Sécurité serveur ovh

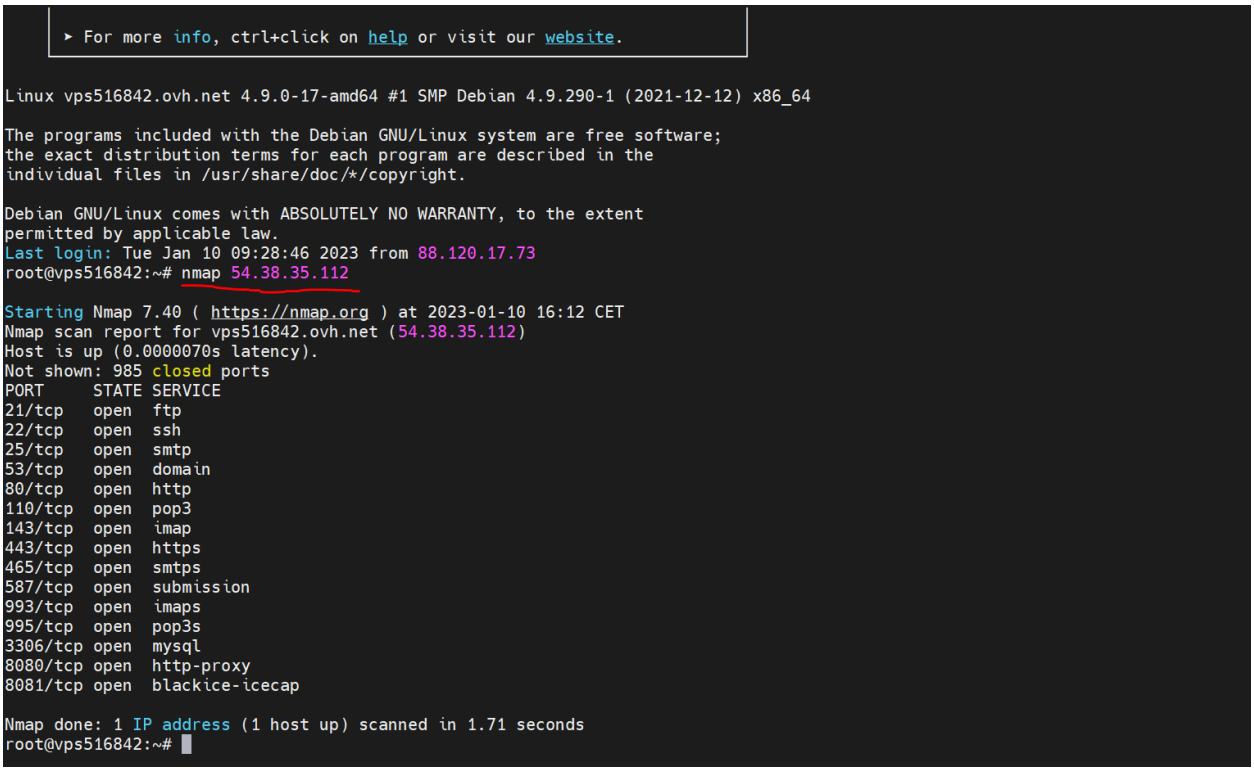
Liste des contrôles de sécurité que nous pouvons effectuer sur le serveur OVH :

- Gestion des ports
- Consultation des logs de connexion
- Mise à jour du système d’exploitation/des applications
- Gestion des privilèges utilisateurs
- Gestion des connexions SSH
- Vérification de la force des mots de passe

Gestion des ports

Tout d’abord nous allons commencer par nous connecter au serveur web pour effectuer les vérifications qui sont à notre portée. La première chose à faire et regarder quels ports sont ouverts et pour quelles raisons ils le sont, nous allons le faire avec la commande “nmap”.

```
nmap 54.38.35.112
```



ci nous avons la liste des ports ouverts avec le service qui leur est dédié, on voit que 15 ports sont ouverts et 985 fermés, ce qui est bon signe. Chacun des ports ouverts a une utilité respective :

