4 - Allez plus loin

Même si la protection ultime n’existe pas, il est toujours possible de protéger encore plus son serveur. Cependant plus on ajoute de protections plus on ajoute de contraintes. Il est donc essentiel d'évaluer votre besoin et de ne pas surcharger votre machine de sécurités inutiles.

Nous verrons ici:

- Garder son serveur à jour contre les failles de sécurité avec cron-apt

- RkHunter pour détecter les backdoors et rootkits ainsi que d'autres problèmes de sécurité.

Et une analyse du traffic en temps réel pour prévenir d'attaques:

- Portsentry, une solution pour se protéger des scans de ports. (Snort peut être une alternative mais ne sera pas traité ici)

Garder son serveur à jour avec cron-apt

Les logiciels de plus en plus complexent comportent parfois de nombreuses failles de sécurités qui ne seront découvertes qu'avec le temps. C'est pourquoi mettre son système à jour régulièrement est important. Mais comment faire si une faille de sécurité est découverte alors que vous êtes en vacances ? Malade ? Ou tout, simplement, n'avez pas de quoi accéder au serveur ?
L'avantage d'un serveur, c'est que tout peut être automatisé ! Il peut donc se mettre à jour tout seul ! Nous allons voir comment avec cron-apt. Il existe également d'autres solutions.

Installation de cron-apt

```
apt install cron-apt
```

Pour ne pas tout casser lors de certaines mise à jour on souhaite ne faire que les mises à jour de sécurité !
On crée un nouveau fichier source en redirigeant la sortie de `grep`:

```
grep security /etc/apt/sources.list > /etc/apt/security.sources.list
```

Configurons ensuite cron-apt:

```
nano /etc/cron-apt/config
```

Puis on y colle à la suite:

```
APTCOMMAND=/usr/bin/apt-get
# Le path vers le fichier source que l'on viens de créer
OPTIONS="-o quiet=1 -o Dir::Etc::SourceList=/etc/apt/security.sources.list"
# L'email pour avertir de la mise à jour MAILTO="mon_email@mail.com" ou ici root
# en accord avec l'alias que nous avons créé précédemment
MAILTO="root"
# Quand envoyer l'email au sujet des résultats de cron-apt:
# Value: error   (send mail on error runs)
#  upgrade (when packages are upgraded)
#  changes (mail when change in output from an action)
#  output  (send mail when output is generated)
```

```
#        always  (always send mail)
#                (else never send mail)
MAILON="upgrade"
```

On vérifie que la ligne `dist-upgrade -d -y -o APT::Get::Show-Upgraded=true` est présente:

```
nano /etc/cron-apt/action.d/3-download
```

Puis on la modifie en retirant le flag -d signifiant Download only. Si elle n'est pas présente on l'ajoute à la fin de notre fichier.

```
# Vérifier que la ligne est bien décommentée et présente sinon la rajouter. Sauvegarder puis quitter
dist-upgrade -y -o APT::Get::Show-Upgraded=true
```

Et voilà c'est fait ! `cron-apt` se lance par défaut toute les nuits à 4h du matin. Il est possible de modifier cela en éditant le fichier `/etc/cron.d/cron-apt`.

Voilà ! Votre serveur se mettra à jour automatiquement sur le point de vue de la sécurité. Cependant cela ne dispense pas de ne jamais vérifier son bon fonctionnement.

Rkhunter (Optionnel)

Rkhunter peut détecter les répertoires généralement utilisés par les rootkits. Les permissions anormales, les fichiers cachés, les chaînes suspectes dans le kernel et peut effectuer des tests spécifiques à Linux. Cela se fait en comparant les hashs de vos fichiers avec des hashs de virus et logiciels connus.

Installation de RKHunter

```
apt install rkhunter -y
```

Chercher ensuite la ligne ou il y a `WEB_CMD="/bin/false"` vous pouvez utiliser Ctrl+W sur nano. Une fois la ligne trouvée commentez la.

```
# WEB_CMD="/bin/false"
```

Puis la ligne ou il y a écris `UPDATE_MIRRORS` et mettez sa valeur à 1. Cela spécifie que le fichier mirroir doit être vérifié pour les mise à jours lors d'une update.

```
UPDATE_MIRRORS=1
```

Puis la ligne `MIRRORS_MODE` et mettre sa valeur à 0. Cela permet de spécifier à rkhunter quel mirroir utiliser lors d'une update.

```
MIRRORS_MODE=0
```

Une fois l'étape précédente terminé et enregistré on va rajouter les notifications par mail:
On ouvre le fichier situé sous `/etc/default/rkhunter` avec notre éditeur préféré:

```
nano /etc/default/rkhunter
```

Puis on modifie si besoin les variables suivantes:

```
# Pour effectuer une vérification chaque jour
CRON_DAILY_RUN="yes"
# L'adresse sur laquelle on veut recevoir les emails, ici celle associé à root
REPORT_EMAIL="root"
```

Vérifions que notre fichier de configuration est correct avec la commande:

```
rkhunter -C
```

Pour mettre à jour rkhunter vous pouvez ensuite utiliser la commande:

```
rkhunter --update
```

