21 21 à 98 libre pour défaut privatifs du constructeur,
99 erreur non répertoriée, autre ou inconnue.

ERI : Ce paramètre est obligatoire.
En classe 1: ERI signale la présence d'une erreur majeure, mineure ou test par un "1". ERI repasse à 0 lorsque toutes les erreurs sont résorbées. Ces trois types d'erreurs agissent de manière unique sur ERI, elles sont néanmoins mémorisées séparément car elles agissent différemment sur le status temps réel.
En classe 2 et 3, il signale le numéro de la dernière (la plus récente) erreur majeure, mineure ou test survenue accompagnée de l'adresse du module concerné. ERI doit pouvoir contenir les défauts sur tous les modules ou sous-modules qui peuvent être détectés en erreur. Son contenant peut être assimilé à un tableau dont une dimension est l'adresse du module défectueux et l'autre la gravité (majeure, mineure ou test). Son contenu est vide s'il n'y a pas d'erreur ou contient l'état de défaut v. (de fsgee/am/v). Il ne peut y avoir plusieurs erreurs majeures à la fois sur un même module. Il peut par contre y avoir à la fois une erreur majeure, une erreur mineure et une erreur test.
ERI signale l'erreur par une amc lorsque l'erreur est générale au module

Code des erreurs ERI=fsgee

f	Code de fonction	"K" (caméra)
S	Code de sous-famille de matériel	(réservé constructeur)
g	Gravité de l'erreur	0 = Erreurs majeures 5 = Erreurs mineures 8 = Commandes TST

Valeur particulière : fsgee ="**TS8ee**" pour une erreur provoquée par une commande TST
Les erreurs majeures conduisent à l'arrêt de tout mouvement et lèvent les bits 3 et 5 du stR0.
Les erreurs mineures et test lèvent le bit 5 du stR0.

Liste des erreurs

g	Ee	Description	Raz par :
0	00	Pas d'erreur	
0	01	Erreur indéterminée	?
0	02	Défaut de communication PIC/Caméra	Communication correcte
0	03	Erreur de Checksum. Le CKS calculé du premier niveau de ST est différent du CKS mémorisé	CKS Correct
0	04	Défaut mémoire des configurations, du status.	Reconfiguration
0	05	Défaut de scrutation par le PC (SPC)	Commande de lecture ou d'écriture sur port quelconque
0	07	Défaut détecté par un algorithme de sécurité interne.	Commande d'écriture acquittée ne déclenchant pas l'algorithme
0	09	Erreur sur un module fonctionnel.	Défaut résorbé
0	99	Réservé constructeur	
5	00	Pas d'erreur mineure	
5	01	Erreur indéterminée	?
5	99	Réservé constructeur	
8	01	Erreur indéterminée	
8	99	Réservé constructeur	

L'indicateur optionnel v peut être restitué pour signaler l'état de défaut du module. Pour un module il pourra prendre les valeurs suivantes:

v	Modules mécaniques	v	
0	Indéterminé	0	
1	Module toujours en mouvement	6	
2	Module arrêté en position intermédiaire	7	
3	Module sur une position incorrecte	8	

EVT : Ce paramètre restitue l'étiquette du dernier paramètre du ST ayant été modifié, suivie de l'horodate de l'événement : EVT=evt:jj/mm/aa hh:mm:ss. Les seuls événements pris en compte sont les suivants :
, BTR, CKS, CTL, EDF, ERI, GAR, GAT, INI, RST et TRM
EVT n'est modifié que si la valeur d'un de ces paramètres est modifiée.

Ex : Lorsqu'une ERI située en milieu de pile (qui n'est pas la plus récente) est supprimée le paramètre ERI du status reste inchangé, il n'y a donc pas modification de EVT.
GAR : Ce paramètre est obligatoire. Il comptabilise les occurrences du chien de garde. C'est un champ de longueur fixe composé de 3 caractères numériques de 000 à 999. Après 999 le compteur repasse à 0.
GAT : Ce paramètre signale un défaut (ouverture) sur un accès physique aux équipement de terrain. Lorsqu'aucun accès n'est surveillé le paramètre est restitué sans argument **(_GAT=).**
GEN : Ce paramètre est obligatoire. Il permet d'identifier le constructeur et la génération matérielle du PIC c'est un champ de longueur fixe composé des 7 caractères suivants :

ccc	Identifiant du constructeur 3 caractères alpha numériques du jeu J3.
".K"	Caractères ASCII 2E16 et .... , séparateur et identifiant d'un PIC
g	Identifiant du PI : 0=prototype, 1=classe 1, 2=classe 2, 3=classe 3.
v	Version matériel, 1 caractère numérique de 0 à 9.

INI : Ce paramètre est obligatoire. Il comptabilise les initialisations réalisées par une commande INIT ou par une mise sous tension du PI. C'est un champ de longueur fixe composé de 3 caractères numériques de 000 à 999. Après 999 le compteur repasse à 0.
LOC : Ce paramètre est obligatoire. C'est un paramètre inscriptible destiné à contenir la localisation de l'équipement. C'est un champ de longueur variable composé de 0 à 14 caractères alphanumériques appartenants au jeu J3.
NST : Ce paramètre est obligatoire. C'est un paramètre inscriptible destiné à contenir un numéro d'équipement. C'est un champ de longueur fixe composé de 4 caractères numériques.
RST : Ce paramètre est obligatoire. Il comptabilise les initialisations manuelles (poussoir RESET). C'est un champ de longueur fixe composé de 3 caractères numériques de 000 à 255. Après 255 le compteur repasse à 0.
TRM : Ce paramètre n'est restitué qu'en classe 2 et 3. Il signale qu'un terminal est connecté sur un port asynchrone du PI (voir chapitre interface physique). La valeur est 0 si le port terminal est inoccupé ou non équipé et 1 si le terminal est branché sur le port terminal
VER : Ce paramètre permet d'identifier la version logicielle du PI. C'est un champ de longueur fixe composé de 3 caractères alphanumériques appartenants au jeu J3.