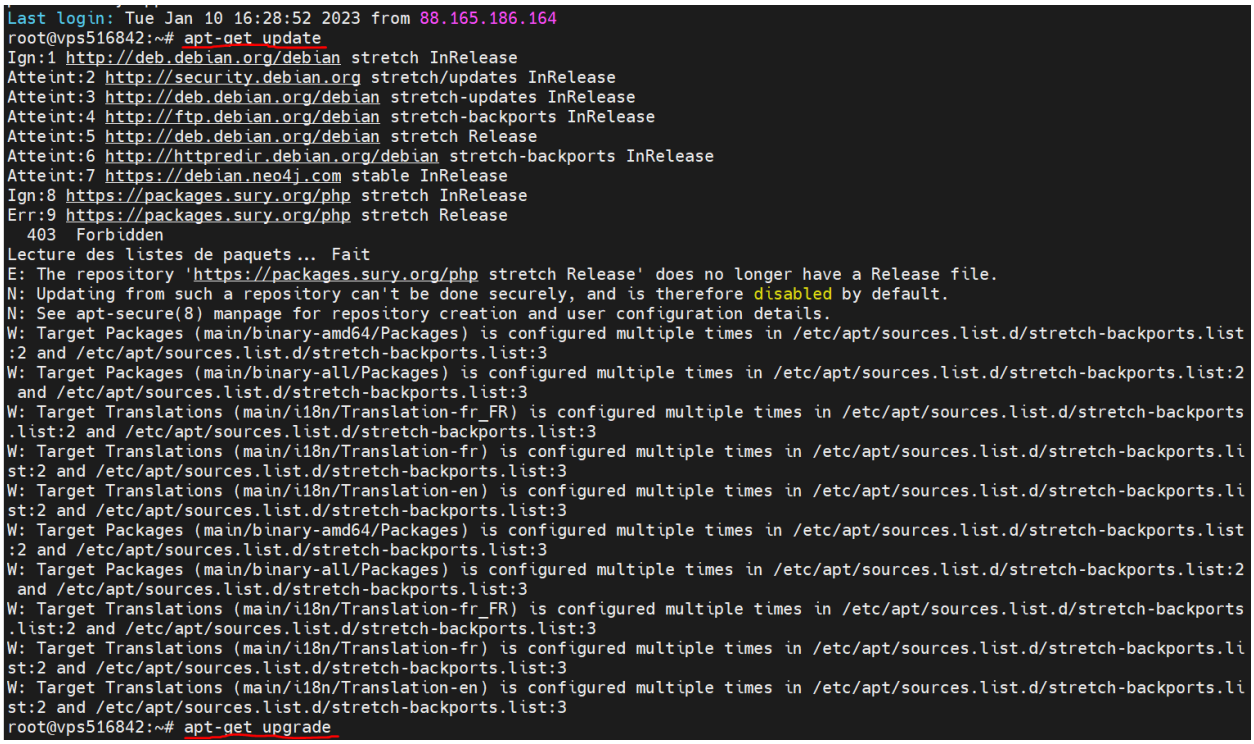
- Le port 21 est utilisé pour les connexions FTP (File Transfer Protocol) pour envoyer et recevoir des fichiers
- Le port 22 est utilisé pour les connexions SSH (Secure Shell) pour la gestion à distance de serveurs et de dispositifs
- Le port 25 est utilisé pour les connexions SMTP (Simple Mail Transfer Protocol) pour envoyer des emails
- Le port 53 est utilisé pour les connexions DNS (Domain Name System) pour la résolution des noms de domaine
- Le port 80 est utilisé pour les connexions HTTP (Hypertext Transfer Protocol) pour afficher des pages web
- Le port 110 est utilisé pour les connexions POP3 (Post Office Protocol version 3) pour récupérer des emails depuis un serveur

- Le port 143 est utilisé pour les connexions IMAP (Internet Mail Access Protocol) pour accéder à des emails sur un serveur
- Le port 443 est utilisé pour les connexions HTTPS (Hypertext Transfer Protocol Secure) pour protéger les communications web et l'échange de données sensibles
- Le port 465 est utilisé pour les connexions SMTP over SSL (Secure Sockets Layer) pour envoyer des emails de manière sécurisée
- Le port 587 est utilisé pour les connexions submission pour envoyer des emails depuis un client vers un serveur
- Le port 993 est utilisé pour les connexions IMAP over SSL pour accéder à des emails de manière sécurisée
- Le port 3306 est utilisé pour les connexions MySQL pour gérer les bases de données
- Le port 8080 est utilisé pour les connexions web proxy ou pour les applications web personnalisées
- Le port 8081 est utilisé pour les connexions similaires à celles du port 8080, généralement utilisé pour des applications web supplémentaires ou pour des services différents.

Mise à jour du système d’exploitation

Un bon moyen de rester à l’abris de l’exploitation de potentielles vulnérabilités est de constamment maintenir son système d’exploitation à jour, nous allons procéder au téléchargement des mises à jour avec “apt-get update” et lancer l’installation avec “apt-get upgrade”.

```
apt-get update
apt-get upgrade
```



Gestion des privilèges utilisateurs

Actuellement toutes les connexions au serveur web par SSH se font sous l’utilisateur Root, ce qui est une mauvaise chose dans la mesure où il les actions effectuées en tant qu'utilisateur root ont un impact global sur le système. Si un utilisateur fait une erreur en utilisant la commande root, il pourrait causer des problèmes importants sur le système, comme la suppression de fichiers importants.
Il faut appliquer le principe du moindre privilège et ainsi ne pas se servir de comptes qui disposent de droits supérieurs à ce dont on a besoin.

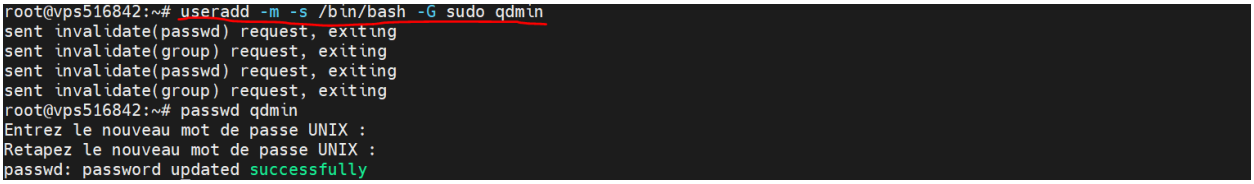
Avec la commande “compgen -u” on voit la liste des utilisateurs présents sur la machine, incluant les utilisateurs dont se servent les services installés, d’où le nombre et les noms des utilisateurs.
On voit ici qu’il n’y a que deux utilisateurs dont on peut réellement se servir, Root et Debian, deux utilisateurs installés par défaut.

```
compgen -u
```

On va commencer par créer un nouvel utilisateur administrateur afin de ne pas avoir à effectuer toutes les tâches en Root et donc de limiter les risques.

