TP 19 decembre 2024

ÉCOLE EUROPÉENNE DE CYBER SÉCURITÉ

TP Systèmes d'Exploitation

L'ensemble des commandes tapées en mode CLI et leurs résultats sont requis. Un document au format texte (réalisé par exemple sous gedit) reprenant les numéros des questions, les commandes et leurs résultats ainsi qu'un document papier pour les représentations graphiques et vos commentaires constitueront les éléments à remettre à l'issue du TP.

Note : Il sera tenu compte de la présentation
JUSTIFIER VOS RÉPONSES

L'objectif du TP est de se familiariser avec quelques commandes essentielles pour gérer et tester des configurations réseaux simples sous Linux.

0/ Prérequis : Machine physique sous Ubuntu 22.04, possédant une interface Ethernet, accès Internet pour installation initiale des packages.
Passer **root** dans un terminal et installer les packages suivants via la commande :

```
apt install -y ethtool wireshark nmap nc openssh-server
```

et ettercap

Récupérer éventuellement les RFCs 1918 et 3927 si vous ne les connaissez pas.

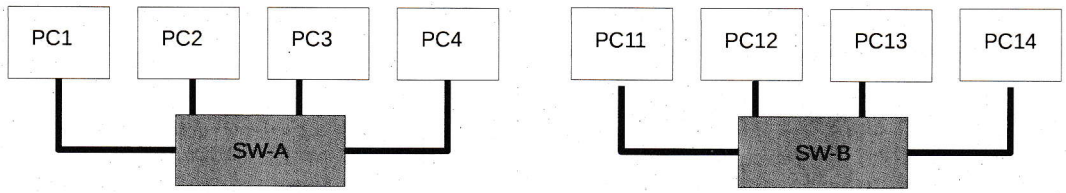
```
Wget https://www.ietf.org/rfc/rfc1918.txt
wget https://www.ietf.org/rfc/rfc3927.txt
```

Désactiver le wifi,
Stopper les services « réseaux » via la commande :

```
systemctl mask avahi-daemon ; systemctl stop ufw ; systemctl stop NetworkManager ;
nmcli radio wifi off
```

1/ Niveaux 1 et 2

1.1 Créer des « îlots » de connectivité en raccordant sur un switch (ou un hub) les machines via des câbles RJ45. Veillez à conserver, a minima, une prise RJ45 disponible sur l'équipement réseau.



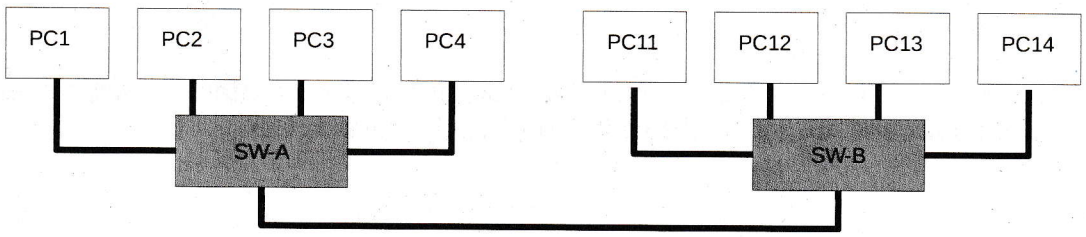
1.2 Rappeler très brièvement les différences entre un switch et un hub
1.3 Les machines d'un îlot peuvent-elles communiquer entre elles ?
1.4 Comment le vérifier ? Proposer des commandes.
La commande **ip link** liste les interfaces réseaux, repérer l'interface Ethernet, des machines, notez son nom et son adresse Ethernet.
1.5 La commande **ethtool** permet de vérifier/modifier de nombreux paramètres d'une interface Ethernet. Appliquer cette commande à votre interface et préciser les paramètres que vous jugez pertinents.
1.6 Proposer un plan d'adressage **IPv4** utilisant les plages du rfc1918 pour chaque îlot en utilisant le masque de sous-réseau (netmask) le plus adapté pour ne pas gaspiller trop d'adresses IP. L'ensemble des îlots utilisera des adresses issues de 2 blocs d'adresses proposés par le rfc1918. Une représentation graphique est requise.
1.7 Appliquer sur chaque machine le plan d'adressage défini précédemment via la commande ip :
ip addr add <ip_address /netmask> **dev** interface

Remarque : ping v6 du lien local, ne pas oublier la mention de l'interface, -I <interface> ou ' %<interface>'

