

GUIDE INSTALL IDS

Snort

Ce guide décrit la mise en place d'une solution de détection d'intrusion réseau en utilisant les outils open source suivants : snort (système de détection d'intrusion réseaux (NIDS)), Barnyard qui est une couche applicative qui exploite les événements générés par Snort au format « unifié », oinkmaster qui est un script simple de gestion et de mise à jour de règles Snort et enfin l'outil Base est une interface web permettant la gestion des alertes générées par snort.





الوكالة الوطنية للسلامة المعلوماتية
Agence Nationale de la Sécurité Informatique

Gestion du document

Version	Date	Modification apportée
1.0	03/08/2009	Première version
1.1	29/01/2010	M.à.J des versions des packages

PLAN

1.	Présentation de snort	3
2.	Installation des dépendances.....	3
3.	Installation.....	4
3.1.	INSTALLATION DE SNORT	4
3.2.	INSTALLATION DE BARNYARD	5
3.3.	INSTALLATION D'OINKMASTER	6
3.4.	INSTALLATION DE LA CONSOLE BASE	6
4.	Exploitation	7
4.1.	Configuration de snort	7
4.2.	configuration de Barnyard.....	9
4.3.	Configuration d'oikmaster	10
4.4.	Configuration de la console base	10
4.5.	Configuration des scripts du nettoyage et du démarrage.....	10
4.6.	Test de fonctionnement.....	11
5.	Annexes.....	11
o	Script clean.sh	11
o	Script snort.sh	12
o	Configuration d'apache	15
o	Script fw.sh	16

1.

1. PRESENTATION DE SNORT

Snort est un système de détection d'intrusion réseau (NIDS) open source, disponible sous licence GPL, fonctionnant sur les systèmes Windows et linux.

Dans ce guide on va installer la dernière version de snort « snort 2.8.5.2 » qui est disponible depuis son site officiel <http://www.snort.org/> sous un système d'exploitation linux debian.

Snort est capable d'effectuer en temps réel des analyses de trafic et de logger les paquets sur un réseau IP. Il peut effectuer des analyses de protocole, recherche/correspondance de contenu et peut être utilisé pour détecter une grande variété d'attaques et de sondes comme des dépassements de buffers, scans, attaques sur des CGI, sondes SMB, essai d'OS fingerprintings et bien plus.

Pour effectuer ces analyses snort se base sur des règles. Celles-ci sont écrites par Sourcefire ou bien contribuées par la communauté. On l'utilise en général pour détecter une variété d'attaques et de scans tels que des débordements de mémoire, des scans de ports, des attaques CGI, des tentatives de déni de service (DOS). Ce guide explique les différentes étapes pour mettre en place un NIDS avec une base de données MySQL pour les logs.

Snort peut également être utilisé avec d'autres projets open sources tels que Barnyard et BASE (qui utilise ACID) afin de fournir une représentation visuelle des données concernant les éventuelles intrusions.

2. INSTALLATION DES DEPENDANCES

```
# pour snort

apt-get install libltdl3 libpcap0.8 libprelude2

#pour admin mysql
```

```
apt-get install phpmyadmin

apt-get install mysql-server mysql-common mysql-client

apt-get install libnet1 libnet1-dev libpcrc3 libpcrc3-dev autoconf automake1.9

apt-get install libpcap0.8-dev libmysqlclient15-dev

apt-get install gcc make libtool libssl-dev gcc-4.1 g++

# pour base

apt-get install libphp-adodb php5 php-pear php5-cli php5-gd
```

3. INSTALLATION

3.1. INSTALLATION DE SNORT

- Ajouter un groupe et un utilisateur snort : snort et créer le répertoire d'installation puis télécharger, compiler et installer snort.

```
groupadd snort

useradd -g snort snort

mkdir /etc/snort

Faire un enregistrement dans le site de snort.org pour avoir une clé oinkmaster et
télécharger la source et les rules de snort dans /root

cd /root

wget http://dl.snort.org/snort-current/ snort-2.8.5.2.tar.gz

tar xvzf snort-2.8.5.2.tar.gz

cd snort-2.8.5.2

./configure --prefix=/etc/snort --with-mysql --enable-dynamicplugin

make
```

```
make install

make clean

mkdir /var/log/snort

chown snort:snort /var/log/snort
```

