Firewall X Fail2Ban

Etape 1

Scriptez votre configuration pour la remettre rapidement d'aplomb à l'avenir !

Il faut installer tous les outils dans la VM :

NFTABLES à la place de IPTABLES : NFTABLES est déjà installé

apt install mariadb-server apache2 logwatch fail2ban proftpd ssh

enable et start proftpd et ssh

{anne – anne}

Ajouter une table pour y mettre les paramètres :

```
nft add table ip mon_filtre
```

```
root@ldap:~# nft add chain ip mon_filtre input { type filter hook input priority 0 \; }

root@ldap:~# nft add chain ip mon_filtre output { type filter hook output priority 0 \; }
```

Pour vérifier :

```
root@ldap:~# nft list table ip mon_filtre
```

```
root@ldap:~# nft list table mon_filtre
table ip mon_filtre {
        chain input {
                type filter hook input priority filter; policy accept;
        }

        chain output {
                type filter hook output priority filter; policy accept;
        }
}
```

Pour paramétrer les entrées et sorties en http, https, et ssh il faut agir sur les ports :

```
root@ldap:~# nft add rule mon_filtre input tcp dport 80 accept
root@ldap:~# nft add rule mon_filtre input tcp dport 443 accept
root@ldap:~# nft add rule mon_filtre input tcp dport 22 accept

root@ldap:~# nft add rule mon_filtre input drop
root@ldap:~# nft add rule mon_filtre output tcp sport 80 accept
root@ldap:~# nft add rule mon_filtre output tcp sport 443 accept
root@ldap:~# nft add rule mon_filtre output tcp sport 22 accept

root@ldap:~# nft add rule mon_filtre output drop
```

Il faut bien penser à toujours autoriser, puis bloquer le reste en finissant par *drop*

```
root@ldap:~# nft list table ip mon_filtre
table ip mon_filtre {
        chain input {
                type filter hook input priority filter; policy accept;
                tcp dport 80 accept
                tcp dport 443 accept
                tcp dport 22 accept
                drop
        }

        chain output {
                type filter hook output priority filter; policy accept;
                tcp dport 22 accept
                tcp dport 443 accept
                tcp dport 80 accept
                drop
        }
}
```

Etape 2

Pour comprendre ce qu'il s'est passé sur le serveur, il faut regarder les logs!

Et pour plus de simplicité il existe des systèmes configurables d'analyse de logs.

Vous allez installer "Logwatch".

Puis pour détecter les tentatives d'intrusion ou permettre à votre système de répondre à des successions de logs spécifiques, vous allez installer "fail2ban".

Étape 1 : installation et configuration Logwatch

```
root@ldap:~# apt-get install logwatch
```

Étape 2 : Backup

À partir des logs recueillis par logwatch, un tri est nécessaire pour ne pas se retrouver avec une multiplicité d'informations non pertinentes en terme de sécurité, encombrantes et inutiles.

Pas question de les oublier, mais de pouvoir les rediriger vers des archives résilientes

que vous stockerez dans un répertoire de votre machine Host.

Il est impératif de recueillir les tentatives de connexion une fois par jour.

Archivez les logs avec Rsync avec un cron quotidien.

Aller dans

/etc/logwatch/logwatch.conf

Pour paramétrer le stockage des logs :

Mettre # devant les mails to et mails from

Paramétrer le fichier de stockage :

Enlever # à filename et indiquer le chemin

Filename =/tmp/logwatch

Configure crontab :

/etc/cron.daily/

Lancer logwatch pour vérifier

Aller dans le fichier nano logwatch dans /tmp

Configuration de .conf :

```
########################################################
# This was written and is maintained by:
#    Kirk Bauer <kirk@kaybee.org>
#
# Please send all comments, suggestions, bug reports,
#    etc, to kirk@kaybee.org.
#
########################################################

# NOTE:
#   All these options are the defaults if you run logwatch with no
#   command-line arguments.  You can override all of these on the
#   command-line.

# You can put comments anywhere you want to.  They are effective for the
# rest of the line.

# this is in the format of <name> = <value>.  Whitespace at the beginning
# and end of the lines is removed.  Whitespace before and after the = sign
# is removed.  Everything is case *insensitive*.

# Yes = True  = On  = 1
# No  = False = Off = 0
```

