TP PF

Auteur: Guillaume Arcas
Date : 13/07/2005
Version : 1.1

Ce document présente le pare feu PF pour systèmes Unix FreeBSD et OpenBSD à partir d'un cas pratique.

L'objectif est d'illustrer les notions abordées dans le cours PF.

Contexte

Pour présenter PF et son utilisation comme outil de filtrage réseau, nous nous appuierons sur le cas suivant : la société Entreprise souhaite utiliser Internet tout en protégeant son réseau interne. Les services qui sont accessibles aux employés depuis les postes de travail du réseau interne sont la messagerie électronique et la navigation Web.

La mise en œuvre de la protection du réseau interne se traduit par :

- un strict cloisonnement du réseau interne vis-à-vis d'Internet : aucun flux ne peut transiter entre ces deux zones ;
- l'utilisation de serveurs mandataires (proxies) hébergés dans une DMZ isolée du réseau interne ;
- l'application sur les flux de messagerie d'un filtrage antivirus et sur les flux Web d'un filtrage d'URL qui empêche l'accès aux sites Internet interdits ainsi que le téléchargement de fichiers non autorisés par la politique de sécurité.

Pour assurer le cloisonnement des trois zones – Internet, DMZ, réseau interne – qui composent le système d'information de la société, il a été décidé de déployer deux pare feux. Ces pare feux sont des serveurs à base de processeurs Intel et utilisent le système d'exploitation FreeBSD 5.3. L'outil utilisé pour mettre en œuvre le filtrage réseau est PF.

Le schéma ci-dessous présente une vue du système d'information cible :

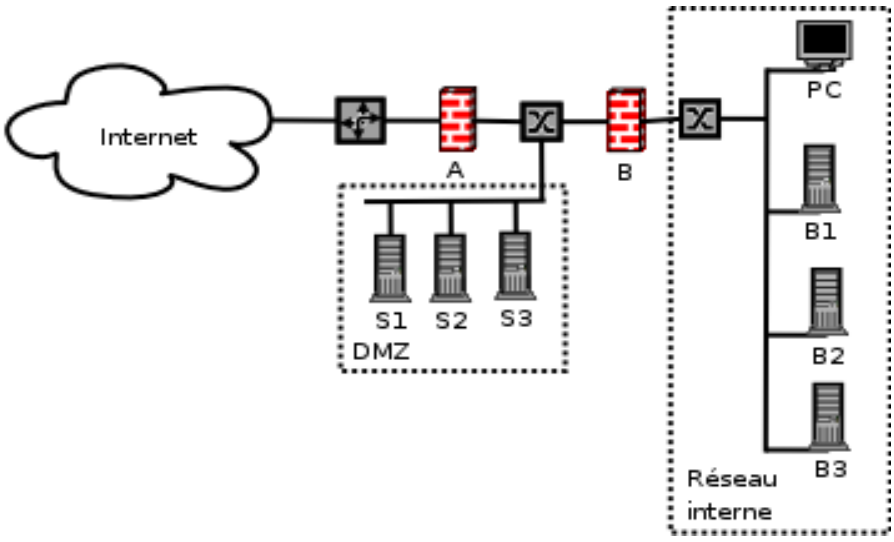

Figure 1 - Schéma du SI

Le tableau suivant présente les trois zones et leurs principales caractéristiques :

Zone Internet	Zone externe également appelée WAN
Zone DMZ	Zone interne destinée à héberger exclusivement des serveurs ayant besoin de se connecter à l'extérieur. On y trouvera : un relais SMTP (S1) un resolver DNS (S2) un proxy http (S3)
Zone Interne	Réseau interne de l'entreprise également appelée LAN. Dans cette zone se trouvent les postes de travail (PC) du personnel et les serveurs suivants : Serveur de sauvegarde (B1) Serveur SMTP (B2) Serveur de fichiers (B3)

Chacune de ces zones est isolée du reste par deux pare-feux (firewall) comme l'illustre la figure 1. Le pare feu A assure le contrôle des flux entre l'extérieur (Internet) et la DMZ. Le B assure le contrôle des flux entre le réseau Interne et la DMZ.

Adressage et routage

Des adresses privées de type RFC1918 sont utilisées pour les zones DMZ et Interne.