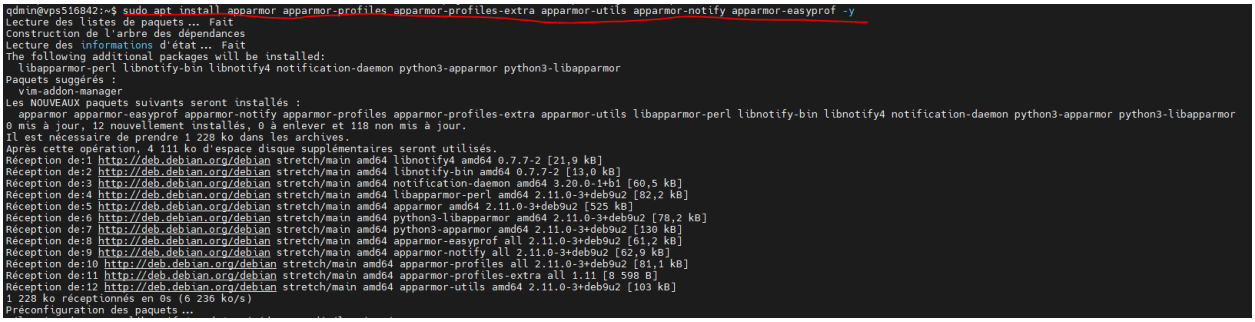
Ici on créer l'utilisateur "qdmin", en le mettant dans le groupe "sudo" afin qu'il soit capable d'utiliser les commandes root lorsque cela est nécessaire. On évite d'utiliser un nom comme "admin" ou "administrateur" afin d'éviter les tentatives d'attaques brute force où les attaquants essaient de se connecter à partir de noms d'utilisateurs répandus.

```
useradd -m -s /bin/bash -G sudo qdmin
```



Afin d'avoir encore plus de contrôle sur les accès utilisateurs nous allons installer une application de gestion de droits, ici AppArmor mais nous aurions aussi pu utiliser SElinux par exemple.
On commence par installer l'application et les différents modules qui nous permettront une utilisation optimale.

```
sudo apt install apparmor apparmor-profiles apparmor-profiles-extra apparmor-notify
apparmor-notify apparmor-easyprof -y
sudo apt install auditd -y
```



```
qdmin@vps516842:~$ sudo apt install auditd -y
Lecture des listes de paquets ... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état ... Fait
The following additional packages will be installed:
  libauparse0
Paquets suggérés :
  audispd-plugins
Les NOUVEAUX paquets suivants seront installés :
  auditd libauparse0
0 mis à jour, 2 nouvellement installés, 0 à enlever et 118 non mis à jour.
Il est nécessaire de prendre 259 ko dans les archives.
Après cette opération, 781 ko d'espace disque supplémentaires seront utilisés.
Réception de:1 http://deb.debian.org/debian stretch/main amd64 libauparse0 amd64 1:2.6.7-2 [53,6 kB]
Réception de:2 http://deb.debian.org/debian stretch/main amd64 auditd amd64 1:2.6.7-2 [205 kB]
259 ko réceptionnés en 0s (6 274 ko/s)
Sélection du paquet libauparse0:amd64 précédemment désélectionné.
(Lecture de la base de données ... 78848 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../libauparse0_1%3a2.6.7-2_amd64.deb ...
Dépaquetage de libauparse0:amd64 (1:2.6.7-2) ...
Sélection du paquet auditd précédemment désélectionné.
Préparation du dépaquetage de .../auditd_1%3a2.6.7-2_amd64.deb ...
Dépaquetage de auditd (1:2.6.7-2) ...
Paramétrage de libauparse0:amd64 (1:2.6.7-2) ...
Paramétrage de auditd (1:2.6.7-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/auditd.service → /lib/systemd/system/auditd.service.
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Traitement des actions différées (« triggers ») pour libc-bin (2.24-11+deb9u4) ...
Traitement des actions différées (« triggers ») pour systemd (232-25+deb9u13) ...
Traitement des actions différées (« triggers ») pour man-db (2.7.6.1-2) ...
qdmin@vps516842:~$
```

On va maintenant vérifier que les services installés sont bien actifs avec la commande

```
systemctl status apparmor
```

```
qdmin@vps516842:~$ sudo apparmor_status
apparmor module is loaded.
apparmor filesystem is not mounted.
qdmin@vps516842:~$ nano /etc/default/grub/etc/default/grub
qdmin@vps516842:~$ sudo mkdir -p /etc/default/grub.d
qdmin@vps516842:~$ sudo cat /etc/default/grub.d
cat: /etc/default/grub.d: est un dossier
qdmin@vps516842:~$ echo 'GRUB_CMDLINE_LINUX_DEFAULT="$GRUB_CMDLINE_LINUX_DEFAULT apparmor=1 security=apparmor"' \
>   | sudo tee /etc/default/grub.d/apparmor.cfg
GRUB_CMDLINE_LINUX_DEFAULT="$GRUB_CMDLINE_LINUX_DEFAULT apparmor=1 security=apparmor"
qdmin@vps516842:~$ sudo ls /etc/default/grub.d
apparmor.cfg
qdmin@vps516842:~$ sudo cat /etc/default/grub.d/apparmor.cfg
GRUB_CMDLINE_LINUX_DEFAULT="$GRUB_CMDLINE_LINUX_DEFAULT apparmor=1 security=apparmor"
qdmin@vps516842:~$ sudo update-grub
Création du fichier de configuration GRUB…
Image Linux trouvée : /boot/vmlinuz-4.9.0-17-amd64
Image mémoire initiale trouvée : /boot/initrd.img-4.9.0-17-amd64
fait
qdmin@vps516842:~$ sudo reboot
```

