Netfilter : le firewall de linux 2.4 et 2.6

- Netfilter: le logiciel, IPTABLES: la commande permettant de le configurer
- netfilter (noyaux 2.4 et premiers noyaux 2.6):
  - filtre à état pour ipv4
  - filtre de paquet sans états pour ipv6 (Arg !)
  - filtre pour decnet, arp et (via des rustines) pour IPX
- Netfilter est un gros progrès par rapport au coupe feu des noyaux 2.2 (ipchain)
  - architecture modulaire
  - filtre à état sur ipv4
  - traduction d'adresses,
  - altération d'entêtes de paquets (mangle)
- configuration/sauver/restaurer les tables

Netfilter

- présent dans les sources du noyau
- la version de l'outil iptables doit être compatible avec celle de netfilter
  - sinon toutes les fonctionnalités ne seront pas accessibles
- patch-o-matic: rustines apportant des fonctionnalités supplémentaires
  - submitted: rustines soumises pour la prochaine version du noyau
  - pending: en attente de soumission
  - base: rustines variées sans conflits entre eux
  - extra: le reste (conflits possibles)

Netfilter

- Thème de cette présentation
  - filtrage à état ipv4 avec netfilter
- 2 bonnes documentations (en français) :
  - « netfilter/iptables: le fonctionnement interne du parefeu selon linux »: linux mag France HS 12, octobre 2002
  - « didacticiel sur iptables » par Oskar Andreasson
    http://www.linux-france.org/prj/inetdoc/guides/iptables-tutorial/

Netfilter: tables et chaînes

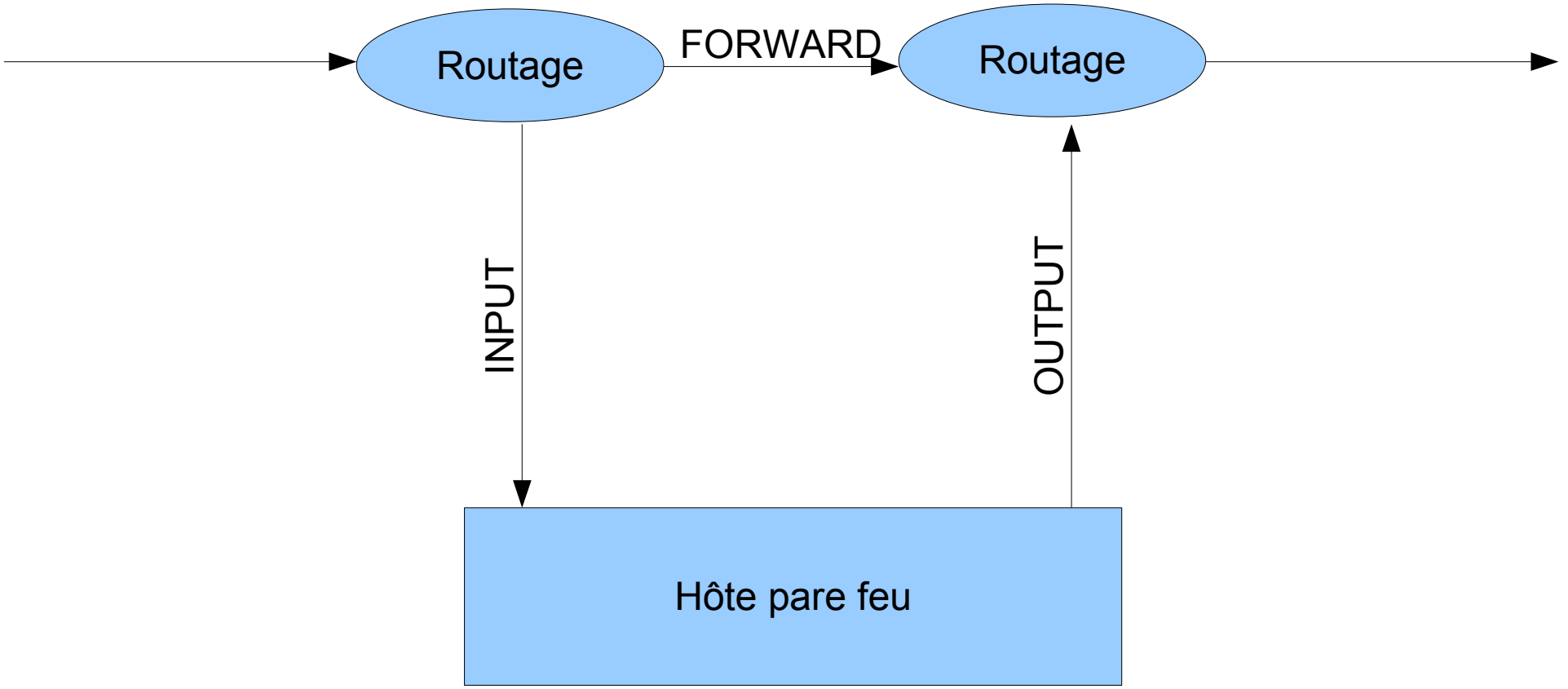
- tables: ensemble de chaînes.
- chaîne: suite linéaires de règles
- règle: constituée
  - d'un motif permettant de reconnaître des paquets selon certaines critère
  - d'un cible indiquant l'action à effectuer sur les paquets reconnus
- un paquet
  - sera traité par certaines chaînes des tables
  - dans ces chaînes, il sera traité consécutivement par toutes les règles jusqu'à en trouver une dont il valide les critères
  - la cible de cette règle sera alors appliquée

