

# Rapport de projet - mineure SSI

Professeur référent : David Soria

---

## Développement d'un honeypot réseau

---

Projet mené par les étudiants en deuxième année en IENAC SITA à  
l'École Nationale de l'Aviation Civile

JAZOULI Othmane  
LEBERT Nathan  
SETTAI Yassine

---

Mai 2020

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>Techniques d'intrusion classiques</b>	<b>4</b>
Scan de ports	4
Empoisonnement ARP	5
<b>Travail réalisé</b>	<b>6</b>
Choix du langage de programmation et capture de paquets	6
Détection d'empoisonnement ARP	8
Une simulation d'attaque d'empoisonnement ARP	9
Scan de ports	10
<b>Conclusion</b>	<b>12</b>

## I. Introduction

La sécurité des réseaux est devenue un enjeu majeur dans les entreprises. Afin de garantir la disponibilité, l'intégrité et la confidentialité de tous les systèmes d'information d'un réseau, une surveillance permanente couplée à de la collecte et de l'analyse d'information est essentielle. De nombreuses solutions existent pour répondre à ces trois problématiques. Certaines s'appuient sur la simple observation d'événements au sein d'un réseau pour tenter d'y répondre tandis que d'autres y jouent un rôle plus actif.

Certaines de ces solutions plus actives présentent un comportement à première vue contre-intuitif se révélant très efficaces pour piéger les pirates ou détecter une intrusion sur le réseau. Parmi elles, on peut citer les *honeypots* réseau. Ces systèmes inclus dans le réseau à protéger visent à tromper les pirates en les attirant jusqu'à lui. Faisant office de "pot de miel" placé à la vue d'un ours, le honeypot est ainsi une cible privilégiée et vulnérable à même d'être le premier système touché par une attaque. Des tentatives d'intrusion classiques au sein du réseau sont ainsi facilement détectables. Les informations recueillies par le honeypot et leur analyse doivent ensuite aider les administrateurs du réseau à renforcer la sécurité des systèmes d'informations dont ils ont la charge.

La mise en place d'un leurre au sein du réseau peut être réalisée de deux manières différentes : par l'intermédiaire d'un système réel ou par la simulation de l'un de ces systèmes. La première solution, appelée *honeypot* à forte interaction comporte des risques importants puisqu'un attaquant a la possibilité d'infiltrer une machine réelle du réseau. La seconde, appelée *honeypot* à faible interaction limite les risques d'intrusion, les services réseaux, le système d'exploitation et ses applications étant notamment simulées.

Notre projet vise ainsi à développer un *honeypot* réseau à faible interaction capable de détecter les tentatives de "mouvements latéraux" d'un attaquant dans un réseau et de remonter les alertes associées. La première partie de notre projet vise à étudier les moyens classiques d'intrusion et le type de trafic légitime sur un réseau. Ces méthodes d'instructions connues, nous avons développé des outils permettant de détecter des comportements anormaux au sein du réseau et de les isoler pour une future analyse avant de s'attacher à vérifier et tester ces fonctionnalités.

## II. Techniques d'intrusion classiques

Les techniques utilisées pour compromettre des données ou des systèmes au sein d'un réseau sont nombreuses et peuvent se révéler très efficaces lorsque la protection des systèmes qu'il comporte n'est pas suffisante. Ces techniques, aussi appelées mouvements latéraux, sont généralement détectables car assez lentes et risquées en cas d'erreur de l'attaquant. La connexion au réseau établie, l'attaquant tente de cartographier les systèmes présents, leurs caractéristiques, les services réseaux disponibles à l'aide d'outils classiques comme *nmap* avant de tenter d'infecter des applications, par exemple. Parmi les moyens usuels pour y parvenir, nous pouvons citer les mouvements latéraux suivants : le scan de port et l'empoisonnement ARP.

### 1. Scan de ports

Un scan de ports ou balayage de ports vise à rechercher les ports ouverts sur un serveur réseau. Cette technique est généralement utilisée par un administrateur pour vérifier la sécurité des serveurs disponibles sur un réseau.

