Shorewall + ipset

A) Introduction

Ce petit tutoriel présente la mise en place d'un pare-feu possédant deux interfaces réseau physiques, placé en tête de réseau.

Même si la logique de Shorewall est relativement simple, et que la syntaxe des fichiers de configuration de Shorewall est bien compréhensible, il n'en reste pas moins que **ce tutoriel s'adresse à des gens ayant déjà une certaine habitude du réseau et de GNU/Linux**, raison pour laquelle certains passages sensés être évidents n'y seront pas détaillés.

L'idée générale est que la plupart des entreprises/particuliers cherchent à se laisser un accès extérieur depuis l'internet au réseau local interne, soit via un **VPN** (généralement *OpenVPN* ou *IPsec*), soit via un accès **SSH** plus restreint. On se placera ici avec l'idée d'un *OpenVPN* sur *UDP 1194* « classique » et d'un port **SSH** sur *TCP 2222* pour éviter les bots des script-kiddies.

Précisons ici le choix assumé d'avoir désactivé l'IPv6 sur le firewall même, soit avec **`ipv6.disable=1`** dans les options de GRUB (/***etc/defaut/grub***) au démarrage, soit dans ***/etc/sysctl.conf***, via

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
net.ipv6.conf.tun0.disable_ipv6 = 1
```

Dans ce dernier fichier, ne pas oublier d'activer la transmission des trames IPv4, via **`net.ipv4.ip_forward=1`**.

On considérera également que la box internet renvoie les deux ports concernés vers notre pare-feu. Le but du jeu sera alors de fermer les accès aux pirates de l'est ET de l'ouest, les seconds ne valant pas mieux que les premiers...

Shorewall est un pare-feu bien connu dans le monde du libre, qui raisonne à coup de zones. On utilisera **loc** pour désigner la zone intranet, **$FW** pour la zone du pare-feux lui-même et enfin **net** pour la zone internet.

B) Politiques de sécurité par défaut

Shorewall utilise un fichier ***/etc/shorewall/policy*** qui indique la politique de sécurité par défaut : en général, on y bloque tout ce qui vient du net, on laisse la zone pare-feu discuter avec les deux autres zones (**loc** et **net**), et on bloque la sortie du réseau local vers l'internet, tout en laissant l'accès du réseau local vers le pare-feu, ce qui donne les règles suivantes :

```
################################
#SOURCE         DEST            POLICY
LOG LEVEL       LIMIT:BURST
################################
```

```
$FW         all          ACCEPT
loc         $FW          ACCEPT
net         all          DROP      info

# THE FOLLOWING POLICY MUST BE LAST
all         all          REJECT       info
```

Attention quand même au niveau de journalisation, ici fixé en *info*, qui peut vite s'avérer lourd en cas d'attaques par déni de service.

C) Exceptions aux règles par défaut

On peut ensuite faire des exceptions aux règles par défaut, via le fichier ***/etc/shorewall/rules***, et une syntaxe très simple, utilisant soit les macros **Shorewall** intégrées au paquet par défaut (cf. ***/usr/share/shorewall/macro.**** pour la liste complète), soit en détaillant les adresses IP/ports source/destination manuellement.

En première approche, ça nous donne :

```
Ping(ACCEPT)   all  $FW
Ping(ACCEPT)   loc  all
#############
# loc -> net
#############
DNS(ACCEPT)    loc  net
NTP(ACCEPT)    loc  net
Whois(ACCEPT)  loc  net
HTTP(ACCEPT)   loc  net
HTTPS(ACCEPT)  loc  net
SSH(ACCEPT)    loc  net
FTP(ACCEPT)    loc  net
POP3S(ACCEPT)  loc  net
IMAPS(ACCEPT)  loc  net
Mail(ACCEPT)   loc  net
# HKP Keys Ubuntu
ACCEPT         loc  net  tcp  11371
#############
# net -> FW
#############
# OpenVPN
ACCEPT         net:+france    $FW udp   1194
# Renvoi SSH entrant vers une autre machine
DNAT           net:+france    loc:192.168.1.22:22
    tcp  2222 -    -    1/min:5
```

Notez ici l'utilisation d'une liste **ipset** nommée ***france***, qui devra être déclarée dans le fichier ***/etc/openvpn/france***, et qui contiendra la liste de toutes les plages IP françaises.

Ainsi Shorewall fermera par défaut l'accès à toute machine hors du réseau français, sachant que cette stratégie a aussi ses limites dans le cas où l'attaquant passerait par un VPN public français, ou disposerait d'une machine compromise dans notre bon pays...

D) Création de la liste ipset

Pour créer notre liste ***/etc/openvpn/france***, c'est assez simple, mais malheureusement manuel si on veut rester dans le gratuit...

Il faut aller sur https://lite.ip2location.com/france-ip-address-ranges, et suivre les liens pour télécharger les plages IPv4 de France dans un format texte, que l'on placera dans ***/etc/shorewall/firewall.france.txt*** pour la suite...

N.B. : à ma grande surprise, je n'ai pas réussi à trouver un site en ligne proposant ces données actualisées en accès libre et gratuit, de manière à pouvoir automatiser la mise à jour via **`wget`** ou **`curl`**. Si jamais vous avez de bonnes adresses en la matière, je suis évidemment preneur...

Notre fichier ***firewall.france.txt*** début avec quelques lignes de commentaires, suivies d'une plage IPv4 par ligne.