On voit que ce n’est pas le cas, le module est bien installé mais pas actif. Pour remédier à ça nous devons autoriser l’utilisation d’AppArmor au kernel.

```
sudo mkdir -p /etc/default/grub.d
echo 'GRUB_CMDLINE_LINUX_DEFAULT="$GRUB_CMDLINE_LINUX_DEFAULT apparmor=1
security=apparmor"' \ | sudo tee /etc/default/grub.d/apparmor.cfg
sudo update-grub
```

```
qdmin@vps516842:~$ systemctl status auditd
● auditd.service - Security Auditing Service
   Loaded: loaded (/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-01-24 13:32:00 CET; 3min 4s ago
     Docs: man:auditd(8)
           https://people.redhat.com/sgrubb/audit/
 Main PID: 31536 (auditd)
   CGroup: /system.slice/auditd.service
           └─31536 /sbin/auditd -n
qdmin@vps516842:~$ systemctl status apparmor
● apparmor.service - AppArmor initialization
   Loaded: loaded (/lib/systemd/system/apparmor.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:apparmor(7)
           http://wiki.apparmor.net/
```

```
qdmin@vps516842:~$ cat /sys/module/apparmor/parameters/enabled
Y
qdmin@vps516842:~$ sudo aa-status
[sudo] Mot de passe de qdmin :
apparmor module is loaded.
48 profiles are loaded.
15 profiles are in enforce mode.
   /usr/bin/freshclam
   /usr/bin/irssi
   /usr/bin/pidgin
   /usr/bin/pidgin//launchpad_integration
   /usr/bin/pidgin//sanitized_helper
   /usr/bin/totem
   /usr/bin/totem-audio-preview
   /usr/bin/totem-video-thumbnailer
   /usr/sbin/apt-cacher-ng
   /usr/sbin/clamd
   /usr/sbin/haveged
   /usr/sbin/named
   /usr/sbin/ntpd
   /usr/sbin/tcpdump
   gst_plugin_scanner
33 profiles are in complain mode.
```

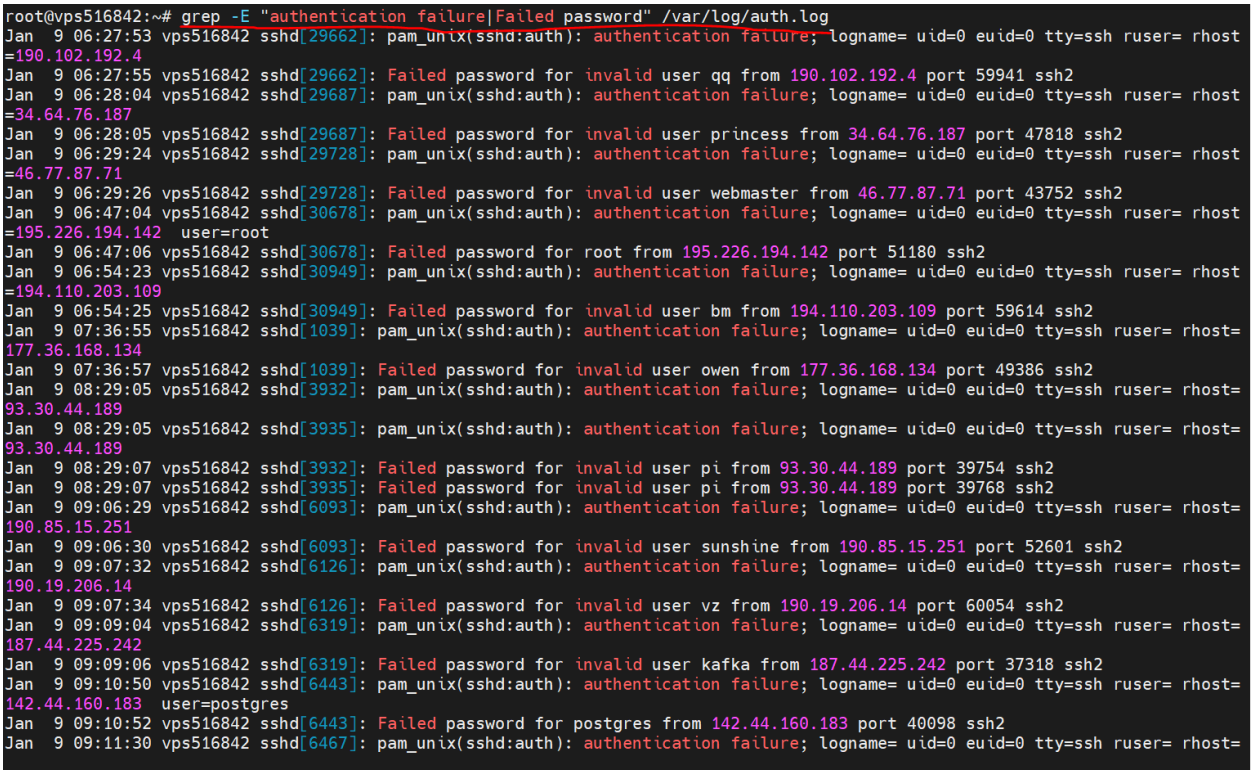
AppArmor est maintenant installé et actif, il suffira de configurer les permissions des prochains utilisateurs à l'aide des différents profils.

