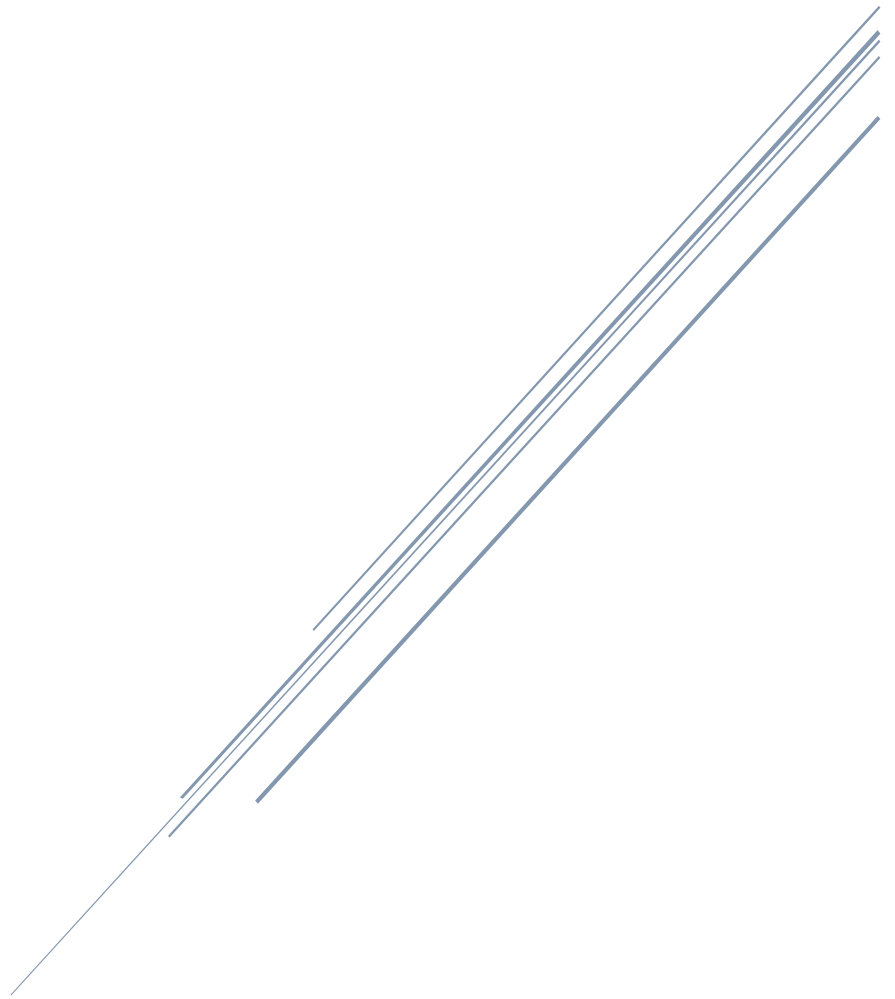


DESCRIPTION D'UNE MISSION

BTS SIO SISR



Matis GAGNEUX

Description d'une mission en entreprise numéro

Sommaire:

Le cahier des charges	3-4
Contexte	3
Expressions des besoins	3
Outils disponibles	3
Description de l'existant.....	4
Délais	4
OpenSSH	5-12
Installation de OpenSSH	5-7
Changement port SSH	8-9
Désactivation du root login	10-12
Fail2ban	13-16
Installation de fail2ban	13-14
Configuration de fail2ban	15-16
Test	17
Bilan.....	18

Le cahier des charges

Contexte :

L'entreprise Madnot a récemment fait l'acquisition de nouveau bureau, avant l'arrivée des salariés, nous devons mettre en place une infrastructure systèmes et réseau fiable et sécurisé. Nous avons reçu la mission de mettre en place un serveur SSH et de sécuriser la connexion pour éviter toute connexion malveillante.

Expression des besoins :

Les besoins de l'entreprise sont les suivants :

- Installation d'un serveur SSH
- Sécurisation de la connexion SSH

Outils disponibles :

Un serveur nous a été mis à disposition, il s'agit d'un HP BL460C G8 (Blade) dans un HP C7000. Nous disposons d'un accès IPMI (ILO) à ce serveur.



