processeur : rôle : calculer, traiter des infos

carte mère : lien entre tous les composants

carte graphique : calculer, traiter des infos graĥiques

ram : mémoire vive, temporaire..

carte réseau : bluetooth, wifi, fibre optique, ethernet..

FIREWALL

firewall Linux

configurer firewall

à quoi ca sert ?

https://www.youtube.com/watch?v=nZrPOsqXF8U

commande pour connaitre l'état du firewall : sudo ufw status

ufw étant le firewall

firewall : en francais pare-feu/ assure la sécurité du réseau/ peu aussi servir de filtre pour l'accès à un site..

Si on ne précise aucune règle au pare-feu, il bloquera toute entrée

Etablir une règle par default : default

exemple : sudo ufw default

Pour interdir tout ce qui vient : sudo ufw default deny incoming

Pour autoriser tout ce qui sort : sudo ufw default allow outgoing

cat /etc/services > permet de voir le port d'un élément

Ici on s'en sert pour autoriser l'accès à internet en regardant le port associer à http ---- c'est 80

La commande est : sudo ufw allow 80

On fait la même chose pour https

Ici c'est de la communication entrante, c'est à dire que ca concerne tout ce qui arrive

Pour activer le firewall : sudo ufw enable

Si on veut reset les règles qu'on a créée ou autre : sudo ufw reset >> on repart donc de 0

sudo ufw status numbered >> permet de numéroter les règles

> sert quand il faut par exemple supprimer une règle : sudo ufw delete "numéro de la règle"

on peut par exemple autoriser l'accès à http depuis une certaine adresse ip : sudo ufw allow http/80 from "ip"

les mots-clés :

deny : refuser

allow : accepter

delete : supprimer

anable : activer

disable : désactiver

reset : rénitialiser

SERVEUR NGINX

C'est un logiciel libre ou serveur Web (http) ainsi qu'un proxy inverse. (proxy inverse : type de serveur habituellement placé au front de serveurs web/ contrairement au serveur proxy qui permet à un utilisateur d'accéder au réseau internet, le proxy inverse permet à un utilisateur d'accéder à des serveurs internes).

configurer le démarrage automatique du serveur nginx : sudo systemctl enable nginx

pour tester si l'on est bien connecté sur notre serveur nginx en local : tapper dans le navigateur 127.0.0.1

la page qui s'affiche dans le navigateur correspond à un fichier stocker dans /var/www/html/

cette page est enfaite associée au site par defaut de nginx que l'on peut modifier comme ca : sudo vim ou nano /etc/nginx/sites-enabled/default

pour nginx on a un dossier de configuration global qui se trouve dans : etc/nginx/nginx.conf

Comment créer notre premier site avec nginx :

1) Créer un dossier dans /var/www/-R

2) Déterminer comme propriétaire du dossier l'utilisateur d'nginx : sudo chown -R www-data:www-data /var/www/ugofirstsite/

on a indiqué à la fin le chemin vers la racine de notre site

3) On applique les droits : sudo chmod 755 /var/www/ugofirstsite/site

4) On créer une page index html à la racine et on y intègre un léger code en html : sudo vim /var/www/ugofirstsite/index.html

5) créer le fichier de configuration dans /etc/nginx/sites-available/ugofirstsite

```
server {                                --- déclarer notre site au sein d'un bloc serveur


   listen 80                            --- accessible sur le port 80 (ipv4)
   listen [::]:80;                       --- IPV 6


   root /var/www/ugofirstsite                  --- racine du site


   index index.html;                        --- directive index
   server_name ugofirstsite www.ugofirstsite.net;       --- nom de domaine associé à ce site, ici ugofirstsite et www.ugofirstsite


   location / {                          --- s'applique à l'ensemble des pages de notre site grâce au /
      try_files $uri $uri/ =404;               --- vérifier l'existence des dossiers ou fichiers passés en paramètre de l'url
   }
}
```

6) Créer un lien symbolique : sudo ln -s /etc/nginx/sites-available/ugofirstsite /etc/nginx/sites-enabled/ugofirstsite