Consultation des logs de connexion

Une autre chose importante à vérifier est l'historique de connexions SSH au serveur, cela va tout simplement nous permettre de lister toutes les tentatives de connexion au serveur, établies ou échouées. Nous trouvons ces journaux dans le dossier "/var/log/auth.log". Nous allons utiliser la commande "grep" pour extraire des chaines de caractères du fichier et n'afficher que ce qui nous intéresse, ici "authentication failure" et

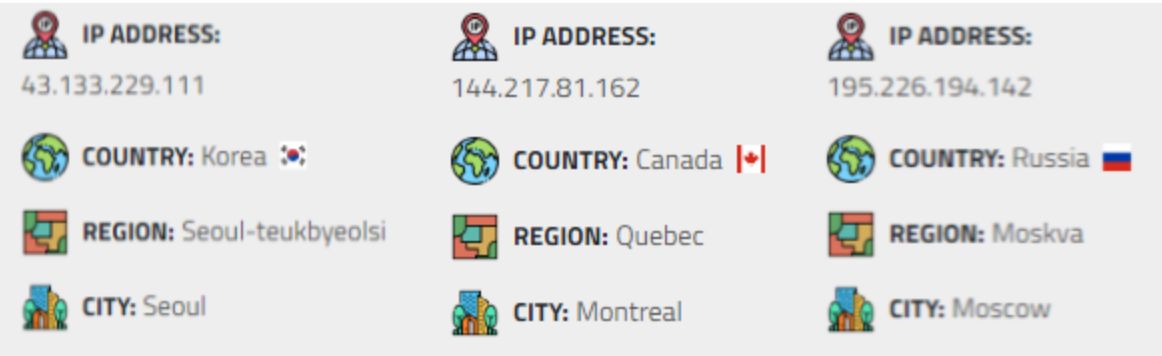
“failed password” afin de voir si il y a eu des tentatives de connexion échouées au serveur.

```
grep -E "authentication failure|Failed password" /var/log/auth.log
```



```
Jan  9 09:20:09 vps516842 sshd[6907]: Failed password for invalid user administrator from 157.230.98.148 port 50534 ssh2
Jan  9 09:20:43 vps516842 sshd[6921]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
35.199.73.100
Jan  9 09:20:45 vps516842 sshd[6921]: Failed password for invalid user jacky from 35.199.73.100 port 60666 ssh2
Jan  9 09:21:02 vps516842 sshd[6928]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
43.156.39.228
Jan  9 09:21:03 vps516842 sshd[6928]: Failed password for invalid user oracle from 43.156.39.228 port 53784 ssh2
Jan  9 09:28:32 vps516842 sshd[7228]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
162.243.172.183
Jan  9 09:28:34 vps516842 sshd[7228]: Failed password for invalid user deploy from 162.243.172.183 port 36470 ssh2
Jan  9 09:30:24 vps516842 sshd[7371]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
34.101.240.144
Jan  9 09:30:26 vps516842 sshd[7371]: Failed password for invalid user sysadmin from 34.101.240.144 port 33912 ssh2
Jan  9 09:30:39 vps516842 sshd[7381]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
201.161.42.34
Jan  9 09:30:41 vps516842 sshd[7381]: Failed password for invalid user elastic from 201.161.42.34 port 41270 ssh2
Jan  9 09:31:07 vps516842 sshd[7405]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
43.133.229.111
Jan  9 09:31:09 vps516842 sshd[7405]: Failed password for invalid user work from 43.133.229.111 port 47616 ssh2
Jan  9 09:33:40 vps516842 sshd[7463]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
189.204.218.146
Jan  9 09:33:41 vps516842 sshd[7463]: Failed password for invalid user isa from 189.204.218.146 port 16432 ssh2
Jan  9 10:31:06 vps516842 sshd[10837]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
=43.134.175.205
Jan  9 10:31:08 vps516842 sshd[10837]: Failed password for invalid user admin from 43.134.175.205 port 49320 ssh2
Jan  9 11:05:39 vps516842 sshd[12877]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
=46.101.29.76
Jan  9 11:05:41 vps516842 sshd[12877]: Failed password for invalid user chenjing from 46.101.29.76 port 38430 ssh2
Jan  9 12:05:51 vps516842 sshd[16138]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
=64.227.190.199  user=root
Jan  9 12:05:53 vps516842 sshd[16138]: Failed password for root from 64.227.190.199 port 50206 ssh2
Jan  9 12:07:29 vps516842 sshd[16188]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
=144.217.81.162  user=root
Jan  9 12:07:30 vps516842 sshd[16188]: Failed password for root from 144.217.81.162 port 39930 ssh2
Jan  9 12:12:26 vps516842 sshd[16539]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
=144.217.81.162  user=root
Jan  9 12:12:28 vps516842 sshd[16539]: Failed password for root from 144.217.81.162 port 37808 ssh2
Jan  9 12:12:46 vps516842 sshd[16549]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost
```