- Faire un enregistrement dans le site de snort.org pour avoir une clé oinkmaster.
- Dans le répertoire « /etc/snort » faire

```
cd /etc/snort

mkdir etc

cd etc

wget http://www.snort.org/pub-bin/oinkmaster.cgi/<cléoinkmaster>/snortrules-
snapshot-2.8.tar.gz

tar xvf snortrules-snapshot-2.8.tar.gz

mv etc snort

mv rules snort/

mv so_rules snort/
```

3.2. INSTALLATION DE BARNYARD

- Télécharger, décompresser et compiler barnyard

```
cd /root

Wget http://garr.dl.sourceforge.net/sourceforge/barnyard/barnyard-0.2.0.tar.gz

Tar xvzf barnyard-0.2.0.tar.gz

cd barnyard-0.2.0
```

```
mkdir /etc/barnyard

./configure --prefix=/etc/barnyard --enable-mysql

make && make install

cp etc/barnyard.conf /etc/snort/etc/snort/
```

3.3. INSTALLATION D'OINKMASTER

- Voici la procédure d'installer

```
apt-get install libcompress-raw-zlib-perl libcompress-zlib-perl libfont-afm-perl libhtml-
format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libio-compress-
base-perl libio-compress-zlib-perl libmailtools-perl libtimedate-perl liburi-perl libwww-
perl

cd /root

wget http://garr.dl.sourceforge.net/sourceforge/oinkmaster/oinkmaster-2.0.tar.gz

tar xzvf oinkmaster-2.0.tar.gz

cd oinkmaster-2.0

cp oinkmaster.pl /etc/snort/bin/

cp oinkmaster.conf /etc/snort/etc/snort/
```

3.4. INSTALLATION DE LA CONSOLE BASE

```
pear config-set preferred_state alpha

pear install Image_Canvas

pear install Image_Color

pear install Numbers_Roman

pear install Image_Graph
```

```
pear install Mail

pear install Mail_Mime

cd /root

wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.4/base-1.4.4.tar.gz/download

tar xvzf base-1.4.4.tar.gz

mv base-1.4.4 /var/www/base

chmod -R 777 /var/www/base
```

4. EXPLOITATION

4.1. Configuration de snort

- Changer le fichier snort.conf comme suit :

```
vim /etc/snort/etc/snort/snort.conf

var RULE_PATH /etc/snort/etc/snort/rules

var HOME_NET <external-net>/external-netmask

var EXTERNAL_NET !$HOME_NET

dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so

# changer /usr/local par /etc/snort dans tous snort.conf
```

- Commenter les lignes suivantes si vous utilisez les règles de VRT SO fournis par défaut, comme ci dessous :

```
#dynamicdetection file /etc/snort/lib/snort_dynamicrules/bad-traffic.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/chat.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/dos.so
```

```
#dynamicdetection file /etc/snort/lib/snort_dynamicrules/exploit.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/imap.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/misc.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/multimedia.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/netbios.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/nntp.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/p2p.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/smtp.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/sql.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/web-client.so

#dynamicdetection file /etc/snort/lib/snort_dynamicrules/web-misc.so
```

- Vérifier le bon fonctionnement de snort

```
/etc/snort/bin/snort -q -u snort -g snort -c /etc/snort/etc/snort/snort.conf

ctrl ^C
```

- Paramétrage de mysql

```
mysqladmin -u root password "mypassword" ; par exemple root

mysql -u root -proot

mysql> create database snort;

mysql> create database snort_archive;

mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort@localhost;

mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort_archive.* to
```

```
snort@localhost;

mysql> SET PASSWORD FOR snort@localhost=PASSWORD('snort');

mysql> exit

cd /root/snort-2.8.4.1/schemas/

mysql snort -u root -proot < create_mysql

mysql snort_archive -u root -proot < create_mysql
```