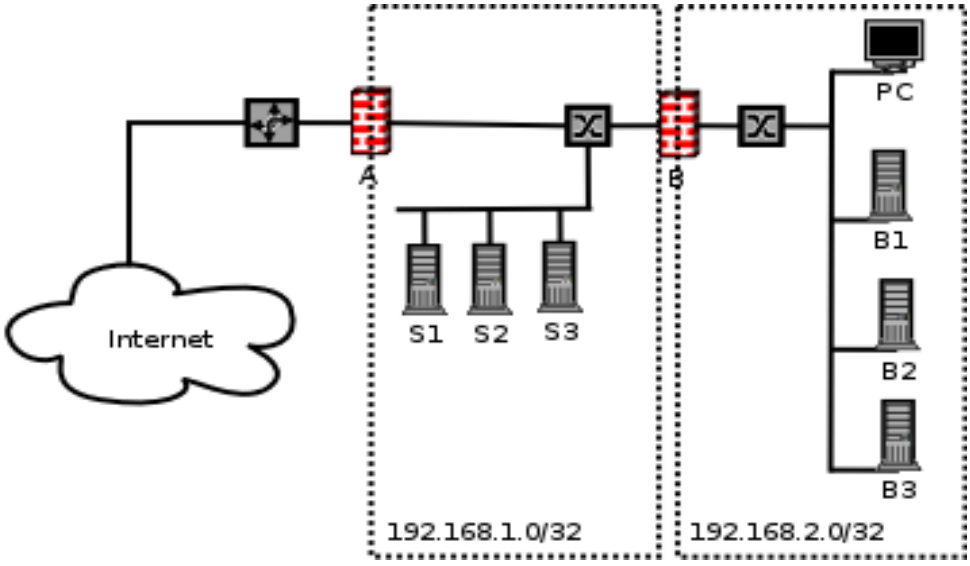

Figure 2 - Plan d'adressage du SI

L'utilisation de ce type d'adresses assure qu'aucun paquet ne sort du S.I. vers Internet (si les routeurs sont configurés correctement) et donne à la société une grande souplesse d'attribution d'IP à des futurs serveurs.

Les pare feux A et B possèdent chacun deux cartes réseaux :

Pare feu	Interface	Adresse	Commentaire
A	fxp0	192.168.0.1	Adresse vers Internet
A	fxp1	192.168.1.1	Adresse vers DMZ
B	fxp0	192.168.1.2	Adresse vers DMZ
B	fxp1	192.168.2.1	Adresse vers réseau Interne

Les routes sont les suivantes :

Pare feu	Route	Commentaire
B	192.168.2.1	Route par défaut des machines du réseau Interne
A	192.168.1.1	Route par défaut des serveurs de la DMZ

La traduction d'adresses IP en mode masquerade est utilisée sur le pare feu B : toutes les machines du réseau interne sont masquées sur une seule IP, celle de la carte fxp0 du pare feu B.

Cette traduction a pour avantages :

- de simplifier le routage entre les zones : les deux seules routes sont celles mentionnées dans le tableau précédent ;
- de camoufler la topologie du réseau interne : même les serveurs de la DMZ n'auront aucune vision complète du nombre ni du genre des machines utilisées sur le réseau interne (application du concept de défense en profondeur).

La DMZ héberge les serveurs suivants :

Serveur	Adresse	Rôle
S1	192.168.1.10	Relais SMTP entrant et sortant + antivirus SMTP
S2	192.168.1.20	Resolver DNS pour les machines de la DMZ et les postes clients du réseau Interne
S3	192.168.1.30	Serveur mandataire (proxy) HTTP + Filtrage d'URL

La zone Interne héberge les serveurs suivants :

Serveur	Adresse	Rôle
B1	192.168.2.10	Serveur de sauvegarde
B2	192.168.2.20	Serveur SMTP
B3	192.168.2.30	Serveur de fichiers

Enfin, les adresses IP publiques suivantes ont été allouées à l'entreprise :

x.x.x.1

x.x.x.2

x.x.x.3

x.x.x.4

x.x.x.5

L'IP x.x.x.1 a été affectée au pare feu A et sert au routage des flux du routeur vers le réseau de l'entreprise. L'adresse publique x.x.x.2 est associée à l'enregistrement DNS de type MX smtp.entreprise.com. Le pare feu A effectue les opérations de traduction d'adresse (NAT) nécessaires pour que cet enregistrement et cette adresse dirigent les flux SMTP vers le relais SMTP S1 de la DMZ. Les autres adresses ne sont pas utilisées pour le moment et sont réservées à des usages ultérieurs.

Description de la politique de sécurité

La politique de sécurité est définie de la manière suivante :

1. Aucun flux ne doit sortir ni entrer directement de la zone Interne vers l'extérieur (Internet).
2. Tous les flux sortant de la zone Interne vers Internet doivent passer par les serveurs de la DMZ.
3. Les seuls flux autorisés en sortie vers la DMZ depuis les postes clients de la zone Interne sont les flux DNS (tcp et udp port supérieur à 1024) vers le resolver S2 (192.168.1.20 tcp et udp port 53) et les flux Web (HTTP, HTTPS et FTP). Pour cela, les navigateurs Internet installés sur les postes clients doivent être configurés pour utiliser le proxy S3 (192.168.1.30, tcp/3128). Une règle de redirection renvoie vers ce proxy les flux HTTP (tcp/80) et HTTPS (tcp/443) pour les navigateurs dont la configuration n'aurait pas été modifiée.
4. Le serveur SMTP B2 de la zone Interne doit utiliser le relais SMTP S1 pour les envois de courrier vers Internet. Les échanges de courrier entre postes clients de la zone Interne sont libres et ne passent pas par le firewall B.
5. Les postes clients peuvent dialoguer entre eux sur le réseau Interne.
6. Aucun flux n'est autorisé depuis ou vers les serveurs B1 et B3 en dehors du réseau Interne.
7. Les flux autorisés en sortie de la DMZ vers Internet sont les flux DNS en provenance du serveur S3 (udp et tcp port 53), les flux Web en provenance du proxy S3 et les flux SMTP en provenance du relais SMTP S1.
8. Les seuls flux en entrée depuis Internet vers la DMZ sont ceux à destination du serveur S1 (relais SMTP) et les flux retour des connexions initiées par les serveurs de la DMZ.
9. Les seuls flux autorisés depuis la DMZ vers la zone Interne sont les flux retour des connexions initiées par les postes clients et le serveur B2.
10. La politique par défaut pour chaque firewall est l'interdiction par défaut.

