



INF8402 – Sécurité des réseaux fixes et mobiles

Automne 2021

TP1 : Outils de sécurité

Mardi 5 octobre 2021

Q1) Combien d'interfaces réseau avez-vous (physique et logique) ? Accordez une attention particulière à la configuration réseau de ces interfaces (Ipv4, serveur DHCP, serveur DNS, serveur WINS, etc) (1 point)

Nous avons deux interfaces physiques Carte Ethernet Ethernet et Carte Ethernet Ethernet 2 et nous avons deux interfaces réseau logiques qui sont Carte Ethernet VMware Network Adapter VMnet1 et Carte Ethernet VMware Network Adapter VMnet8 pour un total de 4 interfaces.

De plus, la Carte Ethernet Ethernet détient 1 serveur DHCP qui fournit une IP sur le réseau de l'école. Cette carte nous montre aussi la présence de 3 serveurs DNS et 1 serveur WINS.

Enfin, chacune des interfaces détient une adresse IPv4 et une adresse IPv6. Les interfaces logiques détiennent des IPv4 privées tandis que les interfaces physiques détiennent des IPv4 publiques.

```
Carte Ethernet Ethernet :
Suffixe DNS propre à la connexion. . . : gig1.polymtl.ca
Description. . . . . : Intel(R) Ethernet Connection I217-V
Adresse physique . . . . . : 08-62-66-4C-7F-A9
DHCP activé. . . . . : Oui
Configuration automatique activée. . . : Oui
Adresse IPv6 de liaison locale. . . . : fe80::2406:3cf2:ebb3:2ebc%15(préfééré)
Adresse IPv4. . . . . : 132.207.29.115(préfééré)
Masque de sous-réseau. . . . . : 255.255.255.0
Bail obtenu. . . . . : 27 juillet 2021 01:10:32
Bail expirant. . . . . : 9 septembre 2021 01:11:55
Passerelle par défaut. . . . . : 132.207.29.1
Serveur DHCP . . . . . : 132.207.180.43
IAID DHCPv6 . . . . . : 134767206
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : 132.207.185.70
                        132.207.180.14
                        132.207.6.11
Serveur WINS principal . . . . . : 132.207.180.14
NetBIOS sur Tcpip. . . . . : Activé
Liste de recherche de suffixes DNS propres à la connexion :
                        gig1.polymtl.ca
                        gi.polymtl.ca

Carte Ethernet VMware Network Adapter VMnet1 :
Suffixe DNS propre à la connexion. . . :
Description. . . . . : VMware Virtual Ethernet Adapter for VMnet1
Adresse physique . . . . . : 00-50-56-C0-00-01
DHCP activé. . . . . : Non
Configuration automatique activée. . . : Oui
Adresse IPv6 de liaison locale. . . . : fe80::e960:9218:4a30:7911%4(préfééré)
Adresse IPv4. . . . . : 192.168.56.1(préfééré)
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . :
IAID DHCPv6 . . . . . : 67129430
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                        fec0:0:0:ffff::2%1
                        fec0:0:0:ffff::3%1
NetBIOS sur Tcpip. . . . . : Activé

Carte Ethernet VMware Network Adapter VMnet8 :
Suffixe DNS propre à la connexion. . . :
Description. . . . . : VMware Virtual Ethernet Adapter for VMnet8
Adresse physique . . . . . : 00-50-56-C0-00-08
DHCP activé. . . . . : Non
Configuration automatique activée. . . : Oui
Adresse IPv6 de liaison locale. . . . : fe80::f169:c2eb:6da:74cd%2(préfééré)
Adresse IPv4. . . . . : 192.168.11.1(préfééré)
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . :
IAID DHCPv6 . . . . . : 134238294
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                        fec0:0:0:ffff::2%1
                        fec0:0:0:ffff::3%1
NetBIOS sur Tcpip. . . . . : Activé
```

```

Carte Ethernet Ethernet 2 :

Suffixe DNS propre à la connexion. . . : 
Description. . . . . : Intel(R) PRO/1000 GT Desktop Adapter
Adresse physique . . . . . : 90-E2-BA-49-FA-60
DHCP activé. . . . . : Oui
Configuration automatique activée. . . : Oui
Adresse IPv6 de liaison locale. . . . : fe80::cca1:d7dc:e8fd:449a%12(préfér  )
Adresse d'autoconfiguration IPv4 . . . : 169.254.68.154(pr  f  r  )
Masque de sous-r  seau. . . . . : 255.255.0.0
Passerelle par d  faut. . . . . : 
IAID DHCPv6 . . . . . : 210821818
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
NetBIOS sur Tcpip. . . . . : Activ  
PS X:\>

```

Q2) Quels sont les param  tre r  seau de la carte Intel I217-V :

a) Avec les 3 premiers octets de l'adresse MAC (OUI Organisation Unique Identifier) on sait que la carte provient du constructeur ASUSTek COMPUTER INC.

OUI	MAC range	Company
08-62-66	08-62-66-00-00-00 - 08-62-66-FF-FF-FF	ASUSTek COMPUTER INC.

Source : <https://hwaddress.com/?q=08%3A62%3A66>

b) Adresse IPv4 : 132.207.29.115

```

Adresse IPv4. . . . . : 132.207.29.115(pr  f  r  )

```

c) Masque r  seau : 255.255.255.0 et il y a sous r  seautage car 8 des 32 bits peuvent   tre utilis  s pour de nouvelles machines.

```

Masque de sous-r  seau. . . . . : 255.255.255.0

```

d) Cette adresse a   t   obtenue gr  ce au serveur DHCP (Dynamic Host Configuration Protocol) de Polytechnique qui permet    une machine de demander une adresse IP dans le r  seau lors de la connexion    ce dernier.

e) Adresse IPv6 : fe80::2406:3cf2:ebb3:2ebc

```

Adresse IPv6 de liaison locale. . . . : fe80::2406:3cf2:ebb3:2ebc%15(pr  f  r  )

```

f) Serveur DHCP : 132.207.180.43

```

Serveur DHCP . . . . . : 132.207.180.43

```

g) Serveurs DNS : 132.207.185.70, 132.207.180.14 et 132.207.6.11

```

Serveurs DNS. . . . . : 132.207.185.70
                      132.207.180.14
                      132.207.6.11

```

h) Server Wins : 132.207.180.14

```

Serveur WINS principal . . . . . : 132.207.180.14

```

Q3) Qu'est-ce qu'un serveur WINS ? Quelle est la différence entre un serveur DNS et un serveur WINS ? (2 points)

Les Windows Internet Name Services (WINS) sont des serveurs de résolution de noms comme les Domain Name Servers (DNS), mais ces derniers ne font pas la résolution de noms de la même manière. Les serveurs WINS font partie de la topologie des réseaux Microsoft et ils permettent d'associer les noms NetBios avec des adresses IP tandis que les serveurs DNS permettent d'associer les noms de domaines TCP/IP à des adresses IP. Ainsi, la grande différence est dans ce qui est associé à une adresse IP, le NetBios pour les WINS et les noms de domaines TCP/IP pour les DNS. Aujourd'hui, les serveurs WINS ne sont plus utilisés, car Microsoft a fait des changements dans le NetBios et donc la plupart des DNS peuvent traiter les requêtes NetBios.

Référence : <https://searchnetworking.techtarget.com/answer/What-is-difference-between-a-WINS-server-and-a-DNS-server>

Q4) Vérifiez que tous les ordinateurs sont dans le même réseau (domaine de diffusion). Faites des tests de connectivité (ping) entre eux, dès que ces tests fonctionnent, écrivez les adresses IP de chaque machine. L'adresse IP de Bitnami est à l'écran. Vous devez inclure des captures d'écran de vos tests de connectivité (0.25 point)

Windows 10 physique et Kali Linux (192.168.11.149) dans le même réseau :

```
Carte Ethernet VMware Network Adapter VMnet8 :
Suffixe DNS propre à la connexion. . . . . : 
Description. . . . . : VMware Virtual Ethernet Adapter for VMnet8
Adresse physique . . . . . : 00-50-56-C0-00-08
DHCP activé. . . . . : Non
Configuration automatique activée. . . . . : Oui
Adresse IPv6 de liaison locale. . . . . : fe80::f169:c2eb:6da:74cd%2(préféré)
Adresse IPv4. . . . . : 192.168.11.149(préféré)
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 
IAID DHCPv6 . . . . . : 134238294
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::3%1
NetBIOS sur Tcpip. . . . . : Activé

root@kali: ~
File Edit View Search Terminal Help
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 2670 bytes 2948287 (2.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 883 bytes 129369 (126.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Windows 10 physique et Bitnami (192.168.11.148) dans le même réseau :

```
Carte Ethernet VMware Network Adapter VMnet8 :
Suffixe DNS propre à la connexion. . . . . : 
Description. . . . . : VMware Virtual Ethernet Adapter for VMnet8
Adresse physique . . . . . : 00-50-56-C0-00-08
DHCP activé. . . . . : Non
Configuration automatique activée. . . . . : Oui
Adresse IPv6 de liaison locale. . . . . : fe80::f169:c2eb:6da:74cd%2(préféré)
Adresse IPv4. . . . . : 192.168.11.148(préféré)
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 
IAID DHCPv6 . . . . . : 134238294
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
NetBIOS sur Tcpip. . . . . : Activé

* Documentation: https://help.ubuntu.com/
bitnami@linux:~$ ifconfig
eth0    Link encap:Ethernet HWaddr 00:0c:29:ff:6c:11
        inet addr:192.168.11.148 Bcast:192.168.11.255 Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:feff:6c11/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:165 errors:0 dropped:0 overruns:0 frame:0
        TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:22819 (22.8 KB) TX bytes:6534 (6.5 KB)
        Interrupt:17 Base address:0x1080

