tetaneutral.net - Evolution #152

STRI iptables arpwatch

20/02/2012 01:03 - Laurent GUERBY

Statut:	Résolu	**Début:**	20/02/2012
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0.00 heure
Version cible:			

Description

Proposer un jeu de regles iptable pour eviter de router des RFC1918. Mettre en place un outil de type arpwatch sur le VLAN 3131 sur h2 ou une VM dediee.

Historique

#1 - 20/02/2012 15:45 - Brian Tur

Bonjour ! j'ai commencé à réfléchir à la première partie pour éviter de router des RFC 1918 (les IP privées).

J'en ai déduis les 4 plages d'adresses à bloquer :

10.0.0.0/8
127.0.0.0/8
172.16.0.0/12
192.168.0.0/16

Je serai tenter d'intégrer du coup les règles suivante sur le routeur h2 :

```
iptables -A INPUT -i br0 -s 10.0.0.0/8 -j DROP
  iptables -A INPUT -i br0 -s 127.0.0.0/8 -j DROP
  iptables -A INPUT -i br0 -s 172.16.0.0/12 -j DROP
  iptables -A INPUT -i br0 -s 192.168.0.0/16 -j DROP
```

Ces 4 règles bloqueraient le trafic entrant sur h2 depuis des adresses privées.

```
iptables -A OUTPUT -o br0 -d 10.0.0.0/8 -j REJECT
  iptables -A OUTPUT -o br0 -d 127.0.0.0/8 -j REJECT
  iptables -A OUTPUT -o br0 -d 172.16.0.0/12 -j REJECT
  iptables -A OUTPUT -o br0 -d 192.168.0.0/16 -j REJECT
```

Ces 4 règles bloqueraient le trafic sortant vers des @Privées.

Toutefois cà me parait être un peu simple à mon goût et comme je débute dans iptables je préfère ne pas prendre de risque et demander votre avis !

En découle quelques questions :

```
- Est-ce que l'interface d'entrée et de sortie du trafic est br0 ou eth0 sur h2 ?
   - Ces règles vous semblent-elles correctes ?
   - Il y a-t-il particulièrement des aspects à faire très attention lors de l'ajout de règles iptables ? (je
ne souhaites pas créer un problème critique sur h2 !!^^)
```

Merci.

#2 - 21/02/2012 09:54 - Laurent GUERBY

Salut Brian,

Note : h2 n'est pas un routeur, les routeurs sont h3 et gw.

Comme tu l'as proposé on va discuter ici avant de faire quoique ce soit sur ces routeurs dont le fonctionnement est critique pour notre réseau :).

Je pense que c'est plutot FORWARD comme chaine non ? De mémoire INPUT et OUTPUT sont pour des paquets générés localement a la machine mais je peux me tromper.

Ce qui nous interesse aussi est d'avoir des logs de qui emet des paquets a des RFC1918 pour pouvoir prevenir nos membres.

#3 - 21/02/2012 10:40 - Brian Tur

Oui,

en effet je ne sais pas pourquoi j'ai bugué en pensant que c'était h2 ...

Pour le FORWARD je suis plutôt d'accord en effet, c'est la chaîne qui autorise ou bloque directement le routage au sein du routeur. INPUT permet de bloquer le trafic à l'entrée du routeur et OUTPUT en sortie.

Je me pose une question quant aux politiques de bases de ces chaînes : Part-on du principe que l'on bloque tout le trafic (POLICY DROP) sur les chaines et on autorise ensuite le trafic voulu au cas par cas avec des iptables ? Ou alors on autorise tout le trafic par défaut (POLICY ACCEPT) et on bloque le trafic non voulu avec des règles drop ?

Ensuite je me suis renseigné sur pourquoi bloquer des paquets RFC1918. J'ai pu comprendre que c'est pour éviter des attaques de type déni de services par Spoofing ou de type usurpation d'identité (@IP) ! Est-ce que je suis à coté de la plaque lol ? il y a t-il d'autres raisons à la mise en place de ces filtres ?

Ca reste un peu vague encore, mais ça commence à s'éclaircir :).

#4 - 21/02/2012 11:55 - Laurent GUERBY

Pour la bonne POLICY je ne suis pas un pro d'iptables, a toi de chercher une solution :)

Pour la raison de bloquer il y en a plusieurs :
- actuellement si tu fais un traceroute 192.168.3.1 depuis une machine ou VM de 91.224.149.0/24 ca va marcher et on veut l'eviter car c'est un VLAN admin
- en effet si on recoit des paquets a destination de RFC1918 depuis des interface edge autant les dropper tout de suite ca ne sert a rien de saturer notre reseau interne

Pense a regarder la partie log / alerte, et ensuite arpwatch :)

#5 - 21/02/2012 21:42 - Philippe Latu

Bonsoir,

Les deux options de politiques de filtrage sont les grands classiques de cours.