Tables NetFilter

- Filter:
  - pour les opérations de filtrage IP.
  - les paquets n'y sont jamais modifiés
  - cibles: ACCEPT, DROP, LOG, REJECT, RETURN, ...
- NAT:
  - pour les opération de traduction d'adresses
  - cibles: SNAT, SAME, DNAT, MASQUERADE, REDIRECT, RETURN, ...
- Mangle:
  - pour modifier les paquets (TTL, TOS, ...)
  - cibles: TTL, TOS, TCPMSS, RETURN, ...

traversée des tables

- cf http://www.linux-france.org/prj/inetdoc/guides/iptables-tutorial/traversingoftables.html



Les 3 chaînes de la table filter

paquet entrant

Étape	Table	Chaîne	Commentaire
1			Sur le câble (ex. Internet)
2			Arrive sur l'interface (ex. eth0)
3	mangle	PREROUTING	Cette chaîne sert normalement à modifier les paquets, i.e. changer les bits de TOS, etc.
4	nat	PREROUTING	Cette chaîne sert principalement au DNAT. Évitez de filtrer dans cette chaîne puisqu'elle est court-circuitée dans certains cas.
5			Décision de routage, i.e. le paquet est-il destiné à notre hôte local, doit-il être réexpédié et où ?
6	mangle	INPUT	Ici, il atteint la chaîne INPUT de la table mangle. Cette chaîne permet de modifier les paquets, après leur routage, mais avant qu'ils soient réellement envoyés au processus de la machine.
7	filter	INPUT	C'est l'endroit où est effectué le filtrage du trafic entrant à destination de la machine locale. Notez bien que tous les paquets entrants et destinés à votre hôte passent par cette chaîne, et ceci quelle que soit leur interface ou leur provenance d'origine.

tableau tiré de
http://www.linux-france.org/prj/inetdoc/guides/iptables-tutorial/traversingoftables.html

paquet sortant

Étape	Table	Chaîne	Commentaire
1			Processus/application local (i.e. programme client/serveur)
2			Décision de routage. Quelle adresse source doit être utilisée, quelle interface de sortie, et d'autres informations nécessaires qui doivent être réunies.
3	mangle	OUTPUT	C'est là où les paquets sont modifiés. Il est conseillé de ne pas filtrer dans cette chaîne, à cause de certains effets de bord. C'est aussi où le traçage de connexion généré localement prend place, nous verrons celà dans le chapitre La machine d'état.
4	nat	OUTPUT	Cette chaîne permet de faire du NAT sur des paquets sortant du pare-feu.
5			Décision de routage, comment les modifications des mangle et nat précédents peuvent avoir changé la façon dont les paquets seront routés.
6	filter	OUTPUT	C'est de là que les paquets sortent de l'hôte local.
7	mangle	POSTROUTING	La chaîne POSTROUTING de la table mangle est principalement utilisée lorsqu'on souhaite modifier des paquets avant qu'ils quittent la machine mais après les décisions de routage. Cette chaîne est rencontrée d'une part par les paquets qui ne font que transiter par le pare-feu, d'autre part par les paquets créés par le pare-feu lui-même.
8	nat	POSTROUTING	C'est ici qu'est effectué le SNAT. Il est conseillé de ne pas filtrer à cet endroit à cause des effets de bord, certains paquets peuvent se faufiler même si un comportement par défaut a été défini pour la cible DROP.
9			Sort par une certaine interface (ex. eth0)
10			Sur le câble (ex. Internet)

tableau tiré de
http://www.linux-france.org/prj/inetdoc/guides/iptables-tutorial/traversingoftables.html

