FAIL2BAN RAPIDAMENTE

Rapida guida per mettere in produzione fail2ba. Non spieghero' cosa e' fail2ban , google ha gia' tutte le risposte. La nostra prima prova viene eseguita su centos 8.

A) Disinstallare il servizio firewalld
B) installare il servizio iptables e renderlo eseguibile all'avvio
C) installare epel release , col comando yum install epel-release
D) installare i seguenti pacchetti
   fail2ban-server-0.10.5-2.el8.noarch
   fail2ban-systemd-0.10.5-2.el8.noarch

fail2ban utilizza iptables per bloccare gli ip dai quali arriva l'attacco. Io mi sono servito del servizio iptables, non ho provato l'utilizzo del servizio firewalld.
Nel mio sistema l'installazione di fail2ban-all.noarch restituisce un errore, quindi mi sono accontentato dei due pacchetti sovracitati.
A questo punto abbiamo il pacchetto fail2ban installato, pronto per essere configurato. I files di configurazione si trovano tutti nella posizione /etc/fail2ban . Come primo comando eseguiamo una copia di un file di configurazione.
cp /etc/fail2ban/jail.con /etc/fail2ban/jail.local
quindi vediamo cosa ci restituisce il comando ls -l /etc/fail2ban

```
drwxr-xr-x. 2 root root  4096 15 feb 17.57 action.d
-rw-r--r--. 1 root root  2817 21 gen 05.11 fail2ban.conf
drwxr-xr-x. 2 root root  4096 21 gen 05.11 fail2ban.d
drwxr-xr-x. 3 root root  4096 15 feb 17.57 filter.d
-rw-r--r--. 1 root root 23681 16 feb 08.51 jail.conf
drwxr-xr-x. 2 root root  4096 15 feb 17.57 jail.d
-rw-r--r--. 1 root root 23712 16 feb 08.58 jail.local
-rw-r--r--. 1 root root  2827 21 gen 05.11 paths-common.conf
-rw-r--r--. 1 root root   930 21 gen 05.11 paths-fedora.conf
```

I files vengono letti ed applicati nel seguente ordine

/etc/fail2ban/jail.conf
/etc/fail2ban/jail.d/*.conf, in ordine alfabetico
/etc/fail2ban/jail.local
/etc/fail2ban/jail.d/*.local, in ordine alfabetico

Il motivo per cui eseguimo la copia di jail.conf e' che detto file potrebbe essere sovrascritto se aggiorniamo il sistema, mentre il file jail.local sicuramente non viene toccato dall'aggiornamento. jail.local sovrascrive jail.conf, quindi le modifiche dell'admin non vanno perse, mantenendo anche la compatibilita'. Per abilitare il controllo sui login in ssh , dobbiamo editare il file jail.local ed assicurarci che nella sezione apposita ci siano le seguenti direttive

```
[sshd]
enabled = yes
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and
details.
#mode   = normal
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

Questo e' gia' sufficiente a bloccare i tentativi di ingresso a forza bruta. Infatti nella sezione [default] , le direttive bantine e maxretry indicano al servizio il numero massimi di tentativi nel tempo prestabilito (direttiva

findtime) , oltrepassato il quale scatta il blocco dell'indirizzo ip.
Per avviare il servizio dobbiamo lanciare il comando systemctl start fail2ban.service. Per stoppare il servizio dobbiamo lanciare il comando systemctl stop fail2ban.service, mentre il comando systemctl status fail2ban.service ci dice se il servizio e' attivo, con qualche altra indicazione aggiuntiva. Ipotizziamo ora di volere mettere sotto controllo il servizio ftp, oggetto di attacco a forza bruta. Il server ftp in esecuzione e' vsftp.
Editiamo il file jail.local fino ad arrivare alla sezione vsftp , ed impostiamo le seguenti direttive

```
[vsftpd]
enabled = yes
# or overwrite it in jails.local to be
# logpath = %(syslog_authpriv)s
# if you want to rely on PAM failed login attempts
# vsftpd's failregex should match both of those formats
port     = ftp,ftp-data,ftps,ftps-data
logpath  = %(vsftpd_log)s
```

Questo non e' sufficiente a rendere operativo il controllo. Se accettiamo i parametri di default, ecco che il servizio fail2ban controlla il file /var/log/vsftp.log alla ricerca di login errati. In realta' il file che contiene tali informazioni e'
/var/log/secure. La maniera piu' semplice per rimediare a questo e' creare un link simbolico dal nome /var/log/vsftp.log che "punti" a /var/log/secure .
Cosi' facendo il servizio fail2ban riesce a leggere i tentativi di login, e riesce ad impostare il blocco.
Fail2ban puo’ intercettare anche attacchi destinati ad altri servizi. Un buon punto di partenza e’ analizzare la directory /etc/fail2ban/filter.d .
Qui vi sono i files di configurazione inerenti il servizio da controllare.
Interessante ed utili sono webmin , apache , phpmyadmin , mariadb .
Ecco alcuni esempi di configurazione del file /etc/fail2ban/jail.local