Mise en œuvre de la politique de sécurité

La politique de sécurité décrite dans le paragraphe précédent est mise en œuvre à l'aide du pare feu PF.

Pour respecter le principe de défense en profondeur, nous utilisons le même outil (PF) mais dans les conditions suivantes :

- le pare feu A est une machine Sun de type Ultra 2 équipée d'un processeur UltraSparc et du système d'exploitation FreeBSD 5.3
- le pare feu B est une machine de type PC équipée d'un processeur Intel et du système d'exploitation OpenBSD 3.7.

Cette diversité matérielle et logicielle (au niveau des OS) assure qu'une vulnérabilité commune à une plate-forme matérielle ou un système d'exploitation ne pourra pas être utilisée avec succès sur les deux pare feu.

Note : il est tout à fait possible d'utiliser Netfilter/IPtables et un système Linux sur un des deux firewall pour une plus grande diversité.

Justification du choix de PF

Le choix de PF pour mettre en œuvre la politique de sécurité au niveau du filtrage des flux Réseau s'explique par le fait que cet outil permet d'utiliser le mode « *stateful inspection* » pour tous les protocoles. Il apporte également des fonctionnalités de filtrage avancées que nous présentons ci-dessous.

Filtrage à états (stateful inspection)

Rappelons que ce mode de filtrage permet de rédiger des règles qui prennent en compte l'état d'un paquet dans une session.

Une session correspond à un dialogue initié par un client avec un serveur. Elle commence au moment où le programme client (navigateur Internet, logiciel de messagerie, etc.) contacte le serveur et se termine au moment où le client ou le serveur informe l'autre que la session est terminée.

Dans le cas d'une session TCP, une session commence par l'envoi par le client d'un paquet qui porte le drapeau SYN et se termine par l'envoi par le client ou le serveur d'un paquet qui porte le drapeau FIN.

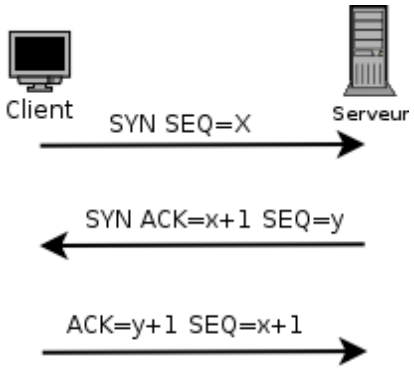

Figure 3 - Ouverture d'une session TCP

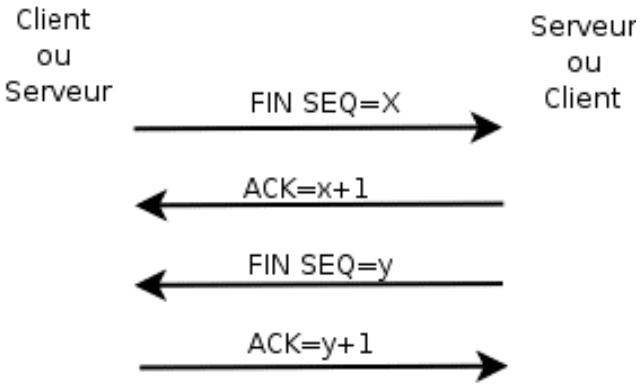

Figure 4 - Fin d'une session TCP

Une session complète commence donc par un paquet TCP SYN et se termine par un paquet ACK.

Pour mémoire, la figure ci-dessous présente les états possibles d'une session TCP.