Puis pour vérifier le système local

```
rkhunter --check
```

Rkhunter peut avoir **de faux positifs** suite à une mise à jour par exemple dans ce cas, il faut mettre la base d'empreintes à jour avec la commande :

```
rkhunter --propupd
```

Ce sera tout pour RKhunter vous pouvez également voir les logs sous `/var/log/rkhunter.log` :)

Portsentry contre les scans de ports

Chaque minute de nombreuses tentatives d'intrusion sur des serveurs sont perpétrés par des pirates. Pour trouver une cible et avant de l'attaquer ils vont passer par une phase de reconnaissance. Certaines de ces phases sont parfois automatisées par des machines zombies. Le scan de port à l'aide de logiciel comme nmap permet de détecter les ports ouverts amenant à de potentielle exploits. Pour prévenir de tels scans rien de mieux que d'analyser le traffic vers notre serveur à l'aide de portsentry !

Nb: Portsentry ne bloque rien par défaut il se contente de logger les scans de votre serveur. Il faut le configurer, ce que l'on va tout de suite faire en commençant par white-lister certaines IP

Installation de portsentry

```
apt-get install portsentry
```

Dans cet exemple nous allons whitelist le range d'IP de Google et les nôtres :

```
nano /etc/portsentry/portsentry.ignore.static
```

Puis ajouter dans le fichier

```
# IP du serveur
x.x.x.x
# Votre IP maison si fixe par sécurité
x.x.x.x
# Plage d'IP Google
66.249.64.0/19
```

Il nous faut ensuite **modifier le fichier de configuration** pour que portsentry puisse bloquer des IPs ce qu'il ne fait pas par défaut.

```
nano /etc/portsentry/portsentry.conf
```

A) On commence par modifier les variables suivantes:

```
##################
# Ignore Options #
##################

# 0 = Do not block UDP/TCP scans.
# 1 = Block UDP/TCP scans.
# 2 = Run external command only (KILL_RUN_CMD)

BLOCK_UDP="1"
BLOCK_TCP="1"
```

B) On va modifier ensuite la section *Dropping routes*
Chercher et vérifier que la lignes suivante est bien décommentée:

```
# Newer versions of Linux support the reject flag now. This
# is cleaner than the above option.
KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
```

C) La section *TCP Wrappers*
Chercher et vérifier que les lignes suivantes sont bien décommentées:

```
###############
# TCP Wrappers#
###############

KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"
```

D) Et ajouter dans la partie *External Command* :

```
###################
# External Command#
###################
KILL_RUN_CMD_FIRST = "1"

KILL_RUN_CMD="/sbin/iptables -I INPUT -s $TARGET$ -j DROP && /sbin/iptables -I INPUT -s
$TARGET$ -m limit --limit 3/minute --limit-burst 5 -j LOG --log-level debug --log-prefix
'Portsentry: dropping: '"
```

Pour finir on change la variable dans la partie *Scan trigger value* :

```
SCAN_TRIGGER = "1"
```

Il vaut mieux utiliser le mode **atcp** et **audp** pour une détection automatique des ports utilisés, il faut donc éditer le fichier `/etc/default/portsentry` :

```
nano /etc/default/portsentry
```

Et on modifie:

```
# /etc/default/portsentry
#
# This file is read by /etc/init.d/portsentry. See the portsentry.8
# manpage for details.
#
# The options in this file refer to commandline arguments (all in lowercase)
# of portsentry. Use only one tcp and udp mode at a time.
#
TCP_MODE="atcp"
UDP_MODE="audp"
```

Puis on redémarre Portsentry et on lui permet de démarrer à chaque redémarrage de notre serveur:

```
systemctl restart portsentry
systemctl enable portsentry
```

Pour jeter un œil aux IPs bloqués.

```
cat /etc/hosts.deny
```

Pour avoir plus d'infos sur le port qui a déclenché le blocage + date/heure vous pouvez voir les différents fichiers dans le répertoire:

```
cd /var/lib/portsentry/
```

Utilitaires

Pour dé-bannir un user bloqué par erreur, cherchez son IP dans `/etc/hosts.deny` puis redémarrez portsentry. Si malgrès cela l'utilisateur reste bloqué vous pouvez tenter:

```
route del -host IP reject
```

Port Knocking

Cela permet de bloquer l'accès au réel port SSH à moins qu'une séquence spécifique de ports ne soit "knock". Ce n'est qu'à ce moment que les règles iptables permettront au port SSH d'être ouvert à l'adresse IP de celui qui aura knock cette séquence de port.

