiée.

-D, --delete

Efface une règle dans la chaine spécifiée.

-R, --replace

Remplace une règle dans la chaine spécifiée.

-I, --insert

Insert une ou plusieurs règles dans une chaine précise.

-L, --list

Liste toutes les règles dans une chaine.

-F, --flush

Efface toutes les règles une par une.

-Z, --zero

Efface les compteurs de paquets et bytes de toutes les chaines.

-N, --new-chain

Pour créer une nouvelle en donnant son nom.

-X, --delete-chain

Supprime une chaine dans une table.

-P, --policy

Applique une politique de traitement des paquetys à une chaine ou cible

-E, --rename-chain

Renonme une chaine.

--destination-port [!] [port[:port]]

Destination du port ou série de ports, --dport est un alias.

--tcp-flags [!] mask comp

Permet de filtrer selon les drapeaux TCP comme : SYN ACK FIN RST URG PSH ALL NONE. Exemple de commande : iptables -A FORWARD -p tcp -tcp-flags SYN,ACK,FIN,RST SYN

Ce qui veut regarder tout les drapeaux, si uniquement SYN alors FORWARD

[!] --syn

Identifie un paquet pour une nouvelle connection, donc TCP avec le drapeu SYN activé et les drapeux ACK et FIN non activés Autre possibilité identique --tcp-flags SYN,RST,ACK SYN.

Le signe "!" peut être utilisé.

--source-port [!] [port[:port]]

Port source ou séries de ports (6680:6699). Voir aussi multiport.

--destination-port [!] [port[:port]]

Idem ci-dessus mais pour la destination.

--icmp-type [!] nomtype

Spécification pour le protocol ICMP, ou «nomtype» correspond un type de requete/réponse ICMP. Les noms complets peuvent être obtenus en faisant : iptables -p icmp -h, ce qui donne :

echo-reply (pong)
destination-unreachable
network-unreachable
host-unreachable
protocol-unreachable
port-unreachable
fragmentation-needed
source-route-failed
network-unknown
host-unknown
network-prohibited
host-prohibited

TOS-host-unreachable
communication-prohibited
host-precedence-violation
precedence-cutoff
source-quench
redirect
network-redirect
host-redirect
TOS-network-redirect
TOS-host-redirect
echo-request (ping)
router-advertisement
router-solicitation
time-exceeded (ttl-exceeded)
ttl-zero-during-transit
ttl-zero-during-reassembly
parameter-problem
ip-header-bad
required-option-missing
timestamp-request
timestamp-reply
address-mask-request
address-mask-reply

--mac-source [!] address

Adresse MAC valable uniquement pour les chaines : PREROUTING, FORWARD et INPUT

Limit

Ce module filtre le nombre de paquet (suivi d'un taux horaire ou valeur par défault), souvent utilisé avec la cible LOG (load-sharing...). Le signe '!' peut être utilisé.

--limit rate

Lié avec limit ,limite le nombre maximum de paquets en un temps : x '/second', '/minute',

'/hour', ou '/day'; la valeur par défault est 3/heures.

--limit-burst number

Le nombre maximum de paquet à traiter dans un temps initial, dès la limite atteinte, le compteur est remis à zéro (default 5).

Exemple : -m limit limit 100/s -limit-burst 5 , va faire que 40 paquets par secondes

multiport

Pour lister une serie de ports (max 15) avec utilisation des options -p tcp ou udp.

--source-port [port[,port]]

Provenance du port à filtrer.

--destination-port [port[,port]]

Inverse de ci-dessus, destination du port

--port [port[,port]]

Filtre si le port origine et destination sont égaux.

mark

Module utilisé pour marquer un paquet, employé pour faire du QoS.

--set-mark mark

Utilisé pour faire du QoS.

--mark value[/mask]

Utilisé pour faire du QoS.

owner

--uid-owner userid permet de spécifier l'UID de l'utilsateur qui à créer le paquet.

--gid-owner groupid idem mais pour le GID

--pid-owner processid permet de spécifier l'ID du processus qui à créer le paquet.

--sid-owner sessionid Idem mais pour le groupe.

state

Module de surveillance/filtrage de connections.

--state state

Comme déjà, peut prendre les valeurs INVALID/ESTABLISHED/NEW et

tos

Module pour modification sur le champ type of service

--tos tos

5 valeurs :

0 Service Normal (paquet classique)

2 Minimize-Cost

4 Maximize-reliability

8 Maximize-Throughput

16 Minimize-Delay

Extensions cibles :

LOG

Module pour écrire via Syslogd des évenements.

--log-level level

Niveau des log valeur de 0 à 7 (après = console)

--log-prefix prefix

Phrase pour identifier le/les logs , maximum 29 caractères.

--log-tcp-sequence

Ecrit la séquence TCP,

--log-tcp-options

Ecrit l'entête TCP.