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:65536 Metric:1
```

Windows 10 physique et Windows 10 virtuel (192.168.11.150) dans le même réseau :

```
Carte Ethernet VMware Network Adapter VMnet8 :
Suffixe DNS propre à la connexion. . . . . : 
Description. . . . . : VMware Virtual Ethernet Adapter for VMnet8
Adresse physique . . . . . : 00-50-56-C0-00-08
DHCP activé. . . . . : Non
Configuration automatique activée. . . . . : Oui
Adresse IPv6 de liaison locale. . . . . : fe80::f169:c2eb:6da:74cd%2(préféré)
Adresse IPv4. . . . . : 192.168.11.150(préféré)
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 
IAID DHCPv6 . . . . . : 134238294
DUID de client DHCPv6. . . . . : 00-01-00-01-26-F7-99-19-08-62-66-4C-7F-A9
Serveurs DNS. . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
NetBIOS sur Tcpip. . . . . : Activé

Carte Ethernet Ethernet 2 :
Suffixe DNS propre à la connexion. . . . . : 
Description. . . . . : Intel(R) 82574L Gigabit Network Connection
Adresse physique . . . . . : 00-0C-29-37-66-7B
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::d84c:5ef0:ced:bc3a%7(Preferréd)
IPv4 Address. . . . . : 192.168.11.150(Preferréd)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 8 septembre 2021 18:07:52
Lease Expires . . . . . : 8 septembre 2021 19:39:28
Default Gateway . . . . . : 192.168.11.2
DHCP Server . . . . . : 192.168.11.254
DHCPv6 IAID . . . . . : 50334761
DHCPv6 Client DUID. . . . . : 00-01-00-01-28-CA-EC-8F-00-0C-29-37-66-7B
DNS Servers . . . . . : 192.168.11.2
Primary WINS Server . . . . . : 192.168.11.2
NetBIOS over Tcpip. . . . . : Enabled
```

PING de Windows physique vers Kali Linux (192.168.11.149), Bitnami (192.168.11.148) et Windows 10 virtuel (192.168.11.150) :

```
PS X:\> ping 192.168.11.149

Envoi d'une requête 'Ping' 192.168.11.149 avec 32 octets de données :
Réponse de 192.168.11.149 : octets=32 temps<1ms TTL=64
Réponse de 192.168.11.149 : octets=32 temps<1ms TTL=64
Réponse de 192.168.11.149 : octets=32 temps<1ms TTL=64
Réponse de 192.168.11.149 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.11.149:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
PS X:\> ping 192.168.11.148

Envoi d'une requête 'Ping' 192.168.11.148 avec 32 octets de données :
Réponse de 192.168.11.148 : octets=32 temps<1ms TTL=64
Réponse de 192.168.11.148 : octets=32 temps<1ms TTL=64
Réponse de 192.168.11.148 : octets=32 temps<1ms TTL=64
Réponse de 192.168.11.148 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.11.148:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
PS X:\> ping 192.168.11.150

Envoi d'une requête 'Ping' 192.168.11.150 avec 32 octets de données :
Réponse de 192.168.11.150 : octets=32 temps<1ms TTL=128
Réponse de 192.168.11.150 : octets=32 temps<1ms TTL=128
Réponse de 192.168.11.150 : octets=32 temps<1ms TTL=128
Réponse de 192.168.11.150 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 192.168.11.150:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
PS X:\>
```

PING de Bitnami (192.168.11.148) vers Kali Linux (192.168.11.149) et Windows 10 virtuel (192.168.11.150) :

```

bitnami@linux:~$ ping 192.168.11.149
PING 192.168.11.149 (192.168.11.149) 56(84) bytes of data.
64 bytes from 192.168.11.149: icmp_seq=1 ttl=64 time=0.342 ms
64 bytes from 192.168.11.149: icmp_seq=2 ttl=64 time=0.349 ms
64 bytes from 192.168.11.149: icmp_seq=3 ttl=64 time=0.323 ms
64 bytes from 192.168.11.149: icmp_seq=4 ttl=64 time=0.324 ms
64 bytes from 192.168.11.149: icmp_seq=5 ttl=64 time=0.330 ms
^C
--- 192.168.11.149 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.323/0.333/0.349/0.022 ms
bitnami@linux:~$ ping 192.168.11.150
PING 192.168.11.150 (192.168.11.150) 56(84) bytes of data.
64 bytes from 192.168.11.150: icmp_seq=1 ttl=128 time=0.637 ms
64 bytes from 192.168.11.150: icmp_seq=2 ttl=128 time=0.311 ms
64 bytes from 192.168.11.150: icmp_seq=3 ttl=128 time=0.382 ms
64 bytes from 192.168.11.150: icmp_seq=4 ttl=128 time=0.347 ms
^C
--- 192.168.11.150 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.311/0.419/0.637/0.129 ms

```

PING de Windows 10 virtuel (192.168.11.150) vers Bitnami (192.168.11.148) et vers Kali Linux (192.168.11.149) :

```

C:\Users\GIGL>ping 192.168.11.148

Pinging 192.168.11.148 with 32 bytes of data:
Reply from 192.168.11.148: bytes=32 time<1ms TTL=64
Reply from 192.168.11.148: bytes=32 time<1ms TTL=64
Reply from 192.168.11.148: bytes=32 time<1ms TTL=64
Reply from 192.168.11.148: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.11.148:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\GIGL>ping 192.168.11.149

Pinging 192.168.11.149 with 32 bytes of data:
Reply from 192.168.11.149: bytes=32 time<1ms TTL=64
Reply from 192.168.11.149: bytes=32 time<1ms TTL=64
Reply from 192.168.11.149: bytes=32 time<1ms TTL=64
Reply from 192.168.11.149: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.11.149:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

PING de Kali Linux (192.168.11.149) vers Windows 10 virtuel (192.168.11.150) et vers Bitnami (192.168.11.148):


```

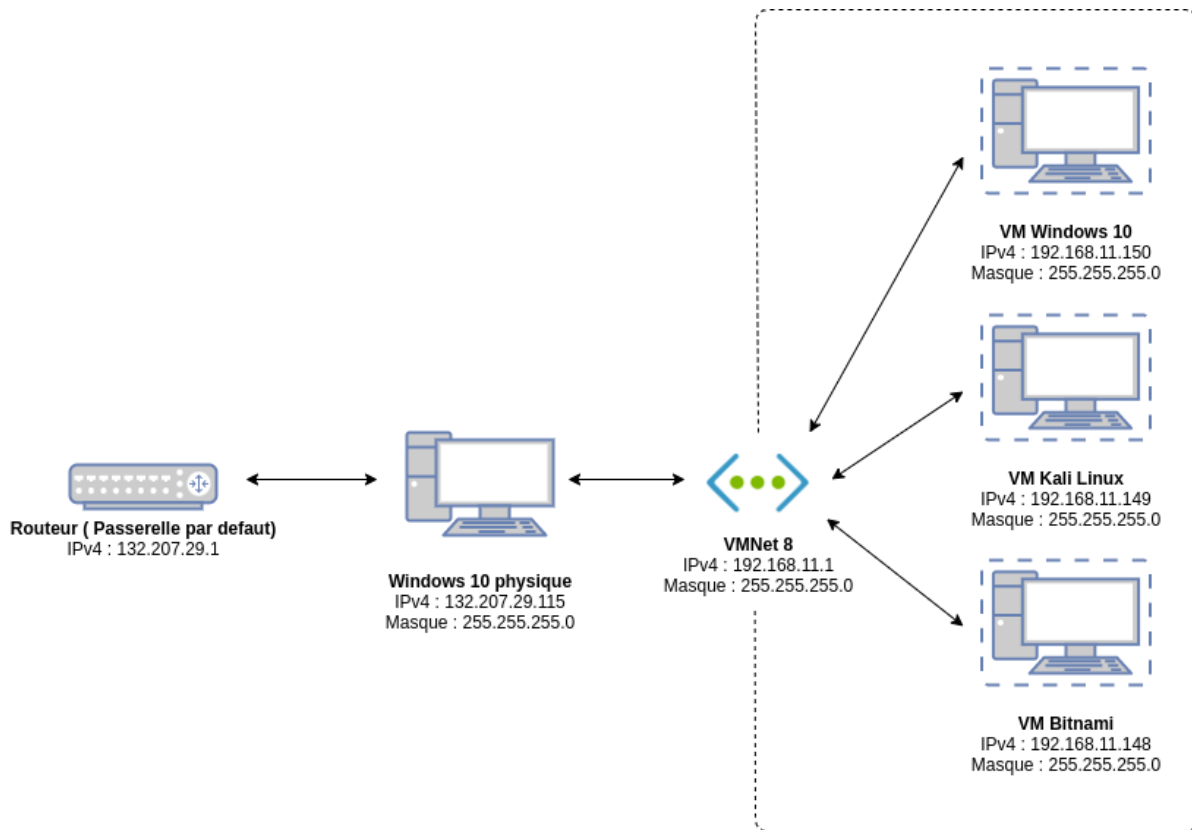
root@kali:~# ping 192.168.11.148
PING 192.168.11.148 (192.168.11.148) 56(84) bytes of data.
64 bytes from 192.168.11.148: icmp_seq=1 ttl=64 time=0.509 ms
64 bytes from 192.168.11.148: icmp_seq=2 ttl=64 time=0.298 ms
64 bytes from 192.168.11.148: icmp_seq=3 ttl=64 time=0.280 ms
64 bytes from 192.168.11.148: icmp_seq=4 ttl=64 time=0.322 ms
64 bytes from 192.168.11.148: icmp_seq=5 ttl=64 time=0.239 ms
64 bytes from 192.168.11.148: icmp_seq=6 ttl=64 time=0.317 ms
^C
--- 192.168.11.148 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5120ms
rtt min/avg/max/mdev = 0.239/0.327/0.509/0.087 ms
root@kali:~# ping 192.168.11.150
PING 192.168.11.150 (192.168.11.150) 56(84) bytes of data.
64 bytes from 192.168.11.150: icmp_seq=1 ttl=128 time=0.362 ms
64 bytes from 192.168.11.150: icmp_seq=2 ttl=128 time=0.333 ms
64 bytes from 192.168.11.150: icmp_seq=3 ttl=128 time=0.346 ms
^C
--- 192.168.11.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.333/0.347/0.362/0.011 ms

