

Réseau

TP - Étude des couches protocolaires

Présentation : utilisation de netkit

Netkit¹ est un projet Open-Source permettant de simuler des réseaux informatiques. Pour cela il utilise les possibilités UML² de Linux.

Netkit propose principalement les commandes suivantes :

vstart qui permet de lancer une machine virtuelle

— Syntaxe d'utilisation : `vstart [options] nom de la machine`

— Options les plus utilisées :

— `--ethN=nom` du HUB connecté, une machine peut posséder plusieurs interfaces réseau

— `-M` taille de la mémoire en Mo

vlist pour avoir la liste des machines virtuelles lancées

vhalt pour arrêter une machine virtuelle depuis la machine réelle (on peut aussi utiliser la commande `halt` depuis la machine virtuelle)

À chaque machine virtuelle est associée une image disque. Cette image est composée de deux fichiers dont le nom est le nom de la machine et les extensions sont `.disk` et `.log`. **Sachant que ces fichiers sont de grande taille, il faut utiliser netkit depuis le répertoire `/tmp`.**

Enfin, depuis une machine virtuelle nous pouvons à notre *home* accéder au répertoire de la machine physique. Pour ce faire il suffit d'utiliser le répertoire `/hosthome`.

1 Recherche documentaire

Pendant ce TP nous allons devoir utiliser les programmes suivants :

— `ifconfig` : que fait ce programme ? quel est son utilisation de base ?

— `ping` : à quoi sert ce programme ? Comment l'utilise-t-on ?

— `tcpdump` : que fait cette commande ? Comment rediriger le flux d'information vers un fichier ?

— `wireshark` : à quoi sert ce programme ? Pourquoi ne peut on pas l'utiliser depuis les machines virtuelles ? Comment va-t-on l'utiliser ?

— `apache` : à quoi sert ce programme ?

— `lynx` : à quoi sert ce programme ?

2 Création du réseau

À l'aide du logiciel Netkit, créez un LAN dont l'adresse est `192.168.0.0/24` composé des éléments suivants³ :

— un *host ml* dont l'adresse IP sera `192.168.0.1` ;

1. <http://www.netkit.org>

2. <http://user-mode-linux.sourceforge.net/>

3. Chaque machine aura une carte ethernet et une mémoire vive de 64Mo

- un *host* *m2* avec 64Mo de RAM dont l'adresse IP sera 192.168.0.2 ;
- un *hub* qui reliera *m1* et *m2*.

Après avoir configuré les adresses IP de chaque *host*, testez le fonctionnement de votre réseau à l'aide de la commande `ping`.

3 Couches protocolaires

L'objectif de cette partie est d'identifier les différentes couches protocolaires utilisées lors d'une navigation *web*. Pour cela nous allons :

- lancer un serveur web (`apache`) sur *m2* ;
- lancer un navigateur web (`lynx`) sur *m1* ;
- lancer `tcpdump` sur *m2* pour capturer le trafic (à enregistrer dans un fichier dans le répertoire `/hosthome` ;
- afficher à l'aide de `lynx` (sur *m1*) la page par défaut du serveur web (sur *m2*) ;
- visualiser le fichier généré par `tcpdump` avec `wireshark`.

Voici la suite des actions à réaliser :

- démarrez sur *m2* le serveur web `apache` (`service apache2 start`⁴).
- lancez sur *m2* le `tcpdump`. Commencer une capture réseau (interface `eth0`)
- lancer sur *m1* le navigateur `lynx`, et demander à obtenir la page d'accueil (commande `g`) ;
- après l'obtention de la page (affichage de « It works ! »), arrêtez la capture réseau et visualisez le contenu du fichier généré.

Questions

- Identifiez les échanges liés à la navigation web (protocole `http`). Combien y en a-t-il ? Pourquoi ?
- Quelles sont les couches protocolaires utilisées dans le premier échange ?
- Quelle commande HTTP est utilisée pour obtenir la page par défaut ?
- Dans quelle couche retrouve-t-on les adresses IP source et destination ?

4. N'hésitez pas à utiliser la touche `tab`