Le scan de port n'est pas une menace immédiate et ne constitue pas forcément une amorce d'attaque. Néanmoins, la recherche de vulnérabilités et l'étude des logiciels en écoute qui peut suivre en constitue une. En effet, une fois les logiciels en écoute connus par l'attaquant, celui-ci peut exploiter des vulnérabilités existantes sur certaines versions.

Elle concerne en grande partie le protocole TCP mais peut également s'effectuer en UDP, grâce à des requêtes spécifiques. Le protocole TCP étant utilisé en majeure partie par les applications, un balayage de port sur ce protocole permettrait à un attaquant de vérifier si un logiciel est en écoute sur un port particulier. Après l'envoi de paquets à destination de ce port, l'étude des réponses permet de déterminer si le port est en écoute et potentiellement le nom du logiciel ainsi que sa version grâce à des logiciels tels que *nmap*. Lorsque le nombre de paquets reçus par la machine qui est scannée devient élevé, il peut s'agir d'une recherche de vulnérabilité : un attaquant peut ainsi être en train d'analyser les réponses à des paquets caractéristiques pour déterminer la version du logiciel qui est visé. Ce type de recherche, dite active, est ainsi assez intrusive et pourrait être détectée. A l'inverse, des techniques plus lentes d'analyse des paquets circulant sur un réseau sont bien plus difficiles à détecter. Il s'agit de *network sniffing*.

Des outils classiques tels que *nmap* permettent de réaliser des balayages de ports très complets et parfois peu visibles. Considérons la requête suivante, réalisant un scan TCP - SYN (option -S) et UDP (option -sU) :

```
nmap -sS -sU -sV 192.168.1.101
```

Grâce à l'option -sV, *nmap* retourne la version des logiciels en écoute sur un port. Ci-après un extrait de ce qui pourrait être renvoyé par la requête précédente.

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	
25/tcp	open	smtp	
53/tcp	open	domain?	
80/tcp	open	http	Microsoft IIS 7.5

Prenons l'exemple du port 80/TCP. Le port est ouvert avec Microsoft IIS 7.5. Le serveur web Microsoft présente des vulnérabilités connues accessibles dans les *Common Vulnerabilities and Exposures* notamment (voir capture d'écran ci-après du site *cvedetails.com*). De telles vulnérabilités sont ensuite facilement exploitables par un attaquant aguerri.

Microsoft » Internet Information Server » 7.5 : Security Vulnerabilities

Cpe Name: cpe:/a:microsoft:internet\_information\_server:7.5

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By: CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2012-2532</a>	200		+Info	2012-11-13	2018-10-12	5.0	None	Remote	Low	Not required	Partial	None	None
Microsoft FTP Service 7.0 and 7.5 for Internet Information Services (IIS) processes unspecified commands before TLS is enabled for a session, which allows remote attackers to obtain sensitive information by reading the replies to these commands, aka "FTP Command Injection Vulnerability."														
2	<a href="#">CVE-2012-2531</a>	200		+Info	2012-11-13	2019-07-03	2.1	None	Local	Low	Not required	Partial	None	None
Microsoft Internet Information Services (IIS) 7.5 uses weak permissions for the Operational log, which allows local users to discover credentials by reading this file, aka "Password Disclosure Vulnerability."														
3	<a href="#">CVE-2010-3972</a>	119	1	DoS Exec Code Overflow +Info	2010-12-23	2019-07-03	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
Heap-based buffer overflow in the TELNET_STREAM_CONTEXT::OnSendData function in ftpsvc.dll in Microsoft FTP Service 7.0 and 7.5 for Internet Information Services (IIS) 7.0, and IIS 7.5, allows remote attackers to execute arbitrary code or cause a denial of service (daemon crash) via a crafted FTP command, aka "IIS FTP Service Heap Buffer Overrun Vulnerability." NOTE: some of these details are obtained from third party information.														
4	<a href="#">CVE-2010-2730</a>	119		Exec Code Overflow	2010-09-15	2019-07-03	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
Buffer overflow in Microsoft Internet Information Services (IIS) 7.5, when FastCGI is enabled, allows remote attackers to execute arbitrary code via crafted headers in a request, aka "Request Header Buffer Overflow Vulnerability."														

