



Compte rendu du projet Administration réseau

Serveur WEB – Hicham CHAHBOUNE

Objectifs du travail:

Ci-suit les différentes étapes à faire :

1. Se connecter à la machine.
2. Créer des comptes utilisateurs pour chaque membre de votre groupe, vous et moi.
 - a. Mon user devra s'appeler "alexis" et vous définirez un mot de passe qui sera indiqué dans votre rapport personnel
3. Donner à votre utilisateur et au mien les droits administrateur en utilisant les groupes.
4. Permettre aux utilisateurs de se connecter à la machine en SSH en utilisant le mot de passe de l'utilisateur et la clef SSH que je vous ai fourni (les deux doivent fonctionner ensemble)
5. Installer et configurer fail2ban pour limiter le nombre de tentative de connexions SSH.
 - a. Exemple de configuration :
FindTime : 3 min | BanTime : 3 min | MaxFailure : 2
6. Configurer un serveur Web(**Apache**)

Introduction:

Un serveur web est un fournisseur de service http. Un serveur http est un serveur hébergeant un ou plusieurs sites Web.

Un serveur web est donc un « simple » logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (**par défaut le port 80**) (**HTTPS sur le port 443**) , et de fournir une réponse avec ce même protocole.

Quelques serveurs HTTP :

- *Apache HTTP Server*
- *Apache Tomcat*
- *BusyBox*
- *lighttpd*

Dans la suite, on va travailler avec Apache HTTP Server pour configurer un serveur web.

Pourquoi Apache est le serveur web le plus célèbre:



En 1998, Le serveur web le plus connu qui a inventé l'idée qu'un seul serveur peut héberger plusieurs sites est Apache. donc les fournisseurs d'hébergement Web ont fixé leurs choix sur Apache pour minimiser les ressources, l'énergie... .

La communauté d'Apache s'est agrandie, grâce à sa plusieurs modules sont développés (module de l'authentification ,module de compression des données...) et aussi il peut supporter tous les langages puissantes aujourd'hui, tout simplement Apache devient le serveur Web le plus pris en charge au monde

1- La connexion avec la machine distante:

Pour établir une connexion avec la machine distante on va juste exécuter la commande suivante:

```
$ ssh -i [chemin_de_la_clé_privé] [utilisateur]@[clé_public_de_la_machine]  
ssh -i C:\Users\pc\Desktop\clé.ppk hicham@13.38.251.125
```

2 - La création des comptes utilisateurs:

La première commande à exécuter est \$apt update afin de maintenir à jour votre liste de paquets disponibles.

1. Création des utilisateurs:

La commande `useradd` avec ces options permet de créer un utilisateur et avec son répertoire personnel.

```
admin@ip-172-31-46-110:/$ sudo useradd -m -d /home/alexis alexis  
admin@ip-172-31-46-110:/$ sudo useradd -m -d /home/hicham hicham  
admin@ip-172-31-46-110:/$ sudo useradd -m -d /home/hamza hamza  
admin@ip-172-31-46-110:/$ sudo useradd -m -d /home/khaoula khaoula  
admin@ip-172-31-46-110:/$ sudo useradd -m -d /home/oussama oussama  
admin@ip-172-31-46-110:/$ sudo useradd -m -d /home/youssef youssef  
admin@ip-172-31-46-110:/$
```

2. Affecter un mot de passe à l'utilisateur "alexis":

`pw=alexis`

```
admin@ip-172-31-46-110:~$ sudo passwd alexis  
New password:  
Retype new password:  
passwd: password updated successfully  
admin@ip-172-31-46-110:~$
```