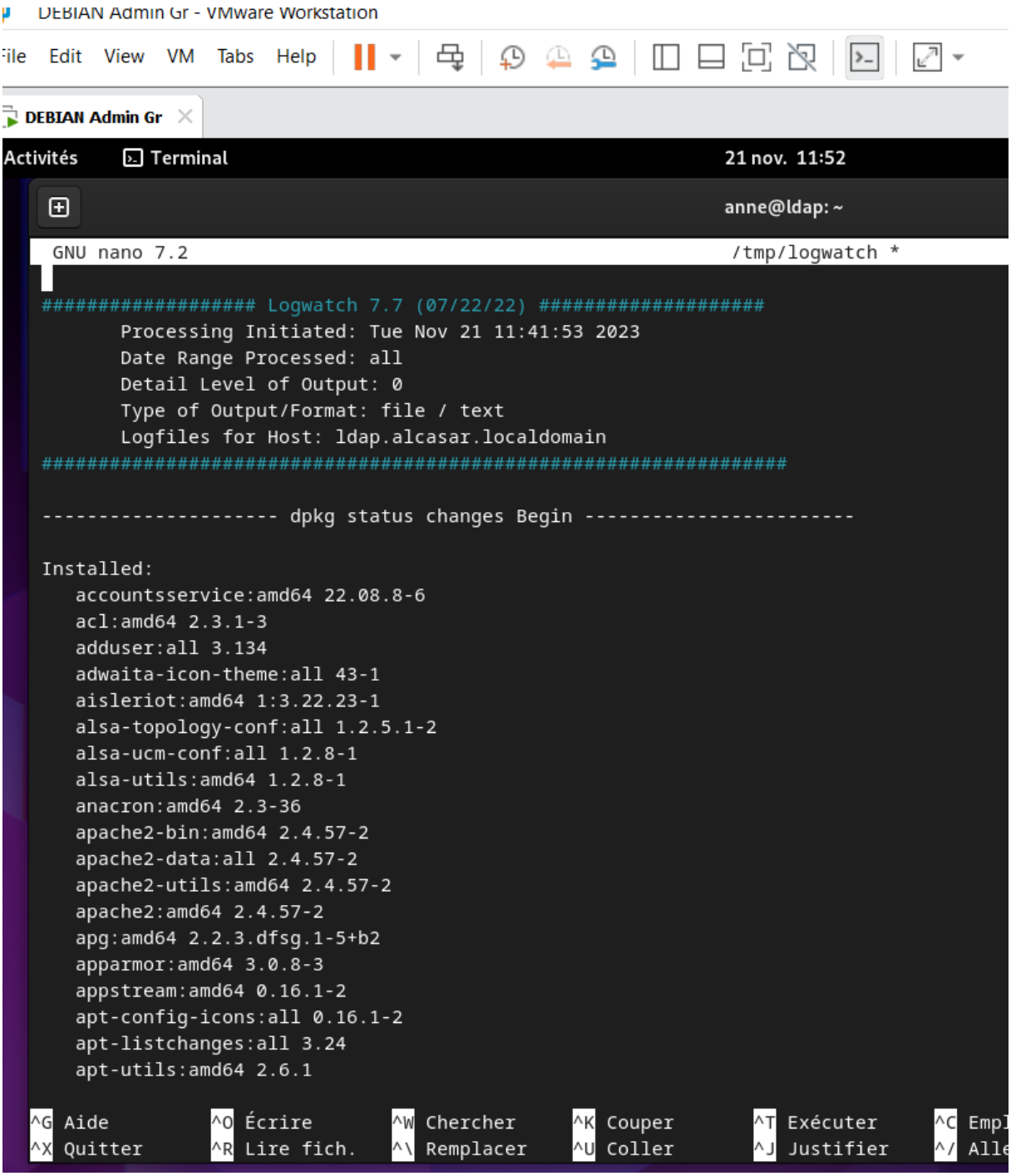
```
# You can override the default temp directory (/tmp) here
TmpDir = /tmp

# Output/Format Options
# By default Logwatch will print to stdout in text with no encoding.
# To make email Default set Output = mail to save to file set Output = file
Output = file
# To make Html the default formatting Format = html
Format = text
# To make Base64 [aka uuencode] Encode = base64
# Encode = none is the same as Encode = 8bit.
# You can also specify 'Encode = 7bit', but only if all text is ASCII only.
Encode = none

# Input Encoding
# Logwatch assumes that the input is in UTF-8 encoding.  Defining CharEncoding
# will use iconv to convert text to the UTF-8 encoding.  Set CharEncoding
# to an empty string to use the default current locale.  If set to a valid
# encoding, the input characters are converted to UTF-8, discarding any
# illegal characters.  Valid encodings are as used by the iconv program,
# and `iconv -l` lists valid character set encodings.
# Setting CharEncoding to UTF-8 simply discards illegal UTF-8 characters.
#CharEncoding = ""

# Default person to mail reports to.  Can be a local account or a
# complete email address.  Variable Output should be set to mail, or
# --output mail should be passed on command line to enable mail feature.
#MailTo = root
# WHen using option --multiemail, it is possible to specify a different
# email recipient per host processed.  For example, to send the report
# for hostname host1 to user@example.com, use:
#Mailto_host1 = user@example.com
# Multiple recipients can be specified by separating them with a space.

# Default person to mail reports from.  Can be a local account or a
# complete email address.
#MailFrom = Logwatch
# or displayed. Be sure to set Output = file also.
Filename = /tmp/logwatch

# Use archives?  If set to 'Yes', the archives of logfiles
# (i.e. /var/log/messages.1 or /var/log/messages.1.gz) will
# be searched in addition to the /var/log/messages file.
# This usually will not do much if your range is set to just
# 'Yesterday' or 'Today'... it is probably best used with Range = All
# By default this is now set to Yes. To turn off Archives uncomment this.
#Archives = No

# The default time range for the report...
# The current choices are All, Today, Yesterday
Range = All

# The default detail level for the report.
# This can either be Low, Med, High or a number.
# Low = 0
# Med = 5
# High = 10
```

```
Detail = Low


# The 'Service' option expects either the name of a filter
# (in /usr/share/logwatch/scripts/services/*) or 'All'.
# The default service(s) to report on.  This should be left as All for
# most people.
Service = All