--log-ip-options

Ecrit les options du paquet IP.

REJECT

rejet à celui qui a envoyé le paquet. Valable dans les chaine : INPUT, FORWARD et OUTPUT

--reject-with type

Type de message (ICMP) qui sera retourné lors d'un rejet :

icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable,

icmp-proto-unreachable, icmp-net-prohibitedor icmp-host-prohibited, pour retourner un message d'erreur approprié. Et sinon port-unreachable est la valeur par défault.

Et dans beaucoup de cas, tcp-reset peut être utilisé.

TOS

Module permettant d'activer des drapeaux spéciaux utilisé dans le routage du paquet.

--set-tos tos

Utilsation d'une valeur numérique ou de la valeur texte

SNAT

Cette cible permet la modification des adrresses IP sources dans la table NAT et dans la chaine POSTROUTING.

--to-source <ipaddr>[-<ipaddr>][:port-port]

Cette option spécifie la nouvelle adresse (source), il est possible d'utiliser une série d'adresses IP.

L'utilisation de ports ou séries de ports est aussi possible.

Uniquement pour : -p tcp et -p udp.

DNAT

Idem pour SNAT mais pour l'adresse de destination, toujours la table NAT et chaines :

PREROUTING et OUTPUT .

--to-destination <ipaddr>[-<ipaddr>][:port-port]

Emplois identique à --to-source

Module capable de modifier l'adresse d'une source sur LAN... et inversement vers une liaison à IP dynamique.

Ce module se retrouve dès qu'un modem/cable est activé. Cette cible se trouve dans la table NAT dans la chaine POSTROUTING

--to-ports <port>[-<port>]

Spécification d'une série de ports avec la chaine SNAT, en complément de -p tcp or -p udp.

REDIRECT

Très pratique redirige un port vers un autre port ou une adresse, en particulier pour le port 80, qui peut être dirigé automatiquement vers le proxy HTTP. Se place dans la table NAT, et en
PREROUTING ou OUTPUT

--to-ports <port>[-<port>]

Specifie le port destination lors d'une redirection.

EXTRA EXTENSIONS

Je passes sur les options EXTRA, libre à vous de les utiliser, elles sont :

ttl / --ttl ttl / --ttl-set ttl / --ttl-dec ttl / --ttl-inc ttl / --ulog-nlgroup <nlgroup> /

--ulog-prefix <prefix> / --ulog-cprange <size> / --ulog-qthreshold <size>

Exemples :

Les possibilités de filtrage étant très nombreuses, un exemple avec accès au port 80 depuis internet est donnée. (Attention, règles à titre d'exemple)

Par adresse IP : input -dport 80 -s 200.1.2.3 -j ACCEPT

-> Toutes connections pour le port 80 de source IP¨200.1.2.3 est accepté.

Par adresse MAC : input -p tcp -dport 80 --mac-source 00:50:FC:54:48:45 -j ACCEPT

-> Toutes connections TCP pour le port 80 d' adresse MAC source 00:50... est accpté.

-> Toutes connections de type TCP en destination du port 80 est acepté.

Par interface : input -i eth0 -p tcp -dport 80 --mac-source 00:50:FC:54:48:45 -j ACCEPT

-> Toutes connections en provenance de ETH0 (TCP+port 80+Adresse MAC) sont accepté.

Par type : input -p all -dport 80 --mac-source 00:50:FC:54:48:45 -j ACCEPT

-> Toutes connections de type all =TCP/ICMP/UDP sur port 80 d'adresse MAC xxx est accepté (exemple ridicule, port 80=TCP)Les types sont "tcp -udp-icmp".

Par type de connection : input -p tcp -dport 80 --mac-source 00:50:FC:54:48:45 -m state -NEW,ESTABLISHED -j ACCEPT

-> Toutes connections (TCP+port 80+adresse MAC) avec état nouvelle ou connection établie est accepté.

En conclusion, les possibilités sont nombreuses et permettent un filtrage précis. Il est donc possible de filtrer les paquets selon :

- Adresse IP source.
- Adresse IP destinataire
- Type de protocole
- Port source.
- Port destination.
- Interface traversée
- Type de connexion.

De là, les paquets sont :

- AUTORISES.
- DÉTRUITS.
- REJETES.
- DETOURNER.

Exemple de configuration :

«/etc/sysconfdir.iptables »: (chmod 700)(root)

Attention, toutes les formes de paquets ICMP ont été listé, il est largement possible d'optimizer le tout, de fàçon plus courte.

J'ai personnellement choisi de faire passer tout le trafic depuis ou vers le VPL/LAN en forward.
Donc les chaines INPUT et OUTPUT ne sont réservées que pour le Firewall.
Lequel possède un serveur de méls, et apache pour le Web.