Exemple :
Q : **ST**
R : STATUS ADR=LNS BTR= CKS=2AF4 COD=ILN59.S EDF=0 ER1=00 ER2=02 ERI=1
EVT=TRM:24/09/97 13:12:10 GAR=003 GEN=SES.P30 INI=012 LOC=Le_Pré_Vert NST=0123
RST=22 TRM=0 VER=101

2.6 stR0 - Lecture du status temps réel par la commande KV

Cette commande permet de lire un statut temps réel sur un seul caractère indiquant l'état de l'équipement :

Fonction	Syntaxes valides de la question
Lecture du status temps réel	**KV**

La réponse à la commande de lecture KV est le status temps réel.

2.6.1 Status temps réel

Le status temps réel permet d'obtenir un compte rendu synthétique de toutes les erreurs propres au PIC en cours. La valeur normale est "à" (code <4/0>). Le détail des erreurs en cours peut être obtenu par la lecture du Status partiel de premier niveau (ST STR) ou du Status complet de premier niveau. Ces derniers contiennent la totalité des éléments ayant une incidence sur la valeur du stR0.
Certains bits du stR0 sont remis à 0 après la réponse à une commande ST, ce qui limite l'utilisation du stR0 dans un contexte multi-utilisateurs.

- Le bit 0 signale les problèmes d'alimentation en énergie externe. Il monte lorsque le paramètre EDF du status de premier niveau prend une valeur différente de 0 et retombe lorsqu'elle retourne à 0.
- Le bit 1 signale la modification d'un des paramètres INI, RST ou GAR du status de premier niveau. Il est remis à 0 par la lecture du status de premier niveau. La RAZ n'a lieu qu'après fourniture de la réponse.

- Le bit 2 signale la prise en main local de l'équipement. Il traduit la présence d'un terminal (présence du DTR sur une interface asynchrone), la prise en main locale d'un ou de plusieurs modules. Il est levé dès que la valeur d'un des paramètres TRM, CTL, ~~MOV~~ du status de premier niveau est différente de 0. Il retombe à 0 lorsque ces trois paramètres sont à 0.

- Le bit 3 est l'indicateur d'erreur majeur. Il est levé dès qu'apparaît une erreur majeure (code g=0). Il est remis à 0 lorsque toutes les erreurs majeures sont résorbées.
- Le bit 4 est levé lorsqu'une alerte requérant un acquittement a été émise. Il est remis à 0 à réception de l'acquittement par la lecture du status ou par une commande INIT.
- Le bit 5 signale l'apparition d'une nouvelle erreur majeure, mineure ou test. Il signale également la disparition d'une erreur majeure. Ceci permet lorsque plusieurs erreurs majeures sont en cours, d'être alerté de leur évolution. Le bit 5 monte également à l'ouverture d'une porte (paramètre GAT), ou sur un défaut des sources d'alimentation interne (paramètre BTR).

Remarque : le bit 5 n'est pas lié directement à l'évolution des paramètres ERI, GAT et BTR mais bien à l'évolution des erreurs correspondantes. En classe 1, par exemple, l'occurrence d'un nouveau défaut ne modifie pas ces paramètres. En classe 2 et 3 la résorption d'une erreur majeure n'a pas d'incidence sur l'ERI de 1er niveau si celle-ci n'était pas la dernière survenue. Dans ces 2 cas le bit 5 du stR0 est levé bien qu'il n'y ait pas eu modification du status de premier niveau.

- Le bit 6 n'est positionné à 0 que lorsque tous les autres bits sont à 1. Le code ASCII du status temps réel reste ainsi toujours ≤ à <4/0>. Le bit 6 n'est positionné à 0 que lorsque tous les autres bits sont à 1. Le code ASCII du status temps réel reste ainsi toujours ≤ à <4/0>.

Exemple

Lecture du status temps réel.