Ci-dessus à gauche le serveur HP BL460C G8 mis à disposition, il s'agit d'une lame qui s'insère dans un châssis ici un HP C7000 (à droite) qui comporte 16 lames.





Description de l'existant :

Un serveur BL460C G8 a déjà été installé avec le système d'exploitation Debian 11, nous pouvons nous y connecter physiquement avec un écran en VGA pour le déroulement de la mission.

Délais :

Pour la réalisation de cette mission nous disposons de la journée, la mission nous a été communiquée à 10h et nous avons donc jusqu'à 18h pour la compléter.

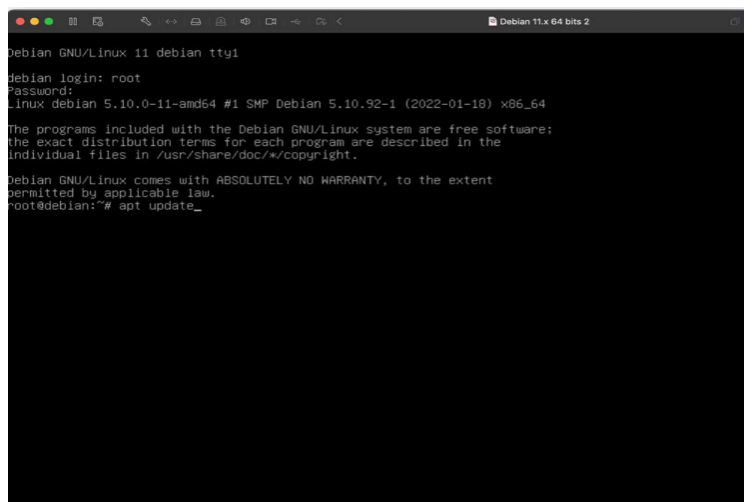
Voici ci-dessous un diagramme de Gantt qui représente l'avancement de la mission :

	10h	14h	18h
Prise en compte de la demande			
Installation du serveur SSH et configuration			
Installation de fail2ban et configuration			
Test			

OpenSSH :

Installation d'OpenSSH :

Tout d'abord nous recherchons les mises à jour disponible :

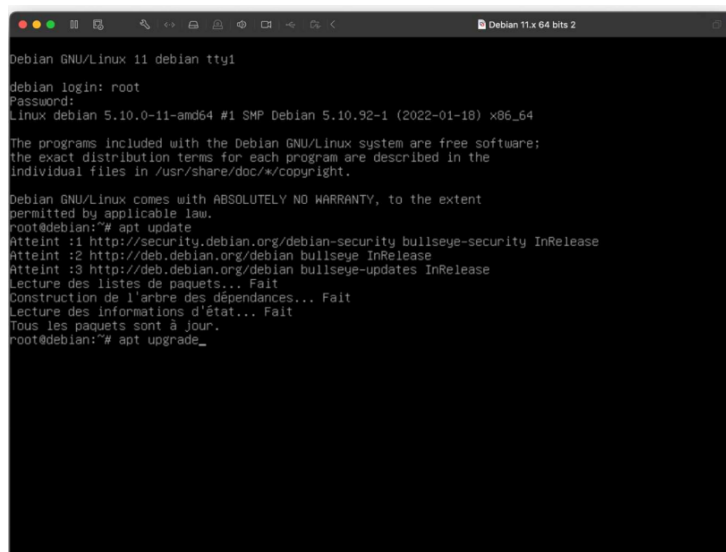


```
Debian GNU/Linux 11 debian tty1
debian login: root
Password:
Linux debian 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# apt update_
```

Ensuite nous pouvons installer les mises à jour disponible :



```
Debian GNU/Linux 11 debian tty1
debian login: root
Password:
Linux debian 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# apt update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@debian:~# apt upgrade_
```

Maintenant que le serveur est à jour, nous pouvons installer OpenSSH :

```
Debian GNU/Linux 11 debian tty1
debian login: root
Password:
Linux debian 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# apt update
Atteint :1 http://security.debian.org/debian-security bullseye-security InRelease
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@debian:~# apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:~# apt install openssh
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
E: Impossible de trouver le paquet openssh
root@debian:~# apt install openssh-server _
```