Lien vers le fichier :

Ecritures des règles/chaines pour IPTABLES :

Il existe de très nombreux exemples sur internet de scripts pour NetFilter. Certains sont mêmes "dynamiques"(bon courage ...). Ensuite, il existe plusieurs façon de réaliser un firewall performant. Comme l'écriture des restrictions ICMP, l'écriture peut être réalisé au moins de 2 façons différentes comme :

Mon fichier de conf :

```
$IPTABLES -A ICMPINBOUND -i $EXTIF -p icmp --icmp-type echo-reply -m limit
--limit 1/s -j ACCEPT
```

Iptables interprete la ligne comme çà :

```
$IPTABLES -A ICMPINBOUND -i ppp0 -p icmp -m icmp --icmp-type 0 -m limit --limit
1/sec -j ACCEPT
```

i*p_list interprete comme ceci :*

```
1 0 0 ACCEPT icmp -- ppp0 * 0.0.0.0/0 0.0.0.0/0 icmp type 0 limit: avg 1/sec burst 5
```

Donc ces 3 lignes (surtout les 2ere) représente la même règle. Le cas inverse est aussi présent, il est possible d'utiliser 21 ou ftp, Iptables comprend les 2.

Ensuite, l'interprétation des règles en utilisant le "PREROUTING" ou en utilisant les filtres INPUT/FORWARD et/ou OUTPUT. Voilà pourquoi ma configuration ne peut être qu'un exemple, de plus j'utilise le logiciel Lopster, pas d'IRC, le port 21 (ftp) est ouvert (bien que non testé)... etc.

Je conseille l'emploi de « /etc.rc.d/init.d/./iptables restart » au lieu de lancer « /etc/sysconfdir/./iptables », bien que ceci fonctionne souvent, au bout de 3 à 4 fois le kernel affiche des registres (mais le système fonctionne !!!).

Compilation et installation d'IPTABLES 1.2.5 et noyau 2.4.17:

La compilation doit assurer l'installation d'Iptables dans le répertoire /usr/sbin, mais il est aussi possible de « patcher » le noyau pour y inclure des modules supplémentaires. Attention, le noyau contient déjà certains modules en version STABLE ... Chose que ne fournit pas forcément les modules du répertoire patch-o-matic d'Ipatbles. Le faite d'apporter un module nouveau dans le noyau peut s'avérer une grande aventure...

Après plusieurs essais, il est soit disant possible d'arriver au résultat final en passant directement à l'étape 3, personnellement çà été un échec (...) . Voici donc les 3 étapes qui fonctionnent, comme d'ailleur indiqué dans le fichier INSTALL d'Iptables.

Pré-requis : Faire un lien du répertoire de votre noyau vers linux, ce qui donne : « ln -sf /usr/sr/linux-2.4.17 linux »- Ce qui simplifie le travail après ...

1. gmake (ou make) si le lien ci-dessus n'est pas présent ajouté KERNEL_DIR=/usr/sr/linux-2.4.17

2. gmake install (ou make install) + KERN..(ev)

3. gmake patch-o-matic +KERN...

Choix des modules (version 1.2.5) : Certains modules sont très interessants, par contre pour les tester cela demande un « certain temps » en plus du risque d'échec. C'est pour cette raison que seul le module « **string.patch »** a été testé.

/iptables-1.2.5/patch-o-matic/base/IPV4OPTSSTRIP.patch
/iptables-1.2.5/patch-o-matic/base/NETLINK.patch
/iptables-1.2.5/patch-o-matic/base/NETMAP.patch
/iptables-1.2.5/patch-o-matic/base/SAME.patch
/iptables-1.2.5/patch-o-matic/base/TTL.patch
/iptables-1.2.5/patch-o-matic/base/ah-esp.patch
/iptables-1.2.5/patch-o-matic/base/ftos.patch
/iptables-1.2.5/patch-o-matic/base/iplimit.patch
/iptables-1.2.5/patch-o-matic/base/ipv4options.patch
/iptables-1.2.5/patch-o-matic/base/mport.patch
/iptables-1.2.5/patch-o-matic/base/nth.patch
/iptables-1.2.5/patch-o-matic/base/pkttype.patch
/iptables-1.2.5/patch-o-matic/base/pool.patch