A noter qu'en limitant la vitesse de balayage et en rendant aléatoire la recherche de ports, un attaquant peut en limiter ses risques d'être détecté. En effet, le faible nombre de paquets envoyés pourrait ne pas être considéré comme du trafic illégitime.

## 2. Empoisonnement ARP

L'empoisonnement ARP consiste à utiliser le protocole de résolution d'adresse ARP, fréquemment utilisé dans les réseaux Ethernet et Wi-Fi. Une telle technique participe à une attaque du type *man-in-the-middle* ("homme du milieu" en français) et vise à intercepter, modifier ou bloquer le trafic entre une machine et une passerelle d'un réseau. Pour ce faire, l'attaquant va tenter d'usurper l'identité d'une passerelle grâce à des requêtes ARP. Prenons l'exemple d'un réseau local aux caractéristiques suivantes et d'une attaque visée sur une machine cible :

192.168.1.1 : passerelle.  
192.168.1.10 : machine cible.  
192.168.1.20 : machine de l'attaquant.

Afin de se faire passer pour la passerelle, l'attaquant envoie un paquet ARP aux caractéristiques suivantes à la machine cible qui fait correspondre, à tort, l'adresse IP de la passerelle à l'adresse MAC de l'attaquant :

adresse IP source : 192.168.1.1 adresse IP destinataire : 192.168.1.10 adresse MAC source : <i>adresse MAC attaquant</i> type de requête ARP : <i>is-at</i>
--

Pour contacter la passerelle, la machine cible sera finalement redirigée vers la machine de l'attaquant plutôt que la passerelle. En effet, l'envoi de paquets nécessite la connaissance des adresses MAC source et destinataire. L'adresse MAC destinataire ayant été usurpée, l'attaquant recevra bien le trafic venant de la machine cible.

La détection d'une tentative d'empoisonnement ARP peut se faire par l'observation précise de toutes les paquets ARP ayant été émises sur le réseau. En comparant les adresses IP et MAC des machines connues du réseau avec les combinaisons données dans les requêtes ARP, il est possible de détecter si un attaquant tente d'usurper des adresses IP.

Les paquets ARP peuvent être émises par un attaquant sur le réseau en *broadcast* (au réseau tout entier) ou en *unicast* (à une machine particulière). Le premier type d'émission est une fonctionnalité prévue pour permettre à une nouvelle machine sur le réseau de s'annoncer aux autres. La connaissance précise du réseau et des adresses MAC uniques des machines du réseau permet de détecter une tentative d'empoisonnement avec émission ARP en *broadcast*. Le second type d'émission est généralement celui utilisé par un attaquant pour usurper une passerelle. En comparant les caractéristiques des requêtes ARP sur le réseau avec les caractéristiques (connues de l'administrateur) des machines et passerelles du réseau, il est facile de déterminer un conflit d'adresses entre la machine de l'attaquant et la passerelle.

### III. Travail réalisé

Afin de détecter les comportements anormaux au sein du réseau, nous nous sommes d'abord intéressés à la capture de paquets échangés sur le réseau. L'analyse de ces paquets a ensuite conduit à l'élaboration de règles expliquant si l'envoi d'un paquet sur le réseau relève d'un comportement anormal ou non.

#### 1. Choix du langage de programmation et capture de paquets

L'objectif de notre projet est de créer un petit honeypot capable de détecter les tentatives de mouvements latéraux classiques sur un réseau de taille raisonnable. Pour cela, nous voulions programmer ce honeypot à l'aide d'un langage bien documenté, assez répandu et présentant déjà de nombreux outils pour travailler sur le réseau. La gestion des erreurs constitue également un point clé dans ce choix de langage. Notre choix s'est ainsi porté sur Java, langage très répandu depuis les années 2000 pour lequel la documentation est très riche. Les outils présents pour la capture de paquets réseau sont également nombreux et documentés.