Q : **KV**

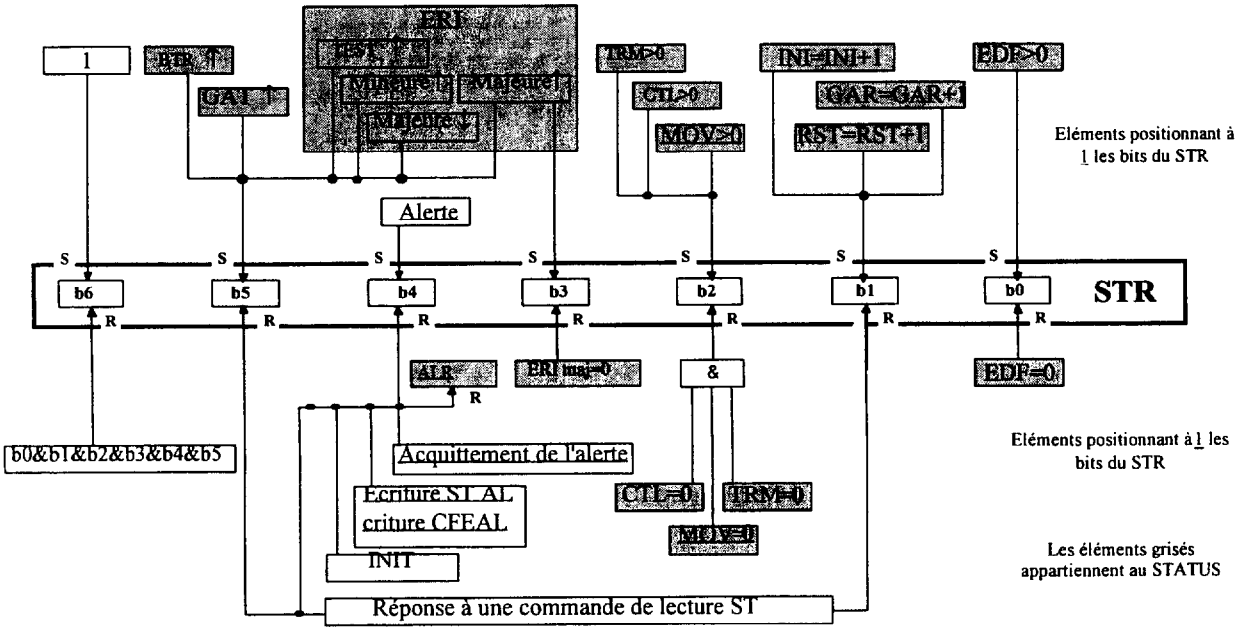
R : @

2.6.2 Récapitulatif du fonctionnement du stR0

Le tableau qui suit synthétise le fonctionnement du stR0.

Les éléments du haut font monter les bits du stR0, ceux du bas les baissent.

+1	le paramètre s'est incrémenté.
>0	il y a une erreur.
=0	il n'y a plus d'erreur
↑	une erreur est survenue (il y en avait peut-être déjà)
↓	une erreur est disparue (il y en a peut-être encore)



Les éléments barrés dans le schéma ci-dessus sont sans objet.

ANNEXE 9 : PROGRAMMATION RESEAU

Découper un numéro de réseau IP en sous-réseaux.

En découpant un réseau IP en deux sous-réseaux séparés, on a alors deux adresses de réseau et deux adresses de diffusion, augmentant le nombre d'adresses 'inutilisables' pour les interfaces (hôtes); créer 4 sous-réseaux crée huit adresses inutilisables. En fait, le plus petit sous-réseau utilisable est composé de 4 numéros IP:

- deux numéros IP d'interface - un pour l'interface du routeur sur ce réseau, et un pour l'unique hôte de ce réseau.
- un numéro de réseau.
- une adresse de diffusion.

En théorie, on peut découper son numéro de réseau IP en 2^n sous-réseaux de tailles égales.
(où n est le nombre de bits d'interface du numéro de réseau moins un).

Calcul du masque de sous-réseau et du numéro de réseau.

Le masque de réseau pour un réseau IP non découpé est simplement un "quadruplet pointé" dont tous les 'bits de réseau' du numéro de réseau sont positionnés à '1', et tous les bits d'interface à '0'.
Pour les trois classes de réseau IP, les masques de réseau sont: classe A (8 bits de réseau): 255.0.0.0
classe B (16 bits de réseau): 255.255.0.0
classe C (24 bits de réseau): 255.255.255.0

Pour mettre en oeuvre le découpage en sous-réseaux, on réserve un ou plusieurs bits parmi les bits d'interface, et on les interprète localement comme faisant partie des bits de réseau. Donc, pour diviser un numéro de réseau en deux sous-réseaux, on réservera un bit d'interface en positionnant à '1' le bit approprié dans le masque de réseau: le premier bit d'interface (pour un numéro de réseau 'normal').
Masque de réseau pour un réseau de classe C: 11111111.11111111.11111111.10000000 ou 255.255.255.128
Sous-réseaux possibles pour le numéro de réseau de classe C 192.168.1.0.

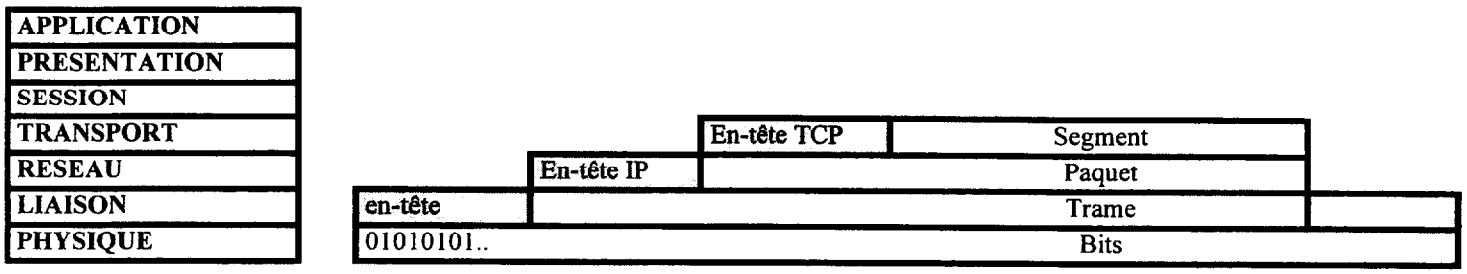
Nombre de sous-réseaux	Nbre d'hôtes par réseau	Masque de réseau	
2	126	255.255.255.128	(11111111.11111111.11111111.10000000)
4	62	255.255.255.192	(11111111.11111111.11111111.11000000)
8	30	255.255.255.224	(11111111.11111111.11111111.11100000)
16	14	255.255.255.240	(11111111.11111111.11111111.11110000)
32	6	255.255.255.248	(11111111.11111111.11111111.11111000)
64	2	255.255.255.252	(11111111.11111111.11111111.11111100)