/iptables-1.2.5/patch-o-matic/base/quota.patch
/iptables-1.2.5/patch-o-matic/base/random.patch
/iptables-1.2.5/patch-o-matic/base/realm.patch
/iptables-1.2.5/patch-o-matic/base/time.patch
/iptables-1.2.5/patch-o-matic/base/ulog.patch
/iptables-1.2.5/patch-o-matic/extra/helper.patch
/iptables-1.2.5/patch-o-matic/extra/ctnetlink.patch
/iptables-1.2.5/patch-o-matic/extra/dropped-table.patch
/iptables-1.2.5/patch-o-matic/extra/eggdrop-conntrack.patch
/iptables-1.2.5/patch-o-matic/extra/ftp-fxp.patch
/iptables-1.2.5/patch-o-matic/extra/pptp-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/extra/recent.patch
/iptables-1.2.5/patch-o-matic/extra/record-rpc.patch
/iptables-1.2.5/patch-o-matic/extra/string.patch
/iptables-1.2.5/patch-o-matic/extra/talk-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/extra/tcp-window-tracking.patch
/iptables-1.2.5/patch-o-matic/extra/tftp.patch
/iptables-1.2.5/patch-o-matic/newnat/0-newnat5.patch
/iptables-1.2.5/patch-o-matic/newnat/h323-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/newnat/irc-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/newnat/talk-conntrack-nat.patch
/iptables-1.2.5/patch-o-matic/pending/mangle5hooks.patch
/iptables-1.2.5/patch-o-matic/submitted/2.4.4.patch
/iptables-1.2.5/patch-o-matic/submitted/TOS-oops-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/conntrack-errormsg.patch
/iptables-1.2.5/patch-o-matic/submitted/ip6tables-export-symbols.patch
/iptables-1.2.5/patch-o-matic/submitted/ip_nat_irc-srcaddr-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/ipchains-redirect-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_LOG.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_MIRROR-ttl.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_REJECT-checkentry.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_mac-fix.patch
/iptables-1.2.5/patch-o-matic/submitted/ipt_unclean-ecn.patch
/iptables-1.2.5/patch-o-matic/submitted/module-license.patch
/iptables-1.2.5/patch-o-matic/submitted/netlink-tcpdiag.patch
/iptables-1.2.5/patch-o-matic/submitted/sackperm.patch
/iptables-1.2.5/patch-o-matic/submitted/skb_clone_copy.patch
/iptables-1.2.5/patch-o-matic/submitted/tcp-MSS.patch

Ensuite devrez s’afficher ceci (fin 3eme étape):

Examining kernel in /usr/src/your_kernel_source_directory

Welcome to Rusty’s Patch-o-matic!
Each patch is a new feature: many have minimal impact, some do not.
Almost every one has bugs, so I don’t recommend applying them all!
---------- (output clipped for brevity)
- At some point during the patching, you will be faced with the following:
Welcome to Rusty’s Patch-o-matic! Each patch is a new feature: many have minimal impact, some do not.
Almost every one has bugs, so I don’t recommend applying them all!

Already applied: 2.4.1 2.4.4 Testing... dropped-table.patch NOT APPLIED (1 missing files)
The dropped-table patch:
Author: Rusty Russell Status: Beta, redesign underway, applies now to 2.4.4-final

This patch adds a ‘drop’ table to iptables, adding a

```
dropped
by the NAT or routing code (among others) will traverse this table,
allowing them to be logged.

THIS PATCH WILL BREAK OTHER PATCHES
(irc-conntrack-nat,talk,NETMAP,SAME,...) Do you want to apply this
patch [N/y/t/f/q/?]
```

Controle du firewall - Règles et scripts :

Personnellement j'utilise « ip_list »(sript), très pratique il affiche tout se dont il est nécessaire. (voir ci-dessous)

Le listing ci-dessous est la même représentation version ip_list que le script /etc/sysconfdir/iptables.

Ce qui est très pratique, est la répresentation du comptage des paquets selon chaques règles, ceci permet de connaitre très précisement ce qui se passe, et éventuellement d'agir, en plus du fichier de logs.

Doc_NetFilter/Ressources/ip_list_listing.html

Le fichier de lancement :

(etc/rc.d/init.d + liens /etc/rc.3 ...) : iptables (chmod 700)(root)

```
#!/bin/sh
#
# chkconfig: 2345 08 92
#

. /etc/init.d/functions

IPTABLES_CONFIG="/etc/sysconfig/iptables"

# check we have the iptables executable
if [ ! -x /sbin/iptables ]; then
exit 0
fi

start() {
# don't do squat if we don't have the script
if [ -f $IPTABLES_CONFIG ]; then
action "Applying iptables firewall rules:" $IPTABLES_CONFIG
echo
touch /var/lock/subsys/iptables
fi
}

stop() {
action "Flushing all chains:" iptables -F
```

```
echo $"Resetting built-in chains to the default ACCEPT policy:"
iptables -P INPUT ACCEPT && \
iptables -P FORWARD ACCEPT && \
iptables -P OUTPUT ACCEPT && \
success "Resetting built-in chains to the default ACCEPT policy" || \
failure "Resetting built-in chains to the default ACCEPT policy"
echo
rm -f /var/lock/subsys/iptables
}

case "$1" in
start)
start
;;

stop)
stop
;;

restart)
# "restart" is really just "start" as this isn't a daemon,
# and "start" clears any pre-defined rules anyway.
# This is really only here to make those who expect it happy
start
;;

status)
iptables --list
;;

panic)
echo $"Changing target policies to DROP: "
iptables -P INPUT DROP && \
iptables -P FORWARD DROP && \
iptables -P OUTPUT DROP && \
success "Changing target policies to DROP" || \
failure "Changing target policies to DROP"
echo
action "Flushing all chains:" iptables -F INPUT && iptables -F
FORWARD && iptables -F OUTPUT
action "Removing user defined chains:" iptables -X
;;

*)
echo "Usage: $0 {start|stop|restart|status|panic}"
exit 1
esac
exit 0
```