Nous avons donc installé notre serveur SSH qui est actif comme nous pouvons le voir ci-dessous :

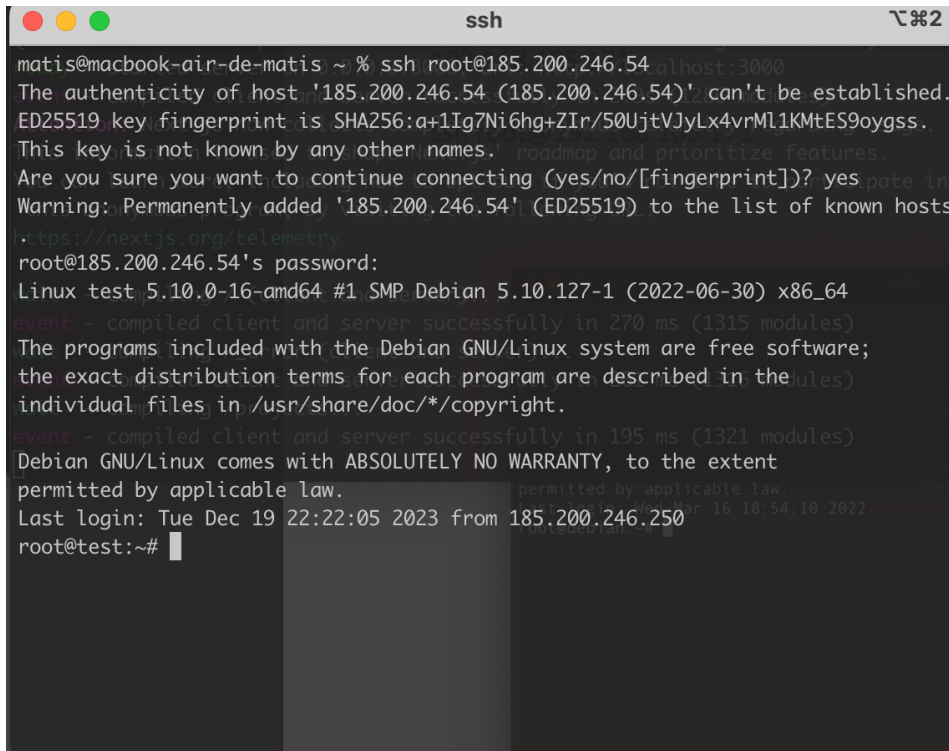
```
Paramétrage de runit-helper (2.10.3) ...
Paramétrage de openssh-sftp-server (1:8.4p1-5) ...
Paramétrage de liburap0:amd64 (7.6.q-31) ...
Paramétrage de openssh-server (1:8.4p1-5) ...

Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:bEsu6mHE9khtMskLo706L2REz1C47dd2I1BPCTNVdx8 root@debian (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:FMCfG311bRBCrzE41+dMRRt4+I27d1VfaF6qPfsAHOE root@debian (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:rdzehLYg/B0mqTTbxH35BHydJ0uwtISm0u+NIXAlju root@debian (ED25519)
Created symlink /etc/systemd/system/ssh.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.31-13+deb11u2) ...
root@debian:~# service ssh status
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-03-07 10:22:48 CET; 1min 47s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 1130 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1131 (sshd)
    Tasks: 1 (limit: 2301)
   Memory: 1.1M
      CPU: 16ms
   CGroup: /system.slice/ssh.service
           └─1131 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

mars 07 10:22:48 debian systemd[1]: Starting OpenBSD Secure Shell server...
mars 07 10:22:48 debian sshd[1131]: Server listening on 0.0.0.0 port 22.
mars 07 10:22:48 debian sshd[1131]: Server listening on :: port 22.
mars 07 10:22:48 debian systemd[1]: Started OpenBSD Secure Shell server.
root@debian:~#
```

Nous pouvons maintenant nous connecter en SSH au serveur, son IP est 192.168.105.171.

```
ssh root@192.168.105.171
```

