

# Altération de Tables ARP (version 1.00)

Traduction française: Jérôme Athias

## Index

Introduction  
Principe d'un Switch  
Paquets (R)ARP  
Altération de Tables ARP  
Entrons dans la réalité  
Les postes de travail sont également vulnérables  
Un scénario pire  
Vaccination de l'empoisonnement ARP  
Conclusion

## Introduction

Ce document est dédié aux tables ARP et comment les altérer à distance. Le document expose également plusieurs implémentations de « ARP poisoning » dans un segment basé sur un pont et un ensemble de méthodes pour s'en protéger.

Comme d'habitude : je ne pourrai être tenu pour responsable de vos actions stupides...

## Principe d'un Switch

Tout d'abord, pour avoir les idées claires, il est très important d'étudier ma/cette théorie.

Nous avons tous entendu parler de cette expression "sniffing"... configurer votre interface réseau en mode promiscuité afin de pouvoir intercepter les paquets destinés à vos voisins, ... Ce que beaucoup de gens ignorent, c'est que vous ne pouvez pas intercepter des paquets sur un segment s'appuyant sur un switch. Et pourquoi cela... ?

Jetons un œil sur la Table suivante :

Hôtes	Port (Sur le Switch)	Adresse MAC	Adresse IP
Hôte1	Port 1	ABCDEF000001	169.254.0.1
Hôte2	Port 2	ABCDEF000002	169.254.0.2
Hôte3	Port 3	ABCDEF000003	169.254.0.3

Exemple:

L'Hôte1 fait tourner un système d'exploitation \*nix avec un serveur FTP, l'Hôte2 est le poste de travail de l'administrateur réseau et l'Hôte3 est mon portable. Ces hôtes sont connectés les uns aux autres par le biais d'un switch de niveau 2. L'expression « niveau 2 » fait référence à la couche liaison (datalink layer) du modèle OSI (Open System Interconnection). Le kernel de l'Hôte2 construit un paquet et la destination sera l' Hôte1, le paquet va ressembler à ceci :

entête TCP  
entête IP  
entête 802.3  
Datagramme

Type de Protocole	IP Source	IP Destination	MAC Source	MAC Destination
entête TCP	X	X	X	X
entête IP	169.254.0.2	169.254.0.1	X	X
entête 802.3	X	X	ABCDEF000002	ABCDEF000001
Datagramme	X	X	X	X

Lorsque le switch sera allumé, il construira une table des adresses MAC de tous les hôtes qui sont physiquement connectés au switch. La table ARP sera (re)construite en envoyant des requêtes ARP sur le réseau, et avec les informations retournées (les réponses ARP), la table ARP sera (re)construite.

Dans mon exemple :

Adresse MAC de l'hôte sur le Port 1 = ABCDEF000001  
Adresse MAC de l'hôte sur le Port 2 = ABCDEF000002  
Adresse MAC de l'hôte sur le Port 3 = ABCDEF000003  
Et ainsi de suite...

Lorsque le switch reçoit un paquet, il ne lira que l'entête 802.x ou etherII car c'est un switch de niveau2... DOH! Ensuite l'adresse MAC destination sera comparée avec sa table ARP locale... et bientôt le switch aura un équivalent. L'hôte derrière le Port1 est la destination et le switch envoie le paquet à l'hôte1. Le paquet ne passera pas à travers l'interface de l'hôte3 !

Avant de continuer avec l'altération de tables ARP, tout d'abord une petite vue d'ensemble des paquets (R)ARP.

## Paquets (R)ARP

Normalement les caches de la base ARP seront mis à jour à partir des réponses ARP. En premier lieu une requête ARP est envoyée sur le réseau et l'hôte approprié répondra avec une réponse ARP. Dans ce paquet se trouve une information nécessaire au rafraichissement du cache ARP. Les paquets ARP sont donc couramment présentés sous deux formats ; les requêtes ARP et les réponses ARP. Voici un dump humainement lisible :



La table MAC va ressembler à ceci:

Adresse MAC de l'hôte 'derrière' le Port3 = ABCDEF000003 & ABCDEF000001

Lorsque l'hôte2 envoie un paquet à l'hôte1, le switch va à nouveau comparer la MAC destination avec sa table MAC et a maintenant 2 correspondances, et maintenant le switch va joyeusement pointer le paquet vers l'hôte1 ET l'hôte3. Je sais qu'il pourrait survenir quelques complications, comme les switches qui mettent à jour leurs tables toutes les 30 secondes en envoyant une requête ARP à tous les hôtes qui y sont physiquement connectés. Et 3COM commercialise quelques switches de niveau2 qui refusent de lier plus d'1 MAC à un Port, le seul inconvénient est leur prix.

### Entrons dans la réalité

Une partie très intéressante pour les scriptkiddies sera celle-ci. Du fait que vous n'aurez pas besoin de beaucoup de connaissances réseau pour utiliser les outils d'empoisonnement ARP. J'ai principalement utilisé l'outil d'empoisonnement ARP version 0.5B de Steve Buer. Il est simple pour envoyer vos propres paquets customisés sur le LAN.