J'utilise aussi mais rarement la commande iptable-save (fichier exemple), existe aussi iptable-restore ...

Ma configuration ethernet :

On s'approche de plus en plus, il faut configurer les cartes ethernet, rien de bien spécial.

```
[root@externe sysconfig]# (la 4eme = ???)
1: lo: < LOOPBACK ,UP> mtu 16436 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 brd 127.255.255.255 scope host lo

2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc
pfifo_fast qlen 100

link/ether 00:50:fc:50:5f:07 brd ff:ff:ff:ff:ff:ff

inet 10.0.0.1/8 brd 10.255.255.255 scope global eth0

3: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc
pfifo_fast qlen 100

link/ether 00:50:fc:54:48:45 brd ff:ff:ff:ff:ff:ff

inet 192.168.0.1/24 brd 192.168.0.255 scope global
eth1

5: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP> mtu 1492
qdisc pfifo_fast qlen 3

link/ppp
inet 213.44.244.157 peer 213.44.255.254/32 scope
global ppp0
```

Donc 2 cartes ethernet, l'une en 10.0.0.1 spécialement dédié à la connection vers Internet et donc au modem dont l'adresse statique est 10.0.0.138 (Modèle ECI ETHernet).

Le routage :

```
[root@externe sysconfig]#ip route
213.44.255.254 dev ppp0 proto kernel scope link src
213.44.244.157
192.168.0.0/24 dev eth1 scope link
10.0.0.0/8 dev eth0 proto kernel scope link src
10.0.0.1
127.0.0.0/8 dev lo scope link
default via 213.44.255.254 dev ppp0
```

Pour les clients, le routage doit se faire vers le firewall (ne pas oublier les navigateurs).

Rien de bien spécial à ce niveau...

Installation des packages (RPM) et logiciels:

Il est de bon ton, de ne garder que le strict minimum afin de réduire au maximum les problèmes de sécurité (intrusion et prise de contrôle).

vers l'administrateur du Firewall (30 cm à coté, mais bon ...) (cryptage des données possibles via GnuPGP ou SSL mais plus lourd).

OpenSSH pour les connexions sécurisées entres PC.

Le noyau à été recompilé, basé sur kernel 2.4.17.

L'IDS Snort scrupte l'interface eth1 (vers VPN).

Tripwire est installé, pour contrôler le cas échéant une intrusion (test d'intégrité sur les fichiers).

Le firewall est équipé d'un arrêt contrôlé (mode esclave), en fonction de l'état d'un onduleur (UPS).

L'horloge interne est mise à jour toutes les heures via NTP.

Le système de fichier est journalisée (EXT3).

Un serveur Apache (1.2.3)

Le logiciel "ntop" très bien pour la représentation des paquets/connections ...etc, mais refuse de se compiler pour une raison inconnue sur le firewall, il tourne sur une poste client, c'est déjà pas mal.

Un morceau de la distribution "clark connect" est installé, il est chargé de représenter les débits ppp0/eth0/1+la charge CPU/Mémoire ...etc

Tout se petit monde représente pas mal de place (+- 450 M) .

Pour mémoire : "rpm -qa" liste les packages installés, "rpm -e xxxxx.rpm" désinstalle un package.

Problème rencontrée : Malgré une mise à jour (rpm --rebuilddb), certains packages restent dans le fichier des packages installés. Par contre toute interrogation d'un tel package renvoit "paquet non installé".Je ne peux donc joindre un listing de mes packages installés ... Le prochain sera sous Debian !!!

Comment contrôler les règles - le trafic - les rejets/drops.

Pour contrôler les règles:

L'utilisation d'un logiciel comme gftp... pour le test FTP suffit, pour les autres mêmes procédures. Attention, il est nécessaire de s'occuper en premier à mon avis du port 53 (résolution DNS), le 22 (SSH) est aussi à prendre en compte.

Lancer la commande “cat /var/log/messages”, et là, libre à vous d’inclure une règle là ou il faut pour auroriser ou tout simplement laissez tel quel dans le cas d’un REJET/DROP volontaire.