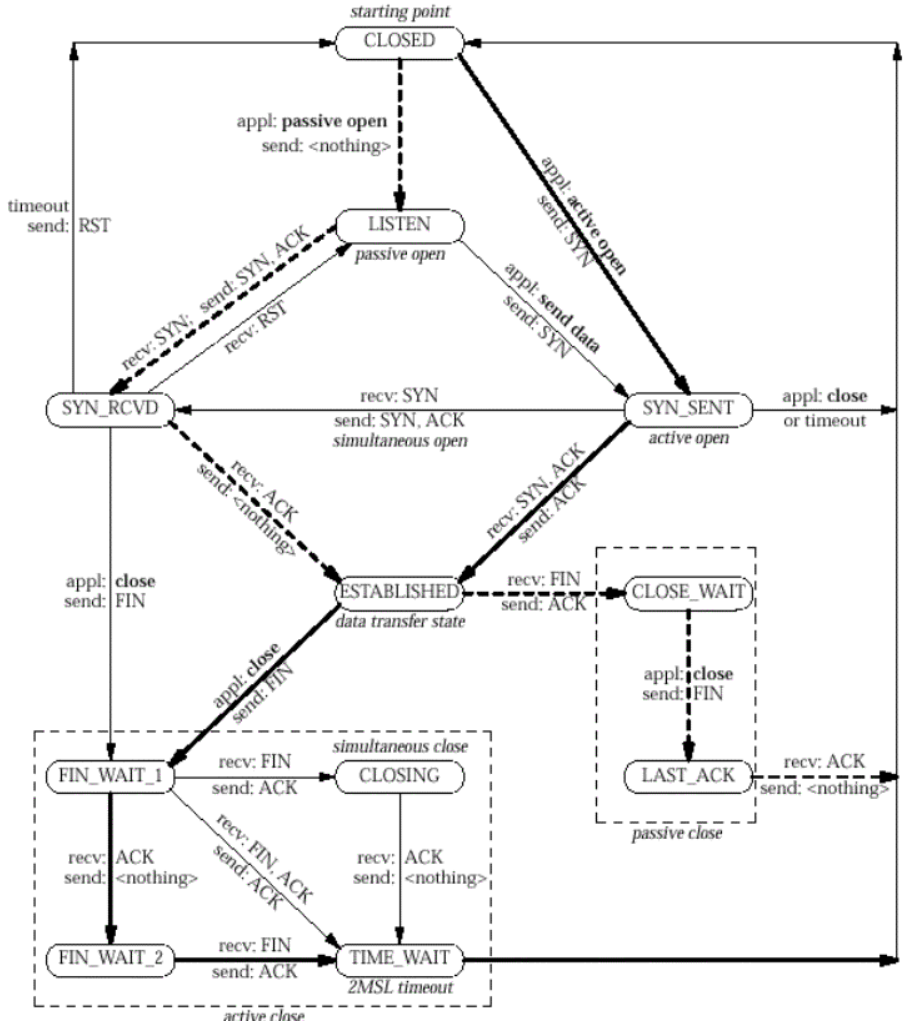

Figure 5 - Etats d'une session TCP

Dans le cadre d'un filtrage avec prise en compte de l'état d'une session, le pare feu utilise les trois états suivants :

- ouverture d'une session (paquet SYN envoyé par un client) ;
- appartenance à une session établie ;

- non appartenance à une session établie.

Pour analyser l'état d'un paquet, PF maintient une table en mémoire dans laquelle sont stockées les caractéristiques des paquets qui constituent les sessions actives.

Avantages du filtrage à états

Le filtrage à états présente de nombreux avantages :

- l'écriture des règles est simplifiée : il suffit d'écrire la règle qui autorise l'ouverture d'une session. Les règles qui correspondent aux flux retour sont automatiquement construites par le pare feu. Dans le cas contraire, il faudrait écrire de manière exhaustive toutes les règles qui correspondent aux flux retour, et ce pour tous les protocoles autorisés. Il faudrait également prévoir tous les cas de figure.
- Les règles sont plus précises : seule la règle qui autorise l'ouverture d'une session est générique. Les règles construites par le pare feu pour autoriser les flux retour ne s'appliqueront qu'au client et au serveur qui font partie de la session.
- Les performances du filtrage sont améliorées : le pare feu n'a qu'à parcourir le tableau des sessions actives pour autoriser ou non le passage d'un paquet une fois la session autorisée établie. Sinon, il lui faudrait comparer le paquet à toutes les règles de filtrage.

Nous allons voir comment dans notre cas pratique, l'utilisation de cette fonctionnalité permet de réduire le nombre de règles nécessaires à l'application du filtrage réseau sur tous les flux.

Rappel sur la notion de direction

Quand on spécifie dans une règle la direction d'un paquet, on utilise avec PF les mots clef `in` et `out`.

Il convient de ne pas oublier que la direction ainsi définie doit être comprise du point de vue de l'interface à laquelle elle s'applique.

Un paquet qui traverse un pare feu est ainsi vu en in sur la première interface et en out sur la seconde :

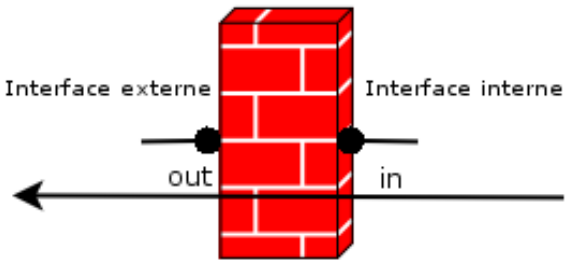

Figure 6 - in/out

Normalisation du trafic

PF permet de normaliser le trafic réseau. Cela signifie qu'avant d'évaluer un paquet pour savoir quelle action prendre (`pass` ou `block`), PF peut :

- reconstituer le trafic fragmenté : la fragmentation de paquets est souvent utilisée dans des attaques en déni de service ou bien pour des actions de reconnaissance de réseaux (avec des outils comme nmap par exemple).
- Eliminer tout paquet anormal : par exemple un paquet TCP dont les drapeaux SYN, ACK et FIN seront positionnés. Une fois de plus, cette technique est utilisée par des outils de reconnaissance de réseaux et de prise d'empreintes de systèmes d'exploitation (technique qui permet à distance de déterminer quel OS est installé sur une machine).