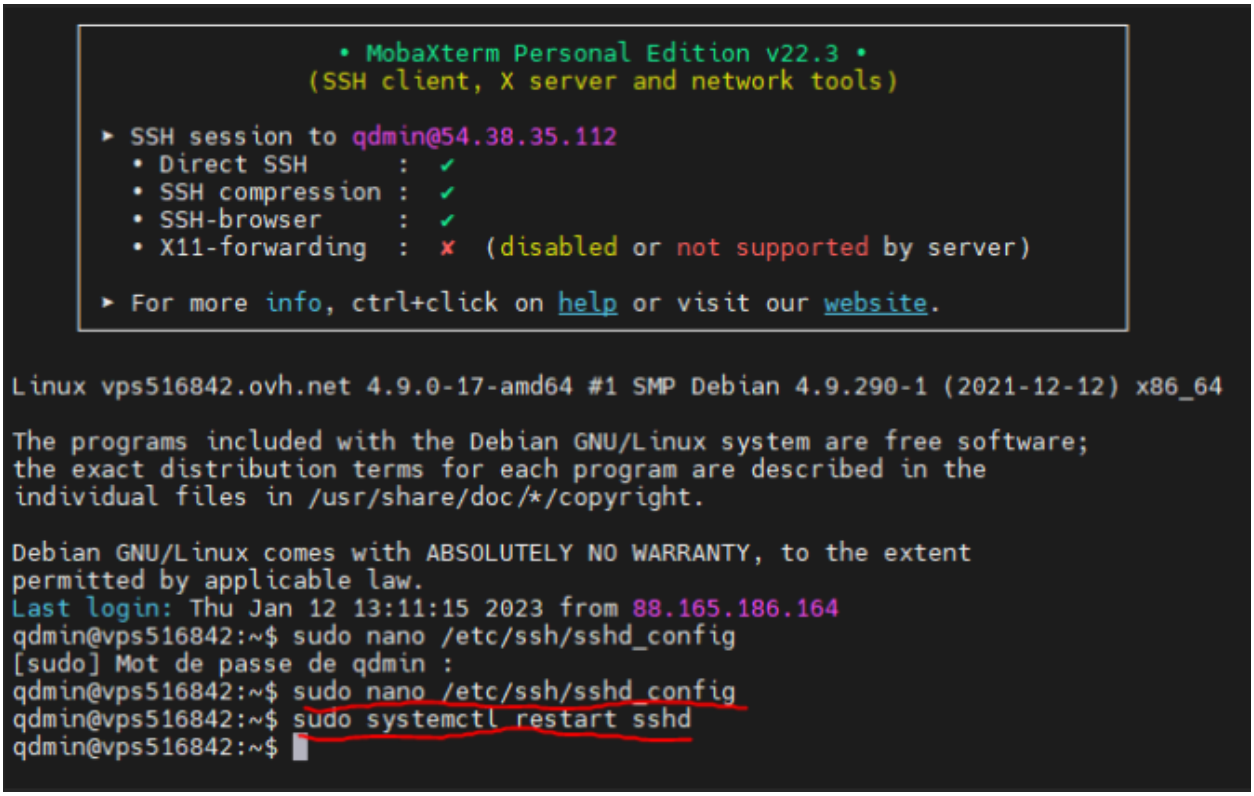
l y a ici un problème, en effet la commande entrée nous montre un nombre élevé de tentatives de connexion échouées (Nom d'utilisateur ou mot de passe incorrect) ce qui implique des personnes ont tenté de se connecter au serveur.
Voici les localisations de quelques de ces adresses :

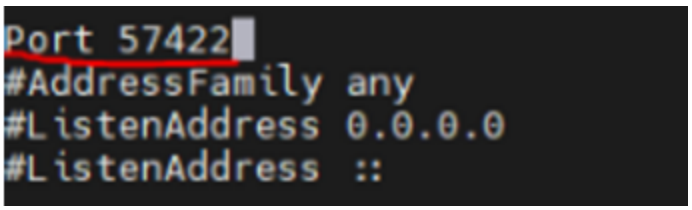


Mesure n°1 : changement du port SSH

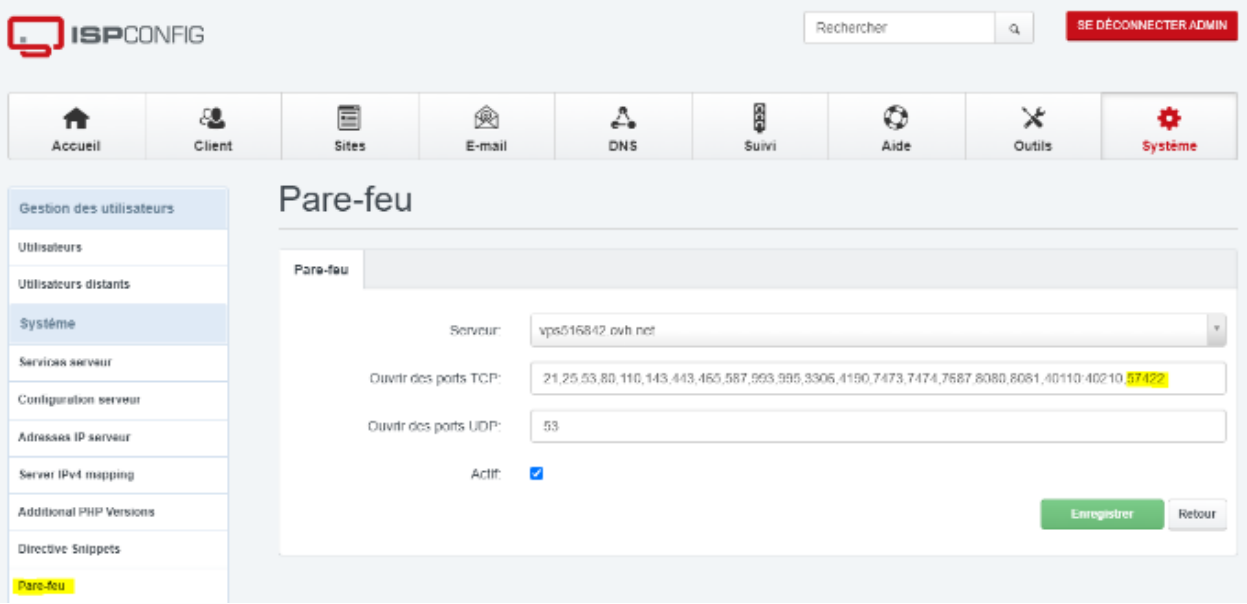
Les attaquants tentent d’accéder au serveur en essayant des combinaisons génériques soit en utilisant des noms d’utilisateurs comme root ou admin et en se connectant au port 22, qui est le port SSH par défaut, pour limites les tentatives de connexion il est possible de changer le port SSH.
La modification du port de connexion SSH est sur le plan technique pas très compliquée cependant il faut être vigilant et veiller à ne pas se bloquer soi-même l’accès au serveur. Ici on commence par éditer le fichier de configuration et redémarrer le service SSH après modifications.

```
sudo nano /etc/ssh/sshd_config
sudo systemctl restart sshd
```