Vous étant décidé sur le masque de réseau approprié, vous devez maintenant trouver quelles sont les différentes adresses de réseau et de diffusion - et l'intervalle de numéros IP pour chacun de ces réseaux. A nouveau, en ne considérant qu'un numéro de réseau IP de classe C et en ne listant que la partie finale (la partie d'interface), on a:

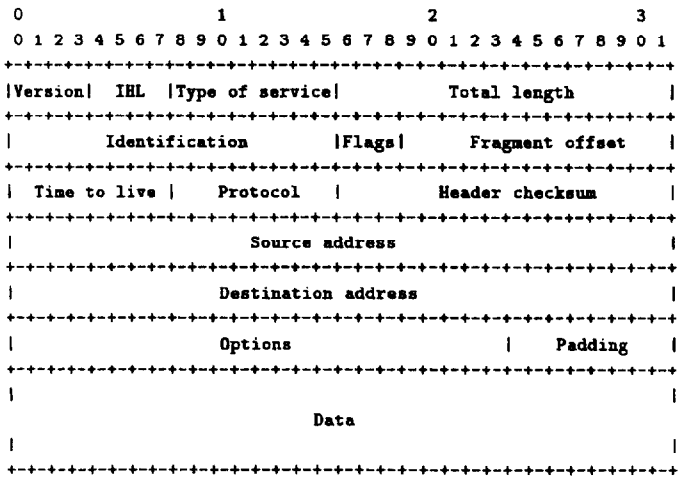
Masque de réseau	Sous-réseaux	Réseau	Diffusion	MinIP	MaxIP	Nbre d'hôtes	Nbre total d'hôtes
128	2	0	127	1	126	126	
		128	255	129	254	126	252
192	4	0	63	1	62	62	
		64	127	65	126	62	
		128	191	129	190	62	
		192	255	193	254	62	248
224	8	0	31	1	30	30	
		32	63	33	62	30	
		64	95	65	94	30	
		96	127	97	126	30	
		128	159	129	158	30	
		160	191	161	190	30	
		192	223	193	222	30	
		224	255	225	254	30	240

Les RFCs 760, 791 et 1812 préconisent de ne pas utiliser le premier et le dernier sous-réseau, et dans certaines installations, soit le dernier sous-réseau, soit le premier et le dernier sont indisponibles. L'utilisation de ces sous-réseaux dépend des protocoles de routages en usage sur le réseau et de l'implémentation IP sur les équipements de routage sur le réseau.

Rappel sur l'encapsulation des données

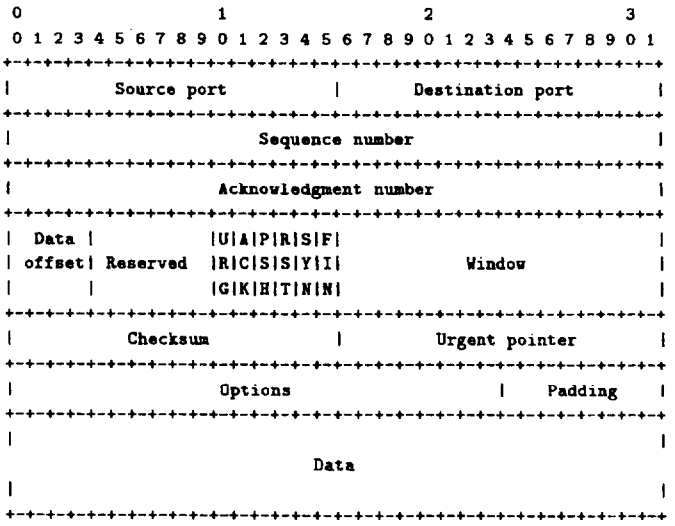


Structure des données IP:



Si le champ IHL (longueur en mot de 32 bits de l'entête) est égale à 5 (longueur minimale), il n'y a pas de champ option ni de champ pading (remplissage pour complément à 32 bits des options).

Structure des données TCP:



Extrait du fichier Services

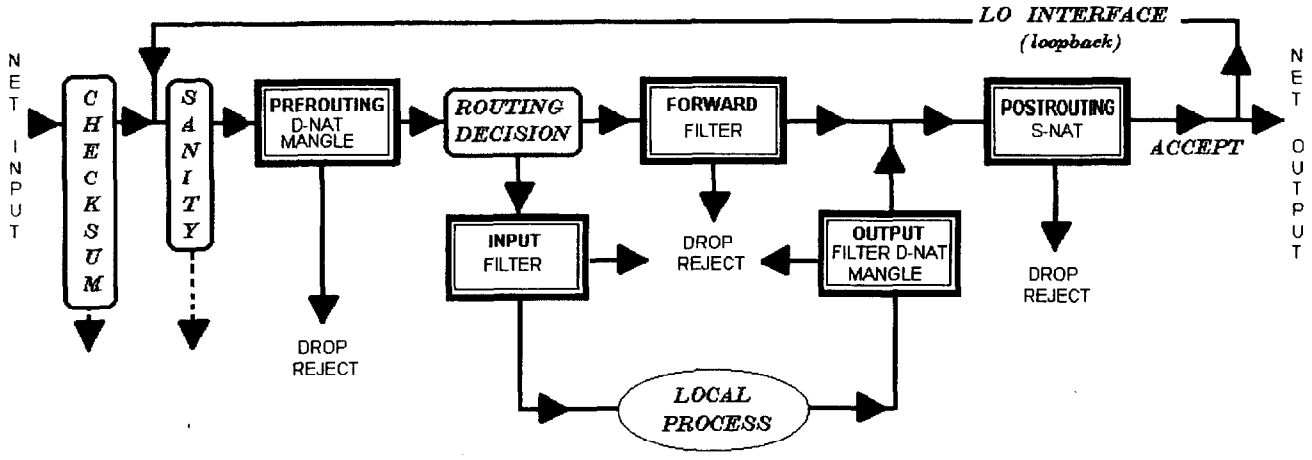
```
# Network services, Internet style
tcpmux          1/tcp     # TCP Port Service Multiplexer
tcpmux          1/udp     # TCP Port Service Multiplexer
compressnet     2/tcp           # Management Utility
compressnet     2/udp           # Management Utility
compressnet     3/tcp           # Compression Process
compressnet     3/udp           # Compression Process
rje             5/tcp           # Remote Job Entry
rje             5/udp           # Remote Job Entry
echo            7/tcp   Echo
echo            7/udp   Echo
discard         9/tcp   Discard sink null
discard         9/udp   Discard sink null

systat          11/tcp  users   # Active Users
systat          11/udp  users   # Active Users
daytime         13/tcp  Daytime # Daytime (RFC 867)
daytime         13/udp  Daytime # Daytime (RFC 867)
netstat         15/tcp          # official Unassigned
qotd            17/tcp  quote   # Quote of the Day
qotd            17/udp  quote   # Quote of the Day
msp             18/tcp          # Message Send Protocol
msp             18/udp          # Message Send Protocol
chargen         19/tcp  ttytst source  # Character Generator
chargen         19/udp  ttytst source  # Character Generator
ftp-data        20/tcp          # File Transfer[DefaultData]
ftp-data        20/udp          # File Transfer[DefaultData]
ftp             21/tcp          # File Transfer [Control]
fsp             21/udp          # official is FileTransfer,
                                          ftp use no udp
ssh             22/tcp          # SSH Remote Login Protocol
ssh             22/udp          # SSH Remote Login Protocol
telnet          23/tcp          # Telnet
telnet          23/udp          # Telnet
smtp            25/tcp  mail    # Simple Mail Transfer
smtp            25/udp  mail    # Simple Mail Transfer
nsw-fe          27/tcp          # NSW User System FE
nsw-fe          27/udp          # NSW User System FE
msg-icp         29/tcp          # MSG ICP
msg-icp         29/udp          # MSG ICP
msg-auth        31/tcp          # MSG Authentication
msg-auth        31/udp          # MSG Authentication
```

Principes de fonctionnement d'Iptables.

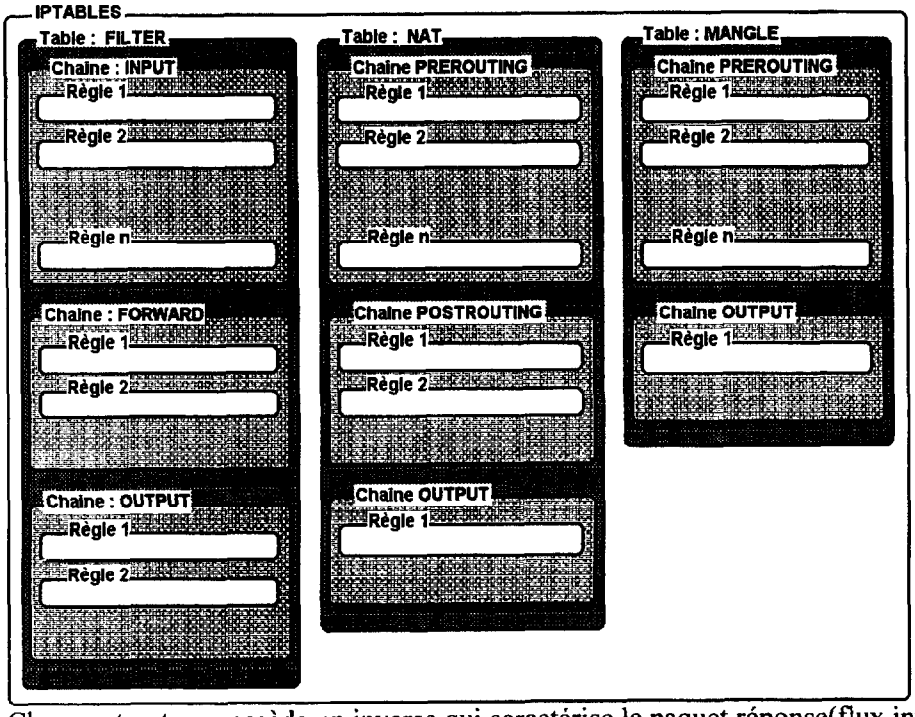
Le noyau de Linux peut filtrer des paquets en provenance et à destination d'un réseau y compris ceux utilisant l'interface loopback, en vérifiant à l'aide de tables de filtrage contenant des chaînes contenant elles-mêmes des règles (conditions) auquel des paquets satisferont ou ne satisferont pas. L'action à faire subir à ce paquet dépendra du résultat de ce test. Iptables permet de gérer, créer, modifier ces chaînes, ces règles contenues dans ces tables, et par conséquent de dicter le sort réservé à ces paquets. Toutes ces règles sont placées directement en mémoire. Afin de conserver ou de réutiliser cette configuration lors du redémarrage de la machine nous pouvons faire appel à deux utilitaires :

- Ipsave pour la sauvegarde dans un fichier
- Iprestore pour le chargement en mémoire.

