
TP Réseau

Netcat et autres utilitaires

Olivier LASNE

2020-11-28

TP Réseau

Introduction

Dans ce TP nous allons voir comment utiliser les fonctionnalités réseau de Linux.

À la fois communiquer en TCP, scanner un réseau pour découvrir les machines sur un réseau et les ports ouverts / services disponibles.

Configuration du réseau hôte

Pour ce TP, nous allons utiliser notre machine Kali et Metasploitable.

Créer le réseau privé hôte

Un *réseau privé hôte* est un réseau virtuel, qui connecte des machines virtuelles et qui est accessible *uniquement aux machines virtuelles de VirtualBox*, et à la machine faisant tourner VirtualBox.

Après avoir ouvert **VirtualBox**, cliquer sur **Fichier > Gestionnaire de réseau hôte**. Normalement la configuration suivante est affichée :

Réseau

Créer Supprimer Propriétés

Nom	Adresse/Masque IPv4	Adresse/Masque IPv6	Serveur DHCP
vboxnet0	192.168.56.1/24	fe80::800:27ff:fe00:0/64	<input checked="" type="checkbox"/> Activer

Interface

Serveur DHCP

☐ Configurer la carte automatiquement
☒ Configurer la carte manuellement

Adresse IPv4 :

Masque réseau IPv4 :

Adresse IPv6 :

Longueur du préfixe du masque réseau IPv6 :

Réinitialiser Appliquer Fermer

FIG. 1: gestionnaire de réseau hôte

Si il n'y a pas d'interface réseau. Cliquer sur **Créer**, pour créer une nouvelle interface vboxnet0.

Configurer les interfaces de Kali

Pour **Kali** on garde une interface en **NAT** de façon à pouvoir accéder à Internet, et on crée une **seconde interface** pour le **réseau privé hôte**.

1. Éteindre la machine virtuelle Kali.
2. Sélectionner la machine virtuelle dans VirtualBox.
3. Appuyer sur le **bouton configuration** (icone en forme d'engrenage)
4. Dans la barre à gauche cliquer sur **Réseau**
5. Cliquer sur *Interface 2*
6. Cocher *activer l'interface réseau*.
7. Dans **mode d'accès réseau**, sélectionner **Réseau privé hôte**.
8. Dans **nom**, sélectionner **vboxnet0**