Les opérations de normalisation s'appliquent aux protocoles IP, ICMP, TCP et UDP. Elles consistent à rejeter tout paquet non conforme ou à modifier certaines caractéristiques des paquets. L'objectif de la normalisation est double :

- protéger le pare feu de certaines attaques basées sur des anomalies dans les paquets transmis ;
- améliorer les performances du filtrage.

Ces opérations sont réalisées par la directive `scrub` que l'on applique sur toutes les interfaces dans les deux sens :

```
scrub in all
scrub out all
```

Normalisation IP

Dans le cadre de la normalisation du protocole IP, seront rejetés par exemple :

- les paquets dont la version IP ne sera pas conforme ;
- ceux dont la valeur du champ longueur d'en-tête (Header Length) sera trop petite ou trop grande ;
- ceux dont la somme de contrôle (checksum) sera incorrecte.

Dans certains cas, les valeurs de certains champs seront modifiés :

- remise à zéro des champs d'options IP (IP Options) ;
- ajustement de la valeur du champ TTL.

Normalisation UDP

Elle s'applique après la normalisation IP. Les paquets dont la somme de contrôle (checksum) est incorrecte ou dont la taille est différente de celle indiquée dans les champs IP sont rejetés.

Normalisation ICMP

Sont rejetés :

- les messages de type Echo Request dont la destination est une adresse de multi diffusion (multicast ou broadcast) ;
- les messages dont la somme de contrôle (checksum) est incorrecte ;
- les messages de type Source Quench.

Normalisation TCP

Les opérations de normalisation effectuées sur les paquets TCP ont une fois encore pour objectif de détecter et corriger les anomalies dans les drapeaux TCP et de rejeter les paquets dont la somme de contrôle est incorrecte.

Règles anti-usurpation (antispoofing)

Il faut distinguer deux types d'usurpation.

Un premier type d'usurpation d'adresse IP consiste, pour un paquet, à utiliser comme adresse source l'adresse IP du pare feu dans l'espoir que celui laissera passer un paquet qu'il considèrera comme venant de lui-même.

Le second type consiste à utiliser comme adresse IP source une adresse privée de type RFC1918.

Pour lutter contre le premier type, PF apporte la fonctionnalité antispoof que l'on applique à une interface :

```
antispoof for fpx0
```

Cette fonction rejette tout paquet qui arrive sur le pare feu et dont l'adresse IP source est celle de l'interface du pare feu sur laquelle le paquet arrive :

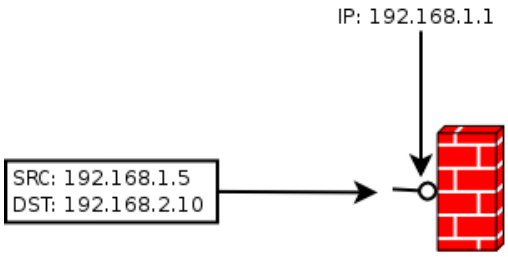

Figure 7 - Spoofing d'interface

Ce paquet sera rejeté. En effet, même si le pare feu utilise le réseau et les protocoles TCP/IP pour des besoins internes, la norme spécifie que dans ce cas il doit utiliser son interface de bouclage interne (loopback) d'adresse 127.0.0.1. Il est donc impossible qu'un paquet normal lui parvienne avec son adresse IP comme source.

Pour lutter contre le second type (utilisation d'adresses privées), nous utiliserons un tableau dans lequel seront consignées les plages d'adresses définies par la RFC1918 et appliquerons une règle de rejet sur le pare feu A (côté Internet).

Traduction d'adresse IP (NAT)

La traduction d'adresse IP consiste à modifier l'adresse source ou destination d'un paquet lorsqu'il traverse le pare feu.

Cette opération permet :

- De masquer un réseau ou un ensemble de serveurs derrière une adresse IP unique ; on parle alors de NAT 1 pour n (avec n > 1).
- D'utiliser des adresses IP privées sur une DMZ et de donner une visibilité publique à certains serveurs en leur attribuant au niveau du pare feu une IP publique, ce qui simplifie les règles de routage. Dans notre cas, cela permet également de respecter le cloisonnement strict des zones : aucun trafic ne sera possible de l'extérieur (Internet) vers la DMZ sans passer par le pare feu A. On parle alors de NAT 1 pour 1 : à une adresse IP publique n'est associé qu'une adresse privée.

- De forcer le passage du trafic en modifiant à la fois l'adresse destination et le port destination. On parle alors de redirection transparente.

Ces trois types de traduction sont mises en œuvre à l'aide des mots clefs suivants :

- nat : traduction 1 pour n ;
- binat : traduction 1 pour 1 ;
- rdr : redirection transparente.

PF maintient, comme pour le filtrage par états, une table en mémoire dans laquelle sont stockées les associations "adresse IP source réelle / adresse de NAT" et qui permettent d'effectuer les opérations de traduction inverse (retour des flux).