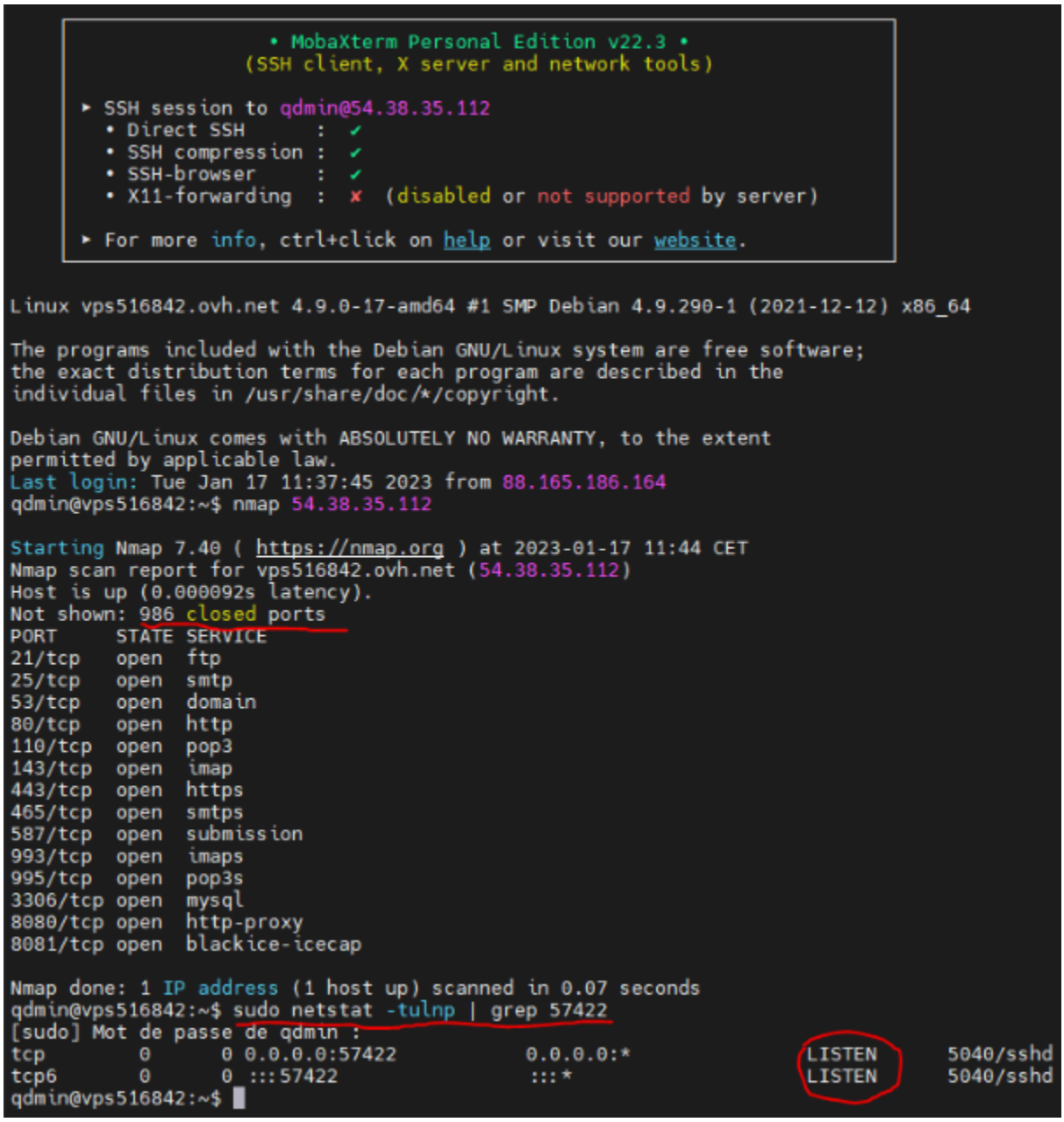


On viens simplement supprimer le “#” devant la ligne du port afin qu’elle soit prise en compte et non considérée comme un commentaire puis on remplace le port qui était à l’origine 22 par un port de notre choix, ici 57422, un port qui n’est pas utilisé par un quelconque autre service.
Par la suite on va également modifier l’ouverture du port sur le pare-feu de manière à pouvoir utiliser le port que l’on vient d’attribuer.



On se rend sur l’ISPconfig, dans les paramètres du pare-feu et on vient supprimer le port 22 de la liste et y ajouter le port 57422 que l’on a attribué aux connexions SSH.

```
nmap 54.38.35.112
sudo netstat -tulnp | grep 57422
```

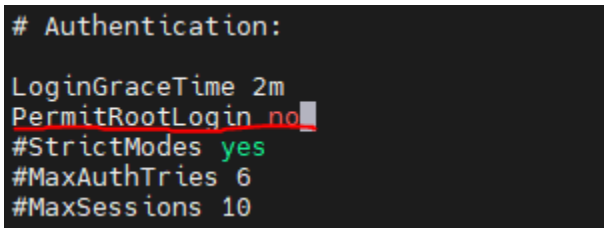


On essaie de se reconnecter au serveur mais cette fois en spécifiant le port 57422 au lieu de 22, la connexion est établie le changement a donc bien été pris en compte.
On vérifie par la suite de nouveau avec la commande “nmap” que le port 22 a bien été fermé, ce qui est le cas car il n’apparait plus dans la liste.
On vérifie que le port 57422 est ouvert, évidemment nous n’aurions pas pu nous connecter s’il ne l’était pas, c’est bien le cas.

