LSN Documents Documentation

Release

Ajinkya Wadekar

Nov 15, 2017

Contents:

We are a Dedicated and Cloud Hosting company. Here you will find information about most of the technologies we use in managing our infrastructure. We use Openstack for our cloud, Ansible for orchestration etc.

CHAPTER 1

Guide

1.1 Limestone_Addon_Services

1.1.1 DDoS_Protection

1.1.2 DNS_rDNS

1.1.3 KVMoIP

1.1.4 NAS

1.1.5 Software

1.1.6 SSL_Certificates

1.1.7 VPN

1.2 OnePortal

1.2.1 API

1.3 OperationgSystem_Support

1.3.1 Linux_Support

Receiving an "ip_conntrack: table full" error.

On OpenVZ/HyperVM machines sometimes the ip_conntrack table will become full and drop packets. You can tell if it is doing this by looking in your /var/log/messages file.

To find out the current limit run:

```
sysctl net.ipv4.netfilter.ip_conntrack_max
```

Then to increase it edit:

```
/etc/sysctl.conf
```

and change the line:

```
net.ipv4.netfilter.ip_conntrack_max = to a higher number
```

Adding **5000** or **10000** to the current max should be fine.

Once you have saved the file, to reload the new configuration run:

```
sysctl -p
```

You should be all set and the machine should not be dropping any packets.

Hardening CentOS

What will this script do?

- Install useful packages such as tcpdump, mtr, zsh, perl and logrotate
- Setup automatic yum updates
- **Set password policies**
  - Passwords will expire every 180 days
  - Passwords may only be changed once a day
- **Set OS policies**
  - Set idle users to be disconnected after 15 minutes
- **Install (if it is not installed) and configure IPTables firewall**
  - Open specified TCP/UDP ports
  - **Set rules to block common attacks**
    - Syn Floods
    - Fragmented Packets
    - Malformed XMAS Packets
    - Drop NULL packets
    - Limit pings to 3 per second and bursts of 25
    - Discourage Port Scanning
  - Set up Connection Tracking
- **Install DDoS Deflate**
  - **More information about DDoS Deflate is available at** http://deflate.medialayer.com/
- **Install CHKROOTKIT**
  - Scheduled to check daily for issues and email your Admin Email

  - **More information about CHKROOTKIT is available at** http://www.chkrootkit.org/
- **Install rkhunter (Root Kit Hunter)**
  - Scheduled to check daily for issues and email your Admin Email
  - More information about rkhunter is available at http://www.rootkit.nl/projects/rootkit_hunter.html
- **Install LSM (Linux Socket Monitor)**
  - Runs in the background and watches for changes in sockets
- **Secure the SSH Daemon**
  - Change the SSH port to a random number
  - Create an "admin" user
  - Make it so only the "admin" user can be logged into over SSH

Downloading the Script:

```
cd /root
wget http://mirror.lstn.net/scripts/hardening/centos.sh
chmod +x centos.sh
```

Modifying the Variables:

```
vim centos.sh
```

```
You may customize TCPPORTS and UDPPORTS, however the defaults in there now
should cover most common processes.
```

Run the Script:

```
./centos.sh
```

What to do afterwards

After it completes, you will get a message like:

```
****************************************
YOUR SERVER IS NOW HARDENED
----------------------------------------
SSH User: admin
SSH Pass: 254457cb9448226
SSH Port: 5575
Admin Email: admin@fake.lstn.net
****************************************

You must now reconnect to this server using the information above
Changing the SSH port has caused this connection to freeze.
BEFORE CLOSING THIS WINDOW please note your information above.
```

How do I set up the local yum repo?

You may now get your CentOS installs and updates locally on Limestone's network. When routed correctly, it will not count against your monthly bandwidth.

Yum Configuration

Edit the config:

```
vi /etc/yum.repos.d/CentOS-Base.repo
```

And put the following:

```
[base]
name=CentOS-$releasever - Base
baseurl=http://centos.mirror.cust.lstn.net/$releasever/os/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[update]
name=CentOS-$releasever - Updates
baseurl=http://centos.mirror.cust.lstn.net/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[addons]
name=CentOS-$releasever - Addons
baseurl=http://centos.mirror.cust.lstn.net/$releasever/addons/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[extras]
name=CentOS-$releasever - Extras
baseurl=http://centos.mirror.cust.lstn.net/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[centosplus]
name=CentOS-$releasever - Plus
baseurl=http://centos.mirror.cust.lstn.net/$releasever/centosplus/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
```

Correctly Route the Traffic

```
ip route add 10.0.0.0/8 via <strong>private-gateway-ip</strong> dev <strong>private-
↪interface</strong>
```

If you want this to save after reboot, add to /etc/sysconfig/network-scripts/route-private-device

```
10.0.0.0/8 dev <strong>private-interface</strong>
```

Example:

```
10.0.0.0/8 dev eth1
```

How can I reset my root password?