```

Q5) Si vous trouvez qu'il n'existe pas de connectivité (ping) entre certains ordinateurs, il faudra expliquer la raison de ce comportement, proposer une solution et finalement appliquer la solution. Justifiez votre réponse à l'aide de captures d'écran. (0.25 point)

Si le ping ne marche pas, cela veut dire que le VMNet des VMs est mal configuré et donc que ces dernières ne sont pas dans le même sous réseau que la machine hôte. Une autre raison peut être la présence d'un pare-feu qui bloque les paquets ICMP. Il faut alors désactiver le pare-feu. Dans notre cas, nous n'avons pas eu de problèmes pour le test de connectivité entre les machines.

Q6) Incluez un diagramme de la configuration du réseau (VM + hôte) en précisant les adresses ipv4 et les masques de sous-réseau de la machine physique ainsi que des machines virtuelles. (1 point)



2.3 TCP/UDP

→	1	0.000000000	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
←	2	0.000179457	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	5	0.902742459	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	6	0.902912809	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	7	1.805804511	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	8	1.805953898	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	18	2.708827806	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	19	2.708964628	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	20	3.612152410	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	21	3.612309325	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	22	4.516137644	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	23	4.516291187	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	26	5.419750505	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	27	5.419884837	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	28	6.323270014	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	29	6.323436849	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	30	7.226407115	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	31	7.226567649	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	32	8.129317826	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	33	8.129424527	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	34	9.032143014	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	35	9.032307247	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	36	9.934668931	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	37	9.934795413	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	38	10.837171726	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	39	10.837334713	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply
	40	11.740561877	192.168.11.1	192.168.11.148	ICMP	74	Echo (ping) request
	41	11.740723033	192.168.11.148	192.168.11.1	ICMP	74	Echo (ping) reply

Q7) Pourquoi pouvez-vous voir cette connexion si la machine n'est ni l'origine ni la destination de la connexion ?

Nous pouvons voir cette connexion si la machine n'est ni l'origine ni la destination de la connexion, car les 3 machines sont sur le même sous réseau virtuel VMNet 8.

Q8) En sélectionnant un paquet dans Wireshark, identifiez les différentes couches du modèle OSI. Quelles sont les informations affichées, consultez l'annexe A ?

```
▶ Ethernet II, Src: Vmware_ff:6c:11 (00:0c:29:ff:6c:11), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
▶ Internet Protocol Version 4, Src: 192.168.11.148, Dst: 192.168.11.1
▼ Internet Control Message Protocol
  Type: 0 (Echo (ping) reply)
  Code: 0
  Checksum: 0x5518 [correct]
  [Checksum Status: Good]
  Identifiant (BE): 1 (0x0001)
  Identifiant (LE): 256 (0x0100)
  Sequence number (BE): 67 (0x0043)
  Sequence number (LE): 17152 (0x4300)
  [Request frame: 1]
  [Response time: 0.179 ms]
▼ Data (32 bytes)
  Data: 6162636465666768696a6b6c6d6e6f707172737475767761...
  [Length: 32]
```

Les paquets étaient composés des éléments suivants :

- **Frame** qui correspond à la couche 1, la couche physique, avec des informations sur les données envoyées notamment le nombre d'octets et la longueur de la trame.
- **Ethernet II** qui correspond à la couche 2, la couche liaison, avec les adresses MAC source et destination.
- **Internet Protocol version** et **Internet Control Message Protocole** font partie de la couche 3, la couche réseau, avec les adresses IP sources et destinations et les informations supplémentaires telle que le type de requête et le temps de réponse.

2.4 FTP

Q9) Discutez, d'un point de vue sécurité, du protocole FTP. (1 point)

Le protocole FTP (File Transfer Protocol) n'a pas été construit pour être sécurisé. Ainsi, ce dernier n'utilise pas d'encryption pour les paquets envoyés sur le réseau. Cela rend toutes les données envoyées par FTP vulnérables à la capture de paquets (sniffing) puisqu'un attaquant au milieu pourrait capturer toutes les données en clair.

Q10) Pouvez-vous voir cette image ? Intercepter l'image. Justifiez votre réponse à l'aide d'une capture d'écran de vos étapes. (1 point)

Oui, comme le trafic des paquets FTP n'est pas crypté nous pouvons reconstruire l'image grâce au TCP Stream. Les étapes pour faire cela se trouvent ci-dessous.

Tout d'abord, nous avons localisé les paquets contenant les données de notre image, puis nous avons utilisé le TCP Stream pour voir le flux des données transitant sur le réseau et nous avons remarqué le string "JFIF" qui est présent dans les entêtes des fichiers JFIF et utilisé par la norme d'encodage JPEG. Ainsi, nous savions que c'était notre image :

```

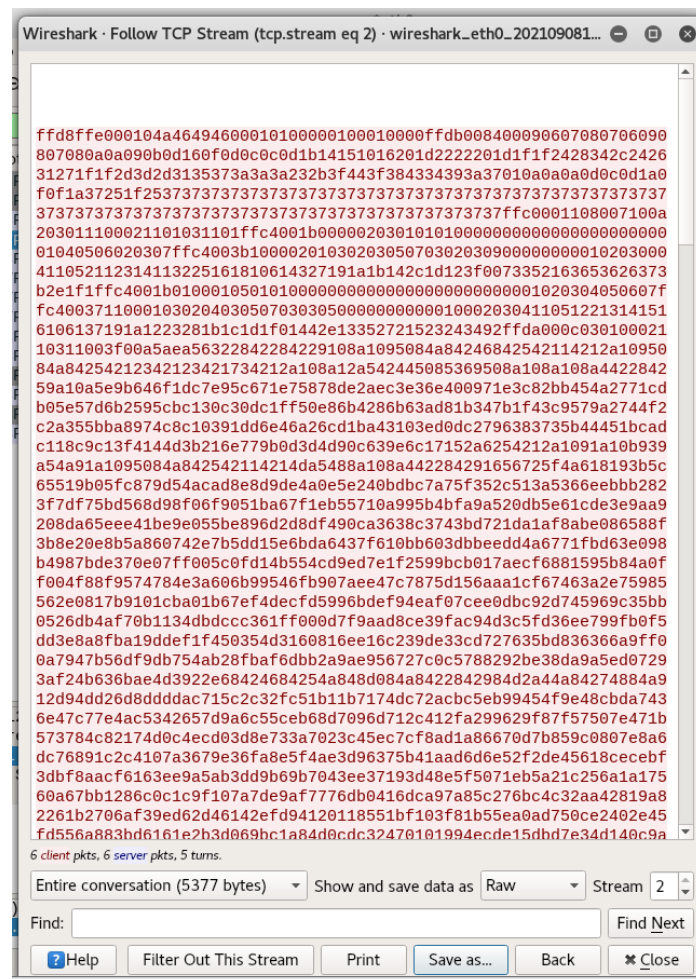
ne 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
Ethernet II, Src: VMware_37:66:7b (00:0c:29:37:66:7b), Dst: VMware_ff:6c:11 (00:0c:29:ff:6c:11)
Internet Protocol Version 4, Src: 192.168.11.159, Dst: 192.168.11.148
Transmission Control Protocol, Src Port: 49881, Dst Port: 39185, Seq: 1, Ack: 1, Len: 1460
Data (1460 bytes data)

