Internet Radio Linking Project (IRLP) Wiki

Keeping the Radio in Amateur Radio

These instructions will help you to make your IRLP Node a lot more secure! What we will do here is tell the IRLP node to lock out further attempts at access the Secure Shell (SSH) if someone or something tries to many times with the wrong password. So a bot is walking along and he decides to guess your password and after he guesses wrong say 4 times he is locked out from trying again for 30 minutes. This will prevent attackers from using millions of combinations to guess the password. So here goes…

THESE INSTRUCTIONS ARE NOT REAL PRETTY BUT I WILL CLEAN THEM UP A BIT LATER… I copied these instructions from another place on the internet but I have used these instructions on multiple occasions to help keep my CentOS installations secure. ~K0KAD

Step One—Install Fail2Ban

Because fail2ban is not available from CentOS, we should start by downloading the EPEL repository:

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
[http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm]
```

Follow up by installing fail2ban:

```
yum install fail2ban
```

Step Two—Copy the Configuration File

The default fail2ban configuration file is location at /etc/fail2ban/jail.conf. The configuration work should not be done in that file, however, and we should instead make a local copy of it.

cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local After the file is copied, you can make all of your changes within the new jail.local file. Many of possible services that may need protection are in the file already. Each is located in its own section, configured and turned off.

Step Three—Configure defaults in Jail.Local

Open up the the new fail2ban configuration file:

```
pico /etc/fail2ban/jail.local
```

The first section of defaults covers the basic rules that fail2ban will follow. If you want to set up more nuanced protection for your virtual private server, you can customize the details in each section.

You can see the default section below.

[DEFAULT]

# “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 separator. ignoreip = 127.0.0.1

# “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

# "maxretry" is the number of failures before a host get banned. maxretry = 3 Write your personal IP address into the ignoreip line. You can separate each address with a space. IgnoreIP allows you white list certain IP addresses and make sure that they are not locked out from your VPS. Including your address will guarantee that you do not accidentally ban yourself from your own virtual private server.

The next step is to decide on a bantime, the number of seconds that a host would be blocked from the server if they are found to be in violation of any of the rules. This is especially useful in the case of bots, that once banned, will simply move on to the next target. The default is set for 10 minutes—you may raise this to an hour (or higher) if you like.

Maxretry is the amount of incorrect login attempts that a host may have before they get banned for the length of the ban time.

Findtime refers to the amount of time that a host has to log in. The default setting is 10 minutes; this means that if a host attempts, and fails, to log in more than the maxretry number of times in the designated 10 minutes, they will be banned.

Step Four (Optional)

Configure the ssh-iptables Section in Jail.Local The SSH details section is just a little further down in the config, and it is already set up and turned on. Although you should not be required to make to make any changes within this section, you can find the details about each line below.

```
[ssh-iptables] enabled = true filter = sshd action = iptables[name=SSH, port=ssh, protocol=tcp] sendmail-whois[name=SSH, dest=root, sender=fail2ban@example.com] logpath = /var/log/secure maxretry = 5
```

Enabled simply refers to the fact that SSH protection is on. You can turn it off with the word "false".

The filter, set by default to sshd, refers to the config file containing the rules that fail2banuses to find matches. The name is a shortened version of the file extension. For example, sshd refers to the /etc/fail2ban/filter.d/sshd.conf.

Action describes the steps that fail2ban will take to ban a matching IP address. Just like the filter entry, each action refers to a file within the action.d directory. The default ban action, "iptables" can be found at /etc/fail2ban/action.d/iptables.conf .

In the "iptables" details, you can customize fail2ban further. For example, if you are using a non-standard port, you can change the port number within the brackets to match, making the line look more like this:

eg. iptables[name=SSH, port=30000, protocol=tcp] You can change the protocol from TCP to UDP in this line as well, depending on which one you want fail2ban to monitor.

log path refers to the log location that fail2ban will track.

The max retry line within the SSH section has the same definition as the default option. However, if you have enabled multiple services and want to have specific values for each one, you can set the new max retry amount for SSH here.

Step Five—Restart Fail2Ban

After making any changes to the fail2ban config, always be sure to restart Fail2Ban:

sudo service fail2ban restart You can see the rules that fail2ban puts in effect within the IP table:

iptables -L

lock_attempts_ssh.txt · Last modified: 2014/08/31 20:32 by k0kad