Pour contrôler le trafic/connections :

Sur un poste client (qui profite pleinement du NET):

```
[root@serveur laurent]# netstat -anp
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:515 0.0.0.0:* LISTEN 2771/lpd Waiting
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 2592/portmap
tcp 0 0 0.0.0.0:6000 0.0.0.0:* LISTEN 3091/X
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 2812/httpd
tcp 0 0 0.0.0.0:32784 0.0.0.0:* LISTEN 2620/rpc.statd
tcp 0 0 127.0.0.1:32786 0.0.0.0:* LISTEN 2749/xinetd
tcp 0 0 0.0.0.0:21 0.0.0.0:* LISTEN 2749/xinetd
tcp 0 0 192.168.0.10:53 0.0.0.0:* LISTEN 2542/named
tcp 0 0 127.0.0.1:53 0.0.0.0:* LISTEN 2542/named
tcp 0 0 0.0.0.0:23 0.0.0.0:* LISTEN 2749/xinetd
tcp 0 0 0.0.0.0:6680 0.0.0.0:* LISTEN 3140/lopster
tcp 0 0 127.0.0.1:953 0.0.0.0:* LISTEN 2542/named
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN 902/master
tcp 0 0 0.0.0.0:443 0.0.0.0:* LISTEN 2812/httpd
tcp 0 0 192.168.0.10:6680 80.0.12.150:1565 ESTABLISHED 3140/lopster
tcp 0 0 192.168.0.10:33160 195.36.162.10:110 TIME_WAIT -
tcp 0 1 192.168.0.10:33158 194.208.138.138:6697 SYN_SENT 3140/lopster
tcp 0 1 192.168.0.10:33157 80.129.78.229:6681 SYN_SENT 3140/lopster
tcp 0 108 192.168.0.10:32816 213.93.78.164:8888 ESTABLISHED 3140/lopster
tcp 0 0 192.168.0.10:32840 192.168.0.1:22 ESTABLISHED 3246/ssh
udp 0 0 0.0.0.0:32770 0.0.0.0:* 2542/named
udp 0 0 0.0.0.0:32771 0.0.0.0:* 2620/rpc.statd
udp 0 0 0.0.0.0:676 0.0.0.0:* 2620/rpc.statd
udp 0 0 192.168.0.10:53 0.0.0.0:* 2542/named
udp 0 0 127.0.0.1:53 0.0.0.0:* 2542/named
udp 0 0 0.0.0.0:111 0.0.0.0:* 2592/portmap
```

Et sur le firewall :

```
[root@externe tmp]# netstat -anp
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name
tcp 0 0 0.0.0.0:6666 0.0.0.0:* LISTEN 965/apcd
tcp 0 0 0.0.0.0:8080 0.0.0.0:* LISTEN 1122/(squid)
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 982/sshd
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN 646/master
tcp 0 0 192.168.0.1:22 192.168.0.10:32840 ESTABLISHED 2929/sshd
udp 0 0 0.0.0.0:32768 0.0.0.0:* 1122/(squid)
udp 0 0 0.0.0.0:514 0.0.0.0:* 544/syslogd
udp 0 0 0.0.0.0:8080 0.0.0.0:* 1122/(squid)
udp 0 0 0.0.0.0:3401 0.0.0.0:* 1122/(squid)
udp 0 0 0.0.0.0:4827 0.0.0.0:* 1122/(squid)
```

Il est donc possible les adresses et ports de toutes les connections, et de là les règles peuvent tomber ...

Termes utilisés :

Packet filter Programme qui examine les entêtes des paquets

Connection tracking Table en mémoire pour la surveillance de connections (SOURCE/DESTINATION/ IP ADRESSE/PORT(SOURCE +DESTINATION)/PROTOCOL/ETAT/TIMEAOUTS)

Stateful Firewall capable d'éffectuer de faire du « connection tracking ».

NAT (Network Address Translation)- Modification au niveau des entêtes IP, comprend 2 sous-ensemble (SNAT)->modification de la source et (DNAT)-> modification du destinataire.

Netfilter Nom du système de filtrage dans le noyau linux 2.4 et supérieur, IPTABLE utilise les possibilités et modules du noyau pour appliquer les règles

Filter tables Table qui contient les chaine INPUT/FORWARD/OUTPUT pour traitement des paquets en : DROP/REJECT/ACCEPT/QUEUE/LOG...

NAT table Table pour utiliser le SNAT/DNAT comporte les chaines PREROUTING et POSTROUTING.

Mangle table Très particulièrement, modification de paquets(TOS...)

PREROUTING and OUTPUT.

Règle Criteres de traitement pour les paquets/interfaces.

Chaine Un ensemble de règles.

User Defined Chain Chaine utilisateur qui peut devenir une cible dans une chaine ou autre chaine.

DNAT Destination Network Address Translation De la table NATet chaine POSTROUTING, pour modification de l'adresse destination (IP).