Pour utiliser UFW à la place des iptables allez directement à la fin ! (UFW)

1 - Installation des packages requis

Commençons par installer le service requis:

```
apt-get install knockd
```

Vous aurez peut être besoin d'installer le package `iptables-persistent`. Il permet de charger de manière automatique les règles `iptables` sauvegardées.

```
apt-get install iptables-persistent


# Save current IPv4 rules ? YES
# Save current IPv6 rules ? YES
```

2-Configuration avec IPTABLES

Avant de mettre en marche knockd nous allons modifier certains paramètres par défauts.
Editons le fichier de configuration:

```
nano /etc/knockd.conf
```

Modifiez ou supprimez ce qui s'y trouve afin de mettre les settings suivantes:

```
[options]
        UseSyslog


[openSSH]
        sequence    = 1555,8888,5555
        seq_timeout = 5
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
        cmd_timeout = 15
```

```
stop_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
```

- `sequence` correspond à la séquence de port à knock dans les 5 secondes.
- `seq_timeout` le temps imparti pour knock les ports
- `command` la commande exécuté pour ouvrir le port ssh à notre IP, Soyez sûr de bien remplacer 22 par votre port SSH.
- `tcpflags` Spécifie qu'il n'acceptera que des segments tcp
- `cmd_timeout` le temps avant que notre IP soit supprimé des iptables, ici, 15 secondes. Cela ne nous déconnectera pas de la session SSH une fois connectée mais préviendra de vous connecter sans knock les ports spécifiés.
- `stop_command` La commande pour supprimer l'accès au port SSH via votre adresse IP. Soyez sûr de bien remplacer 22 par votre port SSH.

Continuons en rajoutant une règle *iptables* afin que les personnes connectées en SSH ne soit pas déconnectées:

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Puis fermons l'accès au port SSH **(Remplacez 22 par le votre)**:

```
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

Démarrons le daemon `netfilter-persistent` associé avec iptables et faisons en sorte qu'il sauvegarde nos règles iptables:

```
systemctl start netfilter-persistent
netfilter-persistent save
netfilter-persistent reload
```

(`sudo netfilter-persistent save` à utiliser à chaque changement de votre firewall que vous voulez sauvegarder)

3 - Mise en route de Knockd

Continuez en allant modifier le fichier `/etc/default/knockd` et mettez `START_KNOCKD` égal à 1.
NB: Vous aurez peut être besoin de changer la command line options afin de mettre le nom de votre interface réseau. Vous pouvez le savoir en tapant "ip addr" par défaut eth1
Redémarrez ensuite knockd `systemctl start knockd`.

Pour vous connectez depuis linux il vous suffit d'installer knockd comme précédemment et d'utiliser la commande:

```
knock -v IP PORT1 PORT2 PORT3 ...
```

Afin d'ouvrir le port SSH. Sous windows vous trouverez de nombreux projets de port knocking disponible sur github facilement installable. Comme: `https://github.com/BetterWayElectronics/port-knocker/releases/tag/1.0` pour le moment (compilez le vous même, on sait jamais).

Pour voir de l'intérieur ce que cela donne une fois que vous avec knock votre port vous pouvez utiliser:

```
tail -f /var/log/syslog
```

(PLEASE READ ME !)- Knockd Bug Not Starting at Boot

Il est probable sur certaines versions de debian comme la 9 (ça m'ai arrivé sur la 11 également □□) que knockd ne redémarre pas au reboot ce qui peut être TRES problèmatique sur un serveur remote. Si cela arrive vous pourrez sûrement vous connecter via la console en ligne de votre host provider.

Pour éviter que cela ce produise vous pouvez réaliser ces étapes:

Identifiez si nous avons le problème:

```
systemctl is-enabled knockd.service
```

cela nous retournera alors `static` !

Pour fix vous allez ensuite ajouter une section [Install] dans le fichier `/lib/systemd/system/knockd.service`:

```
nano /lib/systemd/system/knockd.service
```

Puis rajoutez à la fin:

```
[Install]
WantedBy=multi-user.target
Alias=knockd.service
```

Sauvegardez et activez knockd au démarrage:

```
systemctl enable knockd.service
```

Si vous relancez la commande `systemctl is-enabled knockd.service` vous devriez cette fois avoir la réponse `enabled` !

Hop le tour est joué !

Solutions issu de: https://bugs.debian.org

(UFW) - Si vous utilisez UFW (Ignore this if you used iptables)

Si vous utilisez UFW rien de plus simple commencez par installer le service `knockd`

```
apt-get install knockd
```

Editez le fichier de configuration de la manière suivante:

```
nano /etc/knockd.conf
```

Modifiez ou supprimez ce qui s'y trouve afin de mettre les settings suivantes et remplacez le port 22 par votre port SSH pour la signification des settings remontez à la section **2-Configuration avec IPTABLES**:

```
[options]
        UseSyslog

[openSSH]
        sequence    = 1555,8888,5555
        seq_timeout = 5
        command     = ufw allow from %IP% to any port 22
        tcpflags    = syn
        cmd_timeout = 15
        stop_command = ufw delete allow from %IP% to any port 22
```

Supprimez toute règle déjà mise en place autorisant le traffic depuis votre port SSH. Ces dernières seront gérés directement par UFW.
Puis fermez votre port SSH *(Remplacer 22 par votre port SSH)*.

```
ufw insert 1 deny from any to any port 22
```

Continuez à partir de la section **3- Mise en route de Knockd** !

et voilà :) si jamais n'hésitez pas à me contacter par discord ou mail *(disponible sur mon site)* :)