Mesure n°2 : Désactivation des connexions root

Etant donné qu'on a jouté un nouvel utilisateur ayant les droits d'administration, on va peut l'accès au serveur en SSH avec l'utilisateur root sans risquer de manquer de droits, encore une fois de le but d'éviter les tentatives de compromission du compte par des attaques de force brute. Pour ce faire on se rend dans le fichier de configuration SSH et on passe "PermitRootLogin" de yes à no puis une fois la modification effectuée on redémarre le service SSH.

```
sudo nano /etc/ssh/sshd_config
sudo systemctl restart sshd
```

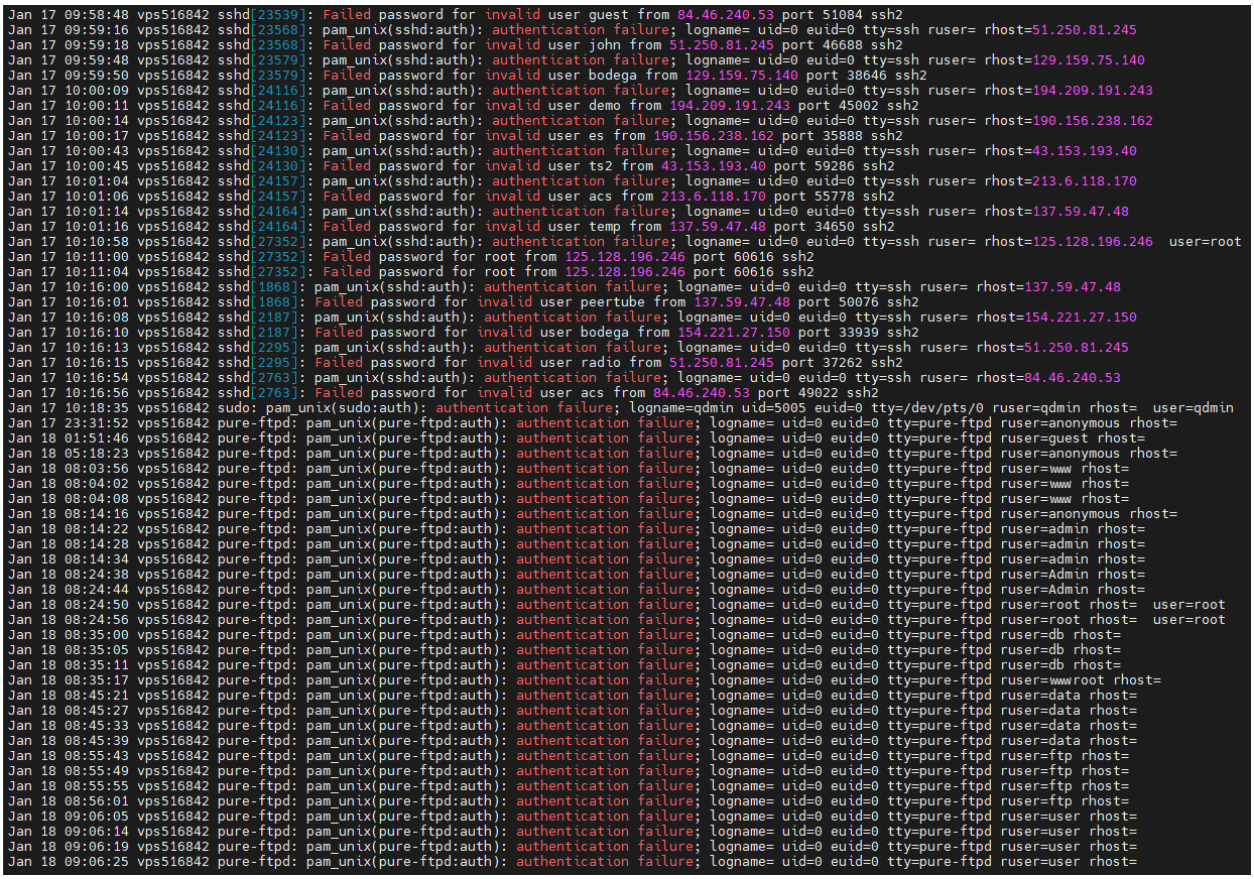


Mesure n°3 : Installation d'une solution de prévention d'intrusion (Fail2Ban)

Se référer à la documentation de Samy

Résultats

Quelques jours après les modifications apportées, voici le contenu des logs de connexion au serveur, qu'on récupère avec la commande utilisée auparavant.



On remarque très vite une chose, à partir de la modification il n’y a plus la moindre tentative d’intrusion par SSH sur le serveur cependant, les attaquants tentent maintenant de se connecter au serveur ftp.
On distingue les deux par le module unix utilité (sshd:auth) pour les connexions SSH et (pure-ftpd:auth) pour les connexions en FTP.

Les résultats des mesures de sécurité mises en places sont concluants mais pas suffisants étant donné qu’il reste des tentatives d’intrusion, ainsi nous allons devoir renforcer la sécurité des connexions FTP.