1. Tout ce qui n'est pas interdit est autorisé

Si on se réfère aux critères académique sur la question, cette politique est à proscrire.
Si on adopte un point de vue plus pragmatique, cette politique est à appliquer en tête de réseau (couche coeur) suivant deux critères :
. On ne dispose pas d'une liste exhaustive des flux valides. On ne peut donc pas appliquer la politique n°2
. On souhaite dégrossir le filtrage en éliminant le «tout venant», ce qui permet d'affiner le filtrage dans les couches inférieures dans lesquelles on pourra appliquer la politique n°2 pour des périmètres de petite taille

Généralement, j'utilise l'analogie des tamis de plus en plus fin au fur et à mesure où l'on se rapproche de «l'actif à protéger» comme on dit maintenant.

2. Tout ce qui n'est pas autorisé est interdit

Si on se réfère aux critères académique sur la question, c'est **la** politique à utiliser.

Elle implique que l'on ait une connaissance exhaustive des flux valides !

Voilà pour le rappel ... de cours.
Dans notre cas, je ne suis pas sûr que Brian ait à sa disposition la fameuse liste exhaustive ...

Voici ce que j'utilise sur Cooper
(Exemples issus du fichier /var/lib/iptables/active appliqué à l'aide de la commande iptables-save)

Table nat : politique n°1 avec traduction d'adresses sources globale vers l'adresse IP publique aaa.bbb.ccc.ddd pour tous les paquets sortants

```
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -p tcp --tcp-flags SYN,RST SYN -m tcpmss --mss 1400:1536 -j TCPMSS --clamp-mss-to-pmtu
-A POSTROUTING -o eth4 -j SNAT --to-source aaa.bbb.ccc.ddd
```

Table filter : politique n°1 uniquement pour la chaîne OUTPUT avec «anti-fuite spécial trafic étudiant»

```
:OUTPUT ACCEPT [0:0]
-A OUTPUT -o eth4 -p tcp --dport 135:139 -j DROP
-A OUTPUT -o eth4 -p tcp --dport 445 -j DROP
-A OUTPUT -o eth4 -p udp --dport 135:139 -j DROP
-A OUTPUT -o eth4 -p udp --dport 445 -j DROP
-A OUTPUT -o eth4 -p tcp -m multiport --sports 20,21,23,25,53,80,110,113 -j DROP
-A OUTPUT -o eth4 -s 127.0.0.0/8 -j LOG --log-prefix "OUTPUT|127/8: "
-A OUTPUT -o eth4 -s 127.0.0.0/8 -j DROP
-A OUTPUT -o eth4 -s 10.0.0.0/8 -j LOG --log-prefix "OUTPUT|10/8: "
-A OUTPUT -o eth4 -s 10.0.0.0/8 -j DROP
-A OUTPUT -o eth4 -s 172.16.0.0/12 -j LOG --log-prefix "OUTPUT|172.16/12: "
```

```
-A OUTPUT -o eth4 -s 172.16.0.0/12 -j DROP
-A OUTPUT -o eth4 -s 192.168.0.0/16 -j LOG --log-prefix "OUTPUT|192.168/16: "
-A OUTPUT -o eth4 -s 192.168.0.0/16 -j DROP
-A OUTPUT -o eth4 -s 224.0.0.0/8 -j DROP
-A OUTPUT -m state --state INVALID -j LOG --log-prefix "OUTPUT|INVALID: "
-A OUTPUT -m state --state INVALID -j DROP
```

Bon, ceci n'est qu'un exemple.

Cordialement,

#6 - 23/02/2012 16:06 - Brian Tur

Bonjour,

voilà ce que j'ai produit pour le moment au niveau d'iptables :