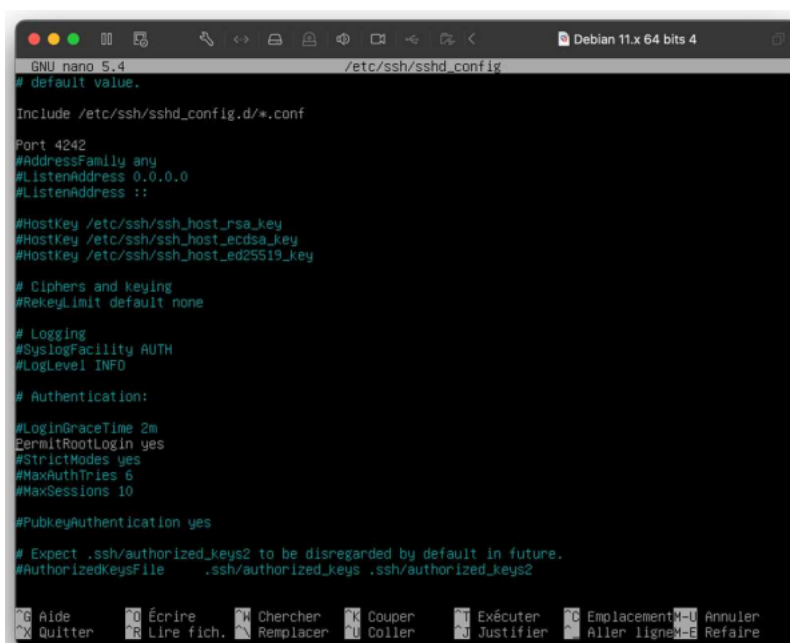


```
ssh
matis@macbook-air-de-matis ~ % ssh root@185.200.246.54 alhost:3000
The authenticity of host '185.200.246.54 (185.200.246.54)' can't be established.
ED25519 key fingerprint is SHA256:a+1Ig7Ni6hg+ZIr/50UjtVJyLx4vrMl1KMtES9oygss.
This key is not known by any other names. roadmap and prioritize features.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes pate in
Warning: Permanently added '185.200.246.54' (ED25519) to the list of known hosts
https://nextjs.org/telemetry
root@185.200.246.54's password:
Linux test 5.10.0-16-amd64 #1 SMP Debian 5.10.127-1 (2022-06-30) x86_64
event - compiled client and server successfully in 270 ms (1315 modules)
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the (ules)
individual files in /usr/share/doc/*/copyright.
event - compiled client and server successfully in 195 ms (1321 modules)
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 19 22:22:05 2023 from 185.200.246.250 permitted by applicable law.
root@test:~#
```

Désactivation du root login :

L'accès SSH à une machine permet d'accéder à cette dernière via un terminal (CLI). Par défaut, on peut utiliser tous les utilisateurs présents sur le système pour s'y authentifier. Néanmoins, il peut être dangereux de laisser l'utilisateur root se loguer en SSH car cela peut ouvrir la porte à des attaques par force brute qui donneront un accès direct au plus haut niveau de privilèges de la machine.

Pour désactiver le root login nous retournons dans la configuration SSH « nano /etc/ssh/sshd_config », à la ligne « PermitRootLogin » :



```
GNU nano 5.4 /etc/ssh/sshd_config
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 4242
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

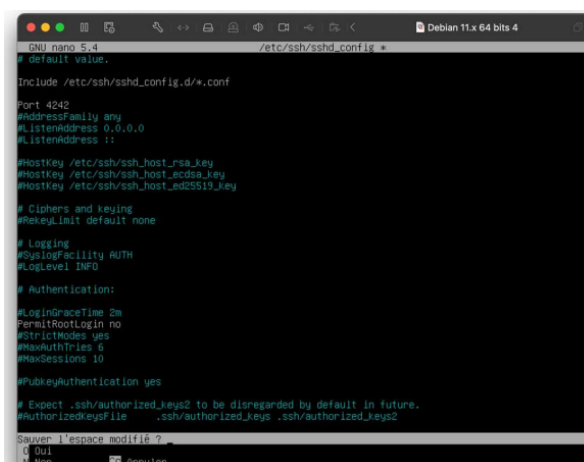
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```

Nous remplaçons yes par no pour désactiver la connexion SSH en root :