La première étape de développement de notre honeypot vise à capturer l'ensemble du trafic sur le réseau pour ensuite l'analyser. A l'aide du module Pcap de *Jnetpcap*, nous récupérons les informations transmises dans les paquets circulant sur le réseau et nous les isolons selon les tests que nous allons réaliser.

```
Network devices found:
#0: \Device\NPF_{3144276A-92E1-4838-9105-7A2333FF13E5} [Microsoft]
#1: \Device\NPF_{4696D819-8D1B-46C0-94C2-A4979894A815} [Intel(R) 82579V Gigabit Network Connection]
#2: \Device\NPF_{2CC75F16-1AE7-4317-9036-B1E614424CB2} [Microsoft]
choose the one device from above list of devices
2
device opened
--> udp packet detected
Udp Source Port :56530
Udp Destination Port :443
Ip4 Source :192.168.1.100
Ip4 Destination :172.217.18.206
--> udp packet detected
Udp Source Port :443
Udp Destination Port :56530
Ip4 Source :172.217.18.206
Ip4 Destination :192.168.1.100
--> udp packet detected
```

La capture d'écran ci-dessus montre des exemples de paquets reçus sur une interface réseau. Pour les paquets UDP reçus dans notre exemple, les informations sont affichées une à une afin de faciliter leur analyse.

## 2. Détection d'empoisonnement ARP

### 2.1 Le mécanisme de détection

La capture des paquets et l'extraction des informations qu'ils contiennent étant maintenant réalisée, nous nous sommes intéressés à détecter d'éventuelles tentatives d'empoisonnement ARP sur le réseau à surveiller. Comme expliqué dans la partie II. 2., la comparaison de la combinaison adresse MAC / adresse IP de l'attaquant avec celle d'adresse MAC / adresse IP de la passerelle.

La combinaison d'adresses de la passerelle étant fixe, on extrait les caractéristiques de la passerelle au lancement du honeypot et on les sauvegarde pour les comparaisons futures. Un potentiel empoisonnement ARP est détecté lorsque l'adresse MAC de l'attaquant est associée dans un paquet à l'adresse IP de la passerelle. La sauvegarde de la combinaison correcte liée à la passerelle nous assure qu'en cas de réception d'une requête ARP *is-at* erronée, notre honeypot sera capable de la détecter. Lorsqu'une tentative d'empoisonnement est détectée, un message d'alerte contenant les adresses IP et MAC incriminées est renvoyée par le honeypot. Dans l'exemple ci-dessous, plusieurs paquets ARP visant à positionner l'adresse mac du gateway à 0:c:29:7f:91:32 (adresse MAC de l'attaquant) sont envoyés sur le réseau.

```
settal@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.10.8 netmask 255.255.255.240 broadcast 172.20.10.15
    inet6 fe80::20c:29ff:fe7f:9132 prefixlen 64 scopeid 0x20<link>
    ether 00:c:29:7f:91:32 txqueuelen 1000 (Ethernet)
    RX packets 280535 bytes 313756853 (313.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 81885 bytes 6162962 (6.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2034 bytes 152348 (152.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2034 bytes 152348 (152.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

settal@ubuntu:~$ sudo arpspoof -i ens33 -t 172.20.10.2 172.20.10.1
[sudo] password for settal:
0:c:29:7f:91:32 0:c:29:3:c0:6 0806 42: arp reply 172.20.10.1 is-at 0:c:29:7f:91:32
0:c:29:7f:91:32 0:c:29:3:c0:6 0806 42: arp reply 172.20.10.1 is-at 0:c:29:7f:91:32
0:c:29:7f:91:32 0:c:29:3:c0:6 0806 42: arp reply 172.20.10.1 is-at 0:c:29:7f:91:32
0:c:29:7f:91:32 0:c:29:3:c0:6 0806 42: arp reply 172.20.10.1 is-at 0:c:29:7f:91:32
0:c:29:7f:91:32 0:c:29:3:c0:6 0806 42: arp reply 172.20.10.1 is-at 0:c:29:7f:91:32
0:c:29:7f:91:32 0:c:29:3:c0:6 0806 42: arp reply 172.20.10.1 is-at 0:c:29:7f:91:32
```

L'application retourne en parallèle un message disant qu'une tentative de *spoofing* a été détectée.

```
choosing <flags=6, addresses=[[addr=[10], mask=[10], broadcast=null, dstaddr=null], [addr=[INET4:172.20.10.2], mask=[INET4:255.255.255.240], broadcast=[INET4:172.20.10.15], dstaddr=null], [addr=[17], mask=null, broadcast=[17], dstaddr=null]], name=ens33, desc=null>
device opened
spoofing attempt detected
spoofing attempt detected
spoofing attempt detected
spoofing attempt detected
```



## 2.2 Une simulation d'attaque d'empoisonnement ARP

Pour pouvoir tester la détection d'empoisonnement ARP, nous proposons de réaliser une attaque arp spoofing avec le programme arpspoof contenu dans le paquet dsniff.