paquet routé

Etape	Table	Chaîne	Commentaire
1			Sur le câble (ex. Internet)
2			Arrive sur l'interface (ex. eth0)
3	mangle	PREROUTING	Cette chaîne est typiquement utilisée pour modifier les paquets, i.e. changer les bits de TOS, etc. C'est ici aussi que le traçage de connexion généré non-localement prend place, nous verrons celà dans le chapitre La machine d'état.
4	nat	PREROUTING	Cette chaîne sert principalement à réaliser du DNAT. Le SNAT est effectué plus loin. Evitez de filtrer dans cette chaîne car elle peut être court-circuitée dans certains cas.
5			Décision de routage, c-à-d. le paquet est-il destiné à votre hôte local, doit-il être redirigé et où ?
6	mangle	FORWARD	Le paquet est alors envoyé à la chaîne FORWARD de la table mangle. C'est utile pour des besoins très spécifiques, lorsque l'on souhaite modifier des paquets après la décision de routage initiale, mais avant la décision de routage finale effectuée juste avant l'envoi du paquet.
7	filter	FORWARD	Le paquet est routé vers la chaîne FORWARD. Seuls les paquets réexpédiés arrivent ici, et c'est ici également que tout le filtrage est effectué. Notez bien que tout trafic redirigé passe par ici (et pas seulement dans un sens), donc vous devez y réfléchir en rédigeant vos règles.
8	mangle	POSTROUTING	Cette chaîne est employé pour des formes particulières de modification de paquets, que l'on veut appliquer postérieurement à toutes les décisions de routage, mais toujours sur cette machine.
9	nat	POSTROUTING	Cette chaîne est employé pour des formes particulières de modification de paquets, que l'on veut appliquer postérieurement à toutes les décisions de routage, mais toujours sur cette machine
10			Sort par l'interface de sortie (ex. eth1).
11			Sort de nouveau par le câble (ex. LAN).

tableau tiré de
http://www.linux-france.org/prj/inetdoc/guides/iptables-tutorial/traversingoftables.html

Chaînes

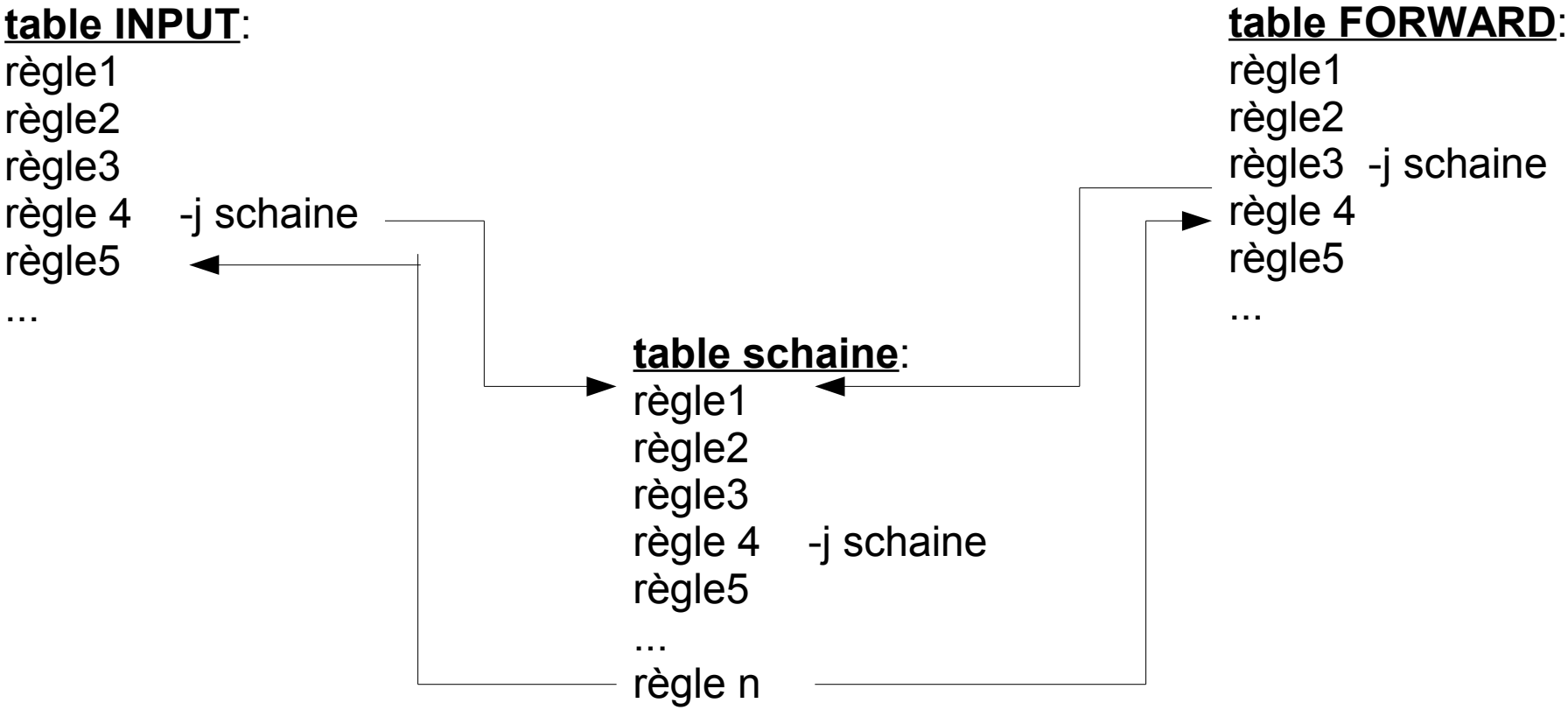
- 2 types de chaînes: par défaut (builtin) et utilisateurs
- chaînes par défaut:
  - propres à certaines tables
    - table Filter: INPUT, OUTPUT et FORWARD
    - table NAT: PREROUTING et POSTROUTING
    - table MANGLE: INPUT, OUTPUT, FORWARD, PREROUTING et POSTROUTING
  - politique par défaut:
    - politique à appliquer en fin de chaîne par défaut: ACCEPT ou DROP
    - commande -P d'iptables: « iptables -P INPUT DROP »