```
GNU nano 5.4 /etc/ssh/sshd_config
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 4242
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```


Nous redémarrons le service SSH avec « service ssh restart » puis nous créons un utilisateur qui pourra se connecter en SSH avec la commande « adduser » l'utilisateur sera « clients » et le mot de passe « NPkgpD9itMpaJig5 »

```
Debian 11.x 64 bits 4
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

root@debian:~# adduser
adduser : Un ou deux noms maximum.
root@debian:~# adduser client
Ajout de l'utilisateur « client » ...
Ajout du nouveau groupe « client » (1001) ...
Ajout du nouvel utilisateur « client » (1001) avec le groupe « client » ...
Création du répertoire personnel « /home/client »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for client
Enter the new value, or press ENTER for the default
  Full Name []: client
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Cette information est-elle correcte ? [0/n]
```

Nous pouvons maintenant voir que la connexion SSH en utilisateur root n'est pas permise par le serveur :

```
root@192.168.105.171's password:
Permission denied, please try again.
root@192.168.105.171's password: ?
```

Nous pouvons nous y connecter via l'utilisateur client que nous venons de créer «ssh clients@192.168.105.171 -p 4242 » avec le mot de passe « NPkgpD9itMpaJig5 »

```
client@192.168.105.171's password:
Linux debian 5.10.0-12-amd64 #1 SMP Debian 5.10.103-1 (2022-03-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
client@debian:~$ █
```

Si nous avons besoin des permissions root nous pouvons utiliser la commande « su » .

Fail2ban :

Installation de fail2ban :

Fail2ban est une application qui analyse les logs de divers services (SSH, Apache, FTP...) en cherchant des correspondances entre des motifs définis dans ses filtres et les entrées des logs. Lorsqu'une correspondance est trouvée, une ou plusieurs actions sont exécutées. Typiquement, fail2ban cherche des tentatives répétées de connexions infructueuses dans les fichiers journaux et procède à un bannissement en ajoutant une règle au pare-feu iptables pour bannir l'adresse IP de la source.

Pour fonctionner, fail2ban a besoin de 2 services, rsyslog qui permet d'obtenir des logs et iptables qui permettra de bannir une IP dès qu'une tentative d'intrusions sera détectée.

Nous allons donc installer ses 2 services :

« apt install rsyslog »

```
root@debian:~# apt install rsyslog
rsyslog: service not installed
root@debian:~# apt install rsyslog
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
rsyslog est déjà la version la plus récente (8.2002-0-2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:~#
```

apt intall iptables

```
3 [40,6 KB]
Reception de :4 http://deb.debian.org/debian bullseye/main amd64 iptables amd64 1.0.7-1 [382 KB]
rsyslog est déjà la version la plus récente (8.2002-0-2).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:~#
```

Maintenant que rsyslog et iptables sont installés nous pouvons installer fail2ban « apt install fail2ban »

```
root@debian:~# apt install fail2ban
+3 [40,6 kB]
Réception de :4 http://deb.debian.org/debian bullseye/main amd64 iptables amd64 1.8.7-1 [382 kB]
472 ko réceptionnés en 0s (1 238 ko/s)
Sélection du paquet libip6tc2:amd64 précédemment désélectionné.
(Lecture de la base de données... 33660 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../libip6tc2_1.8.7-1_amd64.deb ...
Dépaquetage de libip6tc2:amd64 (1.8.7-1) ...
Sélection du paquet libnftnl:amd64 précédemment désélectionné.
Préparation du dépaquetage de .../libnftnl_1.0.1-3+b1_amd64.deb ...
Dépaquetage de libnftnl:amd64 (1.0.1-3+b1) ...
Sélection du paquet libnetfilter-conntrack3:amd64 précédemment désélectionné.
Préparation du dépaquetage de .../libnetfilter-conntrack3_1.0.8-3_amd64.deb ...
Dépaquetage de libnetfilter-conntrack3:amd64 (1.0.8-3) ...
Sélection du paquet iptables précédemment désélectionné.
Préparation du dépaquetage de .../iptables_1.8.7-1_amd64.deb ...
Dépaquetage de iptables (1.8.7-1) ...
Paramétrage de libip6tc2:amd64 (1.8.7-1) ...
Paramétrage de libnftnl:amd64 (1.0.1-3+b1) ...
Paramétrage de libnetfilter-conntrack3:amd64 (1.0.8-3) ...
Paramétrage de iptables (1.8.7-1) ...
update-alternatives: utilisation de « /usr/sbin/iptables-legacy » pour fournir « /usr/sbin/iptables » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-legacy » pour fournir « /usr/sbin/ip6tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/iptables-nft » pour fournir « /usr/sbin/iptables » (iptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ip6tables-nft » pour fournir « /usr/sbin/ip6tables » (ip6tables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/arptables-nft » pour fournir « /usr/sbin/arptables » (arptables) en mode automatique
update-alternatives: utilisation de « /usr/sbin/ebtables-nft » pour fournir « /usr/sbin/ebtables » (ebtables) en mode automatique
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.31-13+deb11u2) ...
root@debian:~# apt install fail2ban
```