1.8 Vérifier via commande **ping** que les machines d'un même îlot peuvent se joindre.
1.9 Que se passe t-il si 2 machines ont la même adresse IP ? Modifier la configuration réseau d'une machine en reprenant la même adresse IP. Expliquer en vous aidant de l'outil d'analyse de trames **wireshark**.
1.10 Que se passe t-il si 2 machines ont la même adresse Ethernet ? Le changement d'adresse MAC s'effectue par commande i**p link set** <interface> **address** <MAC address>
Est ce vraiment gênant si 2 machines dans le monde ont la même adresse MAC ? Justifier.
Remarque : La commande **ethtool -p** permet d'afficher l'adresse originelle permanente.
1.11 Vérifier le cache arp par la commande **ip neigh show dev** <interface>
1.12 Vider les caches arp de chaque machine de l'ilot (**ip neigh flush dev** <interface> puis observer un échange complet « ping » avec l'outil wireshark
1.13 Etablir le 'Sequence Diagram' des trames échangées lors d'un ping entre 2 machines d'un même ilot
1.14 Peux t-on utiliser le nom des machines au lieu de l'adresse IP ? Pourquoi ?
Le fichier **/etc/hosts** permet de déclarer de façon statique le nom avec l'adresse IP.
Modifier ce fichier sur une machine en ajoutant le nom et l'adresse IP des autres machines de l'îlot
1.15 Réaliser un ping avec le nom de machine.

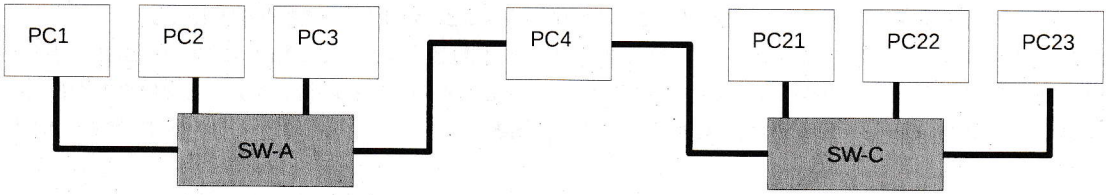
Remarque : La suite du TP se focalisera sur IPv4 mais les principes sont les mêmes en IPv6.

2/ Interconnexion niveau 2
On se propose de raccorder 2 îlots d'un même bloc d'adresses. Utiliser un nouveau câble pour relier deux (ou trois) équipements réseaux.



2.1 Les machines d'îlots différents peuvent-elles se joindre ? Justifier
2.2 Quelles sont modifications à apporter afin que les machines du 'super ilot' puissent communiquer.
2.3 Réaliser ces modifications, ainsi qu'une nouvelle représentation graphique du 'super ilot »
2.4 Vérifier la connectivité par quelques commandes ping.

3/ Interconnexion niveau 3
L'objectif est désormais de faire communiquer les 2 super îlots (« fenêtre » et « mur ») entre eux, chacun ayant une plage d'adresses IP différente.
3.1 Que faut il définir ? Pourquoi ?



3.2 Un adaptateur (USB/Ethernet) permet d'ajouter une interface Ethernet à la machine qui sera désignée comme routeur. Configurer cette nouvelle interface.
3.3 Que faut-il ajouter sur les machines (autres que le routeur) ?

3. Modifier les configurations de 2 machines de super îlots différents afin qu'elles puissent communiquer entre elles (nommées A et B dans la suite de l'énoncé).
L'ajout d'une route s'effectue via la commande **ip route add** <ip/netmask> **via** <ip_gateway>
3.5 Tester avec la commande **ip route get** pour voir comment les paquets seront dirigés.
3.6 Tester la connectivité avec ping et observer les trames avec wireshark
3.7 Que constater vous ?

Par défaut une machine linux (n'est pas un routeur) ne fait pas d'IP forwarding
La commande **sysctl net.ipv4.ip_forward** ou **cat /proc/sys/net/ipv4/ip_forward** affiche 0 si la fonctionnalité ip forwarding est désactivée
3.8 Pour activer l'IP forwarding il suffit de positionner la valeur à 1 via la commande
`sysctl -w net.ipv4.ip_forward=1` ou `echo 1 > /proc/sys/net/ipv4/ip_forward`
3.9 Réaliser (avec soin !) le sequence diagram complet de l'échange ping entre 2 machines de super îlots différents, après vidage des caches arp des machines impliquées en vous aidant de wireshark et en mentionnant sur la figure toutes les valeurs pertinentes.
3.10 sysctl permet de modifier d'autres paramètres noyau réseau. Pour voir les paramètres kernel relatif au réseau **sysctl net**, **sysctl net.ipv4** pour restreindre à ipv4
Choisir 2 paramètres, expliquer leur signification et proposer une méthode de test.

Remarque : Il faut éditer le fichier /etc/sysctl.conf pour rendre les changement permanents. La commande sysctl -p permet d'appliquer les modifications.