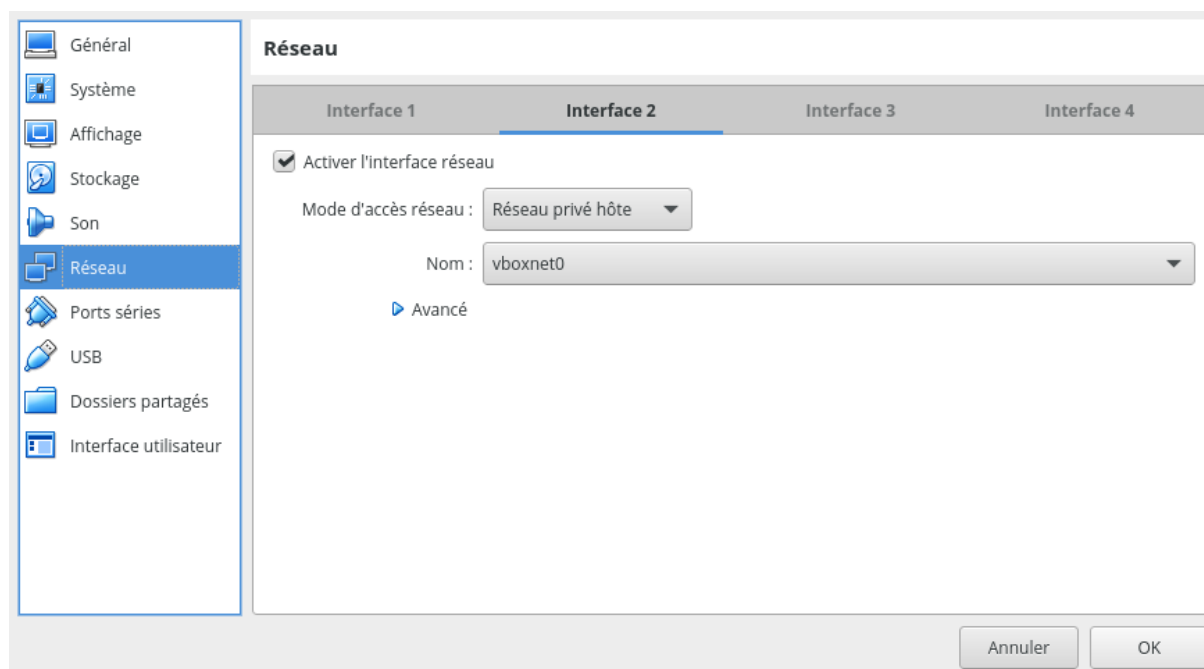


FIG. 2: configuration réseau de Kali

Normalement, nous devrions avoir maintenant deux **interfaces réseau configurées** :

- Une **interface 1** en **NAT**
- Une **interface 2** en **réseau privé hôte**.

Configurer les interfaces de Metasploitable

Faire la même chose sur **Metasploitable**.

Configurer l'interface réseau 1, et définir le **réseau privé hôte** *vboxnet0*. (Il n'est pas nécessaire de garder une interface en NAT pour accéder à Internet.)

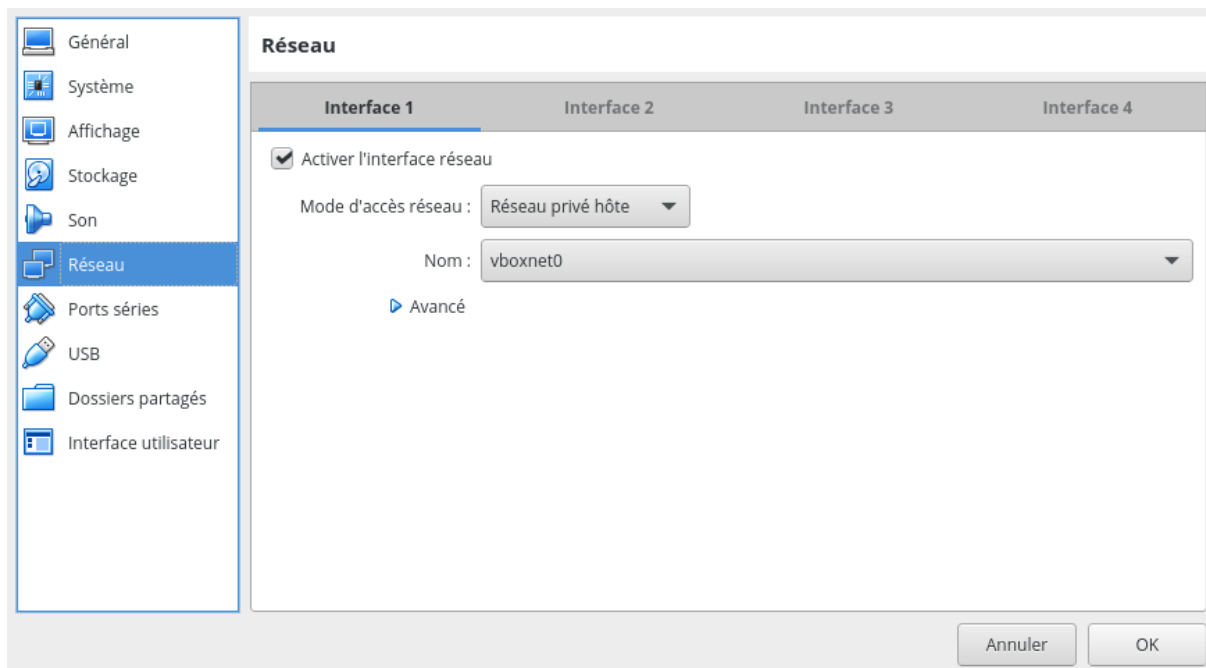


FIG. 3: configuration réseau de Metasploitable

Nous devrions maintenant avoir une seule interface :

- **Interface 1 en réseau privé hôte**

Commandes réseau sous Linux

Ifconfig

Pour connaître son adresse **IP**, on peut utiliser la commande **ifconfig**. Sous Windows, la commande est *ipconfig*.