Cheminement possible d'un paquet.



Les tables.



FILTER

La table FILTER est la table par défaut, elle contient des règles qui sont généralement utilisées pour réaliser le filtrage (d'où son nom). Tous les paquets sont généralement filtrés.

NAT

La table NAT (Network Address Translation) contient des règles qui généralement assureront les redirections d'adresses réseaux. Le code de nat assure également le suivi des connexions.

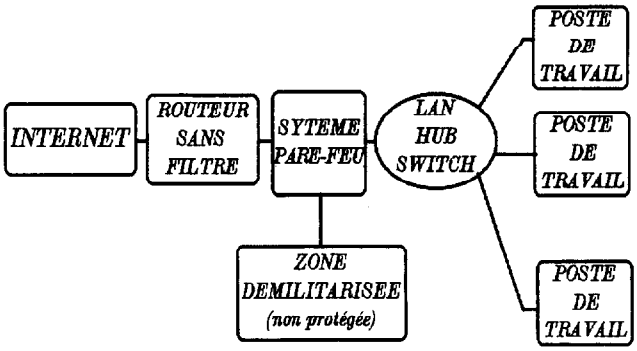
Le suivi de connexions (Connection Tracking) consiste en tout premier lieu à extraire d'un paquet des informations pour renseigner une structure caractérisant la connexion (adresse source destination, ports utilisés...). Cette structure contiendra des informations identiques pour tout paquet d'une même connexion ayant un flux de même sens.

Chaque structure possède un inverse qui caractérise le paquet réponse(flux inverse). Toutes ces informations sont stockées dans une liste utilisant différents pointeurs. Pour traiter un paquet, il suffira de comparer la structure extraite avec celle déjà contenue dans la liste pour savoir s'il s'agit d'une nouvelle connexion ou d'une connexion existante et le sens du flux. S'il s'agit d'une nouvelle connexion, les règles contenues dans la table Nat seront parcourus afin de décider du traitement à lui faire subir. S'il s'agit d'une connexion existante nous n'avons plus besoin de parcourir ces règles car nous connaissons déjà le traitement à lui soumettre(gain de temps).

MANGLE

La table MANGLE contient des règles qui généralement assureront la modification du contenu ou le marquage d'un paquet pour une exploitation ultérieure.

EXEMPLE DE REALISATION.



Configurations de base.

Initialisation des différentes tables à vide :

- ***iptables -F***
- ***iptables -t nat -F***
- ***iptables -t mangle -F***

Suppression de toutes les chaînes utilisateur :

- ***iptables -X***

Mise en place d'une police par défaut, permettant d'ignorer tous les paquets. (On interdit tout, on pourra ainsi par la suite n'autoriser que ceux que l'on souhaite) :

- ***iptables -P INPUT -j DROP***
- ***iptables -P OUTPUT -j DROP***
- ***iptables -P FORWARD -j DROP***

Configurations de la table Filter.

Création des chaînes utilisateurs

Afin de configurer la sécurité du système, nous utilisons le service de iptables avec des chaînes utilisateurs (correspondant aux différentes communications possibles). Pour une chaîne, la première partie du nom désigne l'origine et la seconde partie désigne la destination.

Nous obtenons ainsi la liste de chaîne suivante :

Inter-Serv, Inter-Dmz, Dmz-Serv, Dmz-Inter, Dmz-Prot, Prot-Serv, Prot-Dmz, Prot-Inter

Toutes ces chaînes sont créées grâce à la commande ***iptables -N Nom-Chaîne.***

Installation des règles dans INPUT : (contrôle des entrées).

On autorise l'utilisation de l'interface loopback :

iptables -A INPUT -i lo -j ACCEPT

On ignore le paquet d'une connexion invalide :

iptables -A INPUT -m state --state INVALID -j DROP

On accepte le paquet d'une connexion existante (dans les 2 sens du flux) :

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

On enregistre, dans un fichier input.log, une ligne préfixée par « INPUT TCP NO SYN » qui contient des renseignements relatifs à une tentative de connexion TCP douteuse (paquet ayant un statut NEW, mais ne possédant pas le flag SYN qui est une caractéristique d'une demande de connexion TCP). On limite les enregistrements à 3 par seconde pour éviter la saturation de notre système en cas d'attaques répétées.

iptables -A INPUT -p TCP -m state --state NEW ! --syn -m limit --limit 3/second -j LOG --log-prefix "INPUT TCP NO SYN:"

On ne donne pas suite à cette demande (on l'ignore):

iptables -A INPUT -p TCP -m state --state NEW ! --syn -j DROP

On trie les connexions entrantes en fonction des interfaces vers les chaînes utilisateurs appropriées :

iptables -A INPUT -i eth0 -j Inter-Serv
iptables -A INPUT -i eth1 -j Dmz-Serv
iptables -A INPUT -i eth2 -j Prot-Serv

On ignore les paquets qui parviennent jusqu'ici car ils n'ont pas satisfait à une des règles précédentes leur permettant de poursuivre leurs chemins :

iptables -A INPUT -j DROP

Installation des règles dans OUTPUT : (autorisation de toutes les sorties).

iptables -P OUTPUT ACCEPT

Configurations des tables Nat et Mangle.

