R102 - TP n°5 : ARP poisoning avec Scapy

Durée : 2h

Objectifs

Être capable de réaliser une attaque d'ARP poisoning avec Scapy (uniquement dans un réseau interne personnel).

Matériel nécessaire

Une MV Kali (machine attaquante où Scapy est déjà installé) et une machine victime (D'abord Ubuntu puis vous testerez sur Windows : vous essaierez les deux) dans un réseau interne derrière une passerelle 'RTBox2 Bridge/Interne'.

Il est strictement interdit de faire les manipulations de ce TP en dehors de votre réseau interne de MV.

L'**ARP Poisoning**, aussi connue sous le nom d'**ARP Spoofing** (usurpation ARP), est une attaque réseau sur un LAN très connue. Elle consiste à envoyer des paquets ARP malveillants à une machine de façon à corrompre son cache ARP, c'est-à-dire à mettre de mauvaises associations `(@IP, @MAC)` dans son cache ARP.

Cette attaque permet en particulier de réaliser une attaque MITM (Man in the Middle), c'est-à-dire placer une machine attaquante entre sa victime et les autres machines pour voir le trafic de cette dernière sans que celle-ci le remarque.

Il existe de nombreux outils pour faire cette attaque automatiquement notamment sous Kali avec **arpspoof** https://www.kali.org/tools/dsniff/ ou **bettercap 2** https://www.kali.org/tools/bettercap/. Teasing : vous le ferez en deuxième année en R4.Cyber.09 https://moodle.univ-artois.fr/course/view.php?id=108. Mais ici, dans un but pédagogique de compréhension du protocole ARP, nous le ferons à la main sous Scapy en forgeant nous même nos trames (force à nous).

Déposez en fin de TP un CR illustré sur Moodle contenant vos analyses, manipulations et commandes utilisées pour être capable de refaire ces manipulations (et d'autres) en contrôle TP (CTP). Vos comptes rendus seront vos uniques documents autorisés en CTP.

Préparation

Écrivez dans votre CR les @MAC et @IP de vos 3 machines :

1. `A` = `Machine attaquante Kali`
2. `V` = `Machine victime Ubuntu`
3. `P` = `Passerelle`

Complétez dans votre CR :

1. `@MAC_A` = , `@IP_A` =
2. `@MAC_V` = , `@IP_V` =
3. `@MAC_P` = , `@IP_P` =

Votre objectif va être d'empoisonner les caches ARP de la machine victime V et de la passerelle P :

1. Dans le cache ARP de `V`, `@IP_P` devra être associée à `@MAC_A` (Ainsi `V` pensera envoyer à `P` alors qu'il enverra à `A`).
2. Dans le cache ARP de `P`, `@IP_V` devra être associée à `@MAC_A` (Ainsi `P` pensera envoyer à `V` alors qu'il enverra à `A`).

Pour ce faire, nous allons envoyer des **réponses non sollicitées ARP : @IP is-at @MAC** contenant ces mauvaises associations (une machine peut mettre à jour son cache ARP en recevant ces messages même si non sollicités). En résumé :

Cible empoisonnée	**Message ARP envoyé par l'attaquant à la cible**	**Conséquence**
Victime	@IP_P is-at @MAC_A	La victime envoie tout son trafic destiné à la passerelle vers l'attaquant.
Passerelle	@IP_V is-at @MAC_A	La passerelle envoie tout son trafic destiné à la victime vers l'attaquant.

Vous allez créer ces trames avec Scapy depuis la machine attaquante. Vous aurez besoin des champs ARP Scapy suivants :

Champ Scapy	Valeur	Rôle dans l'attaque
`op = 2`	Réponse (is-at)	Réponse ARP (non sollicitée) pour que la cible mette à jour son cache ARP avec une fausse association.
`psrc`	@IP src	@IP usurpée : @IP victime ou passerelle.
`hwsrc`	@MAC src	@MAC machine attaquante.
`pdst`	@IP dst	@IP de la machine cible recevant le message ARP.
`hwdst`	@MAC dst	@MAC de la machine cible recevant le message ARP.

Mise en place de l'attaque

Rappels

Envois à la machine victime de **réponses ARP non sollicitées : @IP_P is-at @MAC_A**

Et

Envois à la passerelle de **réponses ARP non sollicitées : @IP_V is-at @MAC_A**

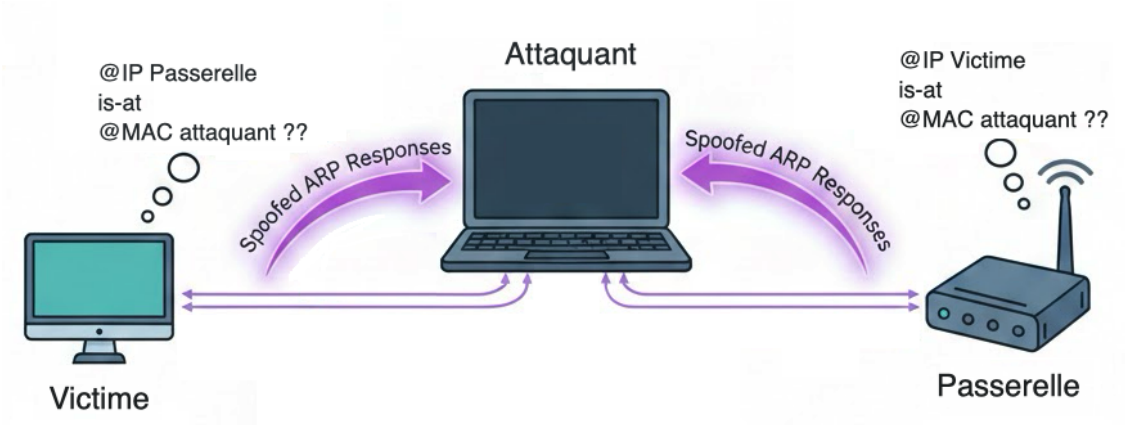
1. Ouvrez un terminal et mettez vous en root :

```
sudo -s
```