Fail2ban est maintenant installé et actif nous pouvons vérifier cela avec « service fail2ban status »

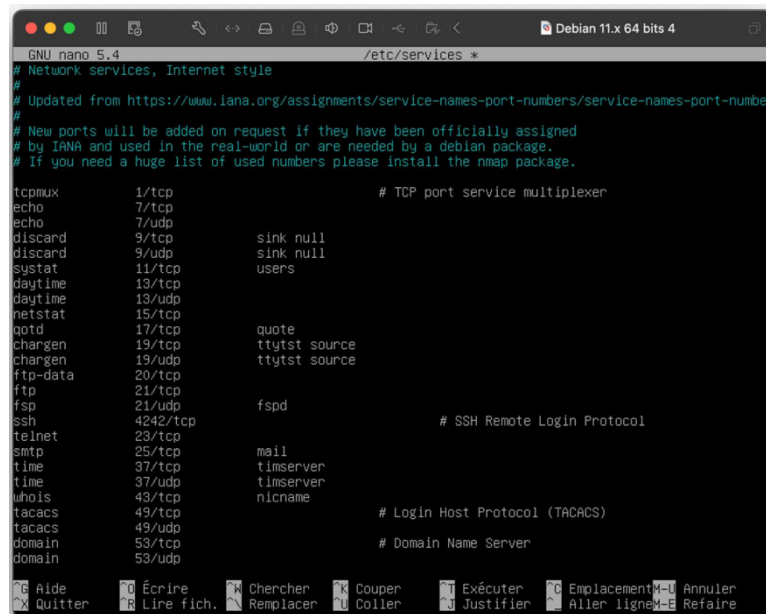
```
root@debian:~# apt install fail2ban
Sélection du paquet fail2ban précédemment désélectionné.
(Lecture de la base de données... 33880 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../fail2ban_0.11.2-2_all.deb ...
Dépaquetage de fail2ban (0.11.2-2) ...
Sélection du paquet python3-pyinotify précédemment désélectionné.
Préparation du dépaquetage de .../python3-pyinotify_0.9.6-1.3_all.deb ...
Dépaquetage de python3-pyinotify (0.9.6-1.3) ...
Sélection du paquet python3-systemd précédemment désélectionné.
Préparation du dépaquetage de .../python3-systemd_234-3+b4_amd64.deb ...
Dépaquetage de python3-systemd (234-3+b4) ...
Sélection du paquet whois précédemment désélectionné.
Préparation du dépaquetage de .../whois_5.5.10_amd64.deb ...
Dépaquetage de whois (5.5.10) ...
Paramétrage de whois (5.5.10) ...
Paramétrage de fail2ban (0.11.2-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/fail2ban.service.
Paramétrage de python3-pyinotify (0.9.6-1.3) ...
Paramétrage de python3-systemd (234-3+b4) ...
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...
root@debian:~# service fail2ban status
fail2ban.service - Fail2Ban Service
Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2022-03-16 19:40:13 CET; 7s ago
Docs: man:fail2ban(1)
Process: 1759 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)
Main PID: 1760 (fail2ban-server)
Tasks: 5 (limit: 2301)
Memory: 17.6M
CPU: 131ms
CGroup: /system.slice/fail2ban.service
└─1760 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

mars 16 19:40:13 debian systemd[1]: Starting Fail2Ban Service...
mars 16 19:40:13 debian systemd[1]: Started Fail2Ban Service.
mars 16 19:40:13 debian fail2ban-server[1760]: Server ready
root@debian:~#
```