4.2. configuration de Barnyard

```
vi /etc/snort/etc/snort/snort.conf

//commenter la ligne suivante:

#output database: log, mysql, user=snort password=snort dbname=snort host=localhost

//et décommenter ces deux lignes:

output alert_unified: filename snort.alert, limit 128

output log_unified: filename snort.log, limit 128

ps -ef | grep snort

kill -9 pid.snort

/etc/snort/bin/snort -Dq -u snort -g snort -c /etc/snort/etc/snort/snort.conf

vi /etc/snort/etc/snort/barnyard.conf

config hostname: localhost

config interface: eth0

//disable tout les outputs plugging sauf:

output log_acid_db: mysql, database snort, server localhost, user snort, password snort,
detail full
```

4.3. Configuration d'oinkmaster

```
vi /etc/snort/etc/snort/oinkmaster.conf

//remplacer la ligne url ....par votre code oinkmaster comme suit:

url=http://www.snort.org/pub-
bin/oinkmaster.cgi/4fb031ee65f9e04d3480027fce3f4323be05c256/snortrules-snapshot-
2.8.tar.gz

mkdir /tmp/oinktest

/etc/snort/bin/oinkmaster.pl -o /tmp/oinktest

ls -al /tmp/oinktest

//test de fonctionnement

vi /tmp/oinktest/pop3.rules

mkdir /tmp/oinkback

/etc/snort/bin/oinkmaster.pl -o /tmp/oinktest -b /tmp/OINKBACK

ls -al /tmp/oinktest
```

4.4. Configuration de la console base

- Par navigateur connecter vous sur <http://10.1.1.25/base> et configurer le système en entrant les paramètres de la base de donnée de donnée.
- Attention le chemin de adodb est /usr/share/php/adodb et n'utiliser pas l'authentification dans la phase 4.
- Pour restreindre l'accès au console *base* modifier le fichier /etc/apache2/sites-enabled/000-default comme dans l'annexe 4 et pour créer un utilisateur nommer *admin* utiliser la commande suivante :

```
htpasswd -s -c /etc/apache2/passwords admin
```

4.5. Configuration des scripts du nettoyage et du démarrage

```
cd /root

cp snort.sh /etc/init.d/

chmod +x /etc/init.d/snort.sh

update-rc.d -n snort.sh defaults

cp clean.sh /etc/cron.daily

chmod +x /etc/cron.daily/clean.sh
```

4.6. Test de fonctionnement

5. ANNEXES

- o Script clean.sh

```
#!/bin/bash

/etc/init.d/snort.sh stop

mysql snort -u snort -psnort <<EOF

DELETE FROM event WHERE timestamp < DATE_SUB(NOW(),INTERVAL 7
DAY);

DELETE FROM data USING data LEFT OUTER JOIN event USING (sid,cid)
WHERE event.sid IS NULL;

DELETE FROM iphdr USING iphdr LEFT OUTER JOIN event USING
(sid,cid) WHERE event.sid IS NULL;

DELETE FROM icmp_hdr USING icmp_hdr LEFT OUTER JOIN event USING
(sid,cid) WHERE event.sid IS NULL;

DELETE FROM tcp_hdr USING tcp_hdr LEFT OUTER JOIN event USING
(sid,cid) WHERE event.sid IS NULL;

DELETE FROM udp_hdr USING udp_hdr LEFT OUTER JOIN event USING
(sid,cid) WHERE event.sid IS NULL;

DELETE FROM opt USING opt LEFT OUTER JOIN event USING (sid,cid)
WHERE event.sid IS NULL;
```

```

DELETE FROM acid_event USING acid_event LEFT OUTER JOIN event
USING (sid,cid) WHERE event.sid IS NULL;

DELETE FROM ag USING acid_ag_alert AS ag LEFT OUTER JOIN event AS
e ON ag.ag_sid=e.sid AND ag.ag_cid=e.cid WHERE e.sid IS NULL;

OPTIMIZE TABLE event, data, iphdr, icmp_hdr, tcp_hdr, udp_hdr, opt,
acid_event, acid_ag_alert;

EOF

rm -rf /var/log/snort/*

/etc/init.d/snort.sh start