chaînes utilisateurs

- les appels aux chaînes utilisateurs peuvent être inclus à une ou plusieurs chaîne par défaut (on utilise le nom de la chaîne utilisateur comme cible)
- à la fin de la chaîne utilisateur, le flot d'exécution reprend à la ligne suivante de la chaîne appelante
- ràf: dessin illustrant l'appel à faire au tableau
- compteurs associés aux règles des chaînes
  - consultation avec l'option -v d'iptables

chaînes utilisateurs

- intérêt :
  - factoriser des règles
  - éviter le passage dans certaines règles à certains paquets



Netfilter: syntaxe

- iptables [-t table] commande [correspondance] [cible/saut]
  - table: table concernée. Par défaut, c'est la table filter qui est utilisée
  - commande: commande iptable (ajout de règle, suppression de règle, ...)
  - correspondance: critères du filtre de sélection de paquets.
  - cible/saut: action à effectuer sur le paquet
- cf « iptables -m correspondance --help» pour plus de détails sur une correspondance
- cf chapitres 9, 10 et 11 du didacticiel d'IPTABLES: http://www.linux-france.org/prj/inetdoc/guides/iptables-tutorial/

Netfilter: correspondance (matches)

- Les critères de base peuvent être enrichis par des modules externes qu'il convient de préciser avec l'option -m
- un protocole sans module spécifique devra se contenter des critères de base
- exemples de modules:
  - -m mac: utiliser l'adresse mac source comme critère
  - -m multiport: pour spécifier plusieurs ports d'un seul coup séparés par une virgule
  - -m state : pour utiliser le suivi de connexion

Netfilter: exemples

- placer une politique par défaut à DROP sur la talbe INPUT:
  - iptables -P INPUT DROP
- détruire les paquets tcp entrants avec un flag SYN seul. Deux solutions produisant les mêmes effets :
  - iptables -A INPUT -p tcp --tcp-flags SYN,ACK,RST,FIN SYN -j DROP
  - iptables -A INPUT -p tcp --syn -j DROP

Netfilter: exemples (2)

- accepter les paquets routés venant d'un source donnée:
  - venant d'un hôte: iptables -A FORWARD -s 192.168.196.246 -j ACCEPT
  - venant d'un sous-réseau: iptables -A FORWARD -s 192.168.196.0/24 -j ACCEPT
- accepter les paquets routés venant d'une adresse MAC source données:
  - iptables -A FORWARD -m mac --mac-source 00-50-56-C0-00-01
  - noter « -m mac » qui active le module mac

Netfilter: exemples (3)

- accepter les paquets entrants appartenant à des connexions déjà établies (ESTABLISHED ou RELATED):
  - iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  - noter le « -m state » qui active le module state
- accepter les paquets tcp routés à destination d'un port donné d'une machine données et venant d'un sous-réseau donné
  - iptables -A FORWARD -p TCP -d 192.168.196.246 --dport 22 -s 192.168.195.0/24 -j ACCEPT