SNAT (Source Network Address Translation) De la table NATet chaine PREROUTING, pour modification de l'adresse source (IP).

performed to cause the appearance that network traffic originates from a translating host instead of from the actual originating host.

dynamique comme DHCP(cable) ou modem via une FAI.

Les commandes IPTABLES :

iptables -A Ajoute une nouvelle règle.

iptables -D Efface une règle.

iptables -F

iptables -I

iptables -L

iptables -N

iptables -P

iptables -X

Diverses commandes utiles :

netstat -i (visualise le trafic/interface)

netstat -ta (visualise les connexions)

netstat -rn (visualise la table de routage)

Tests :

Via internet :

Possible gratuitement . (Voir en fin de document pour d'autres liens).

Voici un résultat fait grace à : http://scan.sygatetech.com/stealthscanf.html

Your system ports are now being scanned and the results will be returned shortly...

Ideally your status should be "Blocked." This indicates that your ports are not only closed, but they are completely hidden (stealthed) to attackers.

Service	Ports	Status	Additional Information
FTP DATA	20	BLOCKED	Used by FTP for data transmission in Passive mode.
FTP	21	**CLOSED**	File Transfer Protocol is used to transfer files between computers. A misconfigured FTP server can allow an attacker to transfer files, trojan horses, and virus programs at will.
SSH	22	BLOCKED	Secure Shell, a encrypted type of telnet. If misconfigured it can allow for brute-force attacks on your administration account.
TELNET	23	BLOCKED	Telnet is used to remotely create a shell (dos prompt), this can allow an attacker to control your system as if he was sitting in front of it.
SMTP	25	BLOCKED	SMTP is used to send email acrost the internet. This allows an attacker to verify user accounts on your system, send anonymous (spam) email, or even access files on your hard drive.
DNS	53	BLOCKED	Domain Name Services are used to resolve host names to IP addresses.
DDC	59	BLOCKED	Used mainly by file transfer and chat programs.
FINGER	79	BLOCKED	Finger offers information about who is currently logged in to your computer.
WEB	80	BLOCKED	HTTP web services publish web pages. A misconfigured web server can not only offer an attacker needed information about his target, but it can allow for various security breaches.
POP3	110	BLOCKED	Post Office Protocol is used to receive email. It can be used by attackers to create fake email addresses, execute programs, and even intercept your private email.
IDENT	113	BLOCKED	Ident is often used for IRC (chat), but also provides information about your system and who is using it.
NetBIOS	139	BLOCKED	NetBios is used to share files through your Network Neighborhood. If you are connected to the internet with this open, you could be sharing your whole hard drive with the world! This is a very dangerous port to have open.
HTTPS	443	BLOCKED	Secure Web Servers are often used by banks and online vendors.

Service	Ports	Status	Additional Information
Server Message Block	445	BLOCKED	In Windows 2000, Microsoft added the possibility to run SMB directly over TCP/IP, without the extra layer of NBT.
SOCKS PROXY	1080	BLOCKED	Socks Proxy is an internet proxy service, many IRC servers will not allow you to log in if you are running an unsecured socks proxy.
WEB PROXY	8080	BLOCKED	HTTP Web Proxy allows other people to bounce their web browser off of your computer to fake their real IP address to web servers.

Comme il est possible de le voir tout les ports sont dans l'état BLOQUE, dont sous contrôle, exception du port 21 (FTP) dans l'état en attente (fermé cas sans connection, active).

Par logiciels :

Les 3 logiciels ci-dessous (ils en existent surement d'autres...), sont parmi des plus connus. NMAP est un scanner pur, il peut détecter les OS. Pour HPING2 et RAIN, ceux sont 2 produits assez similaires, très puissants par contre tout est « à la main ».

NMAP http://www.nmap.org

HPING http://www.hping.org/

RAIN http://freshmeat.net/projects/rain/?topic_id=87%2C73%2C43%2C74%2C253

http://www.tenebrous.com/rain

A citer aussi : ntop

Excellent logiciel qui capture les paquets et permet de visualiser le tout selon de nombreux critères. (stats...)

Conclusions :

- J’espère que que grace aux documents joints et à ce compléments d’informations vous allez pouvoir implémenter un firewall pour chez vous, comme pour tout autre mission, y compris professionnel.
- Soyez patient, la mise au point demande du temps ...

Remerciements :

- A la communauté du libre pour toutes les documentations et conseils que l’on peut trouver sur le NET.

