Published Friday, February 17th, 2012 by Linode

In the Getting Started guide, you learned how to deploy Linux, boot your Linode, and perform some basic system administration tasks. Now it's time to secure your Linode and protect it from unauthorized access. You'll learn how to implement a firewall, SSH key pair authentication, and an automatic blocking mechanism called *Fail2Ban*. By the time you reach the end of this guide, your Linode will be protected from attackers.

Adding a New User

In the Getting Started guide, we asked you to log in to your Linode as the `root` user, the most powerful user of all. The problem with logging in as `root` is that you can execute *any* command - even a command that could accidentally break your server. For this reason and others, we recommend creating another user account and using that at all times. After you log in with the new account, you'll still be able to execute superuser commands with the `sudo` command.

Here's how to add a new user:

1. Open a terminal window and log in to your Linode via SSH.
2. Create the user by entering the following command. Replace *example_user* with your desired username:

```
adduser example_user
```

3. Add the user to the *administer the system* (admin) group by entering the following command. Replace *example_user* with your username:

```
usermod -a -G sudo example_user
```

4. On Debian 7 installations, you will need to install sudo before logging in as the new user:

```
apt-get install sudo
```

5. Log out of your Linode as the `root` user by entering the following command:

```
logout
```

6. Log in to your Linode as the new user by entering the following command. Replace *example_user* with your username, and the example IP address with your Linode's IP address:

```
ssh example_user@123.456.78.90
```

Now you can administer your Linode with the new user account instead of `root`. When you need to execute superuser commands in the future, preface them with `sudo`. For example, later in this guide you'll execute `sudo iptables -L` while logged in with your new account. Nearly all superuser commands can be executed with `sudo`, and all commands executed with `sudo` will be logged to `/var/log/auth.log`.

Using SSH Key Pair Authentication

You've used password authentication to connect to your Linode via SSH, but there's more a secure method available: *key pair authentication*. In this section, you'll generate a public and private key pair using your desktop computer and then upload the public key to your Linode. SSH connections will be authenticated by matching the public key with the private key stored on your desktop computer - you won't need to type your account password. When combined with the steps outlined later in this guide that disable password authentication entirely, key pair authentication can protect against brute-force password cracking attacks.

Here's how to use SSH key pair autentication to connect to your Linode:

1. Generate the SSH keys on a desktop computer running Linux or Mac OS X by entering the following command in a terminal window *on your desktop computer*. PuTTY users can generate the SSH keys by following the instructions in our PuTTY guide.

```
ssh-keygen
```

2. The *SSH keygen* utility appears. Follow the on-screen instructions to create the SSH keys on your desktop computer. To use key pair authentication without a passphrase, press Enter when prompted for a passphrase.

> Two files will be created in your \~/.ssh directory: `id_rsa` and `id_rsa.pub`. The public key is `id_rsa.pub` - this file will be uploaded to your Linode. The other file is your private key. Do not share this file with anyone!

3. Upload the public key to your Linode with the *secure copy* command (`scp`) by entering the following command in a terminal window *on your desktop computer*. Replace `example_user` with your username, and `123.456.78.90` with your Linode's IP address. If you have a Windows desktop, you can use a third-party client like WinSCP to upload the file to your home directory.

```
scp ~/.ssh/id_rsa.pub example_user@123.456.78.90:
```

4. Create a directory for the public key in your home directory (`/home/yourusername`) by entering the following command *on your Linode*:

```
mkdir .ssh
```

5. Move the public key in to the directory you just created by entering the following command *on your Linode*:

```
mv id_rsa.pub .ssh/authorized_keys
```

6. Modify the permissions on the public key by entering the following commands, one by one, *on your Linode*. Replace `example_user` with your username.

```
chown -R example_user:example_user .ssh
chmod 700 .ssh
chmod 600 .ssh/authorized_keys
```

The SSH keys have been generated, and the public key has been installed on your Linode. You're ready to use SSH key pair authentication! To try it, log out of your terminal session and then log back in. The new session will be authenticated with the SSH keys and you won't have to enter your account password. (You'll still need to enter the passphrase for the key, if you specified one.)

Disabling SSH Password Authentication and Root Login

You just strengthened the security of your Linode by adding a new user and generating SSH keys. Now it's time to make some changes to the default SSH configuration. First, you'll disable *password authentication* to require all users connecting via SSH to use key authentication. Next, you'll disable *root login* to prevent the `root` user from logging in via SSH. These steps are optional, but are strongly recommended.

You may want to leave password authentication enabled if you connect to your Linode from many different desktop computers. That will allow you to authenticate with a password instead of copying the private key to every computer.

Here's how to disable SSH password authentication and root login:

1. Open the SSH configuration file for editing by entering the following command:

```
sudo nano /etc/ssh/sshd_config
```

2. If you see a message similar to *-bash: sudo: command not found*, you'll need to install `sudo` on your Linode. To do so, log in as root by entering the `su` command, and type the `root` password when prompted. Next, install `sudo` by entering the following command: `apt-get install sudo`. After `sudo` has been installed, log out as the `root` user by entering the `exit` command.
3. Change the `PasswordAuthentication` setting to `no` as shown below. Verify that the line is uncommented by removing the # in front of the line, if there is one.:

```
PasswordAuthentication no
```

4. Change the `PermitRootLogin` setting to `no` as shown below:

```
PermitRootLogin no
```

5. Save the changes to the SSH configuration file by pressing **Control-X**, and then **Y**.
6. Restart the SSH service to load the new configuration. Enter the following command:

```
sudo service ssh restart
```

After the SSH service restarts, the SSH configuration changes will be applied.

