

## conf-amon - Anomalie #15268

### Agregation 2.4.2 : mauvais résultat dans la table mangle lorsque qu'un lien tombe

02/03/2016 16:47 - équipe eole Academie d'Orléans-Tours

<b>Statut:</b>	Fermé	<b>Début:</b>	02/03/2016
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Emmanuel GARETTE	<b>% réalisé:</b>	100%
<b>Catégorie:</b>		<b>Temps estimé:</b>	1.00 heure
<b>Version cible:</b>		<b>Temps passé:</b>	1.00 heure
<b>Distribution:</b>	EOLE 2.5		

#### Description

Sur un 2.4.2, l'on constate un soucis en table "mangle" lors qu'un lien tombe. Actuellement l'étab en question n'a plus du tout son lien secondaire.

On constate :

```
iptables -t mangle -L -nv
Chain PREROUTING (policy ACCEPT 357K packets, 316M bytes)
 pkts bytes target      prot opt in      out     source        destination
  0     0 RESTOREMARK all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
INVALID, RELATED, ESTABLISHED, UNTRACKED
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state INV
ALID, RELATED, ESTABLISHED, UNTRACKED
  0     0 T1          all  --  eth1     *       192.168.18.0/24  195.83.89.136
state NEW
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  195.83.89.136
state NEW
  0     0 T1          all  --  eth1     *       192.168.18.0/24  195.83.89.137
state NEW
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  195.83.89.137
state NEW
  0     0 T1          all  --  eth1     *       192.168.18.0/24  195.83.89.162
state NEW
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  195.83.89.162
state NEW
  0     0 T1          all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T1 side: dest
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T1 side: dest
  0     0 T2          all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T2 side: dest
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T2 side: dest
  0     0 T2          all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
  0     0 RETURN      all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
statistic mode random probability 0.500000000000 recent: SET name: T2 side: dest
  0     0 T1          all  --  eth1     *       192.168.18.0/24  0.0.0.0/0
state NEW
recent: SET name: T1 side: dest
60881 3370K RESTOREMARK all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state I
INVALID, RELATED, ESTABLISHED, UNTRACKED
60881 3370K RETURN      all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state INV
ALID, RELATED, ESTABLISHED, UNTRACKED
  0     0 T1          all  --  eth2     *       172.20.0.0/16    195.83.89.136
state NEW
  0     0 RETURN      all  --  eth2     *       172.20.0.0/16    195.83.89.136
state NEW
  0     0 T1          all  --  eth2     *       172.20.0.0/16    195.83.89.137
state NEW
  0     0 RETURN      all  --  eth2     *       172.20.0.0/16    195.83.89.137
state NEW
  0     0 T1          all  --  eth2     *       172.20.0.0/16    195.83.89.162
state NEW
  0     0 RETURN      all  --  eth2     *       172.20.0.0/16    195.83.89.162
state NEW
1632 93689 T1          all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T1 side: dest
1632 93689 RETURN      all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T1 side: dest
1205 65945 T2          all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T2 side: dest
1205 65945 RETURN      all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state NEW
recent: UPDATE seconds: 3600 name: T2 side: dest
  47  2991 T2          all  --  eth2     *       172.20.0.0/16    0.0.0.0/0
state NEW
```

```

 25 1351 RETURN      all -- eth2 *      172.20.0.0/16      0.0.0.0/0          state NEW
statistic mode random probability 0.500000000000 recent: SET name: T2 side: dest
 22 1640 T1         all -- eth2 *      172.20.0.0/16      0.0.0.0/0          state NEW
recent: SET name: T1 side: dest

```

Chain INPUT (policy ACCEPT 356K packets, 316M bytes)

```

pkts bytes target      prot opt in      out      source      destination
458K 405M marquage    all -- *      *      0.0.0.0/0    0.0.0.0/0

```

Chain FORWARD (policy ACCEPT 130 packets, 9090 bytes)

```

pkts bytes target      prot opt in      out      source      destination
556 347K marquage    all -- *      *      0.0.0.0/0    0.0.0.0/0

```

Chain OUTPUT (policy ACCEPT 356K packets, 316M bytes)

```

pkts bytes target      prot opt in      out      source      destination
459K 405M marquage    all -- *      *      0.0.0.0/0    0.0.0.0/0

```

Chain POSTROUTING (policy ACCEPT 357K packets, 316M bytes)

```

pkts bytes target      prot opt in      out      source      destination

```

Chain RESTOREMARK (2 references)

```

pkts bytes target      prot opt in      out      source      destination

```

Chain T1 (10 references)

```

pkts bytes target      prot opt in      out      source      destination

```

Chain T2 (4 references)

```

pkts bytes target      prot opt in      out      source      destination

```

Chain marquage (3 references)

```

pkts bytes target      prot opt in      out      source      destination

```

Dans les logs :

```

2016-03-02_15:58:39 L'etat du lien 2 est bien change (essai 3/4)
2016-03-02_15:58:40 Le lien 2 n'est plus actif
2016-03-02_15:58:40 Seul le lien 1 est actif, redirection des flux sur ce lien
2016-03-02_15:59:01 Le lien 2 est tombe

```

Le résultat dans la table mangle pose soucis, car des paquets sont encore marqué en T2 alors que le lien est down, il y a encore 4 référence à cette chaîne.