```

- o Script snort.sh

```

#!/bin/sh

#copyright ANSI

interface="eth0"

pid_snort=`ps ax | awk '{if (match($5, ".*snort$") || $5 == "snort") print $1}'`

pid_barnyard=`ps ax | awk '{if (match($5, ".*barnyard$") || $5 == "barnyard")
print $1}'`

start() {

    if test "$pid_snort" != ""; then

        echo "snort is already running as pid $pid_snort."

    else

        echo "Starting snort..."

        /etc/snort/bin/snort -A full -Dq -u snort -g snort -c
/etc/snort/etc/snort/snort.conf -i ${interface}

        code_retour_snort_started=$?

    if test "$code_retour_snort_started" == "0"; then

        barnyard_timestamp=$(ls -t /var/log/snort/snort.log.* | head -1 | cut -d. -f
3 | tr -d [:blank:])

        echo "/var/log/snort">/etc/snort/etc/snort/bylog.waldo

```

```

echo "snort.log">>/etc/snort/etc/snort/bylog.waldo

echo "$barnyard_timestamp">>/etc/snort/etc/snort/bylog.waldo

echo "0">>/etc/snort/etc/snort/bylog.waldo

if test "$pid_barnyard" != ""; then

    /usr/bin/killall barnyard

fi

/etc/barnyard/bin/barnyard -c /etc/snort/etc/snort/barnyard.conf -D -g
/etc/snort/etc/snort/gen-msg.map -s /etc/snort/etc/snort/sid-msg.map -d
/var/log/snort/ -f snort.log -w /etc/snort/etc/snort/bylog.waldo

code_retour_barnyard_started=$?

if test "$code_retour_barnyard_started" != "0"; then

    echo "barnyard is broken"

    /usr/bin/killall snort && echo "so snort will be stopped."

Fi

else

echo "snort is broken"

fi

fi

}

stop () {

    /usr/bin/killall snort && echo "snort stopped."

    sleep 5

    /usr/bin/killall barnyard && echo "barnyard stopped."

}

status() {

    if test "$pid_snort" != ""; then

```

```
    echo "snort is running as pid $pid_snort."

else

    echo "snort is not running."

fi

if test "$pid_barnyard" != ""; then

    echo "barnyard is running as pid $pid_barnyard."

else

    echo "barnyard is not running."

fi
}
case "$1" in
start)
start
;;
stop)
stop
;;
restart)
stop
start
;;
status)
status
;;
*)
```

```
echo $"Usage: $0 {start | stop | restart | status}"  
  
;;  
  
esac  
  
exit 0
```

- o Configuration d'apache

```
Vi /etc/apache2/sites-enabled/000-default  
  
NameVirtualHost *  
  
<VirtualHost *>  
  
    ServerAdmin webmaster@localhost  
  
    DocumentRoot /var/www/  
  
<Directory />  
  
    Options FollowSymLinks  
  
    AllowOverride None  
  
</Directory>  
  
<Directory /var/www/>  
  
    Options Indexes FollowSymLinks MultiViews  
  
    AllowOverride None  
  
    AuthType Basic  
  
    AuthName "SAHER"  
  
    AuthUserFile /etc/apache2/passwords  
  
    Require valid-user  
  
    RedirectMatch ^/$ /base/  
  
</Directory>  
  
    ErrorLog /var/log/apache2/error.log  
  
    LogLevel warn
```

```
CustomLog /var/log/apache2/access.log combined

ServerSignature On

</VirtualHost>
```