Sur votre machine, la sortie de la commande devrait ressembler à ceci :

```
1 ifconfig
2 eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
4     inet6 fe80::a00:27ff:fee8:d8bf prefixlen 64 scopeid 0x20
5     ether 08:00:27:e8:d8:bf txqueuelen 1000 (Ethernet)
6     RX packets 29 bytes 3325 (3.2 KiB)
7     RX errors 0 dropped 0 overruns 0 frame 0
8     TX packets 52 bytes 4655 (4.5 KiB)
9     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
10
11 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
12     inet 192.168.56.110 netmask 255.255.255.0 broadcast
13         192.168.56.255
14     inet6 fe80::a00:27ff:fe3b:eec1 prefixlen 64 scopeid 0x20
15     ether 08:00:27:3b:ee:c1 txqueuelen 1000 (Ethernet)
16     RX packets 44 bytes 6359 (6.2 KiB)
17     RX errors 0 dropped 0 overruns 0 frame 0
18     TX packets 59 bytes 8642 (8.4 KiB)
19     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
20
21 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
22     inet 127.0.0.1 netmask 255.0.0.0
23     inet6 ::1 prefixlen 128 scopeid 0x10<host>
24     loop txqueuelen 1000 (Boucle locale)
25     RX packets 16 bytes 796 (796.0 B)
26     RX errors 0 dropped 0 overruns 0 frame 0
27     TX packets 16 bytes 796 (796.0 B)
28     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nous avons deux interfaces réseau ici :

- **eth0**

- Elle correspond à l'interface 1 dans Virtual Box
- Nous l'avons configuré en **NAT**
- Elle a pour adresse **IPv4** 10.0.2.15
- Elle possède une adresse **IPv6** fe80 ::a00 :27ff :fee8 :d8bf
- Son **adresse matérielle** est 08 :00 :27 :e8 :d8 :bf (paramètre ether)
- C'est avec elle que nous communiquons à Internet et notre réseau local

- **eth1**

- Elle correspond à l'interface 2 dans Virtual Box
- Nous l'avons configuré en **réseau privé hôte**
- Elle a pour adresse **IPv4** 192.168.56.110
- Elle possède une adresse **IPv6** fe80 ::a00 :27ff :fe3b :eec1
- Son **adresse matérielle** est 08 :00 :27 :3b :ee :c1 (paramètre ether)
- Elle communique uniquement avec les machines de vboxnet0 (ici Metasploitable)

- **lo**

- Il s'agit de la **boucle locale**
- Elle est accessible uniquement à notre machine
- Elle a pour adresse **IPv4** 127.0.0.1
- Elle possède une adresse **IPv6** ::1

Dhclient

Si une de vos interfaces n'a **pas d'adresse IP**, il est possible d'en demander une au serveur DHCP avec la commande **dhclient**.

Exemple avec eth1 :

```
1 sudo dhclient eth1
```

Table de routage

Il est possible de regarder votre table de routage IP avec la commande **route -n**.

La commande *route* permet également de modifier cette table.

```
1 $ route -n
2 Table de routage IP du noyau
3 Destination      Passerelle        Genmask           Indic  Iface
4 0.0.0.0           10.0.2.2          0.0.0.0           UG     eth0
5 10.0.2.0          0.0.0.0           255.255.255.0     U      eth0
6 192.168.56.0      0.0.0.0           255.255.255.0     U      eth1
```

La sortie de la commande se lit de la façon suivante :

Pour toutes les IP entre 192.168.56.1 et 192.168.56.255, envoyer les paquets sur l'interface eth1.

Pour toutes les IP entre 10.0.2.1 et 10.0.2.255, envoyer les paquets sur l'interface eth0.

Pour toutes les autres IP, transmettre les paquets à l'IP 10.0.2.2 (passerelle ou gateway) sur l'interface eth0.

Lister les connexions

Il est possible de lister les connexions avec la commande **netstat**.

On va généralement chercher à lister uniquement les ports en écoute avec **netstat -ltupn**.