1°/ En premier lieu un script de configuration iptables

_____

```
#!/bin/bash
#Script iptables blocage RFC1918

##Règles iptables

##On flush iptables.

/sbin/iptables -F

##On supprime toutes leschapines utilisateurs.

/sbin/iptables -X

##On accepte les trafic

/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT

##On drop ensuite les paquets à destination de RFC 1918.

/sbin/iptables -A OUTPUT -o eth0 -d 10.0.0.0/8 -j LOG --log-level info --log-prefix "Droped Packet OUTPUT|10/8 :"
/sbin/iptables -A OUTPUT -o eth0 -d 10.0.0.0/8 -j DROP
/sbin/iptables -A OUTPUT -o eth0 -d 127.0.0.0/8 -j LOG --log-level info --log-prefix "Droped Packet OUTPUT|127/8 :"
/sbin/iptables -A OUTPUT -o eth0 -d 127.0.0.0/8 -j DROP
/sbin/iptables -A OUTPUT -o eth0 -d 172.16.0.0/12 -j LOG --log-level info --log-prefix "Droped Packet OUTPUT|172.16/12 :"
/sbin/iptables -A OUTPUT -o eth0 -d 172.16.0.0/12 -j DROP
/sbin/iptables -A OUTPUT -o eth0 -d 192.168.0.0/16 -j LOG --log-level info --log-prefix "Droped Packet OUTPUT|192.168/16 :"
/sbin/iptables -A OUTPUT -o eth0 -d 192.168.0.0/16 -j DROP

exit 0
```

_____

Ce qui donne ça après éxécution du script :

```
root@biran:/home/biran/Desktop/TER/iptables# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
LOG        all  --  anywhere             10.0.0.0/8          LOG level info prefix "Droped Packet OUTPUT|10/8 :"
DROP       all  --  anywhere             10.0.0.0/8
LOG        all  --  anywhere             loopback/8          LOG level info prefix "Droped Packet OUTPUT|127/8 :"
DROP       all  --  anywhere             loopback/8
LOG        all  --  anywhere             172.16.0.0/12       LOG level info prefix "Droped Packet OUTPT|172.16/12"
DROP       all  --  anywhere             172.16.0.0/12
LOG        all  --  anywhere             192.168.0.0/16      LOG level info prefix "Droped Packet OUTPUT|192.168/"
DROP       all  --  anywhere             192.168.0.0/16
```

Pour le moment j'ai décidé d'utiliser la politique dans laquelle on autorise tout le traffic et où on interdit au cas par cas, içi j'interdis le trafic sortant à destination des IP RFC 1918. Le script est pour le moment incomplet voire incorrect au niveau des plans d'adressages et des chaînes .... J'attends

vos commentaires à ce sujet.

2° / Journalisation et alerte iptables

la journalisation des paquets dropés s'effectuent dans le script iptables où on aperçoit les --log-level info --log-prefix. J'ai créé un fichier iptables.log dans le dossier /var/log/ de ma machine de test pour y mettre tout les logs des paquets dropés par les règles. Ces logs sont écrits dans ce fichier en rajoutant dans le fichier /etc/rsyslog.conf la ligne suivante : kern.=alert -/var/log/iptables.log

exemple après avoir fait un ping sur une IP RFC 1918 :

---

root@biran:/home/biran# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted

On dispose désormais d'un fichier dédié aux logs des règles précédemment crées dans lequel on peut trouver la ligne suivante pour le ping précédemment fait :

Feb 23 15:46:16 biran kernel: [22962.178213] Droped Packet
OUTPUT|192.168/IN= OUT=eth0 SRC=172.16.80.72 DST=192.168.1.1 LEN=84
TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=ICMP TYPE=8 CODE=0 ID=9573
SEQ=2

Pour la remontée d'alertes on peut utiliser le système de messages instantané kern.=alert nom_user,nom_user2, nom_user3... à remplir dans rsyslog.conf, qui écrit directement dans le terminal de l'équipement concerné par le nom d'user lorsqu'un paquet est dropé. Cependant un problème se pose avec ce système car un simple ping (plusieurs ICMP à la suite) qui tourne pendant un moment suffit à flooder la machine censer recevoir les notifications !

Une petite question me vient : bloque t'on tout le trafic vers les RFC1918 ? ou seulement certains type de trafic (ICMP, ...) ?

Voilà pour ce début, j'attends vos remarques !
Bonne soirée.

#7 - 28/02/2012 15:02 - Brian Tur

Salut,

je viens de passer à la seconde partie sur ARPwatch. J'ai fais une première étude pour bien cerner l'intérêt d'utiliser un tel outil et j'en conclut que c'est pour éviter l'ARP poisoning et l'usurpation d'identité principalement, il y a t'il d'autres buts précis pour tetaneutral?

Tout comme pour h3 je pense qu'il est important de discuter un peu avant de faire quoi que ce soit sur h2. J'ai déjà regardé quelques tuto sur le fonctionnement de ARPwatch, sa configuration etc...
Si j'ai bien compris je vais devoir implanter un outil de type arpwatch sur le VLAN 3131 de h2 c'est à dire ath0.3131, qu'attendez vous exactement de moi à ce niveau là ?