Rappel : pour utiliser les fonctions de traductions d'adresses IP, il faut activer le routage IP sur le pare feu :

```
# sysctl net.inet.ip.forwarding=1
```

Les règles de traduction d'adresses obéissent à la syntaxe suivante :

```
nat [pass] on interface [af] from src_addr [port src_port] to \
   dst_addr [port dst_port] -> ext_addr [pool_type] [static-port]
```

Elles sont appliquées avant les règles de filtrage.

TCP SYN PROXY

PF apporte une fonctionnalité intéressante dans le cadre de la protection contre les dénis de service de type SynFlood. Ce type d'attaque est basée sur l'envoi à un serveur jusqu'à sa saturation de paquets SYN dont l'adresse source est généralement falsifiée. Le serveur attend un paquet ACK qui n'arrivera jamais. Très rapidement, toutes les ressources – notamment les sockets réseau – du serveur seront saturées ce qui le rend inopérant.

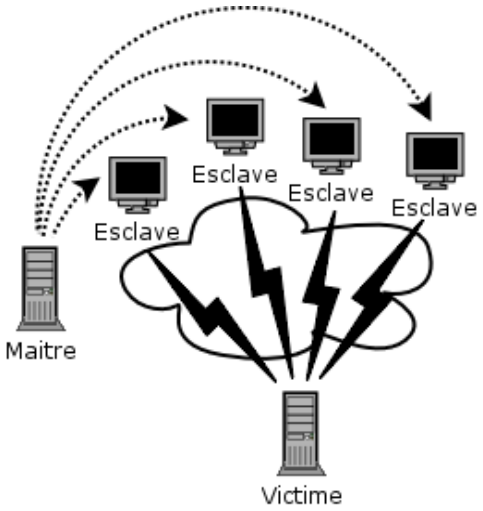

Figure 8 - Attaque DoS SynFlood

Le TCP SYN Proxy de PF permet au pare feu de se substituer au serveur et de ne transférer les paquets que lorsque le pare feu a reçu le paquet ACK en réponse au SYN/ACK :

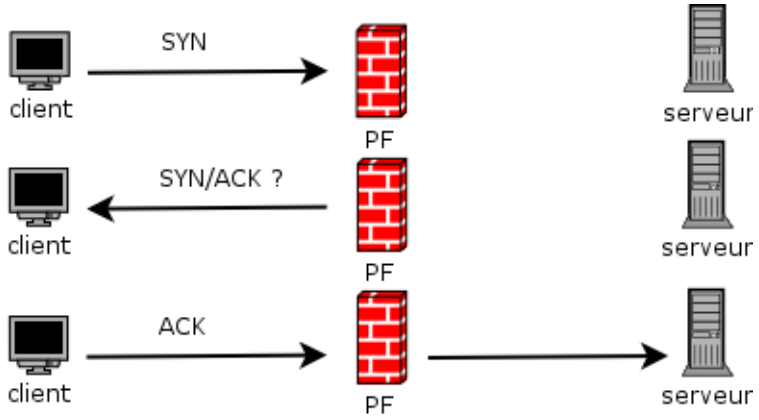

Figure 9 - TCP SYN Proxy

Cette fonction est activée par l'utilisation des mots clefs `synproxy state` dans une règle.

Ecriture des règles

Les règles de filtrage sont stockées sur chaque pare-feu dans un fichier nommé `pf.conf`. Ce fichier est situé dans le répertoire `/etc` de chaque machine.

Pour mémoire, la syntaxe d'une règle PF est la suivante :

```
action [direction] [log] [quick] [on interface] [af] [proto protocol] \
   [from src_addr [port src_port]] [to dst_addr [port dst_port]] \
   [flags tcp_flags] [state]
```

Les champs entre accolades ([ ]) sont optionnels. Si ils ne sont pas renseignés, PF appliquera la règle à toutes leurs valeurs possibles.

Il convient pour écrire les règles de respecter le principe suivant : chaque règle doit être la plus précise possible et doit pour cela utiliser le plus grand nombre de critères possible.

Par exemple, pour autoriser l'ouverture d'une connexion HTTP, nous écrirons la règle suivante :

```
pass in on fxp1 inet proto tcp from $lan port >= 1024 to $wan port 80 \
flags S/SA keep state
```

Plutôt que :

```
pass in proto tcp from $lan to any port 80
```

La première règle n'autorise le passage que des seuls paquets TCP dont l'adresse source appartient au réseau LAN, dont le port est supérieur ou égal à 1024 et dont le drapeau S(YN) est positionné à destination du port 80 d'une machine du réseau WAN. Le mot clef keep state indique que l'on souhaite activer le filtrage avec état sur ce type de flux. Les règles qui autorisent les flux retour sont ainsi construites automatiquement par le pare feu.

La seconde autorise ce flux mais également d'autres flux comme par exemple l'ouverture d'un tunnel entre le port 80 d'une machine du réseau LAN et le même port de n'importe quelle autre machine, interne ou externe.

Pour faciliter la lecture du fichier de règles, nous utilisons des commentaires. Dans un fichier de règles, toute ligne commencée par le caractère # est considérée comme un commentaire.