```
#
------------------------------------------------
-------
# Free IP2Location Firewall List by Country
# Source:
https://www.ip2location.com/free/visitor-blocker
# Last Generated: 17 Mar 2024 20:51:12 GMT
# [Important] Please update this list every
month
#
------------------------------------------------
-------
1.179.112.0/20
2.2.0.0/15
2.4.0.0/14
2.8.0.0/13
...
```

Il faut ainsi le traiter pour le transformer en format **ipset**. C'est le but du script ***/etc/shorewall/ipsec.sh*** suivant :

```
#!/bin/bash

ipset create france hash:net family inet -exist
ipset flush france
while IFS=$'\n' read -r line; do
        [[ "$line" =~ \# ]] && continue
        echo ipset add france $line -exist
        ipset add france $line -exist
done < firewall.france.txt
ipset save france > /etc/shorewall/france
```

Résultat en sortie de script dans ***/etc/shorewall/france*** :

```
create france hash:net family inet hashsize
16384 maxelem 65536 bucketsize 12 initval
0x804d7021
add france 46.105.162.128/25
add france 176.31.7.128/29
add france 213.41.118.216/30
add france 213.200.87.28
add france 54.37.112.140/30
add france 54.38.130.16/30
...
```

Il faut aussi créé le fichier /***etc/shorewall/init*** pour charger la base au démarrage de Shorewall :

```
if [ -f /etc/shorewall/france ]; then
  ipset destroy france
  ipset -file /etc/shorewall/france restore
fi
```

Ainsi au redémarrage de **Shorewall**, on pourra vérifier la présence de la plage **ipset** via :

```
# iptables -t filter -L|grep vpn
ACCEPT     udp  --  anywhere
anywhere             udp dpt:openvpn /* OpenVPN
*/
ACCEPT     udp  --  anywhere
anywhere             udp dpt:openvpn match-set
france src
```

ou encore utiliser :

```
# ipsec list france
```

Bien entendu, les **ipset** peuvent aussi servir à mettre des adresses IP en liste noire ou blanche. L'avantage du format est qu'il permet de manipuler de très grosses bases d'adresses IP rapidement, **Shorewall** se contentant juste d'utiliser ces micro-bases de données dans ses règles.

Ceux qui ont déjà essayé de compiler des milliers de règles à la main, avec le format du fichier ***/etc/shorewall/rules***, savent que **Shorewall** devient très lent au démarrage, ce format n'étant pas du tout optimisé pour gérer un tel volume d'adresses.

Ipset permet ainsi de s'affranchir de ce problème de manière élégante et conviviale, en accélérant la réactivité globale de **Shorewall**.

E) Description des interfaces réseau

Continuons notre configuration en indiquant les interfaces dans ***/etc/shorewall/interfaces*** :

```
###############################
?FORMAT 2
###############################
#ZONE   INTERFACE       OPTIONS
net     NET_IF
dhcp,tcpflags,nosmurfs,routefilter,logmartians,s
ourceroute=0,physical=net1
loc     LOC_IF
tcpflags,nosmurfs,routefilter,logmartians,physic
al=net0
```

Ici, c'est surtout les interfaces **`physical=`** qui doivent reprendre les noms réels de vos deux interfaces réseau (*net1* et *net0* dans cet exemple). Cf. la liste de vos interfaces via :

```
ip -4 -brief address show
```

F) Mise à jour des paramètres

À noter également que les variables *NET_IF* et *LOC_IF* , ici utilisées dans le fichier ***/etc/shorewall/interfaces***, sont à déclarer dans ***/etc/shorewall/params***. Exemple :

```
NET_IF=lajungleenfolie
NET_BCAST=10.255.255.255
LOC_IF=monreseauinterne
LOC_BCAST=192.168.1.255
```

à adapter évidemment à votre configuration...

G) Description des types de zones

Le fichier ***/etc/shorewall/zones*** est le plus simple de tous :

```
#########################
```

```
#ZONE    TYPE     OPTIONS                 IN
OUT
#                                         OPTIONS
OPTIONS
fw       firewall
net      ipv4
loc      ipv4
```

H) Activation de la translation d'adresses

Enfin le fichier ***/etc/shorewall/snat*** est celui qui s'occupe de la translation d'adresses entre zones. Dans notre cas, une seule ligne est nécessaire :

```
?FORMAT 2
###########################################
#ACTION                  SOURCE
DEST             PROTO    DPORT    SPORT    IPSEC
MARK    USER     SWITCH   ORIGDEST
PROBABILITY

MASQUERADE       net0     net1
```

Au démarage de **Shorewall**, on pourra notamment vérifier la prise en compte du NAT via :

