UNIVERSITÉ CAEN NORMANDIE

Les Ports et les Services 15.01.2024

La communication par socket☹

Je vous sers quoi sur la pointe des pieds ?

Auteur : Pascal Fougeray

Port #	Application Layer Protocol	Type	Description
20	FTP	TCP	File Transfer Protocol - data
21	FTP	TCP	File Transfer Protocol - control
22	SSH	TCP/UDP	Secure Shell for secure login
23	Telnet	TCP	Unencrypted login
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP/UDP	Domain Name Server
67/68	DHCP	UDP	Dynamic Host
80	HTTP	TCP	HyperText Transfer Protocol
123	NTP	UDP	Network Time Protocol
161,162	SNMP	TCP/UDP	Simple Network Management Protocol
389	LDAP	TCP/UDP	Lightweight Directory Authentication Protocol
443	HTTPS	TCP/UDP	HTTP with Secure Socket Layer

source : Je ne sais plus ;)

1 Introduction

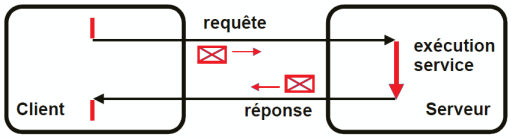
C'est la couche 4 et plus du modèle OSI.

Dans le domaine des réseaux on a les serveurs et les clients. Pour communiquer ils ont besoin de points de connexions que l'on nomme **socket**.

À chaque service délivré par un serveur correspond un numéro de ports.

2 Rappels

Quelques rappels sur la notion Client/Serveur et le modèle en couche ?



2.1 Serveur, Client et Service

Dans une application, les processus utilisés sont divisés en 2 catégories : un ou plusieurs processus serveurs et des processus clients. Ces différents processus échangent des messages : le client adresse à un serveur une requête par l'intermédiaire d'un message et le serveur transmet au client un message de réponse, après avoir satisfait la requête du client. On distingue ainsi :

— le **service** : C'est une tâche particulière dont on peut demander la réalisation ;

— Sous Linux, on peut connaître les services avec le N° de port associé en consultant le fichier **/etc/-services.**

Dont voici un extrait

```
msp             18/tcp                          # message send protocol
msp             18/udp
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
fsp             21/udp          fspd
ssh             22/tcp                          # SSH Remote Login Protocol
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
rlp             39/udp          resource        # resource location
nameserver      42/tcp          name            # IEN 116
whois           43/tcp          nicname
tacacs          49/tcp                          # Login Host Protocol (TACACS)
```

— le **serveur** : C'est une machine sur laquelle un service est réalisé ou un processus (thread) qui rend ce service ;
— le **client** : C'est une machine faisant appel à une machine serveur ou un processus sollicitant un processus serveur.

Question : Quel est le seul protocole où il n'y a qu'un seul client et plein de serveurs ?
Réponse : **SNMP** : ***Simple Network Managment Protocol***
C'est un protocole utilisé par les différents éléments actifs d'un réseau (Ordinateurs, Switchs, Routeurs etc..). Ils sont interrogés régulièrement pas un client (le manager) pour connaître leur état à chacun.
Nous verrons son fonctionnement dans un autre CM/TD/TP.

2.2 Les ports

Il y a 65536 ports différents, le codage sur 16 bits donne 2^{16} possibilités.
Tous les ports n'ont pas les mêmes caractéristiques.
À chaque N° de port peut correspondre un service différent, mais il y a beaucoup plus de N° de ports que de services.

— Le port **0** : Il n'est pas utilisable pour une application, c'est un "jocker" indiquant au système que c'est à lui de compléter automatiquement le numéro entre **49152** et **65535**.
— De 1 à 1023 : Pour utiliser cette zone il faut avoir les droits du root, sinon, à l'exécution le ***bind*** retourne une erreur. Les serveurs "classiques" (***ftp***, ***smtp***, ***telnet***, ***ssh***...) utilisent cette plage de N° de ports.
Vous verrez au TP Protocole ou l'on parle le **SMTP** avec **Telnet : telnet @Ip N°port**
— De **1024** à **49151** est la zone des services enregistrés par l'**IANA** [1] et qui fonctionnent avec des droits ordinaires.
— Port de **49152** à 65535 est la zone d'attribution automatique des ports, pour la partie cliente des connexions (si le protocole n'impose pas une valeur particulière) et pour les tests de serveurs locaux.