```
[apache-multiport]

enabled   = true
port      = http,https
filter    = apache-auth
logpath   = /var/log/apache*/*error.log
maxretry  = 6
```

Con queste direttive indichiamo a fail2ban di tenere sotto osservazione i files presenti in /var/log/apache*/*error , alla ricerca della stringa apache-auth, e di impostare il blocco dopo 6 tentativi errati di login
Dopo l'installazione di fail2ban , abbiamo a disposizione i seguenti comandi
fail2ban-client fail2ban-python fail2ban-regex fail2ban-server
Vediamoli unno ad uno, anche se in modo non troppo approfondito

fail2bac-client

testuale dalla pagina man: "configure and control the server" .
vediamo alcune opzioni (l’elenco non e’ completo)
-c /path/del/file = indica espressamente un determinato file di configurazione
-p /path/pid/file = indica espressamente il path del file di pid
-d = dump configuration, utile per il debug
-t = test , controlla la correttezza del file di configurazione
-i = modalita’ interattiva
-v = aumenta la verbosita’
-q = diminuise la verbosita’
-b = avvia il server sullo sfondo (default)
-f = avvia il server in primo piano
unban -all = elimina il blocco su tutti gli ip bloccati
unban INDIRIZZO_IP elimina il blocco su INDIRIZZO_IP
status = indica l’attuale stato del server

ping = controlla se il server e’ attivo

fail2ban-python

testuale dalla pagina man “ a helper for Fail2Ban to assure that the same Python is used “ . Le opzioni di fail2ban-python non sono significative per il mio scopo, quindi non le analizzo.

Fail2ban-regex

fail2ban legge i file di log, e dall’analisi risultante, blocca o sblocca eventuali indirizzi ip. Fail2ban-regex e’ uno strumento per testare le espressioni regolari di fail2ban. Non significativo per il mio scopo, quindi non lo analizzo.

Fail2ban-server

avvia il server fail2ban. Ecco alcune opzioni (l’elenco non e’ completo)
-c /percorso = percorso della directory contenente i files di configurazione
-p /percorso file pid = indica espressamente la posizione del file pid
-d = dump della configurazione. Utile per il debug
-t = verifica la correttezza del file di configurazione
-x = forza l’esecuzione del server , rimuove eventuali pid orfani

E’ importante anche leggere la pagina man di jail.conf, troppo lunga per essere riportata qui

FAIL2BAN CON DEBIAN 10

installare fail2ban con debian 10 e' davvero semplice. Il comando apt-get install fail2ban svolge tutto il lavoro. Installa il pacchetto ed anche lo script di avvio, locato in /etc/init.d. I files di configurazione si trovano in /etc/fail2ban, ed anche con debian vale la regola di fare una copia di /etc/fail2ban/fail.conf e dargli il nome di /etc/fail2ban/jail.local . Per ottenere il controllo dei login ssh , e' sufficiente editare il file /etc/fail2ban/jail.local e nella sezione apposita indicare le seguenti direttive

```
enabled = yes
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and
details.
#mode   = normal
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

le impostazioni di default ottengono gia' il risultato voluto. Interessante il comando fail2ban-testcase , che esegue un controllo, ma non ho capito cosa viene controllato.

FAIL2BAN CON SLACKWARE 14.2

Anche con slackware 14.2 (la migliore distro) non abbiamo brutte sorprese. il pacchetto da scaricare e' fail2ban-0.9.7-i586-1_slonly.txz . Si installa il pacchetto col solito comando installpkg e poi ci si reca nella posizione /etc/fail2ban ed eseguiamo la solita copia di jail.conf in jail.local.
Con slackware per mettere sotto controllo il servizio ssh , occorre una piccola accortezza. Il file jail.local contiene una imprecisione riguardo il file da mettere sotto osservazione per contare i login errati " ssh " .
Le direttive corrette sono le seguenti

```
[sshd]

enabled = yes
# To use more aggressive sshd filter (inclusive sshd-ddos failregex):
#filter = sshd-aggressive
port    = ssh
# logpath = %(sshd_log)s
logpath = /var/log/messages
backend = %(sshd_backend)s
```

E’ tutto. Prima dei vostri insulti vorrei ricordarvi che questi appunti non sono rivolti alla massa. Mi servono solo per evitare un’altra ora in google.