```
1 $ netstat -ltupn
2
3 Connexions Internet actives (seulement serveurs)
4 Proto  Adresse locale  Adresse distante  Etat  PID/Program name
5 tcp    0.0.0.0:22      0.0.0.0:*        LISTEN  817/sshd
6 tcp6   :::22          :::*             LISTEN  817/sshd
7 udp    0.0.0.0:68      0.0.0.0:*        2627/dhclient
```

Ici nous avons le démon (programme qui s'exécute en arrière-plan) ssh sur port 22. Et un client DHCP.

-l : lister seulement les ports **en écoute**

-t : lister les ports **TCP**

-u : lister les ports **UDP**

-n : donner le **numéro** de port, plutôt que le service associé

-p : affiche les **processus** qui utilise le port (nécessite les droits admin)

Le fichier /etc/hosts

Le fichier */etc/hosts* permet d'associer un **nom à une IP** (de la même manière que le protocole DNS).

Par défaut il contient les lignes suivantes :

```
1 127.0.0.1      localhost
2 127.0.1.1      kali
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1           localhost ip6-localhost ip6-loopback
6 ff02::1      ip6-allnodes
7 ff02::2      ip6-allrouters
```

L'IP et le nom sont séparés par un caractère **Tab**.

Faire la commande **ping localhost** est équivalent à **ping 127.0.0.1**.

Ce fichier est très pratique pour ne pas avoir à retenir l'IP de machine qui n'ont pas de nom DNS.

Exercice : Ajouter une entrée pour la machine *Metasploitable 2* au fichier */etc/hosts* de kali.

Utilitaires réseau

Pour cette partie nous allons avoir besoin d'une seconde machine. **Démarrez Metasploitable 2**. Les identifiants sont **msfadmin/msfadmin**. Attention, le clavier est en **qwerty**. Une fois connectés, trouvez l'IP de la machine avec `ifconfig`.

Ping

Pour voir si une machine est accessible on peut utiliser la commande **ping**.

```
1 $ ping eff.org
2 PING eff.org (173.239.79.196) 56(84) bytes of data.
3 64 bytes from vm1.eff.org (173.239.79.196): icmp_seq=1 ttl=47 time=144
  ms
4 64 bytes from vm1.eff.org (173.239.79.196): icmp_seq=2 ttl=47 time=144
  ms
```

/!\ Windows (hors version serveur) est configuré par défaut pour ne pas répondre à ces requêtes.

SSH

Le protocole **SSH** permet d'obtenir un shell sur une machine distante. La syntaxe est `ssh utilisateur@machine`

```
1 root@kali:~$ ssh msfadmin@192.168.56.102
2 msfadmin@192.168.56.102's password:
3
4 msfadmin@metasploitable:~$ whoami
5 msfadmin
```

Pour une première connexion à la machine, `ssh` demande d'authentifier la machine avec une empreinte RSA. Une fois validé, vous pouvez vous authentifier avec un mot de passe.

Pour fermer la connexion, utiliser **Ctrl + D** ou la commande `exit`.

Netcat

Netcat est un peu le couteau suisse du protocole TCP/IP.

La syntaxe est la suivante : `nc ip port`.

On peut par exemple récupérer la bannière (identifiant de version) de SSH sur Metasploitable avec la commande suivante.

```
1 $ nc 192.168.56.101 22
2 SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
3
4 Protocol mismatch.
5 ^C
```

Il est possible de créer une connexion en écoute avec l'option **-l**. Par exemple pour écouter sur le **port 4444**.

```
1 $ nc -lvp 4444
2 Listening on [0.0.0.0] (family 0, port 4444)
```

Exercice : Ouvrez un port en écoute avec netcat, et connectez-vous dessus.