```

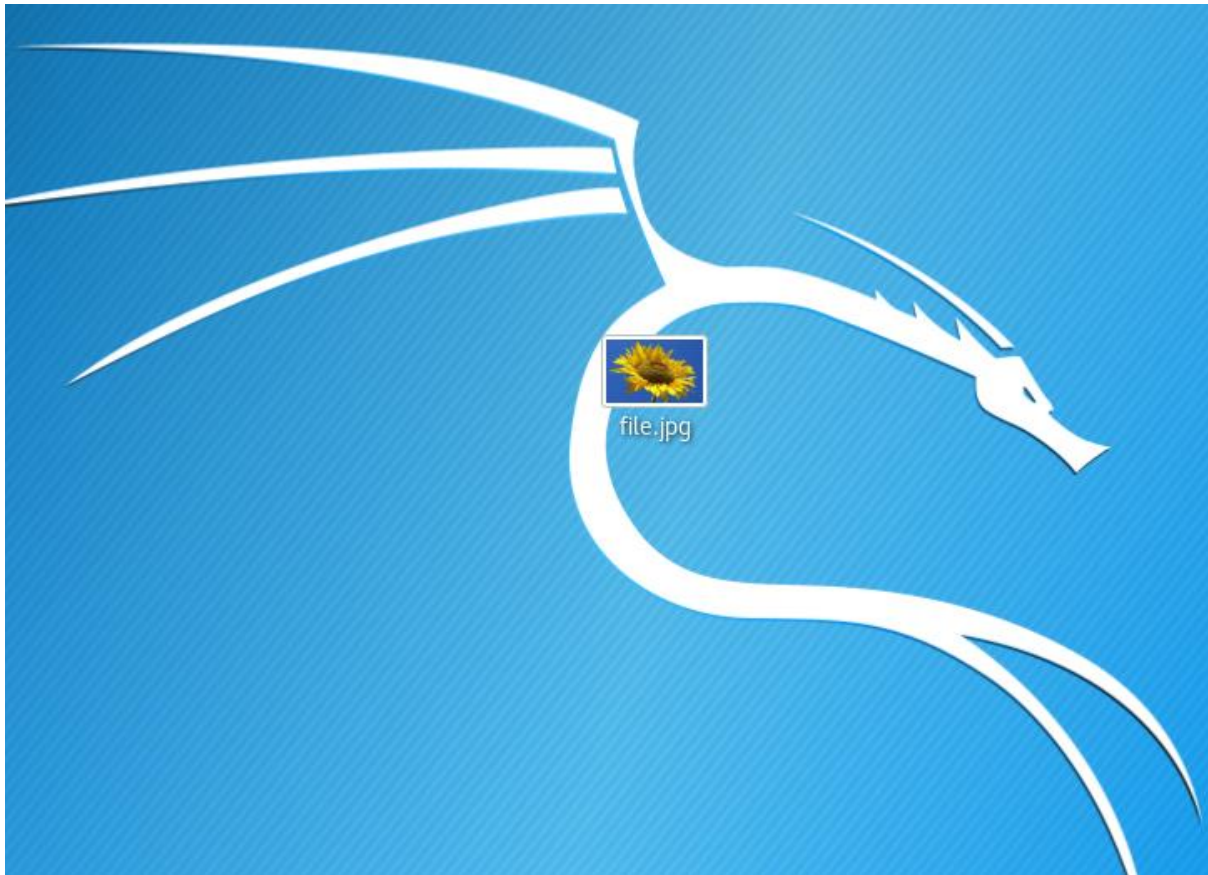
Internet Protocol Version 4 (ip), 20 bytes

input to this VM, move the mouse pointer inside or press Ctrl+G.

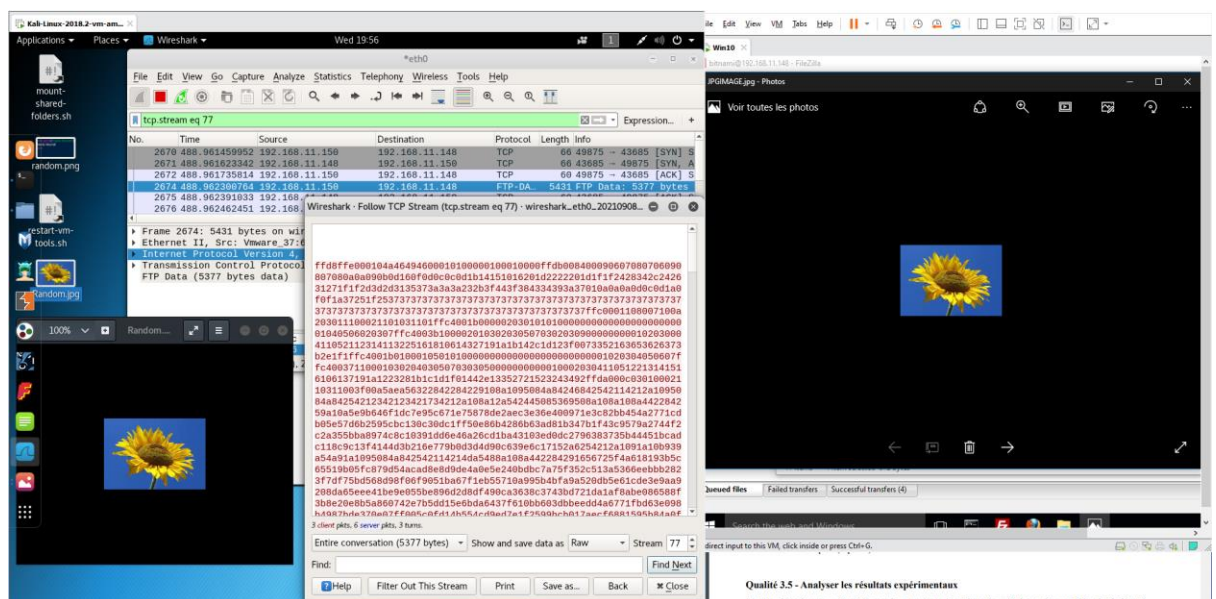
Nous avons ensuite transformé les données ASCII en bytes pour que l'image soit plus facile à extraire :



Après avoir utilisé la fonction “save as” de notre TCP stream pour sauvegarder les bytes dans un fichier JPG, nous avons eu accès à l’image sur notre machine Kali Linux.



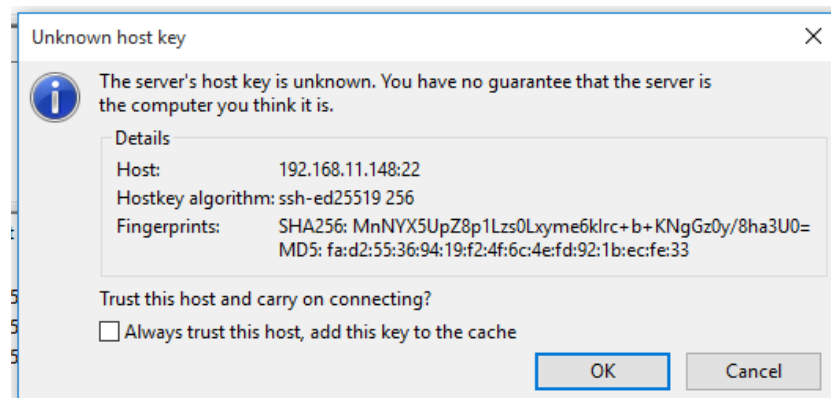
Toutes les opérations sur une même capture d'écran :



2.5 SFTP

Q11) Que signifie l'empreinte digitale affichée lors de la connexion ? (0.5 point)

L'empreinte digitale affichée lors de la connexion est le hash de la clé publique SFTP de la machine Bitnami. Ce hash sert notamment à authentifier la machine Bitnami auprès de la machine Windows.



Q12) Quelle est l'information que vous pouvez trouver de cette connexion dans Wireshark ? (0.5 point)

Nous pouvons voir que le protocole SSH (Secure SHell) a été utilisé pour la communication entre notre machine Windows 10 et la machine Bitnami. Cela a notamment permis des échanges de clés d'encryption et l'envoi de l'image encryptée. Ainsi, nous avons des informations sur la source et la destination (IP, protocole utilisé, etc.), mais rien sur les données des paquets, car ces dernières sont cryptées.

ip.addr == 192.168.11.148						
No.	Time	Source	Destination	Protocol	Length	Info
85	10.869728657	192.168.11.150	192.168.11.148	TCP	66	49653 → 22 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=128 SACK_PERM=1
86	10.869895818	192.168.11.148	192.168.11.150	TCP	66	22 → 49653 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=64
87	10.870116294	192.168.11.150	192.168.11.148	TCP	60	49653 → 22 [ACK] Seq=1 Ack=1 Win=4194304 Len=0
88	10.870399608	192.168.11.150	192.168.11.148	SSHv2	60	Client: Protocol (SSH-2.0-FileZilla_3.34.0)
89	10.870471590	192.168.11.148	192.168.11.150	TCP	60	22 → 49653 [ACK] Seq=1 Ack=27 Win=29248 Len=0
90	10.874319776	192.168.11.148	192.168.11.150	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2)
91	10.874761258	192.168.11.148	192.168.11.150	TCP	1514	22 → 49653 [ACK] Seq=42 Ack=27 Win=29248 Len=1460 [TCP segment of a reassembled PDU]
92	10.874903729	192.168.11.150	192.168.11.148	TCP	60	49653 → 22 [ACK] Seq=27 Ack=1502 Win=4194304 Len=0
93	10.875026534	192.168.11.148	192.168.11.150	SSHv2	242	Server: Key Exchange Init
94	10.875096496	192.168.11.150	192.168.11.148	SSHv2	1254	Client: Key Exchange Init
95	10.880076109	192.168.11.150	192.168.11.148	SSHv2	102	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
96	10.880164782	192.168.11.148	192.168.11.150	TCP	60	22 → 49653 [ACK] Seq=1690 Ack=1275 Win=32128 Len=0
97	10.883233576	192.168.11.148	192.168.11.150	SSHv2	262	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
98	10.936525544	192.168.11.150	192.168.11.148	TCP	60	49653 → 22 [ACK] Seq=1275 Ack=1898 Win=4193792 Len=0
15757	129.622885112	192.168.11.148	192.168.11.150	TCP	60	22 → 49653 [FIN, ACK] Seq=1898 Ack=1275 Win=32128 Len=0
15758	129.622892171	192.168.11.150	192.168.11.148	TCP	60	49653 → 22 [ACK] Seq=1275 Ack=1899 Win=4193792 Len=0
88	10.870399608	192.168.11.150	192.168.11.148	SSHv2	60	Client: Protocol (SSH-2.0-FileZilla_3.34.0)
89	10.870471590	192.168.11.148	192.168.11.150	TCP	60	22 → 49653 [ACK] Seq=1 Ack=27 Win=29248 Len=0
90	10.874319776	192.168.11.148	192.168.11.150	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2)
91	10.874761258	192.168.11.148	192.168.11.150	TCP	1514	22 → 49653 [ACK] Seq=42 Ack=27 Win=29248 Len=1460 [TCP segment of a reassembled PDU]
92	10.874903729	192.168.11.150	192.168.11.148	TCP	60	49653 → 22 [ACK] Seq=27 Ack=1502 Win=4194304 Len=0
93	10.875026534	192.168.11.148	192.168.11.150	SSHv2	242	Server: Key Exchange Init
94	10.875096496	192.168.11.150	192.168.11.148	SSHv2	1254	Client: Key Exchange Init
95	10.880076109	192.168.11.150	192.168.11.148	SSHv2	102	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
Frame 93: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits) on interface 0						
Ethernet II, Src: Vmware ff:6c:11 (00:0c:29:ff:6c:11), Dst: Vmware 37:66:7b (00:0c:29:37:66:7b)						
Internet Protocol Version 4, Src: 192.168.11.148, Dst: 192.168.11.150						
Transmission Control Protocol, Src Port: 22, Dst Port: 49653, Seq: 1502, Ack: 27, Len: 188						
[2 Reassembled TCP Segments (1648 bytes): #91(1460), #93(188)]						
SSH Protocol						
SSH Version 2 (encryption:aes256-gcm@openssh.com mac:<implicit> compression:none)						
Packet Length: 1644						
Padding Length: 10						
Key Exchange						
Message Code: Key Exchange Init (20)						
Algorithms						
Padding String: 000000000000000000000000						
00f0	31	36	30	2c	68	6d 61 63 2d 72 69 70 65 6d 64 31 160,hmac -ripemd1
0000	36	30	40	ef	70 65 6e 73 73 68 2e 63 6f 6d 2c 68 60@opens sh.com,h	
0010	6d 61 63 2d 73 68 61 31 2d 39 30 2c 68 6d 61 63 mac-sha1 -96,hmac					
0020	2d 6d 64 35 2d 39 30 00 00 00 15 6e 6f 6e 65 2c -md5-96,...none,					
0030	7a 6c 69 62 40 ef 70 65 6e 73 73 68 2e 63 6f 6d zlib@ope nssh.com					
0040	00 00 00 15 6e 6f 6e 65 2c 7a 6c 69 62 40 ef 70 ...none,zlib@op					
0050	65 6e 73 73 68 2e 63 6f 6d 00 00 00 00 00 00 00 00 enssh.co m.....					