Définition des variables

Nous avons vu qu'il est possible et souhaitable d'utiliser des variables pour écrire une règle de PF.

L'utilisation de variables facilite la gestion et la modification des règles. Elle permet aussi d'écrire des fichiers génériques que l'on peut appliquer sur différentes machines en ne modifiant que la valeur des variables.

Le fichier `pf.conf` va donc commencer par la déclaration des variables que nous utiliserons ensuite dans le fichier de règles.

Pour chaque pare feu, nous allons définir les variables suivantes :

- nom de chaque interface réseau du pare-feu
- adresse réseau de chaque zone
- adresse IP de chaque serveur de chaque zone
- liste des adresses IP privées de type RFC1918

Définition de la politique par défaut

Avec PF, c'est la dernière règle applicable à un paquet qui s'applique (« last match wins »).

Pour appliquer une politique par défaut, il est possible de placer la règle qui la définit au début ou à la fin du fichier `pf.conf`.

Dans ce document, nous utilisons la première méthode (politique par défaut à la fin du fichier `pf.conf`).

La politique par défaut retenue étant de tout interdire, les règles correspondantes est la suivante :

```
block in all
```

Règles du firewall B

Rappel : ce pare-feu contrôle les flux échangés entre la zone Interne et la DMZ. Il interdit les flux de la zone Interne vers Internet.

La définition des variables constitue l'en-tête du fichier de règles.

Variables

```
# Désignation des interfaces du pare feu B
# interface côté réseau Interne
int_if = "fxp1"
# interface côté DMZ
dmz_if = "fxp0"
# Identification des réseaux
lan = "192.168.2.0/32"
dmz = "192.168.1.0/32"
# Identification des serveurs de la zone Interne
# Serveur de sauvegarde
srv_b1 = "192.168.2.10"
# Serveur SMTP
srv_b2 = "192.168.2.20"
# Serveur de fichiers
srv_b3 = "192.168.2.30"
# Indentification des serveurs de la DMZ
```

```
# Relais SMTP
dmz_s1 = "192.168.1.10"
# Resolver DNS
dmz_s2 = "192.168.1.20"
# Proxy Web
dmz_s3 = "192.168.1.30"
# Liste de tous les serveurs
dmz_srvs = "{" $dmz_s1, $dmz_s2, $dmz_s3 "}"
```

Avant d'écrire les règles spécifiques au filtrage du pare feu B, nous allons activer certaines tâches elles aussi génériques : antispoofing et normalisation du trafic par exemple.

```
# Normalisation du trafic
scrub in all
scrub out all
# Protection contre l'usurpation d'adresse IP
antispoof for $int_if
antispoof for $ext_if
```

Nous pouvons maintenant insérer les règles spécifiques à ce pare feu.

Pour commencer, nous interdisons tout trafic en provenance des serveurs B1 et B3 vers la DMZ :

```
block in quick on $int_if from "{" $srv_b1, $srv-b3 "}"
```

L'utilisation du mot clef `quick` dans cette règle a pour conséquence que tout paquet provenant des serveurs B1 et B3 sera bloqué sans que les autres règles PF ne soient évaluées. L'objectif est d'éviter tout travail inutile à PF.

Ensuite, nous appliquons une règle de traduction de type 1 pour n sur les paquets en provenance de la zone interne vers les serveurs de la DMZ :

```
# Masquage des adresses IP internes
nat on $ext_if  from $lan to $dmz_srvs -> $ext_if
```

Cette règle indique que l'on masque tout le réseau interne derrière l'adresse IP affectée à l'interface externe (côté DMZ) du pare feu.

Nous allons maintenant écrire les règles d'autorisation du trafic du réseau interne vers la DMZ. Nous utiliserons le filtrage à états pour limiter le nombre de règles nécessaires :

```
# Trafic vers le proxy Squid
pass in on $int_if inet proto tcp from $lan port >= 1024 to $dmz_s3\
port 3128 flags S/SA keep state
# Trafic vers le resolver DNS
pass in on $int_if inet proto udp from $lan port >=1204 to $dmz_s2 port 53\
keep state
pass in on $int_if inet proto tcp from $lan port >=1024 to $dmz_s2 port 53\
flags S/SA keep state
```

Les options `flags S/SA` indique que PF doit tester la valeurs des drapeaux TCP et que seuls les paquets dont le drapeau S est positionné sont autorisés à passer. Le mot clef final `keep state` active la filtrage à états pour ces règles.

Le cas du relais SMTP est particulier : seul le serveur SMTP interne est autorisé à dialoguer avec lui :

```
# Dialogue Relais SMTP / Serveur SMTP interne
pass in on $int_if inet proto tcp from $srv_b2 port 25 to $dmz_s1 port 25\
flags S/SA keep state
```

Une fois ces règles définies, nous n'oublions pas de placer nos règles de politique par défaut :

```
# Politique par défaut
block in all
```

Règles du firewall A

La rédaction des règles du pare feu A obéit aux mêmes principes.