# You can also disable certain services (when specifying all)
Service = "-zz-network"     # Prevents execution of zz-network service, which
# prints useful network configuration info.
Service = "-zz-sys"         # Prevents execution of zz-sys service, which
# prints useful system configuration info.
Service = "-eximstats"      # Prevents execution of eximstats service, which
# is a wrapper for the eximstats program.
# If you only cared about FTP messages, you could use these 2 lines
# instead of the above:
#Service = ftpd-messages   # Processes ftpd messages in /var/log/messages
#Service = ftpd-xferlog    # Processes ftpd messages in /var/log/xferlog
# Maybe you only wanted reports on PAM messages, then you would use:
#Service = pam_pwdb        # PAM_pwdb messages - usually quite a bit
#Service = pam             # General PAM messages... usually not many

# You can also choose to use the 'LogFile' option.  This will cause
# logwatch to only analyze that one logfile.. for example:
#LogFile = messages
# will process /var/log/messages.  This will run all the filters that
# process that logfile.  This option is probably not too useful to
# most people.  Setting 'Service' to 'All' above analyzes all LogFiles
# anyways...

#
# By default we assume that all Unix systems have sendmail or a sendmail-like MTA.
# The mailer code prints a header with To: From: and Subject:.
# At this point you can change the mailer to anything that can handle this output
# stream.
# TODO test variables in the mailer string to see if the To/From/Subject can be set
# From here with out breaking anything. This would allow mail/mailx/nail etc..... -mgt
mailer = "/usr/sbin/sendmail -t"

# With this option set to a comma separated list of hostnames, only log entries
# for these particular hosts will be processed.  This can allow a log host to
# process only its own logs, or Logwatch can be run once per a set of hosts
# included in the logfiles.
# Example: HostLimit = hosta,hostb,myhost
#
# The default is to report on all log entries, regardless of its source host.
# Note that some logfiles do not include host information and will not be
# influenced by this setting.
#
#HostLimit = myhost

# Default Log Directory
# All log-files are assumed to be given relative to the LogDir directory.
# Multiple LogDir statements are possible.  Additional configuration variables
# to set particular directories follow, so LogDir need not be set.
#LogDir = /var/log
```

```
#
# By default /var/adm is searched after LogDir.
#AppendVarAdmToLogDirs = 1
#
# By default /var/log is to be searched after LogDir and /var/adm/ .
#AppendVarLogToLogDirs = 1
#
# The current working directory can be searched after the above.  Not set by
# default.
#AppendCWDToLogDirs = 0

# vi: shiftwidth=3 tabstop=3 et
```