Nous considèrerons que la configuration des tables nat et mangle s'effectue de manière similaire à celle de la table filter en utilisant les règles appropriées.

Paramètres de Iptables : *iptables v1.2.6 (extrait)*

Usage:

iptables -[ADC] chaîne règle [options] ou iptables -P chaîne cible [options]

DESCRIPTION

Iptables est utilisé pour mettre en place, maintenir, et inspecter les tables des règles de filtrage des paquets IP du noyau Linux. Plusieurs tables différentes peuvent être définies. Chaque table contient un nombre de chaînes pré-définies, et peut aussi contenir des chaînes définies par l'utilisateur. Chaque chaîne est une liste de règles auxquelles peuvent correspondre un ensemble de paquets. Chaque règle spécifie ce qui doit être fait avec un paquet qui correspond. Cela s'appelle une «cible», qui peut être un saut vers une chaîne définie par l'utilisateur dans la même table.

CIBLES

Une règle de pare-feu spécifie les critères pour un paquets, et une cible. Si le paquet ne correspond pas, la règle suivante de la chaîne est examinée ; si il correspond, la règle suivante est spécifiée par la valeur de la cible, qui peut être le nom d'une chaîne définie parl'utilisateur, ou l'une des valeurs spéciales suivantes : ACCEPT, DROP, QUEUE, ou RETURN.

ACCEPT signifie que le paquet est autorisé à passer.

DROP signifie que le paquet est laissé de coté.

TABLES

-t, --table

Cette option spécifie la table des paquets concordant sur laquelle la commande doit opérer. Les tables prédéfinies sont les suivantes : filter, nat, mangle.

OPTIONS

COMMANDES

-A, --append

Ajoute une ou plusieurs règles à la fin de la chaîne sélectionnée. Lorsque les noms de la source et/ou de la destination résolvent plus d'une adresse, une règle sera ajoutée pour chaque combinaison d'adresses possible.

-P, --policy

Met en place le comportement par défaut (policy) pour la chaîne de la cible fournie. Voir la section TARGETS pour connaître les cibles autorisées. Seules les chaînes pré-définies peuvent avoir des comportements par défaut, et ni les chaînes pré-définies ni les chaînes utilisateur ne peuvent être des cibles (policy targets).

PARAMÈTRES

La plupart des options peuvent être précédées par un ! pour inverser le sens de la correspondance

-p, --protocol [!] protocole

Protocole de la règle ou du paquet à vérifier. Le protocole spécifié est l'un des suivants tcp, udp, icmp, ou all, ou bien cela peut être une valeur numérique représentant l'un de ces protocoles ou un protocole différent. Un nom de protocole du fichier /etc/protocols est aussi autorisé. Un "!" avant le protocole inverse le test. Le nombre zéro est équivalent à all. Le protocole all correspond à tous les protocoles et c'est la valeur par défaut lorsque cette option est omise.

-j, --jump cible

Ceci spécifie la cible de la règle ; c'est-à-dire ce qu'il faut faire si le paquet correspond à la règle. La cible peut être une chaîne définie par l'utilisateur (autre que celle dans laquelle est cette règle), une des cibles pré-définies qui décide du destin du paquet immédiatement, ou une extension (voir EXTENSIONS ci-dessous). Si cette option est omise dans une règle, la correspondance du paquet avec la règle n'aura aucun effet sur le destin du paquet, mis à part le fait que les compteurs de la règle seront incrémentés.

-i, --in-interface [!] [nom]

Nom optionnel de l'interface qui reçoit les paquets (pour les paquets passant par les chaînes INPUT, FORWARD et PREROUTING). Lorsque l'argument "!" est utilisé avant le nom de l'interface, la signification est inversée. Si le nom de l'interface se termine par un "+", toutes les interfaces commençant par ce nom seront concernées. Si cette option est omise, le signe "+" est supposé, ce qui signifie que tous les noms d'interface réseau seront concernés.

-o, --out-interface [!] [nom]
Nom optionnel de l'interface qui envoie les paquets (pour les paquets passant par les chaînes FORWARD, OUTPUT et POSTROUTING). Lorsque l'argument "!" est utilisé avant le nom de l'interface, la signification est inversée. Si le nom de l'interface se termine par un "+", toutes les interfaces commençant par ce nom seront concernées. Si cette option est omise, le signe "+" est supposé, ce qui signifie que tous les noms d'interface réseau seront concernés.

tcp
Ces extensions sont chargées si `--protocol tcp' est spécifié. Elles fournissent les options suivantes :

--source-port [!] [port[:port]]
Spécification d'un port source ou d'un intervalle de port. Cela peut être le nom d'un service ou le numéro de port. Un intervalle inclusif utilisant le format suivant peut aussi être spécifié port:port. Si le premier port est omis, "0" est supposé; si le dernier est omis, "65535" est supposé. Si le second port est plus petit que le premier, il seront intervertis. L'option **--sport** est un alias de cette option.

--destination-port [!] [port[:port]]
Spécification d'un port de destination ou d'un intervalle de port. L'option **--dport** est un alias de cette option.

[!] --syn
Ne sélectionne que les paquets TCP dont le bit SYN est positionné et dont les bits ACK et FIN ne sont pas positionnés. De tels paquets sont utilisés pour les requêtes d'initiation de connexion TCP; par exemple si l'on bloque ce type de paquets entrants sur une interface, cela bloquera les connexions TCP entrantes, mais les connexions TCP sortantes ne seront pas affectées. Cela est équivalent à --tcp-flags SYN,RST,ACK SYN. Si le signe "!" précède le "--syn", le sens de l'option est inversé.

limit
Ce module laisse passer les paquets correspondants à un débit limité, en utilisant un filtre à jetons (token bucket filter) : il peut être utilisé conjointement avec la cible LOG afin de limiter la taille des logs. Une règle utilisant cette extension laissera passer les paquets correspondants jusqu'à ce que cette limite soit atteinte (à moins que le drapeau `!' soit utilisé).