3 Les *sockets*

Mécanisme de communication permettant d'utiliser l'interface de transport TCP-UDP donc la couche 4 du modèle OSI et s'appuyant sur IP.
Ce mécanisme fut introduit dans Unix dans les années 80, c'est un standard aujourd'hui...
Un (ou une...) *socket* est un **point de contact** dans une communication.
Elle permet à 2 entités, généralement un client et un serveur de se retrouver en relation et de pouvoir communiquer.
Pour faire beaucoup plus simple, une *socket* est
— soit un téléphone
— soit une boîte aux lettres...

Une socket c'est une IP, un mode TCP ou UDP et un N° de port

L'IP c'est où l'on veut trouver le service
Le mode c'est comment on y va
Le N° de port c'est qu'elle pièce dans le lieu

On n'utilise pas le même protocole pour manger dans la belle salle à manger à Noël que pour aller prendre une douche ou aller aux WC ☺

1. **Internet Assigned Numbers Authority** : www.iana.org

Et des sockets il peut y en a plein sur une machine!!!
La preuve : la commande ***find -type s | wc -l*** sur la VM vous renverra quelque chose comme cela, idem sur le PC de la FAC que l'on nomme le Host.

```
root@debian95-Rx-Sys-Fougeray:/# find -type s
./tmp/.ICE-unix/1101
./tmp/ssh-wLgg2UM1XGrT/agent.1055
./tmp/.X11-unix/X1
./tmp/.X11-unix/X0
./var/lib/courier/sqwebmail.sock
./root/L3/TD_TP/socket/chapitre_14/local/socket_serveur
./root/L3/TD_TP/socket/local/socket_serveur
./run/openvswitch/ovs-vswitchd.723.ctl
./run/openvswitch/ovsdb-server.687.ctl
./run/openvswitch/db.sock
./run/docker/libnetwork/b917ad3e95e4fbc61658e58db48f52a7f4ec43112c39cd00c1eea1f7ae1d2e1a.sock
./run/docker/metrics.sock
./run/docker/containerd/docker-containerd.sock
./run/docker/containerd/docker-containerd-debug.sock
./run/mysqld/mysqld.sock
./run/uuidd/request
./run/docker.sock
./run/avahi-daemon/socket
./run/dbus/system_bus_socket
./run/cups/cups.sock
./run/user/0/bus
./run/user/0/gnupg/S.gpg-agent.ssh
./run/user/0/gnupg/S.gpg-agent.extra
./run/user/0/gnupg/S.gpg-agent.browser
./run/user/0/gnupg/S.gpg-agent
./run/user/0/systemd/private
./run/user/0/systemd/notify
./run/user/114/gnupg/S.gpg-agent
./run/user/114/gnupg/S.gpg-agent.ssh
./run/user/114/gnupg/S.gpg-agent.extra
./run/user/114/gnupg/S.gpg-agent.browser
./run/user/114/bus
./run/user/114/systemd/private
./run/user/114/systemd/notify
./run/systemd/fsck.progress
./run/systemd/journal/syslog
./run/systemd/journal/socket
./run/systemd/journal/stdout
./run/systemd/journal/dev-log
./run/systemd/private
./run/systemd/cgroups-agent
./run/systemd/notify
./run/systemd/inaccessible/sock
./run/udev/control
root@debian95-Rx-Sys-Fougeray:/#
```

Voyons ce que sont celles qui sont dans /root/L3/TD_TP/socket juste par curiosité!
Un ***ls -l /root/L3/TD_TP/socket/local/socket_serveur***
et un ***file /root/L3/TD_TP/socket/local/socket_serveur***
renvoient

```
./run/udev/control
root@debian95-Rx-Sys-Fougeray:/# find -type s | wc -l
44
root@debian95-Rx-Sys-Fougeray:/# ls -l /root/L3/TD_TP/socket/local/socket_serveur
srwxr-xr-x 1 root root 0 nov.   8 00:17 /root/L3/TD_TP/socket/local/socket_serveur
root@debian95-Rx-Sys-Fougeray:/# file /root/L3/TD_TP/socket/local/socket_serveur
/root/L3/TD_TP/socket/local/socket_serveur: socket
root@debian95-Rx-Sys-Fougeray:/#
```

S comme Socket

UN ou UNE SOCKET c'est donc un fichier sous Linux mais pas que ça!!!!

C'est une @IP, un type (TCP/UDP) et un numéro de port!