Résultat du logwatch :

Tu as bien de la chance car je ne peux pas filmer le fichier, la capture d'écran en film ne fonctionne pas. Une véritable œuvre d'art de 125478632 lignes. Voici un petit extrait 😊 😊 😊



Etape 3 Fail2ban

Il va falloir récupérer les logs et les traiter selon les règles suivantes :

- *Plus de 5 tentatives échouées de connexion à un compte utilisateur*
- *Plus de 10 requêtes d'une adresse IP à destination du serveur FTP*
- *Plus de 20 demandes de connexion au serveur FTP dans un intervalle de 5 minutes*

(même depuis des utilisateurs différents)

sont signes de TBF (Tentative de Brute Force)

Fail2ban est votre bon point de départ pour implémenter ces règles.

Dans Fail2ban il faut créer et paramétrer les fichiers de conf :

Filtres

/etc/fail2ban/filter.d/userauth.conf

```
[Definition]
failregex = authentication failure.*rhost=<HOST>.*
ignoreregex =
```

/etc/fail2ban/filter.d/ftpip.conf

```
[Definition]
failregex = \[<HOST>\].*FTP response: failed
ignoreregex =
```

Actions

/etc/fail2ban/action.d/userauth.conf

```
[Definition]
actionstart =
actionstop =
actioncheck =
actionban = iptables -I fail2ban-<name> 1 -s <ip> -j DROP
actionunban = iptables -D fail2ban-<name> -s <ip> -j DROP
```

/etc/fail2ban/action.d/ftpip.conf

```
[Definition]
actionstart =
actionstop =
actioncheck =
actionban = iptables -I fail2ban-<name> 1 -s <ip> -j DROP
actionunban = iptables -D fail2ban-<name> -s <ip> -j DROP
```

Configuration des règles

/etc/fail2ban/jail.local

```
[userauth]
enabled = true
filter = userauth
logpath = /var/log/auth.log
maxretry = 5
bantime = 3600

[ftpip]
enabled = true
filter = ftpip
logpath = /var/log/vsftpd.log
maxretry = 10
```

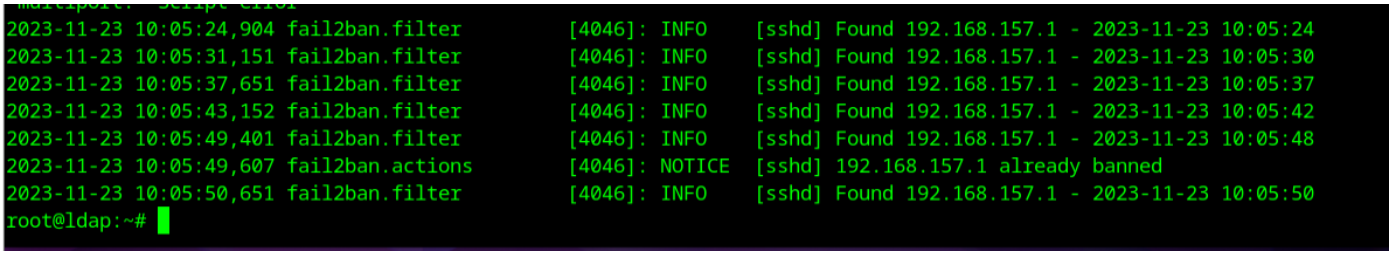
Redémarrer fail2ban

service fail2ban restart

Pour tester tout ça, on va lui donner des éléments « comme si » :

Ouvrir Putty,

Se connecter à la VM par PuttY et faire des erreurs de passwd



Etape 4 Automatisation

Maintenant que votre serveur de test est parfaitement configuré, il va falloir automatiser son déploiement. En cas de panne, cela permettra à votre système en opération de repartir en un clin d'oeil. La tâche n'est pas si simple que ça !
En effet, deux aspects à considérer:

- *la configuration du système telle que vous venez de la faire sur votre VM (installation/configuration des packages, configuration du firewall, gestion des logs/fail2ban, etc…)*
- *mais aussi les données courantes du serveur: comptes utilisateurs, sauvegarde de la base de données, des données du serveur Web, des données du serveur FTP, clefs SSH… bref…*

En partant des cas les plus simples de données pour les services ci-dessus, réfléchissez à un mécanisme de backup/restore qui permet de remettre le serveur en fonctionnement le plus rapidement possible.

Plusieurs possibilités s'offrent à vous ! Faites des recherches !

Hum hum,

Pour ce genre de tâche, il me semble qu'un outil comme Axway Automator soit approprié