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
```

Revision #10
Created 2021-03-20 12:06:57 UTC by Faces
Updated 2022-06-11 09:49:30 UTC by Faces