This article explains how to reset the root password if you no longer know it. You will require KVM access to your server in order to perform these steps.

CentOS/Fedora

- Through the KVM, initiate a restart of your server by sending the shutdown command

```
shutdown -r now
```

- When it comes to the **Loading CentOS/Fedora** Grub Bootloader screen hit **esc**.(Right after you get to the network boot screen it will be at the grub bootloader screen)
- At the grub screen select the default OS & hit **e** to edit.
- Then you should have 3 lines of text. Select the line that starts with **kernel** & ends in **root=label=/**
- Press **e** to edit the line, and at the end of the line add a space and then put a capital **S**.
- Then enter to save & go back to the previous screen.
- Then hit **b** for boot with that line selected.
- It will enter you in single user mode and you get a bash prompt.
- Then you can just type **passwd** enter new password and reboot.

You will not need to re-edit the grub loader it reverts to normal after the next reboot.

Debian

Reboot and edit the Grub kernel line add a space then the following at the end of the line. (like 'Alternate Method' above)

```
init=/bin/sh
```

Then run this command to remount the root partition in read/write mode.

```
mount -o remount,rw /
```

and reset the root password as normal with the **passwd** command.

How do I schedule FSCK to run automatically?

Using cron to schedule an FSCK

- By default, a fsck is forced after 30 reboots or 180 days.
- **To avoid issues such as this, we recommend scheduling fsck to run a basic weekly check on your server to identify and flag**
  - Doing so can prevent unwanted, forced fsck from running in situations such as this one.
  - You can then, plan for a time at which a full system fsck is run.

For more info on running fsck please click https://en.wikipedia.org/wiki/Fsck.

For more info on scheduling tasks under Linux click https://en.wikipedia.org/wiki/Cron.

The following example syntax will add a weekly scheduled **scan-only** fsck and output the results to a log file for review.

```
crontab -e
```

enter the following text substituting *partition* with your root partition.

```
@weekly fsck -nv /dev/*partition* > /var/log/weekly_fsck
```

When saving, do not change the existing file name.

PLEASE NOTE– This does not eliminate the need for a fsck. You will still need to schedule a manual fsck.

To Force a fsck using shutdown command

```
shutdown -rF now
```

Bypass a fsck using shutdown command

```
shutdown -rf now
```

Note: Capital F will Force a FSCK, lowercase f skips a FSCK.

FSCK will sometimes require the root password be entered on the console in order to repair some issues with the filesystem, contact our support department if your server does not respond after a reboot.

How do I set up SSH key authentication?

SSH packets being sent from the SSH client to the server are encrypted with a form of shared-key cryptography, using a random key which is generated for each new connection and thrown away when that connection is over. The client and the server use public-key cryptography to agree on the session key, and either party may request a re-keying of the session at any time.

Once you become familiar with SSH keys, communication and file copying between servers / clients will be secure, quicker, and more convenient.

Here's an example on setting it up between a CentOS Client and CentOS Server:

On the client, do the following:

- Goto the .ssh directory, which is located under /root – full path is /root/.ssh
- Now let's create our private and public keys and put them into a file.

```
ssh-keygen -t rsa
```

This created a 1024 bit key, and creates 2 files.

1. id_rsa – This holds your client's PRIVATE Key.
2. id_rsa.pub – This holds your server's PUBLIC key.

Now, let's place the key **id_rsa.pub** into the servers authorized_keys file. Located at: **/root/.ssh/authorized_keys**, If this file is not already there, we will create it.

Next you need to copy the key to your system. We'll copy the key over via a file copying program called rsync

```
rsync -av -e ssh id_rsa.pub <SERVER_IP>:/root/.ssh/
```

Make sure to change **SERVER_IP** to the servers IP address.

After doing this command, you will be prompted for the root password of the server, type it and press enter.

Now, on the server, do the following:

```
cd /root/.ssh
cat id_rsa.pub >> authorized_keys
chmod 600 authorized_hosts
```

The 2nd command copies the contents of id_rsa.pub into authorized_keys file.

The 3rd command gives it the correct permissions to be run by the system.

Now, back on the client, do the following:

```
cd /root/.ssh
eval `ssh-agent`
ssh-add id_rsa
ssh-add -l
```

2nd command: Starts the SSH agent program.

3rd and 4th command: Adds your private key into memory.

Simply SSH into the server.

```
ssh <server_IP>
```

When prompted, type in the root password. Now exit out and try to SSH into the server from the client once more. This time – you shouldn't be prompted for a password.

How do I use iptables?

Warning

Modifying rules on your server can cause the server to become inaccessible on port 22 (SSH) or your alternate SSH port.

Description / Basic Overview

Everyone in the IT industry is very concerned with security, especially if you're a linux administrator. Many linux distributions come with several services that you may not use or ever need but they're running on your server anyways. This can cause many security threats. With the slightest knowledge of Linux firewalls (iptables) you can secure your linux server very quickly and efficiently. In this article, I will either introduce you into iptables for your first time, or help you become more efficient with iptables if you've worked with them in the past.