--limit taux
Taux maximum de correspondance: spécifié par un nombre, avec un suffixe optionnel `/second', `/minute', `/hour', ou `/day'; la valeur par défaut est 3/hour.

state
Ce module, lorsqu'il est combiné avec "connection tracking", autorise l'accès à l'état du traçage de connexions pour ce paquet.

--state état
Ici état est une liste séparée par des virgules des états de connexions que l'on veut détecter. Les états possibles sont **INVALID** signifiant que le paquet n'est associé avec aucune connexion connue, **ESTABLISHED** signifiant que le paquet est associé avec une connexion qui a vu passer des paquets dans les deux sens, **NEW** signifiant que le paquet a initié une nouvelle connexion, ou sinon qu'il est associé avec une connexion qui n'a pas vu passer de paquets dans les deux sens, et **RELATED** signifiant que le paquet initie une nouvelle connexion, mais qu'il est associé avec une connexion existante, comme un transfert de données FTP, ou une erreur ICMP.

tos
Cette option traite les 8 bits du champ "type de service" dans l'en-tête IP (ceci inclut les bits de priorité).

--tos type_de_service
Cet argument est soit un nom standard, (faire iptables -m tos -h pour voir la liste), soit une valeur numérique à détecter.

LOG

Met en service la journalisation du noyau (kernel logging) pour les paquets correspondants. Lorsque cette option est positionnée pour une règle, le noyau Linux affichera certaines informations sur tous les paquets correspondants à cette règle (comme la plupart des champs de l'en-tête IP) par l'intermédiaire des journaux du noyau (que l'on peut lire avec dmesg ou syslogd(8)).

--log-level niveau

Niveau de journalisation (c'est un nombre ou alors voir syslog.conf(5)).

--log-prefix préfixe

Préfixe les messages de journalisation avec le préfixe spécifié; jusqu'à 29 lettres de long, c'est très utile pour distinguer les différents messages d'un fichier de log.

--log-tcp-sequence

Journalise le numéros de séquence TCP. Ceci peut être un risque pour la sécurité si les fichiers de log sont lisibles par les utilisateurs ordinaires.

--log-tcp-options

Option de journalisation de l'en-tête de paquets TCP.

--log-ip-options

Option de journalisation de l'en-tête de paquets IP.

TOS

Ceci est utilisé pour positionner le champ sur 8 bits du Type de Service dans l'en-tête IP. C'est valable uniquement lorsque vous utilisez la table mangle

--set-tos type_de_service

Vous pouvez utiliser les valeurs numériques des Types de Services, ou utiliser
Minimize-Delay 16 (0x10), Maximize-Throughput 8 (0x08), Maximize-Reliability 4 (0x04), Minimize-Cost 2 (0x02), Normal-Service 0 (0x00).

SNAT

Cette cible n'est valable que dans la table nat, dans la chaîne POSTROUTING

--to-source <adresse-ip>[-<adresse-ip>][:port-port]

qui peut spécifier une seule nouvelle adresse IP source, une série inclusive d'adresses IP, et optionnellement, une série de ports (qui est valide seulement si la règle spécifie aussi -p tcp ou -p udp). Si aucune plage de ports n'est spécifiée, les ports sources inférieurs à 512 seront translatés vers des ports inférieurs à 512 : ceux entre 512 et 1023 inclus seront translatés vers des ports inférieurs à 1024, et les autres seront translatés vers des ports supérieurs ou égaux à 1024. Lorsque cela est possible, aucune translation de ports n'est effectuée.

DNAT

Cette cible n'est valable que dans la table nat , dans les chaînes PREROUTING etOUTPUT, et dans les chaînes définies par l'utilisateur qui sont appelées par celles-ci. Elle spécifie que l'adresse de destination du paquet doit être modifiée (ainsi que tous les paquets à venir dans le cadre de cette connexion), et que les règles doivent cesser d'être examinées. Elle reçoit une option :

--to-destination <adresse-ip>[-<adresse-ip>][:port-port]

qui peut spécifier une seule adresse IP de destination, une série d'adresse IP inclusive, et optionnellement, une plage de ports (qui n'est valide que si la règle spécifie aussi -p tcp ou -p udp). Si aucune plage de port n'est spécifiée, le port de destination ne sera jamais modifié.

MASQUERADE

Cette cible n'est valable qu'avec la table nat, dans la chaîne POSTROUTING assignées dynamiquement (dialup) : si vous avez une adresse IP statique, vous devez utiliser la cible SNAT. Le masquerading revient à spécifier une translation (mapping) vers l'adresse IP de l'interface par laquelle le paquet va sortir, mais implique aussi que les connexions sont perdues lorsque l'interface tombe. C'est le comportement correct, lors du prochain établissement de liaison (dialup) il y a peu de chance d'obtenir la même adresse pour l'interface (et par conséquent les connexions déjà établie seront perdues quand même). Il y a une option :

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

qui spécifie une série de ports source à utiliser, qui prend le pas sur la sélection heuristique de port source SNAT (voir ci-dessus). Ceci n'est valide que si la règle spécifie aussi -p tcp ou -p udp.