L'outil **ping** permet de nombreux tests. On se propose d'illustrer la fragmentation
3.11 Réduire la MTU d'une des interfaces du routeur par la commande
ip link set <interface> **mtu** <valeur>dev ip link set dev <interface> mtu <valeur>
3.12 Depuis la machine A, établir un ping avec la machine B en utilisant les options -M et -s avec des valeurs adéquates et expliquer les réponses obtenues en vous aidant de wireshark.

3.13 L'outil **tracepath** est également très pratique. Expliquer son fonctionnement en réalisant un tracepath de A vers B en vous aidant de wireshark

4/ Communication niveau 4
La communication haut niveau (au sens do modèle OSI) entre machines s'effectue via des ports.
4.1 Vérifier par la commande **ss -ntualp** quels sont les ports ouverts sur votre machine.
L'outil **nc** (netcat) permet de rapidement réaliser des tests TCP et UDP.
4.2 Créer sur machine un serveur à l'écoute sur le port 9000 (si inutilisé) via la commande **nc -l 9000.**
4.3 Depuis une autre machine, créer un client ouvrant une connexion sur le port 9000 via la commande **nc <ip_serveur> 9000**.
4.4 Échanger quelques données en observant les trames avec wireshark.
4.5 Réaliser le sequence diagram de l'échange entre le client et le serveur.
4.6 Créer un fichier sur le client et le transmettre au serveur par les commandes suivantes, en étudiant les trames.
serveur : **nc -l 9001 > fichier.txt**
client : **cat fichier.txt | nc <ip_serveur> 9001**

4.7 La commande **nmap** permet de scanner les ports d'un machine distante.
Depuis une machine, réaliser un scan de port sur une machine distante via la commande **nmap**.

4.8 nmap permet aussi de scanner tout un réseau !
nmap -sn <ip_net/mask> permet de détecter les machines up.
nmap -sS -F <ip_net/mask> donne rapidement les principaux ports ouverts sur un réseau

5/ Niveau 4+

La commande **ssh** permet d'établir une connexion sécurisée sur une machine distante

5.1 Etablir une connexion sur une machine distante via la commande **ssh login@ip,** puis taper les commandes **hostname ; ip a** pour vérifier ainsi que la connexion est établie. Observer les trames avec wireshark et vérifier que la communication est chiffrée.

5.2 Déconnecter via la commande **exit** (ou CTRL-D) puis établir une nouvelle session ssh avec l'option -X, et lancer la commande **xclock**, l'heure affichée correspond à quelle machine ?

5.3 Rétablir le service avahi par la commande **systemtcl unmask avahi-daemon**

avahi permet la résolution de noms dans les petits réseaux (sans usage de DNS).

En suffixant le nom de la machine par .local, avahi va rechercher sur le réseau l'adresse IP correspondante.

5.4 Réaliser une commande ping <nom_machine_de_votre_ilot>.local et observer les traces sous wireshark.

5.5 La commande **avahi-resolve -a** permet de connaître le nom connaissant l'IP. Tester.

5.6 Est ce que cela fonctionne aussi pour les machines de l'autre super-ilot ? Pourquoi ?

Remarque : avahi permet aussi l'annonce et la découverte de services

6/ trafic control

tc est une commande très puissante avec une syntaxe quelque peu complexe. La suite de l'exercice est une illustration très modeste de ses possibilités. Elle permet notamment de simuler des caractéristiques de réseaux longues distances (WAN).

Ajout d'un délai sur une liaison

6.1 Réaliser un ping vers une machine et observer le rtt affiché

ping <ip_machine> **-c 10**

6.2 La commande suivante va appliquer un délai de 200 ms avant de transmettre les paquets

tc qdisc add dev <interface> **root netem delay 200ms**

6.3 Réaliser de nouveau le ping et observer le rtt

6.4 Supprimer le délai par la commande

tc qdisc del dev <interface> **root netem**

Ajout de perte sur une liaison

6.5 Réaliser un ping vers une machine et observer le % de perte affiché

ping <ip_machine> **-c 10**

6.6 La commande suivante va appliquer des pertes de 20 % sur les paquets à transmettre

tc qdisc add dev <interface> **root netem loss 20%**

6.7 Réaliser de nouveau le ping et observer les pertes

6.8 Supprimer les pertes par la commande

tc qdisc del dev <interface> **root netem**

7/ Rétablissement du Wifi

7.1 Réactiver le wifi et connecter la machine au SSID de l'école

7.2 Quelle est le nom et l'adresse ip de l'interface wireless ?

7.3 Comment a été obtenue cette adresse ?

7.4 Quels autres paramètres ont été obtenus ?

7.5 Observer le fichier **/var/lib/dhcp/dhclient.leases**