Q13) Comment utiliseriez-vous Wireshark pour analyser la sécurité d'une entreprise ? (0.5 point)

Nous pouvons utiliser Wireshark pour tester la sécurité d'une entreprise en écoutant sur le réseau et en analysant les paquets de données transitants. On peut alors regarder si les communications des données sensibles sur l'entreprise sont cryptées grâce à l'utilisation de protocoles sécurisés tels que SSH, SFTP. Cela permettrait notamment à l'entreprise d'éviter que les attaquants aient accès à des données sensibles en clair en écoutant le réseau.

Q14) Est-il possible d'intercepter l'image ? Justifiez votre réponse (0.5 point)

Non il n'est pas possible d'intercepter l'image, car SFTP utilise SSH pour faire le transfert de l'image donc toutes les données transitant par ce protocole sont cryptées comme vous pouvez le remarquer dans les captures d'écran suivantes :

```
SSH-2.0-FileZilla_3.40.0
SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2
.....Fv.....c.AC.....curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-
exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,rsa2048-sha256,rsa1024-sha1,diffie-hellman-group1-
sha1...Wssh-ed25519,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-rsa,ssh-dss...aes256-gcm@openssh.com,aes256-
ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc,chacha20-
poly1305@openssh.com,blowfish-ctr,blowfish-cbc,3des-ctr,3des-cbc,arcfour256,arcfour128...aes256-gcm@openssh.com,aes256-ctr,aes256-
cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc,chacha20-
poly1305@openssh.com,blowfish-ctr,blowfish-cbc,3des-ctr,3des-cbc,arcfour256,arcfour128...hmac-sha1-256,hmac-sha1,hmac-sha1-96,hmac-
md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-etm@openssh.com...hmac-sha2-256,hmac-
sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-
etm@openssh.com...none,zlib...none,zlib...d.1...c2...l
...c.D\...0.9...z...curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-
exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1.../ssh-rsa,ssh-dss,ecdsa-sha2-
nistp256,ssh-ed25519...aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-
poly1305@openssh.com,aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-cbc,arcfour,rijndael-cbc@lysator.liu.se...aes128-
ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes128-cbc,
3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-cbc,arcfour,rijndael-cbc@lysator.liu.se...hmac-md5-etm@openssh.com,hmac-sha1-
etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-
ripemd160-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-md5,hmac-
sha1,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-ripemd160,hmac-ripemd160@openssh.com,hmac-sha1-96,hmac-
md5-96...hmac-md5-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-etm@openssh.com,hmac-md5-96-
etm@openssh.com,hmac-md5,hmac-sha1-96,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-ripemd160,hmac-
ripemd160@openssh.com,hmac-sha1-96,hmac-md5-96...none,zlib@openssh.com...none,zlib@openssh.com.....
...b.k...@.i.y...H.k...y.o.I...3...ssh-ed25519...F...5...nUq.*&...t...S.W...@2...t.6.v.3.a.
8.V#q...VKA...S...ssh-ed25519...>.sm.>.d.Q$...p.T...9...1VK<.<8.../...$"<.....
.....
.../.J...9.M...
.I.b...C.3...3...{zn,W?}...Z...6...8...R:WX+.T5.y.j...s...#...7.0...0.E...15UJ.f.6;...\
$.i.i...|...N.i,Z...u...s}.Ty.%...".e.l...PM...9...}.L+A.i...}.s
%9.....5dzk.m.....K...;j...'. $
]
.....pr.[S.(J.m.F.*...c.)...A...iv.n.. '
v..@wq...7R.d...f.o...0U...E.'hv.wk'.2...?H.J0...1.9@{h...!L.W...r@|l.')...@.MG.%
...N.Ze.6.3...qR{v5...ww.g...&...}.Z0@...u.Z
N...<h.N...0s.F...K...M...j!6<5W.K...?<.
D.u...A"-...F2...MF.K.3...M...A...I.K...}...k...S..d.H.f.z@...;'A\...9...}.Pv.n.F..1.Y../".
a..2.IAH.V.
...e..J6...n...s.jf...W.U...7t&.7.5...}...?...\q/2.W..(a...@lZ.ZRUG!hI...T.<...<v...?t...(. .... )...mn)...$.
8...H...l...T...T...>...<IY.B...k.[]...p.X...1.9...@0.y...Q...a,x.Y...|...^...P.e
0...<...<H...}.a."/;...u
...y..>hp...n..
...;..Rz..nj...A...(.MF...<N..u..ko...(. .... )Q r..zd.O.F...i...ku...6..j...}...A$...Br...E.CN.a...e'9...%[.1.5.X...
{.../.OY.Z.1..vi...1...}>1...uu.u..x...6.W.K...[...y.V.K8P...V...e.x.g...n.m.M7.SxNR.8G...;nf.Rv;a...].
8...c[...A.V...u.8#i$.a.0
...H2...a...="...zh...e0...(.F\...0.j...y.f9...4.f...%Z%.U.O...|...}%F.n...$S...r.j...Z^)...
3.../3...Z...B...*2Prb...F.z...oZ<6.m...{Z.yc.Z...1e...4..rY...9...2.m.0...V.f.I.9...pB6vF.HY.c@...>.I..h.IT.P4c.
...",Z.W.
.
m...M...v.r...m.sIHS...h.8...S+..G.A.i.v.QZa%./....}.b...V...?&...x..65m...T@B...!...UbJx;2o!...9e...}...B/
TE.S.X...u.r...d...x]s.@JXh..J"...eA...4...jW...1...Y0...#.LI...+0.N>k...
e]...S...?...b%q#...D..a...<j..l...a...f...Z...y...&...\...m.Q...<9..2;2...:$.H...!...J..P.qk.Z
...J...!...1...@{.6.3@...}.s..9B.s=...j.qY...{...~4..KBmg.T5...H^c...I..o.Tz...[.1.N...a...+d.t.C.c.f...}.t.j.S{b,cn.
5f.B...H...0...B...7.KYY*...}.2.../F...D\J
#v$.#kd...&.q...f...r...%2..n'g.K.y...{.H...;$.A..c&W...}X...
#
...6...}.p...U...w.x...>...a8R$.w6&TP`...R.k.NSS$.B.&..."...+7.....s`C...g...!|{=...D...^..R^...jSc.
9P...F...c.Ot5...!DHV...\<.s..b.A.f.
...F...g.M!
[...Z...J...

```

3.1 Partie B - ASA Adaptive Security Appliance

Capture d'écran de la commande "show running-config" :

```
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
```



```

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

```

Capture d'écran de la commande "show ip" :

```

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
POLYFW01#

```

Capture d'écran de la commande "show running-config" après l'exécution des commandes de configuration :

```
: 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 192.168.199.5 255.255.255.0
!
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
http 192.168.199.0 255.255.255.0 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
```

```

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:3bfbb34belc8050619c09ae2ccbe587c
: end

```

Capture d'écran de la commande "show ip" après l'exécution des commandes de configuration :

```

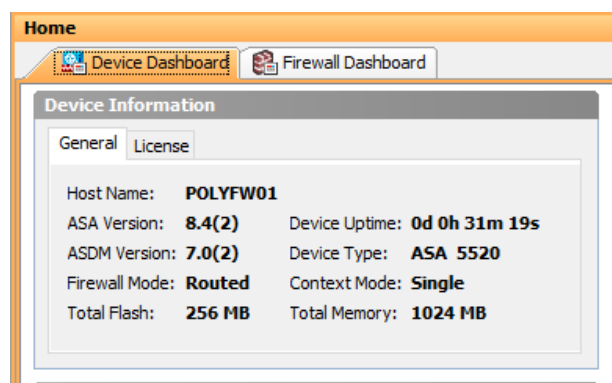
POLYFW01(config-if)# show ip
System IP Addresses:
Interface          Name          IP address      Subnet mask      Method
GigabitEthernet0   INSIDE        192.168.199.5   255.255.255.0    manual
Current IP Addresses:
Interface          Name          IP address      Subnet mask      Method
GigabitEthernet0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

```

3.2 Informations générales et tableau de bord des ASA

**Q15) Quel modèle d'ASA virtuel avez-vous ? Quelle est la version IOS de cet ASA ?
Type de License ? (0.5 point)**

Le modèle d'ASA que nous avons est un pare-feu appartenant à la série 5500-X, plus précisément un ASA 5520. La version IOS de cet appareil est 8.4(2). Enfin, quant au type de licence, nous avons une licence VPN Plus.