Ce programme est lancé de la manière suivante :

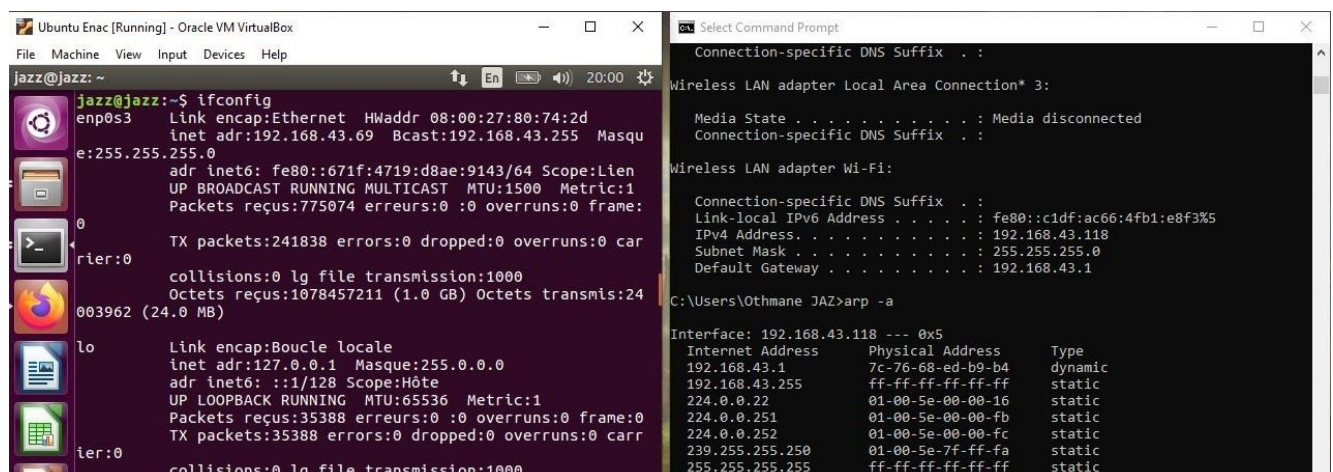
```
# arpspoof -i interface -t target host
```

Avec :

- target est la machine client que nous souhaitons attaquer
- host est la machine serveur pour laquelle nous souhaitons nous faire passer

Ce programme est actif jusqu'à sa fermeture. Nous pouvons observer dans les captures d'écrans ci-dessous que l'adresse mac de l'attaquant a bien été liée à l'adresse ip du local gateway.

