conf-amon - Tâche #14083

Proposition Scénario # 13967 (Classée sans suite): Agregation de lien - mauvais routage des réponses aux paquets entrants vers la DMZ

Intégrer les modification de Jean-Marc Melet

19/11/2015 09:47 - Emmanuel GARETTE

Statut:	En cours	**Début:**	19/11/2015
Priorité:	Normal	**Echéance:**	
Assigné à:	Karim Ayari	**% réalisé:**	0%
Version cible:		**Temps estimé:**	6.00 heures
		Temps passé:	0.00 heure

Description

Je joins notre version du script, il comporte aussi plusieurs autres modifications (envole, réseau global, routes forcées, masque agriates...) mais toutes sont commentée en "# AixMars"

Historique

#1 - 20/11/2015 12:19 - Emmanuel GARETTE

- Fichier agregation.sh supprimé

#2 - 20/11/2015 12:23 - Emmanuel GARETTE

Précision sur les modifications utiles :

Fonction balance:

```
balance () {
    iptablesmangleclear $1
    # AixMars: Si reseau pedago, prise en compte de l'adressage global du reseau (1)
    if [ $1 == "eth2" ] && [ -n "$adresse_global_vlanpeda" ] ; then
        network="$(eval echo \$adresse_global_vlanpeda)/$(eval echo \$netmask_global_vlanpeda)"
    else
        network="$(eval echo \$adresse_network_$1)/$(eval echo \$adresse_netmask_$1)"
    fi
    # AixMars: Forcer Envole sur lien 1 (2)

    if [ $1 == "eth2" ] && [ $ent == "local-peda" ] ; then
        /sbin/iptables -t mangle -A PREROUTING -i $1 -s $ip_scribe_peda -j T1
        /sbin/iptables -t mangle -A PREROUTING -i $1 -s $ip_scribe_peda -j RETURN
        /sbin/iptables -t mangle -I PREROUTING -i $1 -p tcp -m tcp --sport 4200 -j CONNMARK --save-mark
        /sbin/iptables -t mangle -I PREROUTING -i $1 -p tcp -m tcp --sport 4200 -j MARK --set-mark 1
    fi
    #Si non NEW
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state ! --state NEW -j RESTOREMARK
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state ! --state NEW -j RETURN

    # AixMars: regles pour les destinations forçées en insert au lieu de append pour qu'on puisse forcer des r
outes sur le lien 2 pour Scribe même si envole est forcé sur le lien 1 (ex: SMTP) (3)
    #Routes forcees
    #for ip_force1 in ${FORCE1[@]}; do
    #    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -d $ip_force1 -m state --state NEW -j T1
    #    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -d $ip_force1 -m state --state NEW -j RETURN
 ; done
    #for ip_force2 in ${FORCE2[@]}; do
    #    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -d $ip_force2 -m state --state NEW -j T2
    #    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -d $ip_force2 -m state --state NEW -j RETURN
 ; done
    for ip_force1 in ${FORCE1[@]}; do
        /sbin/iptables -t mangle -I PREROUTING -i $1 -s $network -d $ip_force1 -j RETURN
        /sbin/iptables -t mangle -I PREROUTING -i $1 -s $network -d $ip_force1 -j T1 ; done
    for ip_force2 in ${FORCE2[@]}; do
        /sbin/iptables -t mangle -I PREROUTING -i $1 -s $network -d $ip_force2 -j RETURN
        /sbin/iptables -t mangle -I PREROUTING -i $1 -s $network -d $ip_force2 -j T2 ; done
    #Si NEW et recent alors Tag puis RETURN
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -m recent --name T1 --update
```

```
 --rdest --seconds 3600 -j T1
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -m recent --name T1 --update
 --rdest --seconds 3600 -j RETURN
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -m recent --name T2 --update
 --rdest --seconds 3600 -j T2
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -m recent --name T2 --update
 --rdest --seconds 3600 -j RETURN
    #Si NEW sans recent alors Tag puis RETURN
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -j T2
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -m statistic --mode random -
-probability 0.$WCO -m recent --name T2 --set --rdest -j RETURN
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -m recent --name T1 --set --
rdest -j T1

}
```

(1) Evolution pour prise en compte d'un réseau global de machines situées derrière un équipement de routage relié à l'interface eth2 d'Amon. Sur certaines de nos architectures, l'interface eth2 d'Amon est située dans un VLAN d'interconnexion avec un routing switch sur lequel sont reliés plusieurs VLANs du réseau pédagogique. Il a donc fallu prendre en compte ce cas là pour permettre le marquage des trames provenant des IP de l'ensemble de ces VLAN et non seulement du réseau d'interconnexion d'Amon (adresse_network_eth2/adresse_netmask_eth2 par défaut). La même évolution a été apportée aux fonctions wan1 et wan2 pour étendre ce fonctionnement en cas de perte d'un lien et de bascule sur l'autre

(2) Evolution pour forcer envole sur le lien 1: cela a été nécéssaire pour nous car historiquement nos ENT des Scribes se trouvent sur la zone pédagogique et, de la même façon que pour les accès à la DMZ publique, les connexions depuis les Scribes doivent utiliser la table T1 pour sortir sur le même lien que les accès entrants vers l'ENT.