IP correspond à **Où**
Type correspond à **Comment**
Numéro de port correspond à **Quoi**

Si vous allez à la scolarité à cheval et que vous demandez une bière, pas certain que vous soyez bien reçu-e ☺

Ce n'est pas le bon serveur!

4 Les ports

Quand un serveur logiciel tourne sur un serveur physique, il ouvre un port afin que le client puisse s'y connecter.
Je ne vais pas vous énumérer tous les ports qui existent et encore moins les services correspondants.
Néanmoins il vous faut en connaître certains et savoir si les services tournent et si les ports sont ouverts.
Sous Linux il existe deux commandes qui permettent de savoir quel service donc quel port est ouvert.
Ceux sont les commandes **netstat** et **nmap**.

4.1 Netstat

Netstat (**statistiques réseau**) est un outil en ligne de commande qui affiche les connexions réseau entrantes et sortantes, les tables de routage et un certain nombre de statistiques d'interface réseau.

Elle s'utilise en interne sur le serveur!

Si vous **tapez** la commande ***netstat -help***, vous obtiendrez les consignes d'utilisation suivantes :

```
etudiant@debian-11-GNS3:~# netstat -help
usage: netstat [-vWeenNcCF] [<Af>] -r           netstat {-V|--version|-h|--help}
       netstat [-vWnNcaeol] [<Socket> ...]
       netstat { [-vWeenNac] -I[<Iface>] | [-veenNac] -i | [-cnNe] -M | -s [-6tuw] } [dela
        -r, --route              display routing table
```

```
-I, --interfaces=<Iface> display interface table for <Iface>
-i, --interfaces         display interface table
-g, --groups             display multicast group memberships
-s, --statistics         display networking statistics (like SNMP)
-M, --masquerade         display masqueraded connections
-v, --verbose            be verbose
-W, --wide               don't truncate IP addresses
-n, --numeric            don't resolve names
-N, --symbolic           resolve hardware names
-p, --programs           display PID/Program name for sockets
-c, --continuous         continuous listing
-l, --listening          display listening server sockets
-a, --all                display all sockets (default: connected)
```

Je n'ai pas tout mis ...
Les options les plus courantes sont
— **-l** pour listen (en écoute),
— **-n** (Numéro de port),
— **-p** (Permet de voir le PID du processus Daemon attaché à la socket),
— **-t** (TCP) ,
— **-u** (UDP).

Exemples : Si on recherche toutes les connexions établies en UDP.
netstat -natu | grep 'ESTABLISHED'

4.2 NMAP

La commande Nmap signifie "***Mappeur de réseau***". Elle permet la découverte de réseau et l'audit de sécurité.

Elle s'utilise en externe sur le serveur !

Son principe est fort simple : Elle envoie des paquets et analyse la réponse qu'elle obtient pour découvrir des hôtes et des services sur un réseau informatique. C'est l'un des scanners de port disponible les plus utilisé aujourd'hui pour aider à trouver les ports ouverts et à détecter les risques de sécurité sur un réseau.

ATTENTION : À utiliser avec modération sous peine d'être repéré et blacklisté ☺

apt install nmap

On peut **Scanner** une machine en donnant son nom ou son IP, mais on peut aussi scanner un réseau complet en donnant l'IP du Rx plus le masque!

Un Serveur : **nmap 192.168.0.0** ou un réseau **nmap 192.168.0.0/24**
Trois serveurs **nmap 192.168.0.1, 12, 16** ou 9 serveurs **nmap 192.168.0.1-9**
Pour voir des ports spécifiques : **nmap -p 22,80,443 192.168.0.1**
Format de sortie en XML : **nmap -oX scanResult.xml 192.168.0.1**
Analyse TCP SYN [-sS] : **nmap -sS 192.168.0.1**
Analyse UDP [-sU] : **nmap -sU 192.168.0.1**
Quelques exemples
193.55.120.26 c'est **www.unicaen.fr**, on voit que c'est un serveur **Web**