As network packets flow in and out of the network interface card, they are intercepted, analyzed and manipulated as ruled through the Linux firewall. As the packet flows through the firewall rules and it reaches a rule that is matches, it stops there and doesn't continue through the rest of the rule set. For instance, if there is a rule to drop all packets coming in through port 25 and then a rule directly after that says "accept 192.168.1.25 on port 25" That packet will be dropped once it hits the first rule. It won't even know there is a second rule. Read the first example further down this article. There is an example. There have been 3 main linux firewalls widely used, and they are as follows:

History

- Ipfwadm which was merged into Linux 2.0. It can filter TCP, UDP, and ICMP packets only. It also does not support QoS. You can "insert" and "remove" rules. This doesn't make it the most user friendly linux firewall on the planet.
- Ipchains which was merged into Linux 2.2. It supports QoS, Is very flexible with the configuration, as it has "replace" along with "insert" and "remove". This makes ipchains more user friendly. Ipchains also has the ability to filter any IP protocol explicitly, not just TCP, UDP, and ICMP.

- Iptables. This iptables project was begun in 1998 by Rusty Russell. This was merged into Linux 2.3 in 2000, and is still widely used today. It supports stateful IPv4, and IPv6 protocol tracking and IPv4 application tracking. Has built-in PORTFW functionality. It is also very user friendly, as you'll soon find out.

Getting started

Let's take a look at our iptables list, see what is currently under there!

```
/sbin/iptables -L -n
```

That will show you your complete iptables rule list, with as much information as possible about each rule. Let's break down what you're looking at you should see something similar to: (note: the following is an empty table you may have some rules in yours).

```
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
```

Flushing your list of rules can be good if you would like to rewrite your rules completely as I've done plenty of times in the past. You can "flush" every rule under iptables by doing

```
/sbin/iptables -F
```

However you may want to only flush all the rules under the INPUT, FORWARD or OUTPUT chain. You can specify which chain to flush by either of the following:

- /sbin/iptables -F INPUT
- /sbin/iptables -F FORWARD
- /sbin/iptables -F OUTPUT

Additionally, you can save your rules so that when you restart your linux server, the current rules will become active once again. You can save by doing

```
/etc/init.d/iptables save
```

If you would wish for iptables to STOP running, you can initiate the following command

```
#> /etc/init.d/iptables stop

Flushing firewall rules:                                   [  OK  ]
Setting chains to policy ACCEPT: filter                  [  OK  ]
Unloading iptables modules:                                [  OK  ]
Of course, you can START iptables by doing
/etc/init.d/iptables start
Applying iptables firewall rules:                      [  OK  ]
```

As for an example, I have one below:

Example

You want to simply deny every IP address a connection to your SMTP server (tcp port 25) except for the IP: 192.168.1.25.

The two commands for this are as follows

```
/sbin/iptables -I INPUT -p tcp --dport 25 -j DROP
/sbin/iptables -I INPUT -s 192.168.1.25 -p tcp --dport 25 -j ACCEPT
```

The reason I put the "DROP" command in before the "ACCEPT" is because a rule is entered into the database, and then a rule that is added next is added directly above the last one entered. Putting the DROP command before the ACCEPT let's the ACCEPT rule be read before the DROP command. Let's break the rest of these commands down:

The first command:

- **-I** is to insert the rule into the top of the chain. You would use -A to insert it at the bottom of the chain. (Note: you can do "–D" instead to delete the rule from the chain as well.)
- **INPUT** is the chain name. Input is the chain that is followed by "incoming" packets.
- **-p** is the protocol argument, you specify the protocol type with this command, notice the "tcp" after the "-p"
- **–ddport** is what specifies which port to filter. In this case it is 25, because that is what port SMTP runs on (by default).
- **-j** is the argument that specifies what to do with the packet. In this case, it's going to be "DROPPED"

The second command:

The only difference between this command and the first one, is there is a (-s) "src" IP address specified and the –j argument is "ACCEPT".

Since a (-s) "src" address was not specified in the first argument, it assumed that every address is to be dropped.

Ok, let's save our current work.

```
#> /etc/init.d/iptables save
/sbin/iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT          tcp  --  192.168.1.25         0.0.0.0/0           tcp dpt:25
DROP            tcp  --  0.0.0.0/0                0.0.0.0/0           tcp dpt:25
```

Notice: how the "ACCEPT" rule is above the "DROP" rule.

1.3.2 Windows_Support

1.4 Adbuse

1.5 LSN_Cloud_CDN

1.5.1 Cloud_Solutions

Our Cloud is based on Openstack

1.5.2 Content_Delivery_Network

1.6 Resellers

1.7 TOS_AUP

CHAPTER 2

Indices and tables

- genindex
- modindex
- search