Merci ! bonne journée :)

#8 - 29/02/2012 15:52 - Philippe Latu

Pour répondre à ta question :

```
# aptitude install arpwatch
```

Il faut que tu désignes l'interface et le réseau concerné dans le fichier de config.

Ensuite, tu fais une copie de la table ARP valide et ... <troll>tu nettoies ton arme pour te préparer à tirer sur le premier qui se pointe avec une @MAC patibulaire</troll>

#9 - 04/03/2012 20:04 - Brian Tur

Bonjour,

je réfléchis plus en détail sur arpwatch. Je l'ai installé "aptitude install arpwatch" après avoir vérifié la présence du paquet libpcap0.8 pour la découverte arp, et j'ai modifié le fichier de conf dans /etc/arpwatch.conf en rajoutant cette ligne d'instruction :

wlan0 -a -n 192.168.1.0/24 -m root

C'est l'adresse de mon réseau local (chez moi) pour mener mes tests à la maison. Je pense utiliser cette ligne pour tetaneutral sur h2 avec un doute pour l'adresse réseau ...

eth0.3131 -a -n 91.224.148.0/23 -m root

Je redémarre ensuite le service arpwatch "/etc/init.d/arpwatch restart" et la table de correspondance IP/MAC se crée instantanément et on peut la consulter dans "/var/lib/arpwatch/wlan0.dat"

Ensuite on peut constater que lorsque l'on change l'@ IP ou MAC d'une machine c'est à dire la paire IP/MAC enregistrée dans cette table on observe ce changement et cette alerte dans le "/var/log/syslog".

Mar 4 19:36:40 biran arpwatch: new station 192.168.1.11 74:f0:6d:3f:79:b2 wlan0

Dans ce cas la machine biran avait pour adresse 192.168.1.57 lors de son enregistrement dans la table ARP de base et le changement d'adresse avec la meme MAC est donc détecté.

Mon problème est le suivant maintenant je ne vois pas trop comment "tirer" sur quelqu'un qui se pointe avec une @MAC inconnue dans la base ?? Je sais que je peux le trouver et le détecter dans le /var/log/syslog mais comment le rejeter ???

Bonne soirée.

#10 - 04/03/2012 20:08 - Laurent GUERBY

Pour le moment on ne souhaite pas rejeter en automatique, il faut avoir une confiance élevée dans le systeme pour le faire en production :).

Une chose a tester : le conflit d'adresse IP, est-ce que tu peux regarder ce qu'il se passe dans les logs d'arpwatch quand tu mets deux machines avec la meme IP sur ton LAN de test ?

#11 - 04/03/2012 21:08 - Brian Tur

Rebonsoir,

Voilà ce que j'obtiens lorsque 2 MAC différentes ont la même IP (192.168.1.10), j'ai effectué un ping sur chaque machine vers un 3ème hote (192.168.1.11)

Mar 4 21:03:25 biran arpwatch: listening on eth0
Mar 4 21:04:03 biran arpwatch: new station 192.168.1.10 78:e7:d1:ce:05:e9 eth0
Mar 4 21:04:25 biran arpwatch: changed ethernet address 192.168.1.10 00:21:cc:4b:90:64 (78:e7:d1:ce:05:e9) eth0

le premier 'new station' correspond à un ping de machine 1(192.168.1.10) vers machine 3 (192.168.1.11)
le second 'new station' correspond à un ping de machine 2(192.168.1.10) vers machine 3 (192.168.1.11)

Sauf erreur de ma part, on dirait qu'arpwatch ne semble pas détecter qu'il y a deux IP identiques au même moment et voit les choses comme si l'adresse IP est affectée à chacun des 2 PC à chaque requête d'un des 2 PC.

#12 - 04/03/2012 21:37 - Laurent GUERBY

Ok merci pour le test, ca sera facile a reperer dans les logs :)

#13 - 13/03/2012 19:28 - Brian Tur

Bonsoir !

Après l'oral de vendredi dernier, il serait en effet intéressant de pouvoir installer ARPwatch sur h2 et d'observer les logs (mettre en prod).
De plus ça me permettra d'enrichir mon rapport et mon oral avec des mises en applications concrètes !

Qu'en pensez vous ? Comment on peut organiser ça si vous êtes d'accord :p?

Bonne soirée !

#14 - 19/03/2012 08:01 - Laurent GUERBY

Salut Brian,

Oui tu peux y aller sur h2 pour arpwatch :)

#15 - 11/09/2012 14:04 - Laurent GUERBY

Stage terminé