3. Créer un group admins:

```
admin@ip-172-31-46-110:~$ sudo groupadd admins  
admin@ip-172-31-46-110:~$
```

4. Pour mieux gérer les accès et les droits sur la machine distante, on utilise deux groupes (admins et member):

La commande `usermod -aG` permet d'ajouter un utilisateur à un groupe.

```
admin@ip-172-31-46-110:~$ sudo usermod -aG admins alexis  
admin@ip-172-31-46-110:~$ sudo usermod -aG admins hicham
```

```
root@ip-172-31-46-110:/# sudo usermod -aG members hamza  
root@ip-172-31-46-110:/# sudo usermod -aG members khaoula  
root@ip-172-31-46-110:/# sudo usermod -aG members oussama  
root@ip-172-31-46-110:/# sudo usermod -aG members youssef  
root@ip-172-31-46-110:/#
```

- Pour vérifier que les groupes sont bien créés:

`$sudo cat /etc/group`

```
admins:x:1007:alexis,hicham
alexis:x:1001:
hicham:x:1002:
hamza:x:1003:
khaoula:x:1004:
oussama:x:1005:
youssef:x:1006:
ssl-cert:x:114:
members:x:1008:hamza,khaoula,oussama,youssef
```

- Afficher les information d'un utilisateur:

`$id [user]`

```
admin@ip-172-31-46-110:~$ id alexis
uid=1006(alexis) gid=1006(alexis) groups=1006(alexis),1007(admins)
admin@ip-172-31-46-110:~$
```

N.B:

- admins est un groupe secondaire d'alexis
- pour que admins soit groupe principal il faut utiliser la commande

`$usermod -g admins alexis`

2 - Droits d'utilisateurs:

Le fichier sudoers est un fichier utilisé par les administrateurs Linux pour allouer des droits système aux utilisateurs du système.

`$ sudo vi /etc/sudoers`

```
GNU nano 5.4 /etc/sudoers *
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root      ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo     ALL=(ALL:ALL) ALL
%admins    ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "@include" directives:
@include /etc/sudoers.d
```

3- Configuration SSH:

Pour permettre aux nouveaux utilisateurs de se connecter il nous faut modifier la configuration ssh sur le fichier `/etc/ssh/sshd_config`.

1. Autoriser l'authentification en utilisant le mot de pass

\$sudo nano /etc/ssh/sshd_config

```
GNU nano 5.4 /etc/ssh/sshd_config *
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
```

2. Autoriser l'authentification:

```
# AllowTcpForwarding no
# PermitTTY no
# ForceCommand cvs server
ClientAliveInterval 120

AllowGroups admins members sudo
```

4. Configuration Fail2Ban:

1. Installation du fail2Ban:

```
admin@ip-172-31-46-110:/$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify python3-systemd whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify python3-systemd whois
0 upgraded, 4 newly installed, 0 to remove and 56 not upgraded.
Need to get 596 kB of archives.
After this operation, 2819 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://cdn-aws.deb.debian.org/debian bullseye/main amd64 fail2ban all 0.11
```

2. Configuration de fail2ban

Pour configurer fail2ban on utilise cette commande pour modifier le fichier de configuration

```
$sudo nano /etc/fail2ban/jail.conf
```

```
GNU nano 5.4 /etc/fail2ban/jail.conf *
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime = 3m

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 3m

# "maxretry" is the number of failures before a host get banned.
maxretry = 2
```

5-Configuration Serveur WEB:

Pour configurer un serveur web on va utiliser Apache 2 Web Server.