Maintenant je vais vous montrer comment vous pouvez rompre la route entre 2 hôtes distants.

Nous avons 3 hôtes sur un LAN basé sur un switch.

Hôtes	IP	MAC
-------	----	-----

Hôte1:	169.254.0.1	/ 00:10:4B:01:88:F3
--------	-------------	---------------------

Hôte2:	169.254.0.2	/ 00:12:06:19:82:00
--------	-------------	---------------------

Hôte3:	169.254.0.5	/ 00:E0:4C:39:65:37
--------	-------------	---------------------

Nous somme l'Hôte1 et nous voulons mettre fin à la route entre l'Hôte2 et 3.

Il est recommandé de savoir si les deux hôtes sont en ligne; envoyons quelques requetes ICMP sur eux.

```
GateKeeper:~/arpoison # ping 169.254.0.2
PING 169.254.0.1 (169.254.0.1): 56 data bytes
64 bytes from 169.254.0.2: icmp_seq=0 ttl=255 time=2.009 ms
64 bytes from 169.254.0.2: icmp_seq=1 ttl=255 time=1.103 ms
[1]+ Stopped ping 169.254.0.2
```

```
GateKeeper:~/arpoison # ping 169.254.0.5
PING 169.254.0.5 (169.254.0.5): 56 data bytes
64 bytes from 169.254.0.5: icmp_seq=0 ttl=128 time=2.924 ms
64 bytes from 169.254.0.5: icmp_seq=1 ttl=128 time=0.990 ms
[2]+ Stopped ping 169.254.0.5
```

Les deux hôtes sont donc en ligne (pour l'instant)...

Après avoir compilé la source de l'outil d'empoisonnement ARP, remplissez les bonnes informations.

```
GateKeeper:~/arpoison # ./arpoison
```

```
Usage: ./arpoison -i <device> -d <dest IP> -s <src IP> -t <target MAC> -r <src MAC>
```

Il importe peu quel hôte nous allons empoisonner, car les deux sont vulnérables. Je vais empoisonner l'Hôte2... rappelez-vous que l'information que vous rentrerez après "-r" ne doit pas être l'adresse MAC réelle!!! Sinon vous aurez construit un paquet valide et la route ne sera pas rompue.

```
GateKeeper:~/arpoison # ./arpoison -i eth1 -d 169.254.0.2 -s 169.254.0.5 -t
```

```
00:12:06:19:82:00 -r AA:BB:CC:DD:EE:FF
```

```
ARP packet sent via eth1
```

Après environ 30 secondes, la route sera rompue entre ces deux hôtes distants. Ce que nous avons fait est très simple à expliquer. Nous avons envoyé une réponse ARP malformée à l'Hôte2, ce paquet contient des informations invalides (l'adresse MAC) sur l'Hôte3.

Lorsque l'Hôte2 va vouloir se connecter à l'Hôte3, le kernel de l'Hôte2 va construire un paquet et l'information comme la MAC destination sera prise dans la table ARP locale. Seulement cette information est corrompue et le paquet n'arrivera jamais à l'Hôte3.

Notez que tous les systèmes d'exploitation vont rafraichir la table ARP toutes les 30 à 60 secondes. En d'autres termes, si vous voulez interrompre définitivement la route entre deux hôtes distants, vous devrez envoyer une réponse ARP falsifiée toutes les 30 à 60 secondes.

Un autre point intéressant est que cet utilitaire va falsifier (spoofer) les vraies adresses IP et MAC de l'attaquant, ainsi il sera très difficile pour les administrateurs de déterminer la provenance réelle de ces paquets.

## Les stations de travail sont également vulnérables

Tout comme les switches et les routeurs, les machines \*nix, Macintosh et Windows ne sont pas invulnérables à l'empoisonnement ARP. Tous ces systèmes d'exploitation ont le même problème... ils vont joyeusement mettre à jour leurs tables ARP (avec des informations erronées). Je sais que les machines BSD et \*nix peuvent être rendues quasi invulnérables lorsque le kernel a été recompilé avec quelques options spécifiques. Mais je suppose que peu d'utilisateurs vont recompiler le kernel à partir des sources s'il fonctionne correctement.

Ci dessous, j'ai réalisé différentes captures des tables ARP de différents systèmes d'exploitation...

OS: Linux SuSE 7.1

```
GateKeeper:~ # arp -nv -i eth0
```

```
Address      HWtype HWaddress      Flags Mask      Iface
212.187.0.1  ether  00:30:7B:94:31:C8 C              eth0
```

```
Entries: 4    Skipped: 3    Found: 1
```

```
GateKeeper:~ # arp -nv -i eth1
```

```
Address      HWtype HWaddress      Flags Mask      Iface
169.254.0.101 ether  00:E0:4C:39:65:6D C              eth1
```

```

169.254.0.6 ether 00:C0:4F:A7:63:81 C eth1
169.254.0.2 ether 00:12:06:19:82:00 C eth1
Entries: 4 Skipped: 1 Found: 3

```

OS: Windows 98 (seconde édition)