Configuration de fail2ban :

Tout d'abord notre objectif est le suivant : après 3 tentatives échouées de connexion en SSH sur notre serveur en moins de 3 minutes l'IP source de la requête sera bannie par iptables pendant 30 minutes et ne pourras donc plus faire de requêtes SSH pendant cette durée.

Tout d'abord nous devons changer le port sur lequel fail2ban va écouter : « nano /etc/service » nous remplaçons dans la ligne ssh 22 (port par défaut d'SSH) par 4242 (le port que nous avons défini).



```
GNU nano 5.4 /etc/services *
# Network services, Internet style
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers
#
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux      1/tcp                # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp                sink null
discard     9/udp                sink null
systat      11/tcp               users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
gotd        17/tcp               quote
chargen     19/tcp               ttytst source
chargen     19/udp                ttytst source
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp                fspd
ssh         4242/tcp              # SSH Remote Login Protocol
telnet     23/tcp
smtp       25/tcp                mail
time       37/tcp                timserver
time       37/udp                timserver
whois      43/tcp                nickname
tacacs     49/tcp                # Login Host Protocol (TACACS)
tacacs     49/udp
domain     53/tcp                # Domain Name Server
domain     53/udp
```

Nous pouvons voir dans les logs dans nano /var/log/auth.log que fail2ban écoute maintenant sur le port 4242

```
Mar 16 19:07:04 debian sshd[1281]: Server listening on 0.0.0.0 port 4242.
Mar 16 19:07:04 debian sshd[1281]: Server listening on :: port 4242.
```

Nous devons maintenant configurer ce qui est appelé la « jail », qui correspond au fichier de configuration de fail2ban, elle se trouve dans, etc/fail2ban/jail.conf

Nous souhaitons qu'après 3 requêtes de connexion SSH échouée en moins de 3 minutes l'IP source soit bannie pour une durée de 30minutes.

Pour cela nous allons donc modifier le fichier jail.conf avec les paramètres suivant :

```
# "bantime" is the number of seconds that a host is banned.
bantime = 30m

# 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 = 3
```

« Bantime » correspond au temps de bannissement d'une IP après qu'elle ait été déclarée comme tentative d'intrusions. Dans notre cas, 30 minutes.

« Findtime » l'IP source est bannie si elle échoue le nombre maximum de connexions SSH définies dans « maxretry » ici 3 essaie en moins du temps défini dans « findtime » ici 3 minutes.

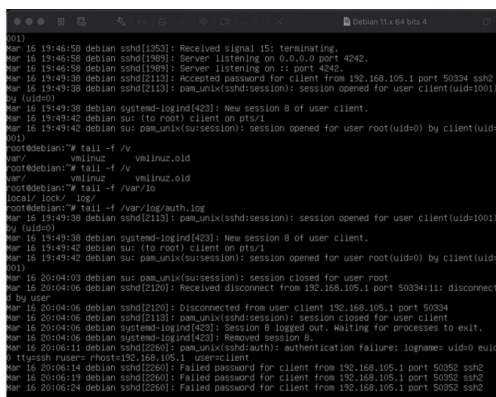
Nous sauvegardons la configuration et redémarrons le service fail2ban « service fail2ban restart

Test :

Pour tester notre solution, nous allons simuler une tentative d'intrusions, nous allons échouer volontairement 3 fois le mot de passe lors de notre connexion SSH et vérifier via les logs que la connexion est bien enregistrée puis bannie par fail2ban une fois nos 3 tentatives échouées en moins de 3 minutes.