Sur un AMON 2.3 dans la même situation, on voit que mangle est bien géré :

```
iptables -t mangle -L -nv
```

Chain PREROUTING (policy ACCEPT 75M packets, 58G bytes)

```

pkts bytes target      prot opt in      out      source      destination      state
1589K 537M RESTOREMARK all -- eth1 *      10.18.43.0/24    0.0.0.0/0          state IN
VALID,RELATED,ESTABLISHED,UNTRACKED
1589K 537M RETURN      all -- eth1 *      10.18.43.0/24    0.0.0.0/0          state INVA
LID,RELATED,ESTABLISHED,UNTRACKED
97209 6240K T1             all -- eth1 *      10.18.43.0/24    0.0.0.0/0          state NEW
6401K 793M RESTOREMARK all -- eth2 *      172.20.112.0/22    0.0.0.0/0          state IN
VALID,RELATED,ESTABLISHED,UNTRACKED
6401K 793M RETURN      all -- eth2 *      172.20.112.0/22    0.0.0.0/0          state INVA
LID,RELATED,ESTABLISHED,UNTRACKED
266K 20M T1          all -- eth2 *      172.20.112.0/22    0.0.0.0/0          state NEW
1031K 1057M RESTOREMARK all -- eth3 *      10.118.43.160/28    0.0.0.0/0          state IN
VALID,RELATED,ESTABLISHED,UNTRACKED
1031K 1057M RETURN      all -- eth3 *      10.118.43.160/28    0.0.0.0/0          state INVA
LID,RELATED,ESTABLISHED,UNTRACKED

```

```

6183 425K T1      all  --  eth3  *      10.118.43.160/28  0.0.0.0/0      state NEW
Chain INPUT (policy ACCEPT 70M packets, 55G bytes)
pkts bytes target  prot opt in      out     source      destination
46M  36G marquage  all  --  *      *      0.0.0.0/0  0.0.0.0/0
Chain FORWARD (policy ACCEPT 5232K packets, 2594M bytes)
pkts bytes target  prot opt in      out     source      destination
3628K 1822M marquage  all  --  *      *      0.0.0.0/0  0.0.0.0/0
Chain OUTPUT (policy ACCEPT 70M packets, 61G bytes)
pkts bytes target  prot opt in      out     source      destination
45M  39G marquage  all  --  *      *      0.0.0.0/0  0.0.0.0/0
Chain POSTROUTING (policy ACCEPT 74M packets, 63G bytes)
pkts bytes target  prot opt in      out     source      destination
0      0      RETURN  all  --  *      *      0.0.0.0/0  0.0.0.0/0      LAYER7 17p
roto bittorrent
8219 606K      all  --  *      *      0.0.0.0/0  0.0.0.0/0      LAYER7 17p
roto edonkey
Chain RESTOREMARK (3 references)
pkts bytes target  prot opt in      out     source      destination
1638K 987M RETURN  all  --  *      *      0.0.0.0/0  10.0.0.0/8
6156K 1189M RETURN  all  --  *      *      0.0.0.0/0  172.16.0.0/12
19653 1436K RETURN  all  --  *      *      0.0.0.0/0  192.168.0.0/16
0      0      RETURN  all  --  *      *      0.0.0.0/0  161.48.0.0/19
1208K 210M CONNMARK  all  --  *      *      0.0.0.0/0  0.0.0.0/0      CONNMARK r
estore
Chain T1 (3 references)
pkts bytes target  prot opt in      out     source      destination
75624 5231K RETURN  all  --  *      *      0.0.0.0/0  10.0.0.0/8
252K  17M RETURN  all  --  *      *      0.0.0.0/0  172.16.0.0/12
8823 2189K RETURN  all  --  *      *      0.0.0.0/0  192.168.0.0/16
0      0      RETURN  all  --  *      *      0.0.0.0/0  161.48.0.0/19
33450 2160K MARK    all  --  *      *      0.0.0.0/0  0.0.0.0/0      MARK xset
0x1/0xffffffff
33450 2160K CONNMARK  all  --  *      *      0.0.0.0/0  0.0.0.0/0      CONNMARK s
ave
Chain T2 (0 references)
pkts bytes target  prot opt in      out     source      destination
0      0      RETURN  all  --  *      *      0.0.0.0/0  10.0.0.0/8
0      0      RETURN  all  --  *      *      0.0.0.0/0  172.16.0.0/12
0      0      RETURN  all  --  *      *      0.0.0.0/0  192.168.0.0/16
0      0      RETURN  all  --  *      *      0.0.0.0/0  161.48.0.0/19
0      0      MARK    all  --  *      *      0.0.0.0/0  0.0.0.0/0      MARK xset
0x2/0xffffffff
0      0      CONNMARK  all  --  *      *      0.0.0.0/0  0.0.0.0/0      CONNMARK s
ave

```

Aucune référence pour T2, ce qui est logique.