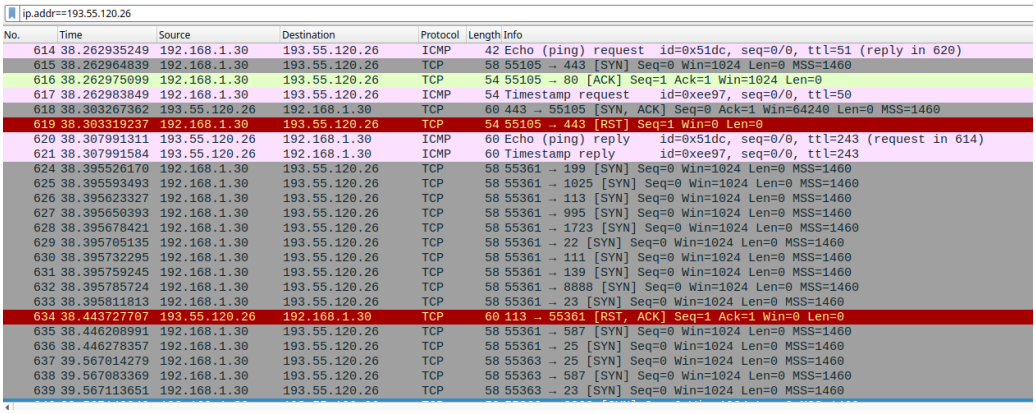
```
etudiant@debian-11-GNS3:~# nmap 193.55.120.26
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-06 10:02 CET
Nmap scan report for rp5.unicaen.fr (193.55.120.26)
Host is up (0.046s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE  SERVICE
80/tcp   open   http
113/tcp  closed ident
443/tcp  open   https
843/tcp  closed unknown
8443/tcp closed https-alt
```

193.55.120.31 c'est **smtp.unicaen.fr**, on voit que c'est un serveur Mail de type **SMTP**

```
etudiant@debian-11-GNS3:~# nmap 193.55.120.31
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-06 10:04 CET
Nmap scan report for smtp.unicaen.fr (193.55.120.31)
```

```
Host is up (0.051s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident
465/tcp open   smtps
```

Voici le résultat d'une capture wireshark après avoir lancé la commande **nmap 193.55.120.26**



On peut voir que la machine d'adresse IP 192.168.1.30 fait en premier un **ping** pour voir si la destination existe (Si elle ne répond pas que se passe-t'il ?).

Et ensuite elle fait des SYN sur plein de ports, ici 443, 80, 199, 1025, 22, 23, 25 etc ...

On voit que pour le port 443 la machine distante répond donc le port est ouvert !

5 Sur les routeurs Mikrotik

Les routeurs Mikrotik que nous allons utiliser dans les TP offrent des services. Pour les connaître, il suffit de taper la commande :

[admin@MikroTik] /ip service> /**ip service print** qui renvoie :

```
Flags: X - disabled, I - invalid
 #    NAME                 PORT ADDRESS
 0    telnet               23
 1    ftp                  21
 2    www                  80
 3    ssh                  22
 4 X  www-ssl              443
```

6 Les ports à connaître !

Voici les services et les ports que nous allons étudier cette année.

Port	Service	Définition	Port	Service	Définition
21	**ftp**	Transfert de fichiers	**443**	**https**	Web sécurisé
22	**ssh**	Shell distant sécurisé	**465**	**smtps**	Envoi Mail sécurisé
23	**telnet**	Shell distant			
25	**smtp**	Envoi Mail			
53	**dns**		**989**	**ftps**	Transfert de fichiers sécurisé
67	**bootps**				
68	**bootpc**		**992**	**telnets**	Shell distant sécurisé
80	**http**	Web	**993**	**imaps**	Lecture Mail sécurisé
110	**pop3**	Lecture Mail	**995**	**pops**	Lecture Mail sécurisé
123	**ntp**	Temps réseau			
143	**imap**	Lecture Mail	1664	toto ☺	
161	**snmp**	Management			
162	**snmp-trap**	Management	**2049**	**nfs**	Système de fichiers en réseau
514	**syslog**	Les logs	**3306**	**mysql**	Base de données

Si vous voulez apprendre les caractéristiques d'un port et d'un service je vous conseille ce site : https://www.speedguide.net

7 Conclusion

— **Les ports sont liés aux services et vice-versa.**
— Une (un) socket est un point de connexion
— Un N° de port correspond à un service. S'il est ouvert c'est que le service est disponible.
— Un service correspond à un processus **Daemon**.
— Le client et le serveur doivent avoir le même mode (TCP/UDP) et le même N° de port pour pouvoir communiquer !
— De nombreux services sont installés sur de nombreuses machines tournant sur internet par exemple les routeurs que nous utiliserons en TP et vous verrez qu'ils font tourner des serveurs Web , SSH, MySQL etc... en plus de leurs fonctions de bases.