**La situation dans le réseau avant d'effectuer l'attaque :**



```
jazz@jazz:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:80:74:2d
        inet adr:192.168.43.69  Bcast:192.168.43.255  Masqu
e:255.255.255.0
        adr inet6: fe80::671f:4719:d8ae:9143/64 Scope:Lien
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        Packets reçus:775074 erreurs:0 :0 overruns:0 frame:
        TX packets:241838 errors:0 dropped:0 overruns:0 car
        collisions:0 lg file transmission:1000
        Octets reçus:1078457211 (1.0 GB) Octets transmis:24
003962 (24.0 MB)

lo       Link encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        adr inet6: ::1/128 Scope:Hôte
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        Packets reçus:35388 erreurs:0 :0 overruns:0 frame:0
        TX packets:35388 errors:0 dropped:0 overruns:0 carr
        collisions:0 lg file transmission:1000

C:\Users\Othmane JAZ>ipconfig /all

Wireless LAN adapter Local Area Connection* 3:

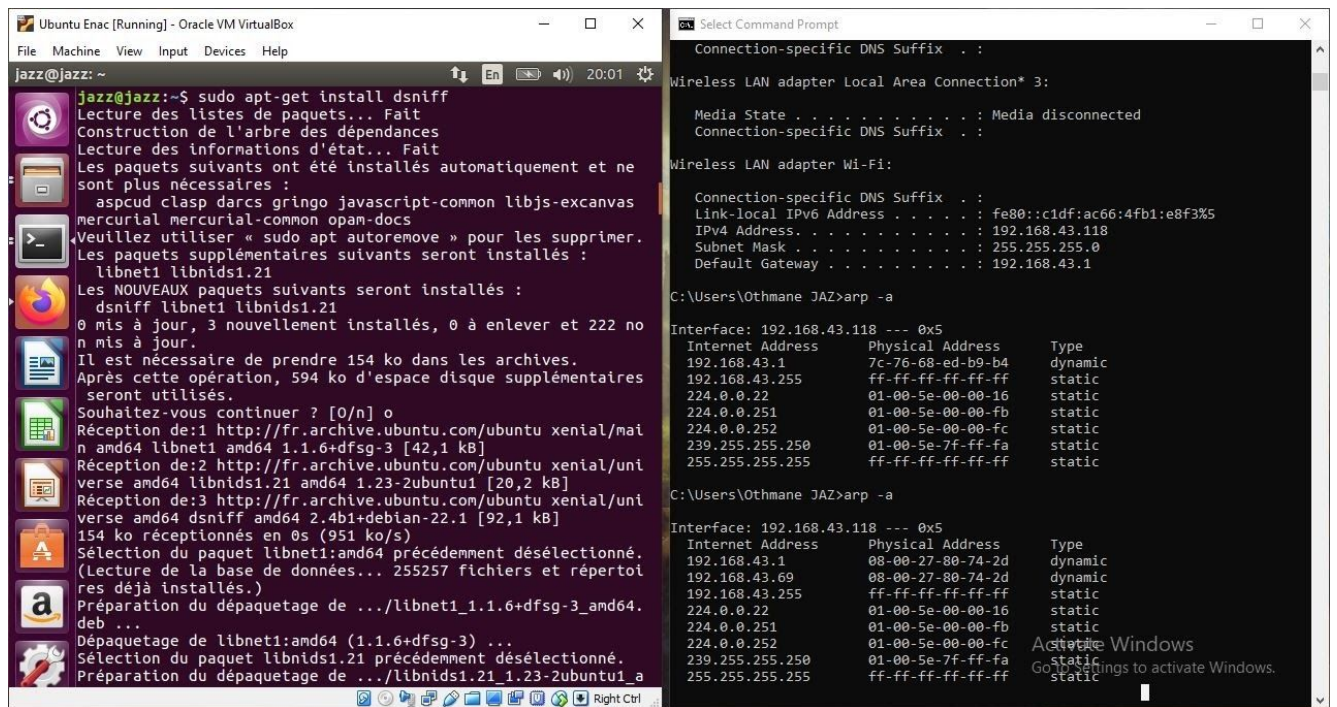
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::c1df:ac66:4fb1:e8f3%5
    IPv4 Address. . . . . : 192.168.43.118
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.43.1

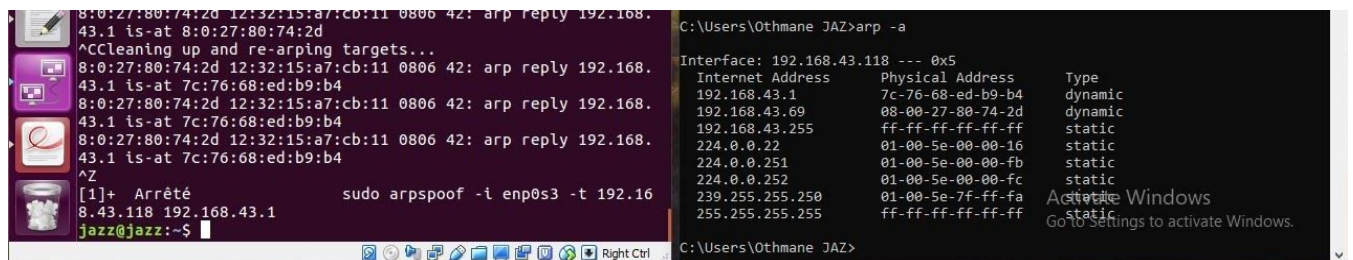
C:\Users\Othmane JAZ>arp -a

Interface: 192.168.43.118 --- 0x5
Internet Address      Physical Address      Type
192.168.43.1          7c-76-68-ed-b9-b4    dynamic
192.168.43.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251          01-00-5e-00-00-fb    static
224.0.0.252          01-00-5e-00-00-fc    static
239.255.255.250      01-00-5e-7f-ff-fa    static
255.255.255.255      ff-ff-ff-ff-ff-ff    static
```