- o Script fw.sh

```
vi /etc/init.d/fw.sh

#!/bin/bash

export IPT="/sbin/iptables"

export IPTS="/sbin/iptables-save"

export IPTR="/sbin/iptables-restore"

export LO_IF="lo"

export EXT_IF="eth2"

export LO_IP="127.0.0.1"

export EXT_IP=`ifconfig $EXT_IF | grep inet | cut -d : -f 2 | cut -d \ -f 1`

export ADMIN_LIST="192.168.6.150 192.168.6.170 192.168.6.110
192.168.6.112"

if [ -e /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses ]; then

    echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

    echo "Ignorer les messages ICMP buggé [OK]"

else

    echo "Ignorer les messages ICMP buggé [FAILED]"

fi

if [ -e /proc/sys/net/ipv4/conf/all/accept_redirects ]; then

    echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects

    echo "Déctivation des redirections ICMP [OK]"

else

    echo "Déctivation des redirections ICMP [FAILED]"

fi
```

```

fi

if [ -e /proc/sys/net/ipv4/conf/all/log_martians ]; then

    echo "1" > /proc/sys/net/ipv4/conf/all/log_martians

    echo "Log des adresses impossibles          [OK]"

else

    echo "Log des adresses impossibles          [FAILED]"

fi

if [ -e /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts ]; then

    echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

    echo "Ignore les requêtes broadcast d'éo ICMP [OK]"

else

    echo "Ignore les requêtes broadcast d'éo ICMP [FAILED]"

fi

if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]; then

    echo "1" > /proc/sys/net/ipv4/conf/all/rp_filter

    echo "Activation des filtres contre le spoofing [OK]"

else

    echo "Activation des filtres contre le spoofing [FAILED]"

fi

if [ -e /proc/sys/net/ipv4/conf/all/accept_source_route ]; then

    echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route

    echo "Déctivation des sources de routage      [OK]"

else

    echo "Déctivation des sources de routage      [FAILED]"

fi

```

```

if [ -e /proc/sys/net/ipv4/tcp_ecn ]; then
    echo "0" > /proc/sys/net/ipv4/tcp_ecn
    echo "D  ctivation de notif. de congestion TCP      [OK]"
else
    echo "D  ctivation de notif. de congestion TCP      [FAILED]"
fi

if [ -e /proc/sys/net/ipv4/ip_conntrack_max ]; then
    echo "16384" > /proc/sys/net/ipv4/ip_conntrack_max
fi

if [ -e /proc/sys/net/ipv4/tcp_syncookies ]; then
    echo "1" > /proc/sys/net/ipv4/tcp_syncookies
    echo "Protection contre le SYN Flooding          [OK]"
else
    echo "Protection contre le SYN Flooding          [FAILED]"
fi

if [ -e /proc/sys/net/ipv4/conf/all/secure_redirects ]; then
    echo "1" > /proc/sys/net/ipv4/conf/all/secure_redirects
    echo "secure redirection              [OK]"
else
    echo "secure redirection              [FAILED]"
fi

/sbin/modprobe ip_tables || exit 1

/sbin/modprobe ipt_owner || exit 1

/sbin/modprobe ip_conntrack || exit 1