```
iptables -t nat -L
```

Et bien entendu le reste des règles usuelles via :

```
iptables -L
```

I) Blocage dynamique d'adresses IP

Shorewall permet également de bloquer immédiatement des IP ou des plages d'IP en cas d'attaque, depuis la ligne de commande

Une fois l'attaquant détecté, on peut par exemple « s'inspirer » de scripts **portsentry** pour exclure rapidement des IP en mode dynamique :

```
# cat /usr/local/bin/portsentry.sh
#!/bin/bash

# Usage: portsentry.sh <bad_ip> <bad_port>

# Set appropriate variables (easy to customize
on different systems).
DROP_INTERVAL_DAYS=5
HOSTNAME="leparefeu"
#NOTIFY_EMAIL="contact@alsatux.com"

# Get the attacker's IP address and probed port
from the command
# parameters. DO NOT CHANGE THIS!
BAD_IP=$1
BAD_PORT=$2

# Block the bad guy.
/usr/sbin/shorewall drop $1
/usr/sbin/shorewall save

# Unblock him X days after midnight tonight.
```

```
#echo "/usr/sbin/shorewall allow $1" | at
midnight + $DROP_INTERVAL_DAYS days

# Mail me a note to notify me of each block.
# TEMPORARILY ENABLED.

echo "Portsentry has blocked $BAD_IP (`host
$BAD_IP`) on `date`" >>
/var/log/portsentry.blocked.log

#echo "Portsentry has blocked $BAD_IP (`host
$BAD_IP`) on `date`, \
#from now until $DROP_INTERVAL_DAYS days from
midnight tonight. At this \
#point `at -l | wc -l` hosts are blocked ." |
mail -s "$HOSTNAME: \
#Portsentry blocked $BAD_IP on $BAD_PORT"
$NOTIFY_EMAIL
```

Attention quand même à bien activer l'option :

```
DYNAMIC_BLACKLIST=Yes
```

dans le fichier de configuration ***/etc/shorewall/shorewall.conf***.

J) Conclusion

Le support d'**ipset** dans **Shorewall** vient lever la limitation que beaucoup reprochaient à **iptables**, à savoir que l'outil devenait très lent au démarrage avec des milliers d'adresses ou de plages IP à traiter.

Quelque part, **ipset** jette donc un pavé dans la mare de ceux qui voyaient déjà disparaître le paquet **iptables** au profit de **nftables**, donné pour plus « moderne » dans sa syntaxe et son approche.

Les deux outils ayant chacun leurs avantages et inconvénients (cf. https://www.it-connect.fr/chapitres/protection-dun-serveur-web-diptables-a-nftables/ pour les plus curieux), on s'abstiendra ici de prendre part dans cette nouvelle querelle de clochers entre libristes gaulois...

En attendant, le fait que **Shorewall** s'appuie toujours sur **iptables** ne porte en rien préjudice à sa robustesse et sa fiabilité, bien au contraire...

Une fois sa logique comprise, **Shorewall** reste un outil facile à mettre en place, offrant un excellent niveau de sécurité.

On regrettera surtout qu'aucun organisme public en France - et on pensera notamment à ceux dont c'est pourtant le rôle - ne fournisse gratuitement les informations de plages IP, déjà par pays, à défaut de plus précis.

Nous n'avons pas ici étudié les **blrules** qui permettent de mettre en liste blanche ou noire des plages entières.

Ainsi à moins d'un bug peu probable dans le logiciel, ou d'une grossière erreur de configuration des administrateurs, toujours possible, stopper « l'ennemi extérieur » n'est plus vraiment le problème aujourd'hui.

IOT, appareils communicants connectés en USB, imprimantes réseau renifleuses de données, etc – le véritable danger aujourd'hui est surtout à l'« intérieur » du réseau.

La nouvelle mission du pare-feu n'est plus seulement de bloquer les intrus en entrée, mais de couper les communications en sortie lorsqu'un trafic suspect est détecté, quitte à bloquer temporairement une connexion en cours.

Le nouveau défi est donc d'être capable de surveiller étroitement les usages et les volumes en sortie, et en temps réel, sachant que :

- ➔ la fibre offre désormais une autoroute 4 voies aux pirates en trafic montant
- ➔ les trafics sont quasiment tous chiffrés aujourd'hui
- ➔ l'encapsulation et le chiffrement des VPN rend l'analyse interne des données difficile, voire impossible
- ➔ aucun réseau sans-fil n'est supposé inviolable
- ➔ les seules informations restant « accessibles » sont essentiellement les sites web visités, les fichiers de travail ouverts, les quotas réseau triés par port/protocole, etc.

Ainsi **Shorewall** ne remplacera pas les ressources humaines indispensables pour analyser les données réseau, et en déduire des scenarii et des procédures.

Comme toujours, la clé de la sécurité est dans les compétences humaines, et non les matériels ou les logiciels supposés « intelligents ».

Sur ce, merci d'avoir lu ce petit tutoriel, en espérant qu'il vous aura donné des idées ou des voies de réflexion pour vos futures réalisations !