Creating a Firewall

Now it’s time to set up a *firewall* to limit and block unwanted inbound traffic to your Linode. This step is optional, but we strongly recommend that you use the example below to block traffic to ports that are not commonly used. It’s a good way to deter would-be intruders! You can always modify the rules or disable the firewall later.

Here’s how to create a firewall on your Linode:

1. Check your Linode’s default firewall rules by entering the following command:

```
sudo iptables -L
```

2. Examine the output. If you haven’t implemented any firewall rules yet, you should see an *empty ruleset*, as shown below:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

3. Create a file to hold your firewall rules by entering the following command:

```
sudo nano /etc/iptables.firewall.rules
```

4. Now it’s time to create some firewall rules. We’ve created some basic rules to get you started. Copy and paste the rules shown below in to the `iptables.firewall.rules` file you just created.

/etc/iptables.firewall.rules

```
*filter

#  Allow all loopback (lo0) traffic and drop all traffic to 127/8 that doesn't use lo0
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 -j REJECT

#  Accept all established inbound connections
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#  Allow all outbound traffic - you can modify this to only allow certain traffic
-A OUTPUT -j ACCEPT

#  Allow HTTP and HTTPS connections from anywhere (the normal ports for websites and SSL).
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT

#  Allow SSH connections
#
#  The -dport number should be the same port number you set in sshd_config
#
-A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT

#  Allow ping
-A INPUT -p icmp -j ACCEPT

#  Log iptables denied calls
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7

#  Drop all other inbound - default deny unless explicitly allowed policy
-A INPUT -j DROP
-A FORWARD -j DROP
```

```
COMMIT
```

5. Edit the rules as necessary. By default, the rules will allow traffic to the following services and ports: HTTP (80), HTTPS (443), SSH (22), and ping. All other ports will be blocked.

   Be sure to revise these rules if you add new services later.

6. Save the changes to the firewall rules file by pressing Control-X, and then Y.
7. Activate the firewall rules by entering the following command:

```
sudo iptables-restore < /etc/iptables.firewall.rules
```

8. Recheck your Linode’s firewall rules by entering the following command:

```
sudo iptables -L
```

9. Examine the output. The new ruleset should look like the one shown below:

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
REJECT     all  --  anywhere             127.0.0.0/8          reject-with icmp-
port-unreachable
ACCEPT     all  --  anywhere             anywhere             state
RELATED,ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
ACCEPT     tcp  --  anywhere             anywhere             state NEW tcp
dpt:ssh
ACCEPT     icmp --  anywhere             anywhere
LOG        all  --  anywhere             anywhere             limit: avg 5/min
burst 5 LOG level debug prefix "iptables denied: "
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

10. Now you need to ensure that the firewall rules are activated every time you restart your Linode. Start by creating a new script with the following command:

```
sudo nano /etc/network/if-pre-up.d/firewall
```

11. **CentOS users:** If you are using CentOS 6.2 or higher, save your current iptables rules with the following command:

```
/sbin/service iptables save
```

12. Copy and paste the following lines in to the file you just created:

/etc/network/if-pre-up.d/firewall

```
#!/bin/sh
/sbin/iptables-restore < /etc/iptables.firewall.rules
```

13. Press Control-X and then press Y to save the script.
14. Set the script’s permissions by entering the following command:

sudo chmod +x /etc/network/if-pre-up.d/firewall

That’s it! Your firewall rules are in place and protecting your Linode. Remember, you’ll need to edit the firewall rules later if you install other software or services.

Installing and Configuring Fail2Ban

Fail2Ban is an application that prevents dictionary attacks on your server. When Fail2Ban detects multiple failed login attempts from the same IP address, it creates temporary firewall rules that block traffic from the attacker’s IP address. Attempted logins can be monitored on a variety of protocols, including SSH, HTTP, and SMTP. By default, Fail2Ban monitors SSH only.

Here’s how to install and configure Fail2Ban:

1. Install Fail2Ban by entering the following command:

```
sudo apt-get install fail2ban
```

2. Optionally, you can override the default Fail2Ban configuration by creating a new `jail.local` file. Enter the following command to create the file:

```
sudo nano /etc/fail2ban/jail.local
```

3. To learn more about Fail2Ban configuration options, see this article on the Fail2Ban website.
4. Set the `bantime` variable to specify how long (in seconds) bans should last.
5. Set the `maxretry` variable to specify the default number of tries a connection may be attempted before an attacker's IP address is banned.
6. Press `Control-x` and then press `y` to save the changes to the Fail2Ban configuration file.

Fail2Ban is now installed and running on your Linode. It will monitor your log files for failed login attempts. After an IP address has exceeded the maximum number of authentication attempts, it will be blocked at the network level and the event will be logged in `/var/log/fail2ban.log`.

Next Steps

Good work! You have secured your Linode to protect it from unauthorized access. Next, you'll learn how to host a website. Start reading the Hosting a Website quick start guide to get going