1. installer apache2

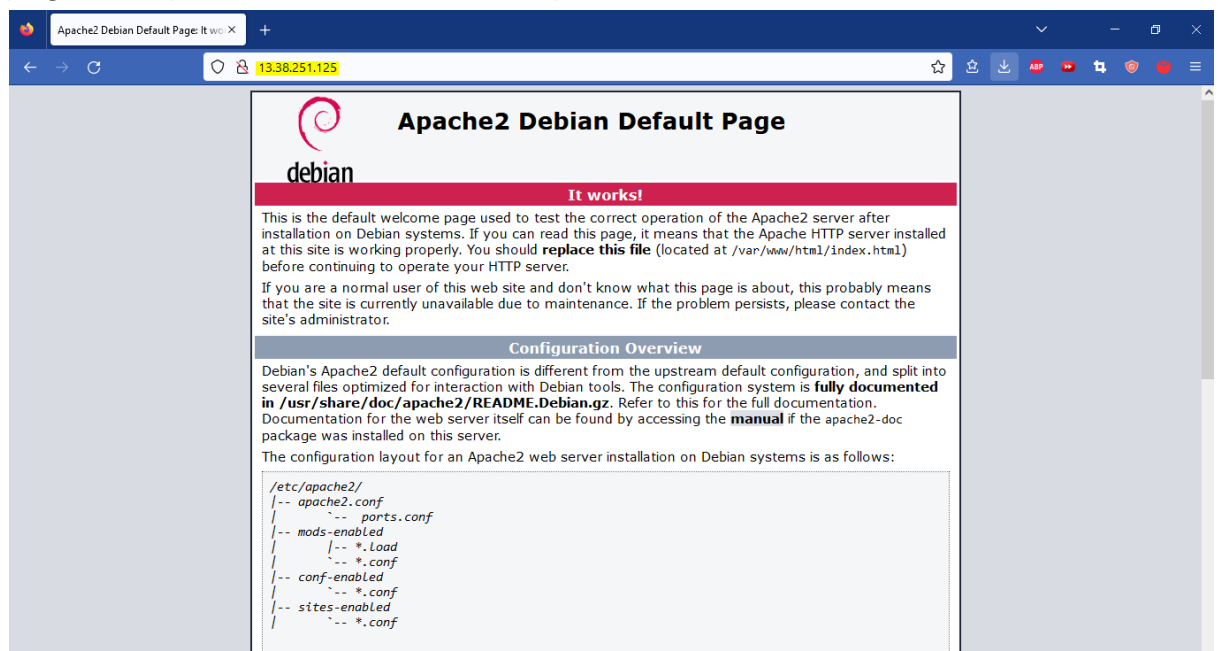
```
$ sudo apt install apache2
[sudo] password for hicham:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libgdbm-compat4 libjansson4
  liblua5.3-0 libperl5.32 mailcap mime-support perl perl-modules-5.32 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
  bzip2-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl
  make libtap-harness-archive-perl
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libgdbm-compat4 libjansson4
  liblua5.3-0 libperl5.32 mailcap mime-support perl perl-modules-5.32 ssl-cert
0 upgraded, 18 newly installed, 0 to remove and 5 not upgraded.
Need to get 9893 kB of archives.
After this operation, 56.0 MB of additional disk space will be used.
```

2. Pour démarrer le service au démarrage :

```
$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

```
$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Sat 2022-12-03 12:03:20 UTC; 4min 24s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 23667 (apache2)
       Tasks: 55 (limit: 1123)
      Memory: 9.1M
         CPU: 44ms
      CGroup: /system.slice/apache2.service
              └─23667 /usr/sbin/apache2 -k start
                └─23669 /usr/sbin/apache2 -k start
                  └─23670 /usr/sbin/apache2 -k start
lines 1-12/12 (END)
```

3. Pour assurer que l'installation est bien passée, apache dispose d'une page test qui va être servie dans le port 80.



Pour modifier le contenu de cette page test, on va just modifier le fichier index.html

\$sudo nano /var/www/html/index.html

```
GNU nano 5.4 /var/www/html/index.html
<h1>Bienvenue à Calais!</h1>
```



Bienvenue à Calais!

4. La configuration:

- Par défaut deux fichiers qui sont autorisés pour servir des fichiers, pour modifier les permissions il faut modifier le fichier apache2.conf.