Rappelons que la politique de sécurité définit que seul le serveur SMTP a une IP publique, que cette IP est gérée sur le pare feu par une règle de traduction bidirectionnelle et que tout trafic entrant à l'exception de celui à destination du relais est interdit s'il ne correspond pas à une session ouverte par un serveur de la DMZ.

L'en-tête du fichier pf.conf sera la suivante :

```
# Désignation des interfaces du pare feu A
# interface côté réseau Interne
int_if = "fxp1"
# interface côté DMZ
dmz_if = "fxp0"
# Identification des réseaux
dmz = "192.168.1.0/32"
# Indentification des serveurs de la DMZ
# Relais SMTP
dmz_s1 = "192.168.1.10"
# Resolver DNS
dmz_s2 = "192.168.1.20"
# Proxy Web
dmz_s3 = "192.168.1.30"
# Liste de tous les serveurs de la DMZ
dmz_srvs = "{" $dmz_s1, $dmz_s2, $dmz_s3 "}"
# Tableau des adresses privées
table <rfc1918> const { 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8 }
# IP publique du relais SMTP
ip_smtp_pub = "x.x.x.2"
```

Note : la zone externe (Internet) sera désignée à l'aide du mot clef `any`.

Nous avons défini une table des adresses privées qui sera utilisée pour l'antispoofing de type RFC1918 sur l'interface externe du pare feu.

```
# Normalisation du trafic
scrub in all
scrub out all
```

```
# Protection contre l'usurpation d'adresse IP
antispoof for $int_if
antispoof for $ext_if
# Protection contre l'usurpation RFC1918
block in quick on $ext_if from <rfc1918>
```

Attention : il faut spécifier dans cette règle le nom de l'interface sur laquelle elle doit s'appliquer. Sinon, PF applique la règle à toutes les interfaces du pare feu, y compris l'interface interne côté DMZ. Or sur ce réseau, nous utilisons des adresses privées. La conséquence d'une telle erreur serait d'interdire tout trafic en provenance de la DMZ.

Nous allons définir les règles de traduction d'adresses. Rappelons que les serveurs DNS et Squid seront masqués (nat) alors que le relais SMTP se verra attribuer un IP publique unique (binat).

```
# Masquage des adresses du resolver et du proxy
nat on $ext_if  from "{" $dmz_s2; $dmz_s3 "}"  to any -> $ext_if
# Traduction 1 pour 1 pour le relais SMTP
binat on $ext_if inet from $dmz_s1 port 25 to any port 25-> $ip_smtp_pub
```

Nous allons maintenant insérer les règles qui autorisent le trafic sortant en provenance de la DMZ. Nous utiliserons une fois encore le filtrage à états pour limiter le nombre de règles, mais cette fois-ci, pour renforcer le niveau de sécurité, nous emploierons le mot clef `modulate state` : PF va alors modifier la valeurs de certains champs (comme par exemple les ID des paquets IP et les ISN des paquets TCP) pour leur attribuer une valeur réellement aléatoire et non prévisible (protection contre le détournement de session) :

```
# Envoi de courrier depuis le relais
pass in on $int_if inet proto tcp from $dmz_s1 port 25 to any port 25\
flags S/SA modulate state
# Résolution DNS
pass in on $int_if inet proto udp from $dmz_s2 port 53 to any port 53\
modulate state
pass in on $int_if inet proto tcp from $dmz_s2 port 53 to any port 53\
flags S/SA modulate state
# Proxy Web
pass in on $int_if inet proto tcp from $dmz_s3 port >= 1024 to any \
flags S/SA modulate state
```

Nous n'indiquons pas de port destination pour cette dernière règle : si la majorité des sites Internet sont accessibles sur les port 80 (HTTP) et 443 (HTTPS), certains utilisent des ports non standards (8080, 8000, etc.).

Il nous fait maintenant autoriser les flux entrants à destination du relais SMTP. Nous allons aussi appliquer la protection anti DoS en utilisant le synproxy PF :

```
# Accès depuis Internet au relais SMTP
pass in on $ext_if inet proto tcp from any port 25 to $dmz_s1 port 25\
flags S/SA synproxy state
```

Enfin, nous plaçons la règle de politique par défaut :

```
# Politique par défaut
block in all
```

Nous venons ainsi, à l'aide de quelques dizaines de règles PF, de mettre en œuvre la politique de sécurité retenue pour le S.I. de la société Entreprise.

Questions

1°) Nous avons utilisé la fonction antispoof de PF pour appliquer la protection d'usurpation d'adresse sur les interfaces des pare feux. Mais nous aurions tout aussi bien pu écrire une règle PF « classique » équivalente. Quelle aurait été cette règle ?

2°) Il n'y a aucune règle en « out ». Pourquoi ?

3°) Quelles règles faut-il ajouter si l'on positionne une politique par défaut de type « block out all » aux règles du pare feu B ?

4°) Il manque sur le pare feu B un règle de redirection du trafic Web vers le proxy de la DMZ. Ecrivez les règles qui forcent la redirection vers le proxy dmz_s3 des flux TCP à destination des ports 80 et 443 en provenance du réseau interne à destination de l'extérieur.