Device Information			
General		License	
License:	VPN Plus	GTP/GPRS:	Disabled
Encryption:		Physical Interfaces:	Unlimited
VLANs:	100	VPN Peers:	0
Failover:	Disabled	SSL VPN Peers:	5000
Security Contexts:	0	More Licenses	

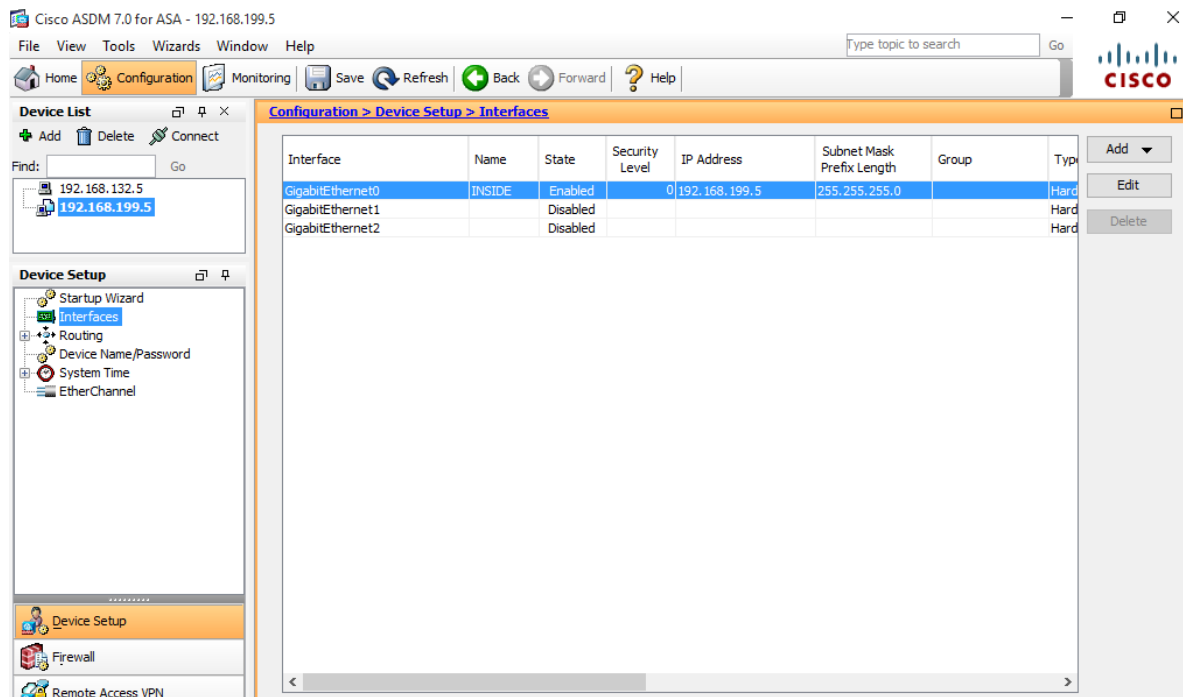
Q16) Quelle est l'utilité d'un système ASA? Aidez-vous en consultant les informations disponibles dans le tableau de bord de l'ASA. (0.5 point)

Le système ASA permet de filtrer les accès entrant et sortant d'un réseau. Il permet de sécuriser les entreprises de l'extérieur. De manière générale, un ASA est un appareil de sécurité combinant les fonctions suivantes : pare-feu, antivirus, prévention d'intrusion et VPN. C'est un système proactif qui procure une défense contre les attaques qui permet d'éviter que ces attaques se propagent à travers le réseau.

Q17) Combien d'interfaces et de zones sont actuellement configurées (précisez-leur nom dans votre rapport) sur le ASA et combien peuvent être créés dans cet ASA virtuel ? (0.5 point)

Il peut y avoir au maximum 3 interfaces, INSIDE, DMZ et OUTSIDE. Dans notre cas, seule INSIDE était configurée.

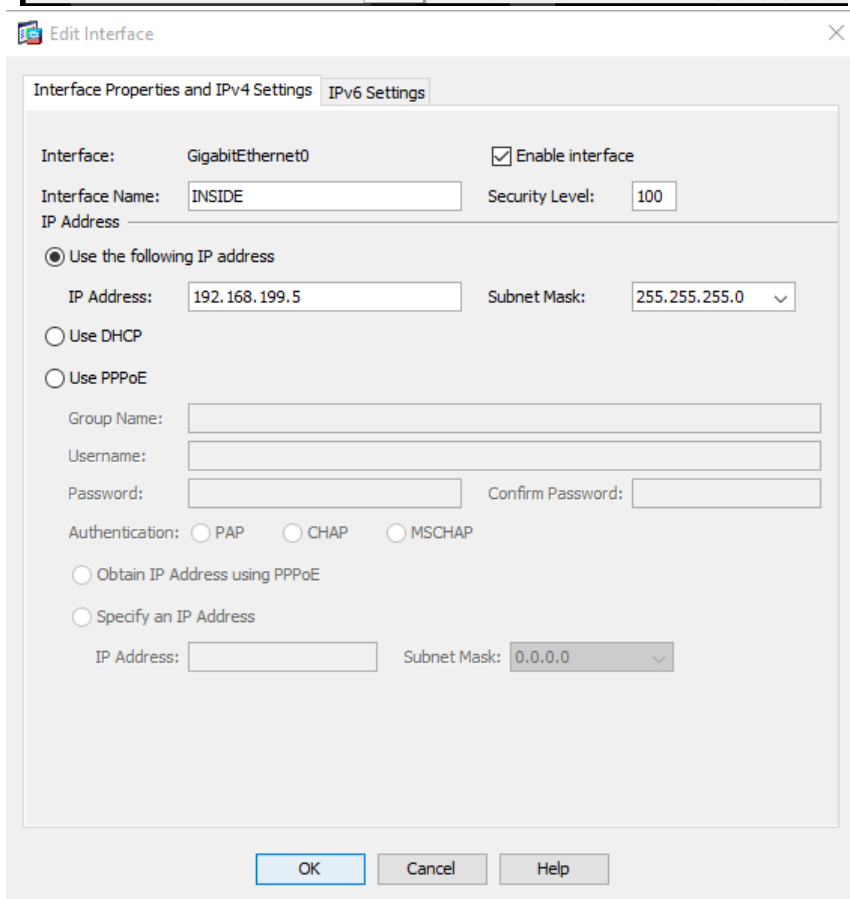
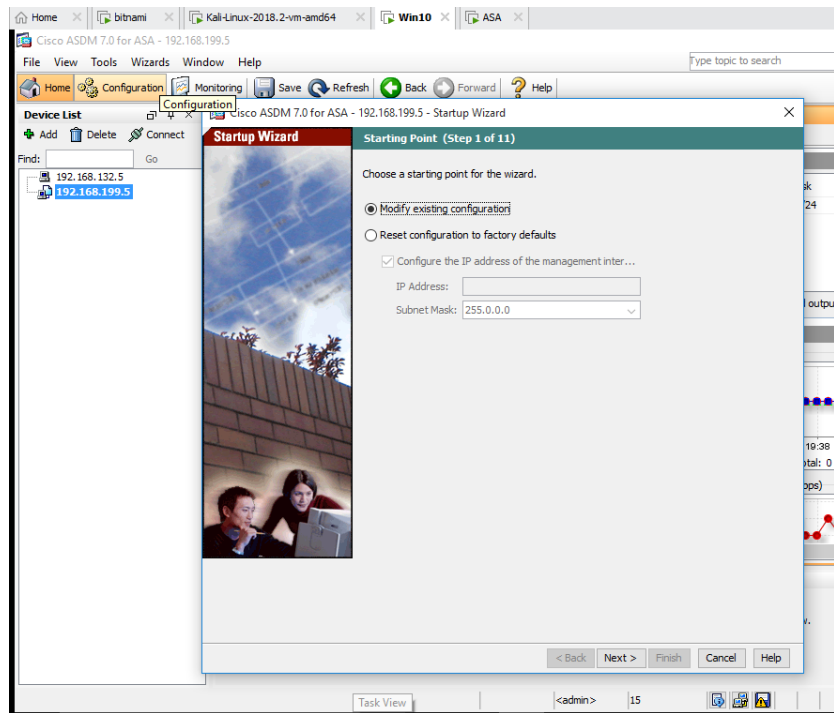
Interface Status				
Interface	IP Address/Mask	Line	Link	Kbps
INSIDE	192.168.199.5/24	⬆ up	⬆ up	3
Select an interface to view input and output Kbps				