Je n'ai pas trouvé exactement le soucis dans le script de la version 2.4, mais en AMON 2.3, on a les fonctions wan1 () et wan2 () qui clear la table mangle et ajoutent seulement leur propre marquage. Ces fonctions sont utilisés lors de la détections de liens down/up. Je ne retrouve pas ce type d'appel dans la version 2.4 du script.

Merci d'avance,

Nicolas

#### Demandes liées:

Lié à Distribution EOLE - Tâche #20057: Proposer un paquet 2.5 contenant les ...

Fermé

03/04/2017

---

## Révisions associées

---

### Révision 4a1fdf90 - 17/03/2017 16:55 - Karim Ayari

Règles mangle seulement pour le lien OK

Dans le cas d'une bascule de lien.

Ref: #15268

### Révision 11e2e9fd - 05/04/2017 10:48 - Karim Ayari

Règles mangle seulement pour le lien OK

Dans le cas d'une bascule de lien.

Ref: #15268

---

## Historique

---

### #1 - 03/03/2016 09:13 - Gérald Schwartzmann

- *Sujet changé de agregation 2.4.2 à Agregation 2.4.2 : mauvais résultat dans la table mangle lors qu'un lien tombe*

### #2 - 07/03/2016 11:54 - Scrum Master

- *Assigné à mis à Emmanuel GARETTE*

Demander l'avis au contributeur.

### #3 - 07/03/2016 15:06 - Karim Ayari

je vais tester cela dès que possible.

### #4 - 16/03/2016 11:11 - Gérald Schwartzmann

- *Sujet changé de Agregation 2.4.2 : mauvais résultat dans la table mangle lors qu'un lien tombe à Agregation 2.4.2 : mauvais résultat dans la table mangle lorsque qu'un lien tombe*

### #5 - 06/04/2016 14:57 - Joël Cuissinat

- *Projet changé de Amon à conf-amon*

### #6 - 08/04/2016 11:07 - Emmanuel GARETTE

- *Statut changé de Nouveau à En attente d'informations*

Des nouvelles ?

### #7 - 24/05/2016 18:44 - Karim Ayari

je vais avoir un peu de temps pour regarder.

### #8 - 04/07/2016 15:05 - équipe eole Academie d'Orléans-Tours

Un rapport avec <https://dev-eole.ac-dijon.fr/issues/10017> ? Comme tu l'as conseillé Karim l'on va repousser le script V2.3 sur les 2.4 et 2.5 via notre variante en attendant.



```
+ '[' -n '' ]'  
+ let i++  
++ /sbin/iptables -t mangle -S PREROUTING  
++ wc -l  
+ '[' 9 -le 14 ]'  
++ /sbin/iptables -t mangle -S PREROUTING 9  
++ grep ' -i eth1.10 '  
+ '[' -n '' ]'  
+ let i++  
++ /sbin/iptables -t mangle -S PREROUTING  
++ wc -l  
+ '[' 10 -le 14 ]'  
++ /sbin/iptables -t mangle -S PREROUTING 10  
++ grep ' -i eth1.10 '  
+ '[' -n '' ]'  
+ let i++  
++ /sbin/iptables -t mangle -S PREROUTING  
++ wc -l  
+ '[' 11 -le 14 ]'  
++ /sbin/iptables -t mangle -S PREROUTING 11  
++ grep ' -i eth1.10 '  
+ '[' -n '' ]'  
+ let i++  
++ wc -l  
++ /sbin/iptables -t mangle -S PREROUTING  
+ '[' 12 -le 14 ]'  
++ /sbin/iptables -t mangle -S PREROUTING 12  
++ grep ' -i eth1.10 '  
+ '[' -n '' ]'  
+ let i++  
++ /sbin/iptables -t mangle -S PREROUTING  
++ wc -l  
+ '[' 13 -le 14 ]'  
++ grep ' -i eth1.10 '  
++ /sbin/iptables -t mangle -S PREROUTING 13  
+ '[' -n '' ]'  
+ let i++  
++ /sbin/iptables -t mangle -S PREROUTING  
++ wc -l  
+ '[' 14 -le 14 ]'  
++ /sbin/iptables -t mangle -S PREROUTING 14  
++ grep ' -i eth1.10 '  
+ '[' -n '' ]'  
+ let i++  
++ /sbin/iptables -t mangle -S PREROUTING  
++ wc -l  
+ '[' 15 -le 14 ]'
```

je vais revoir la fonction iptablesmangleclear

**#11 - 22/03/2017 11:45 - Karim Ayari**

- *Tracker changé de Demande à Anomalie*
- *Statut changé de En attente d'informations à Résolu*
- *% réalisé changé de 0 à 100*
- *Temps estimé mis à 1.00 h*
- *Distribution mis à EOLE 2.5*

la fonction `active_link_to` qui va flusher les règles de PREROUTING n'était jamais appelée lors de la bascule de lien c'est corrigé

**#12 - 22/03/2017 11:45 - Karim Ayari**

- *Statut changé de Résolu à Fermé*

**#13 - 03/04/2017 14:24 - Joël Cuissinat**

- *Lié à Tâche #20057: Proposer un paquet 2.5 contenant les modifications réalisées sur les scripts d'agrégation ajouté*