Sur notre serveur nous entrons cette commande qui permet de voir les logs en direct et nous commençons notre simulation d'intrusions

```
tail -f /var/log/auth/.log
```



```
root@debian:~# tail -f /v
root@debian:~# tail -f /v
root@debian:~# tail -f /v
root@debian:~# tail -f /var/lo
local/ lock/ log/
root@debian:~# tail -f /var/log/auth.log
Mar 16 19:46:58 debian sshd[1953]: Received signal 15; terminating.
Mar 16 19:46:58 debian sshd[1969]: Server listening on 0.0.0.0 port 4242.
Mar 16 19:46:58 debian sshd[1969]: Server listening on :: port 4242.
Mar 16 19:49:38 debian sshd[2113]: Accepted password for client from 192.168.105.1 port 50394 sshd
Mar 16 19:49:38 debian sshd[2113]: pam_unix(sshd:session): session opened for user client(uid=1001)
by (uid=0)
Mar 16 19:49:38 debian systemd-logind[429]: New session 0 of user client.
Mar 16 19:49:42 debian su: (to root) client on pts/1
Mar 16 19:49:42 debian su: pam_unix(su:session): session opened for user root(uid=0) by client(uid=
1001)
root@debian:~# tail -f /v
root@debian:~# tail -f /v
root@debian:~# tail -f /v
root@debian:~# tail -f /var/lo
local/ lock/ log/
root@debian:~# tail -f /var/log/auth.log
Mar 16 19:49:38 debian sshd[2113]: pam_unix(sshd:session): session opened for user client(uid=1001)
by (uid=0)
Mar 16 19:49:38 debian systemd-logind[429]: New session 0 of user client.
Mar 16 19:49:42 debian su: (to root) client on pts/1
Mar 16 19:49:42 debian su: pam_unix(su:session): session opened for user root(uid=0) by client(uid=
1001)
Mar 16 20:04:03 debian su: pam_unix(su:session): session closed for user root
Mar 16 20:04:06 debian sshd[2120]: Received disconnect from 192.168.105.1 port 50394:11: disconnect
ed by user
Mar 16 20:04:06 debian sshd[2120]: Disconnected from user client 192.168.105.1 port 50394
Mar 16 20:04:06 debian sshd[2113]: pam_unix(sshd:session): session closed for user client
Mar 16 20:04:06 debian systemd-logind[429]: Session 0 logged out. Waiting for processes to exit.
Mar 16 20:04:06 debian systemd-logind[429]: Removed session 0.
Mar 16 20:06:11 debian sshd[2260]: pam_unix(sshd:auth): authentication failure; logname: uid=0; exit
d: ttysah ruser=rhost=192.168.105.1 user=client
Mar 16 20:06:14 debian sshd[2260]: Failed password for client from 192.168.105.1 port 50352 sshd
Mar 16 20:06:19 debian sshd[2260]: Failed password for client from 192.168.105.1 port 50352 sshd
Mar 16 20:06:24 debian sshd[2260]: Failed password for client from 192.168.105.1 port 50352 sshd
```

Nous pouvons voir que nos tentatives de connexion sont bien enregistrées. Nous pouvons voir qu'au bout de 3 tentatives nous ne pouvons plus envoyer de requêtes SSH au serveur, notre IP est bien bannie, pour voir les IP bannies nous pouvons faire la commande « iptables -L », nous pouvons voir que notre IP (192.168.105.1) a bien été bannie suite à notre tentative d'intrusions.



```
root@debian:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            multiport dports ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain f2b-sshd (1 references)
target     prot opt source                destination            reject-with icmp-port-unreachable
RETURN    all  --  anywhere             anywhere
```

Test :

La mise en place d'un serveur SSH et la sécurisation de la connexion à celui-ci m'ont permis d'approfondir ma connaissance en termes de sécurité des services informatiques, en effet la mise en place de ce serveur SSH et toutes mes recherches autour de cette mission m'ont permis d'en apprendre plus sur les enjeux liés à la cyber sécurité et les solutions existantes contre les cyber attaques dont le nombre ne cesse d'augmenter au fil du temps.

Les connaissances en termes de cyber sécurité sont très importantes dans mon activité, les besoins dans ce domaine sont de plus en plus importants et la mise en place de solution de sécurité informatique est aujourd'hui indispensable pour n'avoir aucun problème lié à la sécurité