C:\>arp -a

```

Interface: 169.254.0.2 on Interface 0x2000002
Internet-adres Fysiek adres Type
169.254.0.1 00-10-4b-01-88-f3 dynamisch
169.254.0.101 00-e0-4c-39-65-6d dynamisch

```

C:\>

OS: FreeBSD 4.3

```

unreal@NederWiet:~ > arp -a
? (169.254.0.2) at 0:50:bf:5d:4e:6a [ethernet]
? (169.254.0.3) at 0:20:18:3a:fa:12 [ethernet]
? (169.254.0.5) at 48:54:e8:90:5f:cc [ethernet]
unreal@NederWiet:~ >

```

## Un scénario pire

Les lignes suivantes que j'ai écrites ne sont pas basées sur la réalité. Je veux seulement démontrer quels dommages pourraient être causés à partir d'informations comme celles-ci combinées avec les bons outils entre de mauvaises mains. Notez que toutes les informations ont été testées dans un environnement isolé (LAN).

Imaginez qu'un FAI possède un segment de routes avec 32 clients, ces 32 clients ont été divisés en groupes de 16 clients dans 2 segments pontés.

Voici une représentation sommaire :

```

#####
# Internet #
#####
  /\
  |
  \/
Routeur 1 (niveau 3) [169.254.0.1 / 00:10:A3:BC:D5:01]
  /\
  | les switche sont connectés séparément au routeur |
  \/
Switch 1 (niveau 2) <non connectés entre eux> Switch 2 (niveau 2)

```

Client 01  
Client 02  
Client 03  
etc...

Client 17  
Client 18  
Client 19  
etc...

Admettons que le Client 12 (sur le switch 1) veuille interrompre la route entre le Client 3 (également sur le switch 1) et le routeur (également sur le switch 1). S'il réussit, la victime ne pourra rien faire mis à part contacter les hôtes sur son propre segment ponté.

L'attaquant dispose de 2 ou 3 possibilités pour accomplir ceci:

- 1) Empoisonnement ARP du Client 03.
- 2) Empoisonnement ARP du routeur (la plupart du temps la **smartest** méthode).
- 3) Empoisonnement ARP du switch 1.

Un autre exemple...

Admettons que le Client 19 (sur le switch 2) veuille interrompre la route entre le Client 2 (sur le switch 1) et le routeur. Notez que cet exemple diffère réellement du premier du fait que l'attaquant est localisé sur un autre segment. Et vous rappelez-vous que les requêtes ARP ne peuvent pas passer à travers les routeurs ? En fait c'est très simple d'accomplir cette tâche. L'attaquant peut toujours réaliser un empoisonnement ARP du routeur avec des informations erronées sur l'hôte 2.

Dernier exemple...

Pour ceux qui comprennent toujours mon histoire...

Vous pouvez même interrompre une route entre 2 hôtes distants qui sont par exemple distants de 15 sauts... Cela va juste prendre plus de temps.

Car vous aurez besoin d'être sur le segment où l'hôte cible se trouve. Et la seule manière d'accomplir une telle tâche sera que vous possédiez/piratiez le routeur de ce segment.

## **Vaccination à l'empoisonnement ARP**

Ci dessus j'ai décrit deux méthodes pour "utiliser" l'empoisonnement ARP de LANs, évidemment, il existe quelques astuces pour prévenir de telles "mises à jour".

La manière la plus simple pour prévenir l'empoisonnement ARP de stations de travail et serveurs avec des systèmes d'exploitation Open Source est de M-lock le cache ARP ligne par ligne. Cela signifie que lorsque la table ARP contient une entrée valide comme celle-ci:

```
212.187.0.1      ether  00:30:7B:94:31:C8  C          eth0
```

Vous verrouillez cette entrée en tapant:

```
arp -v -i eth0 -s 212.187.0.1 00:30:7B:94:31:C8
```

Vérifiez le cache ARP une nouvelle fois en tapant:

```
arp -nv -i eth0
```

le résultat sera:

```
212.187.0.1      ether 00:30:7B:94:31:C8  CM          eth0
```

Vous voyez la différence? :)

Aussi longtemps que vous ne déverrouillerez pas le cache ARP, relancez les interfaces eth ou relancez le système, personne ne pourra rafraichir l'entrée ci dessus.

Une autre manière serait d'installer un pare-feu (de niveau 2 !!) sur la station de travail, mais la seule différence entre ceci et ma méthode sera le prix. Le pare-feu fera exactement la même chose, cela ne rend pas pour autant votre système invulnérable !

## **Conclusion**

Avec l'empoisonnement ARP vous pouvez faire différentes choses, la première de toute est le sniffing au niveau de segments basés sur des switchs en empoisonnant les hôtes distants ou les switchs.

La seconde, et la plupart du temps la plus méchante est d'altérer les tables ARP des routeurs, ce qui rend les segments de LAN isolés des autres.

Je pense fortement qu'en peu de temps ce genre d'attaques va croitre rapidement...

Remerciements:

Pool, NederWiet, DarkWhite, KD, ssuzeJJ

Copyright (C) 2001, DataWizard, The Netherlands.

Traduction française: Jérôme Athias