```
GNU nano 5.4 /etc/apache2/apache2.conf
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

- Le port par défaut est 80:

\$sudo nano /etc/apache2/ports.conf

```
GNU nano 5.4 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

5. Permettre aux utilisateurs de créer leurs websites:

Les utilisateurs peuvent créer des sites Web sous leur propre répertoire personnel.

- il faut activer cette option : **a2enmod userdir**

```
root@ip-172-31-46-110:/# sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@ip-172-31-46-110:/# systemctl restart apache2
root@ip-172-31-46-110:/#
```

- ajouter un fichier **index.html** le répertoire personnel d'utilisateur



hello in my website

6. Héberger plusieurs sites

- Pour héberger plusieurs sites dans un seul web server:
 - Créer les fichiers des sites web dans /var/www
 - **/var/www/**
 - site1
 - index.html
 - site2
 - index.html
 - Créer les fichiers de configurations pour chaque site
 - **/etc/apache2/sites-available/**
 - 000-default.conf
 - site1.conf
 - site2.conf

```
GNU nano 5.4 site2.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName site2.com

    ServerAdmin hichamchah.jb.mc@gmail.com
    DocumentRoot /var/www/site2

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

fichier de configuration pour site2

- Activer le site1 et le site 2
- \$a2ensite site1.conf**

```
$a2ensite site2.conf
```

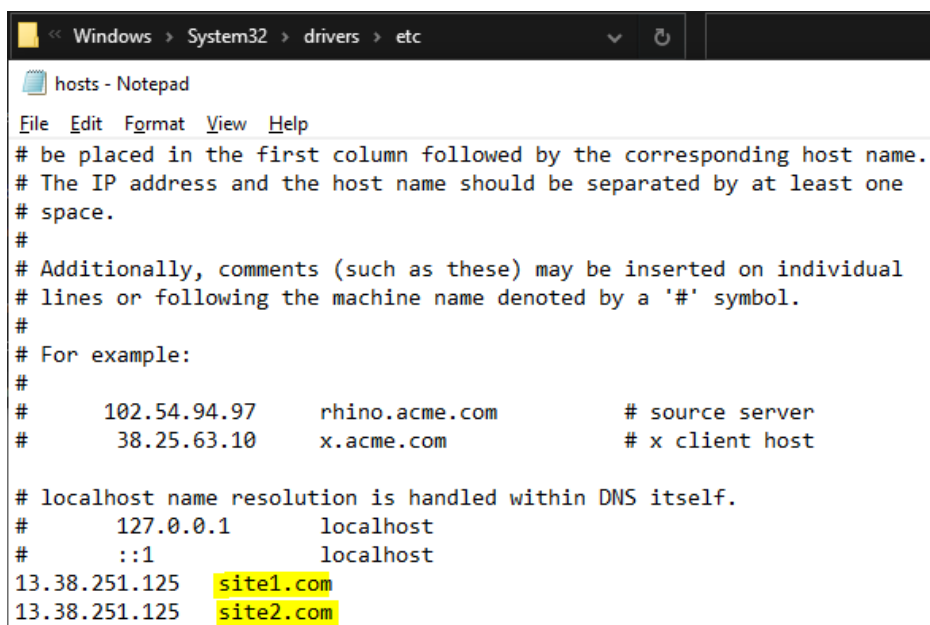
```
root@ip-172-31-46-110:/etc/apache2/sites-enabled# a2ensite site1.conf
Enabling site site1.
To activate the new configuration, you need to run:
systemctl reload apache2
```

-

```
$systemctl reload apache2
```

```
root@ip-172-31-46-110:/etc/apache2/sites-enabled# a2ensite site1.conf
Enabling site site1.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Dans le fichier hosts (pour windows):



```
File Edit Format View Help
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10       x.acme.com               # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost
13.38.251.125 site1.com
13.38.251.125 site2.com
```

Le contenu du fichier index.html de site2 est : "Bienvenue Site2"

Le contenu du fichier index.html de site1 est : "Bienvenue Site1"



Bienvenue Site2



Bienvenue site1!