Sécurisation d'un serveur

PERIER--PICARD VALENTIN

Table des matières

Sécurisation des connexions au serveurs	2
Comptes et utilisateurs	2
Changement de port	2
Sécurisation des entré et sortie	3
Fermons tous les ports.....	3
Fail2ban une solution automatique	4

Sécurisation des connexions au serveurs

Comptes et utilisateurs

Pour empêcher tout connexion de Brute force sur root nous allons désactiver la connexion route en ssh.

Nous allons éditer le fichier `/etc/passwd` et modifier le compte root pour qu'il exécute `/usr/sbin/nologin` au lieu de `bin/bash`

```
GNU nano 7.2 /etc/passwd
root:x:0:0:root:/root:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534:/:nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:usr/sbin/nologin
messagebus:x:100:107:/:nonexistent:/usr/sbin/nologin
avahi-autoipd:x:101:109:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
sshd:x:102:65534:/:run/sshd:/usr/sbin/nologin
valete:x:1000:1000:valete,,,:/home/valete:/bin/bash
```

Nous allons créer un nouveau compte et l'ajouter dans le groupe sudo

Changement de port

Nous allons changer le port ssh afin d'éviter qu'il script kiddie puisse brute force le serveur avec un scripte trouver sur internet.

Il suffit d'aller dans le fichier config de sshd `/etc/ssh/sshd_config` et d'y modifier le port :

```
GNU nano 7.2
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

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

Port 3456
#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
```

C'est ici aussi qu'on peut modifier le serveur ssh pour qu'il n'écoute les demandes de connexion seulement pour un ip local, nous n'allons pas le faire maintenant.

Sécurisation des entré et sortie

Fermons tous les ports

Nous allons utiliser iptables pour fermer tous les ports et n'autoriser que le ssh sur le nouveau port.

```
sudo -A INPUT -i lo -j ACCEPT
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A OUTPUT -o lo -j ACCEPT
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A INPUT -p tcp -m state --state NEW --dport 3456 -j ACCEPT
```

Nous allons aussi faire une règle pour limiter le nombre de nouvelle connexion ssh d'une même ip et bloqué l'utilisateur 60s en cas de spam.

```
sudo iptables -I INPUT -p tcp --dport 3456 -m state --state NEW -m recent --update --seconds 60 --hitcount 4 -j DROP
```

```
valet@nessus:~$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1 DROP tcp -- anywhere anywhere tcp dpt:3456 state NEW recent: UPDATE seconds: 60 hit_count: 4 name: DEFAULT side: source mask: 255.255.255.255
2 ACCEPT all -- anywhere anywhere
3 ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED
4 ACCEPT tcp -- anywhere anywhere state NEW tcp dpt:3456

Chain FORWARD (policy ACCEPT)
num target prot opt source destination

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
1 ACCEPT all -- anywhere anywhere
```

Nous avons bien tous nos règles nous pouvons les sauvegarder.

```
sudo iptables-save
```

Nous avons maintenant une sécurité basique sur notre serveur.

Nous pouvons si besoin ouvrir le port 443 pour le web et y mettre la même règle que pour le ssh.

Fail2ban une solution automatique

Pour approfondir notre sécurité nous allons utiliser Fail2ban.

L'installation est très rapide et la configuration est très clair.

Un simple sudo apt update et sudo apt install fail2ban pour l'installer.

```
valet@nessus:~$ sudo systemctl status fail2ban.service
* fail2ban.service - Fail2Ban Service
   Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-05-14 15:21:21 CEST; 1s ago
     Docs: man:fail2ban(1)
   Main PID: 1296 (fail2ban-server)
    Tasks: 5 (limit: 9472)
   Memory: 18.5M
      CPU: 590ms
   CGroup: /system.slice/fail2ban.service
           └─1296 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

May 14 15:21:21 nessus systemd[1]: Started fail2ban.service - Fail2Ban Service.
May 14 15:21:22 nessus fail2ban-server[1296]: 2024-05-14 15:21:22,085 fail2ban.configreader [1296]: WARNING 'allowipv6' not defined in 'Definition'. Using default one: 'auto'
May 14 15:21:22 nessus fail2ban-server[1296]: Server ready
```

Fail2ban est bien démarré et fonctionne normalement.

Nous allons modifier le temps de ban de 1h à 10min et le nombre d'essai max a 5.

Pour modifier nous allons ajouter la nouvelle configuration au fichier /etc/fail2ban/jail.local

```
GNU nano 7.2
[sshd]
backend=systemd
enabled=true
[DEFAULT]
findtime = 10m
maxretry = 5
```

Nous pouvons redémarrer le service Fail2ban :

```
sudo systemctl restart fail2ban.service
```

Faisons un test de ban

Nous allons nous connecter à la machine avec un mauvais compte

```
ssh 192.168.56.110 -l efsgdfg -p 3456
```

Après plein d'essai :

```
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
sdfsdf
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
sfsfsf
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
sdfsdfsdfPermission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
sdfsdfsdfPermission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
efsgdfg@192.168.56.110: Permission denied (publickey,password).
valete@Valetе-Debian12-A1:~$ ssh 192.168.56.110 -l efsgdfg -p 3456
efsgdfg@192.168.56.110's password:
Permission denied, please try again.
efsgdfg@192.168.56.110's password:
```

```
valete@nessus:~$ sudo fail2ban-client status
Status
|- Number of jail:      1
- Jail list:    sshd
valete@nessus:~$ sudo fail2ban-client get sshd banned
['192.168.56.102']
```

On voit bien l'utilisateur banni temporairement du serveur