(3) Modification des regles pour les destinations forçées: passage de append en insert pour que ces regles soient prioritaires sur celles pour forcer envole sur le lien 1. Ceci a été nécessaire dans notre cas pour pouvoir faire sortir les flux SMTP depuis les Scribes sur le lien 2 car le SMTP du FAI du lien 1 imposait une authentification pour envoyer des mails.

Fonction wan1:

```
wan1 () {
    iptablesmangleclear $1
    # AixMars: Si reseau pedago, prise en compte de l'adressage global du reseau (4)
    if [ $1 == "eth2" ] && [ -n "$adresse_global_vlanpeda" ] ; then
        network="$(eval echo \$adresse_global_vlanpeda)/$(eval echo \$netmask_global_vlanpeda)"
    else
        network="$(eval echo \$adresse_network_$1)/$(eval echo \$adresse_netmask_$1)"
    fi
    # AixMars: Si DMZ publique, marquage en T1 pour tous les paquets (pas de suivi de marquage possible) (5)
    if [ $1 == "eth4" ] ; then
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -j T1
    else
    #Si non NEW
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state ! --state NEW -j RESTOREMARK
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state ! --state NEW -j RETURN
    #Marques sur T1
    /sbin/iptables -t mangle -A PREROUTING -i $1 -s $network -m state --state NEW -j T1
    fi
}
```

(4) Comme pour (1), prise en compte d'un réseau global de machines situées derrière un équipement de routage relié à l'interface eth2 d'Amon. Idem pour la fonction wan2

(5) Forcer le marquage sur la chaine T1 pour tous les paquets. Ceci nous a apparu utile notamment pour les retours de connexion depuis l'extérieur car dans ce cas là, les regles "Si non NEW" présentes essayent de restorer une marque sur des paquets liés à une connexion mais pour laquelle aucune marque n'existe puisqu'ils appartiennent à des connexions venues de l'extérieur (pas de marquage sur ces paquets)

Regles de routage:

```
# Chargement des regles de routage
/sbin/ip route del default

ipruleclear T1
/sbin/ip rule add from $WAN1 table T1
/sbin/ip route add $NET1 dev eth0 src $WAN1 table T1
```

```
# AixMars: ajout des routes vers les réseaux internes pour la table de routage T1 (6)
/sbin/ip route add "$(eval echo \$alias_network_eth0)/$(eval echo \$alias_netmask_eth0)" dev eth0 src $alias_i
p_eth0 table T1
/sbin/ip route add "$(eval echo \$adresse_network_eth1)/$(eval echo \$adresse_netmask_eth1)" dev eth1 src $adr
esse_ip_eth1 table T1
[ $nombre_interfaces -ge 3 ] && /sbin/ip route add "$(eval echo \$adresse_network_eth2)/$(eval echo \$adresse_
netmask_eth2)" dev eth2 src $adresse_ip_eth2 table T1
[ $nombre_interfaces -ge 4 ] && /sbin/ip route add "$(eval echo \$adresse_network_eth3)/$(eval echo \$adresse_
netmask_eth3)" dev eth3 src $adresse_ip_eth3 table T1
[ $nombre_interfaces -eq 5 ] && /sbin/ip route add "$(eval echo \$adresse_network_eth4)/$(eval echo \$adresse_
netmask_eth4)" dev eth4 src $adresse_ip_eth4 table T1
/sbin/ip route add default via $GW1 table T1
/sbin/ip rule add fwmark 1 table T1
```

(6) Evolution pour permettre le routage entre les zones internes dans la table T1. Je ne sais pas si c'est encore nécessaire mais à l'époque, nous avions du l'ajouter pour permettre le routage des flux entre les zones internes. Idem pour la table T2.

Chaines de marquage:

```
## creation de la chaine marquage pour agregation de lien
chaine_T1=$(iptables-save | grep ":T1")
if [ -z "$chaine_T1" ] ; then
/sbin/iptables -t mangle -N T1
# AixMars: Exclusion des destinations des reseaux internes du marquage (7)
/sbin/iptables -t mangle -A T1 -d "$(eval echo \$adresse_network_eth1)/$(eval echo \$adresse_netmask_eth1)" -j
 RETURN
/sbin/iptables -t mangle -A T1 -d "$(eval echo \$adresse_network_eth2)/$(eval echo \$adresse_netmask_eth2)" -j
 RETURN
/sbin/iptables -t mangle -A T1 -d "$(eval echo \$adresse_network_eth3)/$(eval echo \$adresse_netmask_eth3)" -j
 RETURN
/sbin/iptables -t mangle -A T1 -d "$(eval echo \$adresse_network_eth4)/$(eval echo \$adresse_netmask_eth4)" -j
 RETURN
# AixMars: Modification du masque de sous-réseau du tunnel agriates en 10.0.0.0/16 pour éviter d'englober les
DNS privés (8)
#/sbin/iptables -t mangle -A T1 -d 10.0.0.0/8 -j RETURN
/sbin/iptables -t mangle -A T1 -d 10.0.0.0/16 -j RETURN
/sbin/iptables -t mangle -A T1 -d 172.16.0.0/12 -j RETURN
/sbin/iptables -t mangle -A T1 -d 192.168.0.0/16 -j RETURN
/sbin/iptables -t mangle -A T1 -d 161.48.0.0/19 -j RETURN
/sbin/iptables -t mangle -A T1 -j MARK --set-mark 1
/sbin/iptables -t mangle -A T1 -j CONNMARK --save-mark
fi
```

