Preventing SSH Dictionary Attacks With DenyHosts (for openSUSE 10.1)

Version 1.0
Author: Falko Timme <ft(AT)falkotimme(DOT)com>
Last edited: 02/07/2006
Modified by: heathenx <heathenx(AT)gmail(DOT)com>
Last edited: 07/30/06

A note from heathenx

99% of this how to was written by Falko Timme and published on www.howtoforge.com. I have only changed to focus from Debian Sarge to Suse 10.1. I wanted to be able to reference this how to for a Suse system so I change only those things that related.

Start of the tutorial

In this HowTo I will show how to install and configure .DenyHosts. DenyHosts is a tool that observes login attempts to SSH, and if it finds failed login attempts again and again from the same IP address, DenyHosts blocks further login attempts from that IP address by putting it into /etc/hosts.deny. DenyHosts can be run by cron or as a daemon. In this tutorial I will run DenyHosts as a daemon.

From the DenyHosts web site:

"DenyHosts is a script intended to be run by Linux system administrators to help thwart ssh server attacks.

If you've ever looked at your ssh log (/var/log/messages on Suse 10.1) you may be alarmed to see how many hackers attempted to gain access to your server. Hopefully, none of them were successful (but then again, how would you know?). Wouldn't it be better to automatically prevent that attacker from continuing to gain entry into your system?

DenyHosts attempts to address the above... "

This tutorial is based on a SUSE 10.1 system (the free openSuse 10.1 to be exact), however, it should apply to other distributions with almost no modifications.

I want to say first that this is not the only way of setting up such a system. There are many ways of achieving this goal but this is the way I take. I do not issue any guarantee that this will work for you!

1 Installation

DenyHosts is written in Python, therefore we must install Python and also the Python development files first:

apt-get install python python-devel (my version at the time of this tutorial is python-2.4.2-18 and python-devel-2.4.2-18)

Then we download and install DenyHosts like this:

```
cd /tmp
wget http://mesh.dl.sourceforge.net/sourceforge/denyhosts/DenyHosts-2.0.tar.gz
tar xvfz DenyHosts-2.0.tar.gz
cd DenyHosts-2.0
python setup.py install
```

This installs DenyHosts to /usr/share/denyhosts.

2 Configuration

Now we have to create the DenyHosts configuration file /usr/share/denyhosts/denyhosts.cfg. We can use the sample configuration file /usr/share/denyhosts/denyhosts.cfg-dist for this:

```
cd /usr/share/denyhosts
cp denyhosts.cfg-dist denyhosts.cfg
```

Then we must edit denyhosts.cfg with our favourite editor such as kwrite, for example. Only change the section that I have here. Mine looks like this:

```
########################################################################
#
# SECURE_LOG: the log file that contains sshd logging info
# if you are not sure, grep "sshd:" /var/log/*
#
# The file to process can be overridden with the --file command line
# argument
#
# Redhat or Fedora Core:
#SECURE_LOG = /var/log/secure
#
# Mandrake, FreeBSD or OpenBSD:
#SECURE_LOG = /var/log/auth.log
#
# SuSE:
SECURE_LOG = /var/log/messages
#
########################################################################
```

and also...

```
########################################################################
#
# LOCK_FILE
#
# LOCK_FILE=/path/denyhosts
# If this file exists when DenyHosts is run, then DenyHosts will exit
# immediately.  Otherwise, this file will be created upon invocation
# and deleted upon exit.  This ensures that only one instance is
# running at a time.
#
# Redhat/Fedora:
#LOCK_FILE = /var/lock/subsys/denyhosts
#
# Debian
#LOCK_FILE = /var/run/denyhosts.pid
#
# Misc
LOCK_FILE = /tmp/denyhosts.lock
#
########################################################################
```

Make sure you set SECURE_LOG and LOCK_FILE to the correct values for your distribution! For SuSE 10.1, these are:

```
SECURE_LOG = /var/log/messages
LOCK_FILE = /tmp/denyhosts.lock
```

As we want to run DenyHosts as a daemon, we need the daemon control script /usr/share/denyhosts/daemon-control. Again, we can use the sample script /usr/share/denyhosts/daemon-control-dist to create the needed file:

```
cp daemon-control-dist daemon-control
```

Edit /usr/share/denyhosts/daemon-control and make sure you set the correct values for DENYHOSTS_BIN, DENYHOSTS_LOCK, and DENYHOSTS_CFG. For Suse 10.1, these are:

```
DENYHOSTS_BIN = "/usr/local/bin/denyhosts.py"
DENYHOSTS_LOCK = "/tmp/denyhosts.lock"
DENYHOSTS_CFG = "/usr/share/denyhosts/denyhosts.cfg"
```

Do not change anything else in /usr/share/denyhosts/daemon-control.

Next we have to make that file executable:

```
chown root daemon-control
chmod 700 daemon-control
```

Afterwards, we create the system bootup links for DenyHosts do that it is started automatically when the system is booted:

```
cd /etc/init.d
ln -s /usr/share/denyhosts/daemon-control denyhosts
```

Finally, we start DenyHosts:

/etc/init.d/denyhosts start (you can ctrl+esc under KDE to see that denyhosts.py is running)

DenyHosts logs to /var/log/denyhosts, if you are interested in the logs. The SSH daemon logs to /var/log/messages on Suse 10.1. You can watch both logs and try to log in with an invalid user or with a valid user and incorrect password, etc. via SSH and see what happens. After you have crossed the threshold of incorrect login attempts, the IP address from which you tried to connect should get listed in /etc/hosts.deny, like this:

```
########################################################################
# /etc/hosts.deny
# See 'man tcpd' and 'man 5 hosts_access' as well as /etc/hosts.allow
# for a detailed description.

http-rman : ALL EXCEPT LOCAL

sshd: 192.168.0.103
########################################################################
```

This means that the system with the IP address 192.168.0.103 cannot connect anymore using SSH.

You can specify if/when IP addresses are removed again from /etc/hosts.deny - have a look at the PURGE_DENY variable in /usr/share/denyhosts/denyhosts.cfg. You must start DenyHosts with the --purge option to make the PURGE_DENY variable effective, like this:

```
/etc/init.d/denyhosts start --purge
```

However, you can also remove IP addresses manually from there, and as soon as they have got removed, these IP addresses can try to log in again via SSH. I prefer the manual method because this forces me to keep an eye on things.

Links

- DenyHosts: http://denyhosts.sourceforge.net

Credits

- www.howtoforge.com: http://www.howtoforge.com/preventing_ssh_dictionary_attacks_with_denyhosts
- Author: http://www.falkotimme.com/, email: ft(AT)falkotimme(DOT)com