3.3 Assistants (Wizards)

Q18) Quelle est la signification du niveau de sécurité de l'interface d'ASA ? Configurez le niveau de sécurité de l'interface INSIDE à 100 et DMZ à 50, et ajoutez une capture d'écran. (2.5 points)

Le niveau de sécurité de l'interface INSIDE est de 100. C'est le niveau le plus haut. La zone de l'interface INSIDE permet notamment de sécuriser les données internes à l'entreprise et donne un accès très limité depuis l'extérieur de l'entreprise. Voici la configuration de ce dernier dans le wizard :



L'interface DMZ a un niveau de sécurité de 50. Cela signifie une sécurité moins intense dans la DMZ pour que les utilisateurs externes aient accès à certaines ressources de l'entreprise telles que les serveurs web. Voici sa configuration dans le wizard :

Edit Interface

Interface Properties and IPv4 Settings | IPv6 Settings

Interface: GigabitEthernet1 ☒ Enable interface

Interface Name: DMZ Security Level: 50

IP Address

☒ Use the following IP address

IP Address: 192.168.126.5 Subnet Mask: 255.255.255.0

☐ Use DHCP

☐ Use PPPoE

Group Name:

Username:

Password: Confirm Password:

Authentication: ☐ PAP ☐ CHAP ☐ MSCHAP

☐ Obtain IP Address using PPPoE

☐ Specify an IP Address

IP Address: Subnet Mask: 0.0.0.0

OK Cancel Help

Enfin l'interface OUTSIDE a un niveau de sécurité 0 puisque cet environnement est destiné à communiquer avec internet et donc avec l'extérieur de l'entreprise. Voici sa configuration dans le wizard :

Startup Wizard

Cisco ASDM 7.0 for ASA - 192.168.199.5 - Startup Wizard

Outside Interface Configuration (Step 3 of 11)

Interface Settings | IPv6 Interface Settings

Configure the outside interface of the ASA. Check with your ISP to determine which option to use.

Interface Properties

Interface: GigabitEthernet2 ☒ Enable interface

Interface Name: OUTSIDE Security Level: 0

IP Address

☒ Use the following IP address

IP Address: 192.168.11.5 Subnet Mask: 255.255.255.0

☐ Use DHCP

The ASA will obtain an IP address from a DHCP server. Ensure that a DHCP server is configured on your corporate network or by your ISP.

☐ Obtain default route using DHCP

☐ Use PPPoE

The ASA will obtain its IP address from a PPPoE server if you do not specify an IP address in next step. Ensure that a PPPoE server is configured by your ISP.

OK Cancel Help

Voici un sommaire des différentes interfaces configurées :

Configuration > Device Setup > Interfaces				
Interface	Name	State	Security Level	IP Address
GigabitEthernet0	INSIDE	Enabled	100	192.168.1.1
GigabitEthernet1	DMZ	Enabled	50	192.168.1.2
GigabitEthernet2	OUTSIDE	Enabled	0	192.168.1.3

3.4 Règles de NAT, L4 pour pare-feu ASA

Q19) Si un pare-feu n'a pas de règles, est-ce que les paquets sont réacheminés ou jetés ? (0.25 point)

Par défaut, si aucune règle n'est spécifiée, les paquets sont automatiquement jetés.

Q20) Quelle est la différence entre NAT et PAT ? (0.25 point)

Le **NAT (Network Address Translation)** et le **PAT (Port Address Translation)** traduisent des adresses IP privées en adresses IP publiques. La différence ici est qu'avec un PAT, plusieurs adresses IP privées vont pouvoir partager la même adresse de translation publique puisque cette dernière sera associée à plusieurs ports. En effet, avec le PAT, chaque IP privée et chaque IP publique vont avoir des ports qui leur sont associés. Ainsi, la translation dans la table de translation ne se fait pas seulement de IP privée à IP publique comme pour un NAT, mais cela prend aussi en compte le port de l'adresse IP privée et de l'adresse IP publique.

Référence :

<https://theithollow.com/2013/03/05/nat-vs-pat/>

Q21) Pourquoi existe-t-il des règles pare-feu (Access Rules), des règles NAT (NAT Rules) et des règles de services politiques (Service Policy Rules). (0.5 point)

Les règles de pare-feu servent à contrôler le flux d'entrées et sorties du trafic Internet entre ce dernier et le réseau local. Elles servent à traiter les paquets selon des critères définis.

Les règles NAT servent à configurer et contrôler le trafic entre le réseau privé et le réseau public. Généralement, Ces dernières sont appliquées sur les paquets qui sont passés et ont été acceptés par les règles de pare-feu.

Les règles de services politiques appliquent des services au trafic que nous autorisons, en d'autres mots, tout trafic autorisé par les règles d'accès peut avoir des politiques de service appliquées et donc recevoir un traitement spécial.

Référence :

<https://www.nextiva.com/support/articles/what-are-firewall-access-rules.html#:~:text=Firewall%20Access%20Rules%20control%20the,traffic%20on%20the%20local%20network.>

<https://docs.sophos.com/nsg/sophos-firewall/18.0/Help/en-us/webhelp/onlinehelp/nsg/sfos/concepts/NATRules.html>

<https://www.cisco.com/c/en/us/td/docs/security/asa/asa98/asdm78/firewall/asdm-78-firewall-config/inspect-service-policy.html>

Q22) Créez un NAT pour permettre à tous les utilisateurs des réseaux « *INSIDE* » et « *DMZ* » d'aller vers internet (ajoutez une capture d'écran à chaque étape). Vérifiez que la machine virtuelle Windows 10 peut aller sur internet. Expliquer aussi ce qu'est une route statique (1.5 points)

Le routage statique est un type de routage dans lequel il faut spécifier explicitement et manuellement le chemin entre deux routeurs. Les routes de ce type de routage ne peuvent pas être mises à jour automatiquement, contrairement au routage dynamique, il faut passer par des tables de routage créées manuellement pour cela. Les routes statiques ont pour avantages de consommer moins de bande passante et de ressources en CPU, mais ces dernières sont faites uniquement pour les réseaux de petite taille dont la topologie est très simple.

Référence :

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus3000/sw/unicast/503_u1_2/nexus3000_unicast_config_qd_503_u1_2/l3_route.html

Les étapes que nous avons suivi pour que la machine virtuelle Windows 10 puisse aller sur internet sont les suivantes :

- Tout d'abord, nous avons créé une règle de NAT statique. Cela nous a permis depuis l'interface *INSIDE* d'atteindre l'interface *OUTSIDE* qui est connectée à internet.

Edit NAT Rule

Match Criteria: Original Packet

Source Interface: **INSIDE** Destination Interface: **OUTSIDE**

Source Address: **any** Destination Address: **any**

Service: **any**

Action: Translated Packet

Source NAT Type: **Static**

Source Address: **OUTSIDE** Destination Address: **-- Original --**

☐ PAT Pool Translated Address: Service: **-- Original --**

☐ Round Robin

☐ Fall through to interface PAT

Options

☒ Enable rule

☐ Translate DNS replies that match this rule

☐ Disable Proxy ARP on egress interface

☐ Lookup route table to locate egress interface

Direction: **Both**

Description:

OK **Cancel** **Help**

- Nous avons ensuite créé une route statique sur l'interface OUTSIDE avec comme gateway le serveur WINS de VMNET8. Cela nous a permis de diriger le trafic vers le serveur WINS de VMNET8.

Configuration > Device Setup > Routing > Static Routes

Specify static routes.

Filter: ☒ Both ☐ IPv4 only ☐ IPv6 only

Interface	IP Address	Netmask/ Prefix Length	Gateway IP	Metric/ Distance	Options
OUTSIDE	0.0.0.0	0.0.0.0	192.168.11.2	1	None

- Voici le sommaire de la règle de NAT que nous avons rajouté:

Configuration > Firewall > NAT Rules

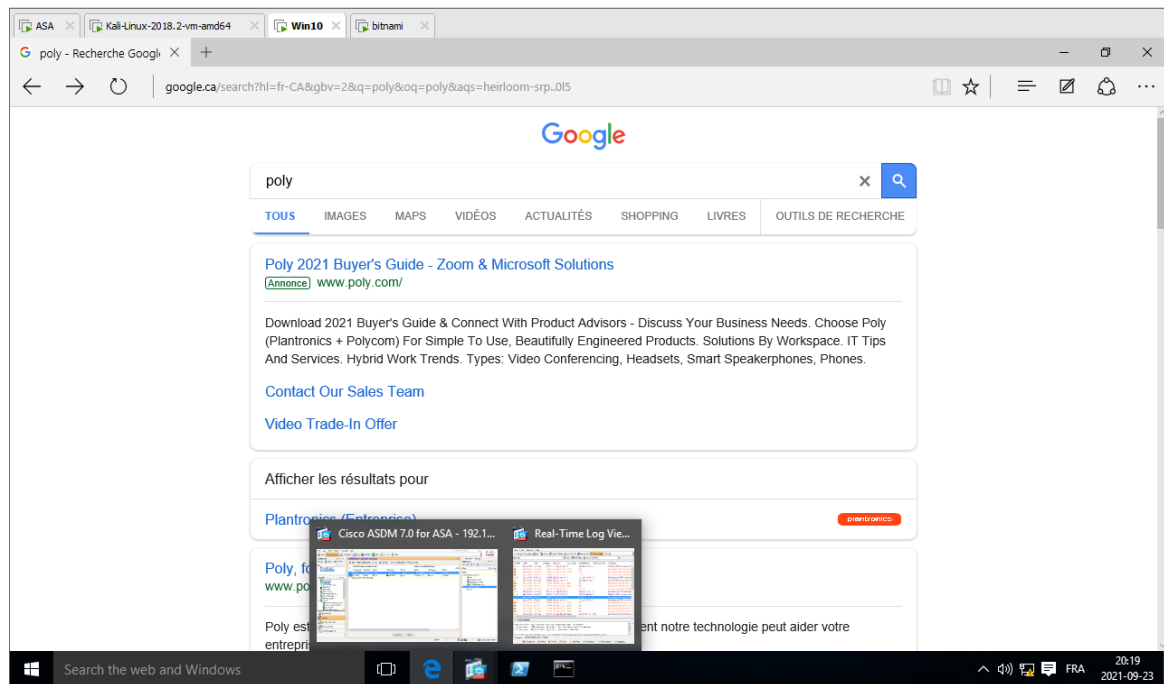
Monitoring Save Refresh Back Forward Help

Add Edit Delete Up Down Copy Paste Find Diagram Packet Trace

#	Match Criteria: Original Packet					Action: Translated Packet			Options
	Source Intf	Dest Intf	Source	Destination	Service	Source	Destination	Service	
1	INSIDE	OUTSIDE	any	any	any	OUTSIDE (S)	-- Original --	-- Original --	
	OUTSIDE	INSIDE	any	OUTSIDE	any	-- Original -- (S)	any	-- Original --	

"Network Object" NAT (No rules)

- Grâce à cela, comme vous pouvez le remarquer dans la capture d'écran ci-dessous, la VM Windows 10 avait accès à internet.