(7) Exclure du marquage les trames pour les destinations vers les réseaux internes, même si à priori ça ne doit pas être utile car il y a déja les regles qui excluent les classes d'adresses privées. Idem pour les chaines T2 et RESTOREMARK
(8) Il a fallu modifier le masque de la classe privée A en 10.0.0.0/16 car nous avons des flux vers des DNS de collectivité en IP privée qui étaient englobés et envoyés dans le tunnel VPN agriates. Idem pour les chaines T2 et RESTOREMARK

Envoi des mails d'alerte:

```
if [ "$ag_active_mail" == "oui" ] ; then
    MssG="Seul le lien 2 est actif, redirection des flux sur ce lien"
    SubJ="Liaison $nom_domaine_local_supp  ($numero_etab)"
    # AixMars: export valeurs dico pour mutt (9)
    export libelle_etab=$libelle_etab
    export nom_domaine_local=$nom_domaine_local
    if [ -z "${CC[@]}" ] ; then
        # AixMars: Modif sender commande mutt (10)
        #echo "$MssG"|mutt -s "$SubJ" "$DEST"
        echo "$MssG"|mutt -n -e 'my_hdr From:Agregation Amon $libelle_etab <amon-agregation.$nom_domaine_local
@ac-aix-marseille.fr>' -s "$SubJ" "$DEST"
    else
        # AixMars: Modif sender commande mutt (11)
        #echo "$MssG"|mutt -s "$SubJ" "$DEST" -c "${CC[@]}"
        echo "$MssG"|mutt -n -e 'my_hdr From:Agregation Amon $libelle_etab <amon-agregation.$nom_domaine_local
@ac-aix-marseille.fr>' -s "$SubJ" "$DEST" -c "${CC[@]}"
    fi
```

```
fi
```

(9),(10),(11) Modifications de l'affichage du sender dans l'envoi des mails d'alerte (cosmétique). Modif faite pour le basculement sur le lien 1, lien 2 et rechargement sur les 2 liens.

Répartition de charge:

```
if [ $ag_mode == "mode_lb" ] ; then
    # Iproute2 sur les 2 liens
    /sbin/ip route replace default proto static nexthop via $GW1 dev eth0 weight $W1 nexthop via $GW2 dev eth0
 weight $W2

    # retablit les destination forcees lien 1
    for ip_force in ${FORCE1[@]} ; do
        /sbin/ip route replace $ip_force via $GW1 dev eth0
        done
        # retablit les destination forcees lien 2
        for ip_force in ${FORCE2[@]} ; do
            /sbin/ip route replace $ip_force via $GW2 dev eth0
        done

    # Mangle sur les 2 liens
    balance eth1
    [ $nombre_interfaces -ge 3 ] && balance eth2
    # AixMars: Redirection sur le lien 1 pour les DMZ (12)
    #[ $nombre_interfaces -ge 4 ] && wan2 eth3
    #[ $nombre_interfaces -eq 5 ] && wan2 eth4
    [ $nombre_interfaces -ge 4 ] && wan1 eth3
    [ $nombre_interfaces -eq 5 ] && wan1 eth4
    # AixMars: forcer la destination du réseau interco WAN sur la chaine T1 (13)
    iptables -t mangle -I PREROUTING -d "$(eval echo \$adresse_network_eth0)/$(eval echo \$adresse_netmask_eth
0)" -j RETURN
    iptables -t mangle -I PREROUTING -d "$(eval echo \$adresse_network_eth0)/$(eval echo \$adresse_netmask_eth
0)" -j T1
```

(12) Pour le mode load balancing, utilisation de la table T1 pour les DMZ pour les forcer sur le lien 1 et assurer notamment le retour de connexion pour la DMZ publique. Idem pour le mode fail over

(13) Pour le mode load balancing, forcer le marquage sur la chaine T1 pour les paquets à destination du réseau sur l'interface eth0. Il a fallu ajouter ces regles pour pouvoir toujours atteindre le réseau de l'interface eth0 depuis les réseaux internes, ce qui permet aussi d'éviter un problème pour la diffusion du wpad (sans cela les requetes des postes clients pour récupérer le wpad sur le reverse proxy sur eth0 échouaient lorsque les paquets étaient marqués en T2). Idem pour le mode fail over.

#3 - 25/03/2017 09:57 - Karim Ayari

- Statut changé de Nouveau à En cours

6 : corrigé par #19643
12 : corrigé par #14123
13 : pas de problème constaté chez nous en 2.5.2 par exemple que ce soit en fo ou lb wpad est toujours accessible
je regarderais le reste

#4 - 25/03/2017 09:57 - Karim Ayari

- Assigné à mis à Karim Ayari