# Inhaltsverzeichnis

Das in Python geschriebene Tool '`fail2ban`' verfolgt das Ziel, Serverdienste gegen DoS Angriffe abzusichern. Es prüft Logdateien nach vordefinierten Mustern und sperrt bei wiederholtem fehlgeschlagenem Zugriff temporär die entsprechenden IP-Adressen. Dieser Artikel zeigt, wie Sie einen Debian-basierten Server mit fail2ban absichern. Die eingesetzte Version von fail2ban ist '`0.9.6-2`' unter '`Debian 9.1`'.

#### Problem

In der Logdatei „/var/log/auth.log“ treten mehrere fehlgeschlagene Loginversuche mit dem Protokoll SSH auf, die nicht von Ihnen stammen.

```
1. Feb 19 09:21:15 servername sshd[22796]: pam_unix(sshd:auth):
   authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   rhost=218.207.xx.xx  user=root
2. Feb 19 09:21:17 servername sshd[22796]: Failed password for root from
   218.207.xx.xx port 22 ssh2
```

#### Erklärung

* Der entfernte Benutzer hat (versehentlich) eine falsche Server IP verwendet und irrtümlicherweise versucht sich an Ihrem Server anzumelden. Die Anzahl der Loginversuche ist hier meistens gering. * Sie sind Opfer einer Brute Force Attacke, bei der automatisch ein Login mit Benutzer root und verschiedenen Passwörtern (z.b. aus sogenannten Wörterbuchdateien) versucht werden. Die Anzahl der Loginversuche ist hier erkennbar hoch.

#### Lösung

Sichern Sie Ihren SSH Login mit dem Tool fail2ban ab, verbieten Sie direkten Root Login oder melden Sie sich nur mit Public-Key-Verfahren an.

#### Was ist Fail2Ban

Fail2Ban ist ein in Python geschriebenes Programm, welches verschiedene Serverdienste gegen unbefugten Zugriff absichern kann. In dem Konfigurationsbeispiel unten, wird eine IP Adresse für 1 Stunde gesperrt, nachdem von dieser 4 fehlgeschlagene Anmeledeversuche für SSH stattgefunden haben.

#### Installation von Fail2Ban

```
1. sudo apt install fail2ban
```

**Konfiguration Fail2Ban**

Im Ordner `/etc/fail2ban/` finden Sie die globale Konfigurationsdatei `jail.conf`. Diese jedoch nicht bearbeiten, da sie bei jeder Paketaktualisierung überschrieben wird. Die eigene Konfiguration geschieht in der „jail.local“.

```
# To avoid merges during upgrades DO NOT MODIFY THIS FILE
# and rather provide your changes in /etc/fail2ban/jail.local>
```

Hierzu kopieren Sie die „jail.conf“ nach „jail.local“.

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Prüfen Sie die die Einstellungen zur lokalen IP Adresse Ihres Servers. Die Zeit, wie lange eine IP geblockt werden soll wird in unserem Beispiel auf eine Stunde erhöht und die Anzahl der Versuche, nach denen geblockt werden soll, wird auf 3 verringert. Diese Konfiguration ist in der folgenden Sektion der `jail.local` vorzunehmen:

```
[...]
[DEFAULT]

#
# MISCELLANEOUS OPTIONS
#

# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
# ban a host which matches an address in this list. Several addresses can be
# defined using space (and/or comma) separator.
ignoreip = 127.0.0.1/8

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime  = 3600

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime  = 600
```

```
25.
26. # "maxretry" is the number of failures before a host get banned.
27. maxretry = 3
28. [...]
```

Sie können die Parameter dann für einzelne Dienste (wie hier in dem Artikel der SSH Daemon) separat anpassen.

Ergänzen Sie nun weiter unten in der eigenen, vorher kopierten, Konfigurationsdatei `jail.local` im Abschnitt zum SSH Daemon die erforderlichen Parameter um ihn per fail2ban zu überwachen:

```
 1. [...]
 2. #
 3. # SSH servers
 4. #
 5.
 6. [sshd]
 7.
 8. enabled = true
 9. port    = ssh
10. filter    = sshd
11. logpath    = /var/log/auth.log
12. maxretry = 4
13. [...]
```

Starten Sie anschließend fail2ban neu, damit die Änderungen übernommen werden.

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