**Au moment de l'exécution de l'attaque :**



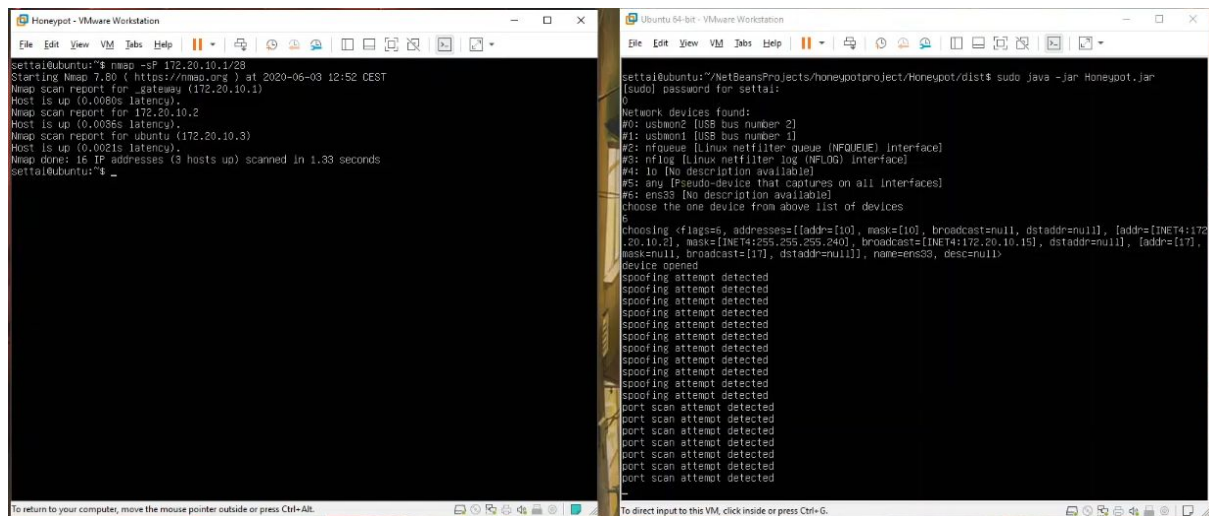
### Fermeture du programme d'attaque :



### 3. Scan de ports

Les trames du scans du port peuvent être des trames tcp/ip en mode syn comme avec l'outil nmap ou encore des paquets arp avec un opcode request comme avec l'arp-scan de linux.

Dans cette simulation, on a essayé de simuler un scan de port avec l'outil nmap et on peut le détecter; si on reçoit de trafic de type tcp ip avec le flag `syn == true`, avec une source autre que la passerelle et honeypot comme destination.



## IV. Conclusion

Notre projet visait à développer un *honeypot* réseau capable de détecter les tentatives de “mouvements latéraux” d’un attaquant dans un réseau et de remonter les alertes associées. L’application que nous avons développée en Java nous a permis de capturer le trafic réseau et d’extraire les informations contenues dans les paquets. L’analyse des paquets ARP a permis de détecter les tentatives d’empoisonnement ARP. L’analyse des paquets UDP et des paquets autres que TCP SYN S nous ont permis de renvoyer la liste des machines suspectes et de détecter d’éventuels scans de ports.

Durant le développement de ce honeypot réseau, nous avons éprouvé des difficultés à tester notre application. En effet, la réalisation d’attaques de type empoisonnement ARP ou le scan de ports sur un réseau tel que celui de l’ENAC n’était pas réalisable. Le matériel à notre disposition constituait un frein à la réalisation de nos tests. Nous avons donc utilisé des machines virtuelles lancées sur Linux et simulé les deux types de mouvements que notre honeypot était susceptible de détecter.