Adresses utiles :

TOS : http://www.linuxdoc.org/LDP/nag2/x-087-2-firewall.tos.manipulation.html

Netfilter Related

http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html
(step-by-step tutorial on setting up a firewall based on Iptables/Netfilter)

http://www.knowplace.org/netfilter/

http://www.linuxguruz.org/iptables/

http://monmotha.mplug.org/

http://www.ecst.csuchico.edu/~dranch/LINUX/ipmasq-beta/c-html/index.html

http://www.prismnet.com/~aef (St. Elmo’s Firewall - Entirely for Netfilter/Linux)

http://hogwash.sourceforge.net (IPTables sub-project going on)

http://project.honeynet.org

http://www.cert.org/nav/index_red.html

http://www.incidents.org

http://www.ecst.csuchico.edu/~dranch/LINUX/TrinityOS/cHTML/TrinityOS-c.html

http://www.chkrootkit.org

http://netfilter.gnumonks.org/documentation/

Adresses pour tester depuis le net: (gratuit et sans téléchargement)

http://scan.sygatetech.com/stealthscan.html

http://www.dslreports.com/scan

Autres :

Firewall-HOWTO
Firewall Policy Guide (ICSA)
Internet Firewalls Frequently Asked Questions
IPCHAINS-HOWTO
Network Ingress Filtering (RFC 2827)
Help Defeat Denial of Service Attacks: Step-by-Step (SANS)
Packet Filtering for Firewall Systems

Service Port Numbers & Log Info

Port Numbers (IANA)
FAQ: Firewalls: What am I seeing?
Commonly Probed Ports

Attack Information

Denial of Service Attacks
Email Bombing and Spamming
Smurf Attacks
Spoofed/Forged Email
TCP SYN Flooding and IP Spoofing Attacks
UDP Port Denial-of-Service Attack

Compromise Detection & Recovery

Intruder Detection Checklist
Recovering from a UNIX Root Compromise

UNIX Configuration

Eliminate SANS' Top 10 Security Threats
Passwd File Protection
Securing Internet Servers
Securing X Windows
Site Security Handbook (RFC 2196)
UNIX Configuration Guidelines

FTP

Anonymous FTP Configuration Guidelines
Anonymous FTP Abuses
FTP Port Command Security Hole

Web/CGI Scripts

CGI Script Security Holes
Managing Web Server Security

Security Information Sites

Apache Web Server Security Tips
BugTraq
CERT Coordination Center
CGI & Perl
CIAC - Computer Incident Advisory Capability

CIAC Security References
COAST Hotlist: Computer Security, Law & Privacy
COAST Hotlist: Internet Firewalls
COAST Security Archive
Dave Dittrich's Security Page

DShield.org (Shared Intrusion Detection Information)
HackerWacker
InfoSysSec (Information System Security Portal)
IP Masquerading Site
Lachlan Cranswick's Security Homepage

Lance Spitzner's Security Publications
Linux Administrator's Security Guide
Linux Security Resources
Matt's Unix Security Page

NIH: Computer Security Information
NIPC: National Infrastructure Protection Center
Packet Storm
Red Hat Security Page

Security Focus
Security Portal
Security Search Engine
WWW Security Resources

Firewall Tools

ipmenu - iptables configuration tool based on ncurses
Mason - interactive, dynamic iptables configuration tool

floppyfw - static router/firewall on a floppy
IP Filter Kernel Module
Ipfwadm Dotfile Module
Ipfwadm/Ipchains Home Site (X/OS)
Isinglass PPP Firewall
Linux Router Project
SINUS Firewall
SmoothWall Firewall
TIS Internet Firewall Toolkit
Trinux Linux Security Toolkit

NAT/Masquerading Tools

IP Masquerade Resource
ipmasqadm (kernelnotes.org)
ipmasqadm (RedHat)
Linux IP Masquerade HOWTO
Masq Apps
Port Forwarding

iptables Logging Tools

fireparse
pdumpq
ulogd

Static IP Networking Tools

IP Sub-Networking Mini-Howto
Bridge + Firewall + DSL Mini-HOWTO
Linux Bridge+Firewall Mini-HOWTO
Bridge Filter Kernel Patch
Bridge Configuration Tool

Software Sites

Security Related Software

COAST Tools & Projects
Freefire Project
Insecure.Org (Nmap)
IPSEC: Linux FreeS/WAN Project

LEAF (Linux Embedded Appliance Firewall)
logcheck
Nmap
Saint
Secure-Me (Automated Security Scanning)

SOCKS Home Site
SSH Home Site
SSL (SSLeay and SSLapps)
SunSite Security Area
TripWire

General Software

Applications & Utilities
Linux Applications
Linux Software Encyclopedia
Netscape (ftp)
Sendmail.Org
Squid Object Cache
Sunet (ftp)
SunSite (www)
XNTP/NTP Time Synchronization Server
Tucows Linux

Standards Sites

IANA - Internet Assigned Numbers Authority
IANA - IPv4 Address Space
IANA - Port Assignments
IETF - Internet Engineering Task Force
RFCs - Request for Comments

Whois Databases

ARIN whois
APNIC whois
RIPE whois