SCR.2.1 TP 13 ⊥ :

Les commandes `ip`, `nc/ncat/netcat`, `ss`, `tracepath/traceroute`

Les utilitaires `tcpdump`, `tshark/wireshark`

— L'utilitaire `tcpdump` permet de visualiser le trafic sur le réseau. Il donne une description du contenu des paquets sur une interface réseau. Cet utilitaire est un standard de toutes les installation Linux. Son utilisation est, par défaut, réservé à l'utilisateur root.

— L'application graphique `wireshark` permet aussi de décrire le trafic sur un réseau ; `tshark` en est la version ligne de commande.

↪ Aussi bien pour `tcpdump` que pour `tshark`/`wireshark`, les paquets dont on veut obtenir une description peuvent être filtrés par des expressions. On peut faire `man pcap-filter` pour avoir la manière de construire ces expressions.

↪ Aussi bien pour `tcpdump` que pour `tshark`, si on veut avoir le résultat, à la fois à l'écran et dans un fichier, on pipelinera avec la commande `tee <trace-file.txt>` (sans les `< >`, bien sûr.)

Filtre pour `tcpdump` et `wireshark`.

Il n'est pas pratique d'avoir les affichages polués par la description de paquets qui ne nous intéressent pas. Ainsi, avant de lancer la commande dont on veut analyser le traffic généré sur le réseau, commencer par "tailler" au mieux le filtre de `tcpdump`/`tshark`.

Pour cela, tester plusieurs lignes de commandes `tcpdump`/`tshark`, en réglant le filtre jusqu'à ce que `tcpdump`/`tshark` ne "parlent" que lorsqu'on aura soumis la commande dont on veut étudier le trafic généré. Par exemple, si on veut anlayser le traffic dû à un `ping` de A vers B, on ajoute dans le filtre l'expression `host <ip_A> and host <ip_B>` (sans les `< >`, bien sûr).

I. La commande `ip`. Elle permet de montrer (et de manipuler si on possède des privilèges suffisants) les interfaces réseau, la table de routage, le cache arp, etc. Consulter la page du manuel de la commande `ip` et réaliser les tâches suivantes uniquement à l'aide de la commande `ip` :

1. Afficher les informations sur toutes les interfaces réseau de sa machine.
2. Afficher les informations de l'interface *loopback* uniquement.
3. Afficher les informations du *niveau liaison de données* sur toutes les interfaces réseau de sa machine. Quelle est la valeur de la mtu pour chacune des interfaces ?
4. Afficher les informations du *niveau liaison de données* de l'interface principale de sa machine uniquement (celle qu'utilisent les autres machines pour communiquer avec.)
5. Afficher les entrées du cache arp.
6. Afficher les entrées du cache arp en donnant les voisins par leurs noms DNS au lieu des adresses ip. Quelle est l'adresse mac de `gatekeeper.arda`/`fireworm.iutsf.lan` ?
7. Afficher la table de routage utilisée par sa machine. Quelle est l'adresse ip de la passerelle de son segment de réseau d'attachement ?

Dans la suite, **on n'utilisera pas** les commandes `ifconfig, arp, netstat, route`.

II. Sur le cache arp.

1. L'option `-c` de `ping` permet de préciser le nombre de paquets `ECHO_REQUEST` au bout duquel on arrête le `ping`. Choisir l'adresse ipv4 d'une machine allumée et qui n'est pas dans le cache arp. Faire `ping` dessus. Afficher les entrées du cache arp juste après. Expliquer.
2. Préparer dans une fenêtre une ligne de commande `tcpdump` pour obtenir la description de l'activité sur le réseau engendrée par un `ping` vers une machine non listée dans le cache arp.
3. Par défaut, `tcpdump` n'affiche pas l'en-tête Ethernet. Quelle option faut-il passer si on veut la description de l'en-tête Ethernet aussi ?
4. Maintenant, dans une autre fenêtre, donner la commande `ping` vers une machine qui n'est pas dans le cache arp. Expliquer ce que montre `tcpdump`.
5. Dans `tshark`, par quelle option présenter à la ligne de commande, l'expression qui constitue le filtre de capture ?
6. Refaire 2 et 4, en utilisant `tshark` au lieu de `tcpdump`.

III. Préparer une ligne de commande `tcpdump` pour répondre aux questions suivantes. On va recueillir le résultat de `tcpdump` également dans un fichier en pipelinant avec la commande `tee`.

1. Faire un `ping` sur l'interface *loopback* avec **un seul** `ICMP ECHO_REQUEST` et en envoyant 9216 octets de données. Combien de paquets ont été envoyés ? Quelle interface précise-t-on à `tcpdump` par l'option `-i` ?
2. Faire un `ping` avec les mêmes arguments mais avec une autre machine comme destination. Combien de paquets ont été envoyés ? Y a-t-il une différence avec la question précédente ? Expliquer.