Q23) Effectuez les mêmes étapes afin de donner accès à Internet à la machine virtuelle bitnami présente sur le VMnet 2. (1.5 points)

Nous avons suivi les mêmes étapes que dans la question précédente. Ainsi, voici notre règle de NAT statique pour que les paquets venant de la DMZ transitent jusqu'à l'interface OUTSIDE :

Edit NAT Rule

Match Criteria: Original Packet

Source Interface: **DMZ** Destination Interface: **OUTSIDE**

Source Address: **any** Destination Address: **any**

Service: **any**

Action: Translated Packet

Source NAT Type: **Static**

Source Address: **OUTSIDE** Destination Address: **-- Original --**

☐ PAT Pool Translated Address: **any** Service: **-- Original --**

☐ Round Robin

☐ Fall through to interface PAT

Options

☒ Enable rule

☐ Translate DNS replies that match this rule

☐ Disable Proxy ARP on egress interface

☐ Lookup route table to locate egress interface

Direction: **Both**

Description:

OK **Cancel** **Help**

Voici le sommaire des règles de NAT que nous avons jusqu'à présent avec la nouvelle règle ajoutée :

Configuration > Firewall > NAT Rules

[Add](#)
[Edit](#)
[Delete](#)
[Up](#)
[Down](#)
[Copy](#)
[Paste](#)
[Find](#)
[Diagram](#)
[Packet Trace](#)

#	Match Criteria: Original Packet					Action: Translated Packet			Options
	Source Intf	Dest Intf	Source	Destination	Service	Source	Destination	Service	
1	INSIDE	OUTSIDE	any	any	any	OUTSIDE (S)	-- Original --	-- Original --	
	OUTSIDE	INSIDE	any	OUTSIDE	any	-- Original -- (S)	any	-- Original --	
2	DMZ	OUTSIDE	any	any	any	OUTSIDE (S)	-- Original --	-- Original --	
	OUTSIDE	DMZ	any	OUTSIDE	any	-- Original -- (S)	any	-- Original --	
"Network Object" NAT (No rules)									

Enfin, pour tester la connexion internet sur la machine Bitnami, nous avons utilisé la commande "wget", car le ping utilise le protocole ICMP, mais nos règles d'accès ne laissaient pas encore passer ce type de paquets sur le réseau. Comme vous pouvez le voir dans la capture ci-dessous, nous sommes arrivés à télécharger la page index.html de google.com grâce à la commande "wget" donc la machine bitnami avait bien accès à internet.


```

bitnami@linux:~$ wget google.com
--2021-09-24 01:19:05-- http://google.com/
Resolving google.com (google.com)... 172.217.13.142, 2607:f8b0:4020:805::200e
Connecting to google.com (google.com)|172.217.13.142|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2021-09-24 01:19:06-- http://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.13.132, 2607:f8b0:4020:805::2004
Connecting to www.google.com (www.google.com)|172.217.13.132|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

[ <=> ] 14,434 --.-K/s in 0.009s

2021-09-24 01:19:06 (1.57 MB/s) - 'index.html' saved [14434]

bitnami@linux:~$

```

Nous avons aussi regardé à l'intérieur de index.html et ce dernier était bien complet comme vous pouvez le voir ci-dessous :


```

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-CA"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="d7yq2T8o0AQGzU20p+Mxfg==">(function(){window.google={kEI:'iidNYcLmB5DjUqPSjegK',kEXPI:'0,1302536,56873,1709,4349,207,4804,925,1391,383,246,5,1354,5250,1122516,1197738,634,328895,51223,16115,28684,17572,4859,1361,9291,3028,17580,4020,978,13228,3847,10622,14763,4284,2775,919,5081,1593,1279,2212,241,289,149,1103,840,1983,4314,108,3406,606,2023,2297,14670,3227,1989,856,9,5597,6755,5096,15767,553,908,2,941,2614,3783,9359,3,576,1014,1,5444,149,11323,2652,4,1528,2304,1236,5226,577,74,1983,2626,2015,6573,7038,2725,2039,2658,4243,3114,30,13628,2305,638,1494,16786,5818,2539,992,3102,3138,6,908,3,3541,1,14710,1814,283,912,5992,1161,14170,2,1395,1715,2,13033,989,1931,784,255,2870,1680,743,5853,2050,6823,447,1143,1160,1365,5335,2378,2720,2985,1053,2,506,3,148,3899,681,745,4635,3640,2,6,1,949,693,5308,32,735,1933,2635,243,2335,10,3122,546,2072,1919,3,583,793,1265,91,175,632,2248,720,2,535,1855,2632,2875,114,77,41,197,693,957,992,1951,1084,847,1506,681,583,60,1824,40,504,1119,411,99,47,2,142,1153,971,8,187,276,1100,300,785,197,103,251,64,380,4,1549,213,387,573,285,339,600,119,154,1003,9,664,275,23,20,90,274,62,422,296,254,5570997,446,72,88,153,1802576,4193459,521,2800696,882,444,1,2,80,1,1796,1,9,2553,1,889,795,2,561,1,4265,1,1,2,1331,4142,2609,155,17,13,72,139,4,2,20,2,169,13,19,46,5,39,96,548,29,2,2,1,2,1,2,2,7,4,1,2,2,2,2,2,353,513,186,1,1,158,3,2,2,2,2,2,4,2,3,3,269,551,7,5,3,20,2,4,10,28,10,25,2,6,3,5,4,10,38,5,2,1,41,1,1,1,1,13,23654059,299865,4041351,276,62,3,2414,448,2,1041,9,3239,1274,262,883,636,219,775,806354',kBL:'lszj'};google.sn='webhp';google.kHL='en-CA'}})();(function(){
"index.html" [Incomplete last line][converted] 18 lines, 14436 characters

```

Q24) Ajoutez les règles de pare-feu, soit des règles d'accès (Access Rules), permettant à la machine Windows 10 (INSIDE) d'effectuer un ping sur la machine bitnami (DMZ) sans que la machine Kali linux (OUTSIDE), située à l'extérieur du réseau, puisse le faire. (2 points)

Tout d'abord, nous avons commencé par créer des "Network Object". Cette étape n'était pas nécessaire, mais nous trouvions que ça rendait les choses plus claires. Nous en avons donc créé pour la machine Bitnami, Windows 10 et Kali Linux comme vous pouvez le voir dans les captures d'écrans ci-dessous :

 Edit Network Object ✕

Name:


Type:

IP Version: ☒ IPv4 ☐ IPv6

IP Address:

Description:

NAT ⌵

 Edit Network Object ✕

Name:


Type:

IP Version: ☒ IPv4 ☐ IPv6

IP Address:

Description:

NAT ⌵

 Edit Network Object ✕

Name:

Type:















IP Version: ☒ IPv4 ☐ IPv6

IP Address:

Description:

NAT ⌵

Ensuite nous avons créé des règles permettant aux paquets ICMP de transiter entre la VM Windows 10 et Bitnami, car la commande “ping” utilise le protocole ICMP. Le résumé de ces règles se trouve ci-dessous :

#	Enabled	Source Criteria:		Destination Criteria:	Service	Action	Hits	Logging	Time
		Source	User	Destination					
DMZ (1 incoming rule)									
1	<input checked="" type="checkbox"/>	 bitnami		 win10	 icmp	 Permit	 7		
INSIDE (1 incoming rule)									
1	<input checked="" type="checkbox"/>	 win10		 bitnami	 icmp	 Permit	 4		
OUTSIDE (0 implicit incoming rules)									
Global (1 implicit rule)									
1		 any		 any	 ip	 Deny			

Avec les règles définies, nous pouvons voir que la VM Windows 10 peut ping la machine Bitnami :

```
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>
```

On peut aussi voir que la machine Kali Linux ne peut pas atteindre la machine Bitnami avec un ping :

```
root@kali:~# ping 192.168.126.100
PING 192.168.126.100 (192.168.126.100) 56(84) bytes of data.
^C
--- 192.168.126.100 ping statistics ---
87 packets transmitted, 0 received, 100% packet loss, time 88049ms
root@kali:~#
```

Ainsi, comme demandé, la machine Windows 10 (INSIDE) peut effectuer un ping sur la machine bitnami (DMZ) sans que la machine Kali linux (OUTSIDE) ne le puisse.