2. Activez l'IP FORWARDING sur votre machine attaquante avec la commande suivante :

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

Par défaut, si une machine reçoit un paquet qui ne lui ai pas destiné, elle l'ignore. Ici, on veut qu'elle le réachemine (renvoie) (en anglais forward) vers sa destination. Sans cela, les paquets non destinés à l'attaquant seraient simplement abandonnés.



3. Lancez ensuite un terminal interactif Python 3

```
python 3
```

4. Importez Scapy

```
>>> from scapy.all import *
```

5. Réalisez ensuite les instructions pour spoofer (usurper, empoisonner) le cache ARP de la machine victime

```
>>> pv = ARP(op=2, psrc=@ip_passerelle, hwsrc=@mac_A,
    pdst=@ip_v, hwdst=@mac_v)
```

Envois à la machine victime de **réponses ARP non sollicitées : @IP_P is-at @MAC_A** : on veut faire croire à la machine victime que la passerelle est à l'adresse mac de la machine attaquante kali.

Les adresses se notent entre guillemets ou quotes (cf TP4) dans l'instruction Scapy : ce sont des chaînes de caractères.

6. Envoyez cette réponse ARP

```
>>> send(pv, verbose=False)
```

puis consulter le cache ARP modifié de la victime Ubuntu

```
ip n
```

Au bout d'un moment la bonne association reviendra, il faudra envoyer régulièrement ces réponses ARP pour garder le spoofing actif.

7. Réalisez ensuite les instructions pour spoofer le cache ARP de votre passerelle

```
>>> pp = ARP(op=2, psrc=@ip_v, hwsrc=@mac_A, pdst=
    @ip_passerelle, hwdst=@mac_passerelle)
```

Envois à la passerelle de **réponses ARP non sollicitées : @IP_V is-at @MAC_A** : on fait croire à la passerelle que la machine victime est à l'adresse mac de la machine attaquante kali.

puis envoyez cette réponse ARP

```
>>> send(pp, verbose=False)
```

puis constatez le cache ARP de la passerelle modifiée

```
ip n
```

Idem, au bout d'un moment la bonne association reviendra, il faut envoyer régulièrement les réponses ARP frauduleuses `pp` et `pv` pour garder les deux caches empoisonnés (dans un script on ferait une boucle)

8. Les deux caches étant empoisonnés, ouvrez Wireshark sur la machine attaquante kali et montrez que vous pouvez voir une connexion de la machine victime, par exemple consultez le site iut-rt via firefox sur la machine victime.
9. Lancez une machine victime Windows et testez également cette attaque sur elle.

   Rappels commandes Windows :

   ```
   ipconfig /all
   ```

   ```
   arp -a
   ```

10. Recherchez les contre-mesures existantes à cette attaque

Il existe de nombreuses contre-mesures à cette attaque comme l'ARP statique qui consiste à associer manuellement les adresses IP aux adresses MAC sur les machines critiques comme les passerelles, cela empêche toute modification via ARP, mais c'est seulement possible dans un petit réseau. Sur un switch il peut y avoir le DAI (Dynamic ARP Inspection) et le Port Security qui limite le nombre d'adresses MAC par port (bloque les appareils non autorisés, empêche l'attaquant de se faire passer pour la passerelle). Il est aussi possible de détecter les ARP spoofings avec des Pare-feux IDS/IPS ou des outils spécifiques de détection comme Arpwatch, ArpON ou Ettercap (qui peut faire les deux : spoofing ou detection)

Synthèse

À l'issue de ce TP :

- Vous êtes capable de réaliser une attaque ARP poisoning avec Scapy.
- Vous êtes capable de concevoir des contre-mesures.

A = Machine attaquante Kali
V = Machine victime Ubuntu
P = Passerelle

@MAC_A = 08:00:27:73:37:c6 , @IP_A = 192.31.25.12
@MAC_V = 08:00:27:42:dd:07 , @IP_V = 192.31.25.13
@MAC_P = 08:00:27:e7:98:2c , @IP_P = 192.31.25.1

Cible empoisonnée	**Message ARP envoyé par l'attaquant à la cible**	**Conséquence**
Victime	@IP_P is-at @MAC_A	La victime envoie tout son trafic destiné à la passerelle vers l'attaquant.
Passerelle	@IP_V is-at @MAC_A	La passerelle envoie tout son trafic destiné à la victime vers l'attaquant.

Vous allez créer ces trames avec Scapy depuis la machine attaquante. Vous aurez besoin des champs ARP Scapy suivants :

Champ Scapy	**Valeur**	**Rôle dans l'attaque**
`op = 2`	Réponse (is-at)	Réponse ARP (non sollicitée) pour que la cible mette à jour son cache ARP avec une fausse association.
`psrc`	@IP src	@IP usurpée : @IP victime ou passerelle.
`hwsrc`	@MAC src	@MAC machine attaquante.
`pdst`	@IP dst	@IP de la machine cible recevant le message ARP.
`hwdst`	@MAC dst	@MAC de la machine cible recevant le message ARP.