```

```
/sbin/modprobe ip_conntrack_ftp || exit 1

/sbin/modprobe ip_conntrack_irc || exit 1

/sbin/modprobe ipt_LOG || exit 1

/sbin/modprobe ipt_REJECT || exit 1

/sbin/modprobe ipt_MASQUERADE || exit 1

/sbin/modprobe ipt_TOS || exit 1

/sbin/modprobe ipt_TCPMSS || exit 1

/sbin/modprobe ipt_MARK || exit 1

/sbin/modprobe ipt_REDIRECT || exit 1

/sbin/modprobe iptable_mangle || exit 1

/sbin/modprobe ip_nat_ftp || exit 1

/sbin/modprobe ip_nat_irc || exit 1

/sbin/modprobe ip_nat_snmp_basic || exit 1

/sbin/modprobe ip_queue || exit 1

/sbin/modprobe iptable_filter || exit 1

/sbin/modprobe iptable_nat || exit 1

/sbin/modprobe ipt_ttl || exit 1

/sbin/modprobe ipt_limit || exit 1

/sbin/modprobe ipt_mac || exit 1

/sbin/modprobe ipt_multiport || exit 1

/sbin/modprobe ipt_length || exit 1

for T in filter nat mangle ; do

    $IPT -t $T -F

    $IPT -t $T -X

done
```

```
$IPT -P INPUT DROP

$IPT -P OUTPUT DROP

$IPT -P FORWARD DROP

if [ "$1" = "stop" ]
then
    echo "Firewall désactivé."

    $IPT -t filter -F

    $IPT -t filter -X

    $IPT -t filter -P INPUT ACCEPT

    $IPT -t filter -P OUTPUT ACCEPT

    $IPT -t filter -P FORWARD ACCEPT

    $IPT -t nat -F

    $IPT -t nat -X

    $IPT -t nat -P PREROUTING ACCEPT

    $IPT -t nat -P OUTPUT ACCEPT

    $IPT -t nat -P POSTROUTING ACCEPT

    $IPT -t mangle -F

    $IPT -t mangle -X

    $IPT -t mangle -P PREROUTING ACCEPT

    $IPT -t mangle -P INPUT ACCEPT

    $IPT -t mangle -P OUTPUT ACCEPT

    $IPT -t mangle -P FORWARD ACCEPT

    $IPT -t mangle -P POSTROUTING ACCEPT

    for Filter in /proc/sys/net/ipv4/conf/*/rp_filter; do

        echo 0 > $Filter
```

```
done

echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all

echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

echo 0 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

$IPT -t nat -A POSTROUTING -o $EXT_IF -j SNAT --to 196.203.67.170

echo 1 > /proc/sys/net/ipv4/ip_forward

exit 0

fi

if [ "$1" = "bloc" ]

then

    $IPT -t filter -F

    $IPT -t filter -X

    $IPT -t filter -P INPUT DROP

    $IPT -t filter -P OUTPUT DROP

    $IPT -t filter -P FORWARD DROP

    $IPT -t nat -F

    $IPT -t nat -X

    $IPT -t nat -P PREROUTING DROP

    $IPT -t nat -P OUTPUT DROP

    $IPT -t nat -P POSTROUTING DROP

    $IPT -t mangle -F

    $IPT -t mangle -X

    $IPT -t mangle -P PREROUTING DROP

    $IPT -t mangle -P INPUT DROP

    $IPT -t mangle -P OUTPUT DROP
```

```

$IPT -t mangle -P FORWARD DROP

$IPT -t mangle -P POSTROUTING DROP

for Filter in /proc/sys/net/ipv4/conf/*/rp_filter; do

    echo 1 > $Filter

done

echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

exit 0

fi

if [ "$1" = "start" ]

then

$IPT -N CHECK

$IPT -F CHECK

$IPT -N MAC_IP

$IPT -F MAC_IP

$IPT -A OUTPUT -j ACCEPT

$IPT -A OUTPUT -j CHECK

$IPT -A INPUT -j CHECK

$IPT -A FORWARD -j CHECK

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A CHECK -m limit --limit 1/s -m state --state NEW -p TCP --tcp-flags !
ALL SYN -j LOG --log-prefix="INVALIDE_SYNC "

$IPT -A CHECK -m state --state NEW -p TCP --tcp-flags ! ALL SYN -j DROP

$IPT -A CHECK -m limit --limit 1/s -m state --state INVALID -j LOG --log-
prefix="INVALID_CONNECTION "

```

```
$IPT -A CHECK -m state --state INVALID -j DROP

$IPT -A CHECK -m state --state RELATED,ESTABLISHED -j ACCEPT

cat < /etc/mac_ip | while true
do
    read ligne
    if [ "$ligne" = "" ]; then break; fi
    source=$(echo $ligne | cut -d \ -f 1)
    mac=$(echo $ligne | cut -d \ -f 2)
    $IPT -A MAC_IP -s $source -m mac --mac-source $mac -j RETURN
done
$IPT -A MAC_IP -j DROP
$IPT -A INPUT -i $EXT_IF -j MAC_IP
$IPT -A INPUT -i $EXT_IF -p tcp --dport 22 -j ACCEPT
fi
```