GNU/Linux Administration - Support #350

Securely Setting Up An External Server to Join ISPConfig Administration Node

03/25/2014 09:39 AM - Daniel Curtis

Status:	Closed	**Start date:**	03/25/2014
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Mail Server	**Estimated time:**	1.50 hour
Target version:		**Spent time:**	1.00 hour

Description

While setting up the internal mail server, I found that I am restricted in sending email due to the fact the web services provided are from a dynamic IP address; I can assume this is to fight spam. With this restriction, I have set up a VPS for testing email functionality from a static IP. I use ISPConfig to manage provisioning of Web and email, which uses MySQL as a backend. Since I need to access the administration nodes' MySQL database to join to ISPConfig proper, I need to create an SSH tunnel to create a connection to the administration node. This guide is what I did to create that secure connection.

Note that the commands and scripts in this document assume that you are connecting to a MySQL database

1. Named "db"
2. On host example.com
3. On port port 3306
4. Using username "dbuser"
5. Using password "PASS"
   You will need to change these values to match your own setup.

Create the local shell account

On the database server, as root, create a user to handle the server side of all tunnels. This user will have no valid shell (might not work on all operating systems?)

```
mkdir /home/mylockdown
useradd -s /bin/false mylockdown
mkdir /home/mylockdown/.ssh
touch /home/mylockdown/.ssh/authorized_keys
chown -R mylockdown:mylockdown /home/mylockdown/
chmod 755 /home/mylockdown/.ssh
chmod 600 /home/mylockdown/.ssh/authorized_keys
```

Setup the SSH Tunnel

To set up a tunneled connection to a MySQL server at example.com, you can issue the following two commands on any client:

```
ssh -fNg -L 3307:127.0.0.1:3306 mylockdown@example.com
mysql -h 127.0.0.1 -P 3307 -u dbuser -p db
```

The first command tells ssh to log in to example.com as mylockdown, go into the background (-f) and not execute any remote command (-N), and set up port-forwarding (-L localport:localhost:remoteport). In this case, we forward port 3307 on localhost to port 3306 on example.com.

The second command tells the local MySQL client to connect to localhost port 3307 (which is forwarded via ssh to remotehost.com:3306). The exchange of data between client and server is now sent over the encrypted ssh connection.

Connections via the tunnel will look like they are coming from 127.0.0.1, so you need to update the GRANT tables in the database:

```
USE mysql;
GRANT ALL ON db.* TO dbuser@127.0.0.1 IDENTIFIED BY 'PASS';
FLUSH PRIVILEGES;
```

To make the same connection using PHP's built-in MySQL client:

```
<?php
 $smysql = mysql_connect( "127.0.0.1:3307", "dbuser", "PASS" );
 mysql_select_db( "db", $smysql );
?>
```

Obviously, this approach requires that you have a shell account on the remote (database) server that you can log into in order to set up the forwarded port. And it requires a MySQL login that is allowed to connect to the database in question from host 127.0.0.1

Congrats, your server is ready to accept tunnels.

Setup key-based logins

Because you won't be around to type a password, you'll need to set up an RSA key pair for ssh on each client.
The first step is to set up an RSA key pair as root on each client, leaving all questions blank:

```
ssh-keygen -t ecdsa
```

> Generating public/private ecdsa key pair.
> Enter file in which to save the key (/var/root/.ssh/id_ecdsa):
> Enter passphrase (empty for no passphrase):
> Enter same passphrase again:
> Your identification has been saved in /var/root/.ssh/id_ecdsa.
> Your public key has been saved in /var/root/.ssh/id_ecdsa.pub.
> The key fingerprint is:
> 23:02:1b:f7:55:2b:35:cf:0c:5d:ad:21:c7:e1:78:ca root@localexample.com

The next step is to append the rsa public key file to mylockdown's .ssh/authorized_keys on the server. First we copy the key, then we shell in and append it to the authoized list:

```
scp /var/root/.ssh/id_ecdsa.pub root@example.com:/tmp/myhost.local_ecdsa.pub
ssh example.com
cat /tmp/myhost.local_ecdsa.pub >> /home/mylockdown/.ssh/authorized_keys
exit
```

Now you can set up the connection on the client, without typing mylockdown's password:

```
ssh -fNg -L 3307:127.0.0.1:3306 mylockdown@example.com
```

Making It A Daemon

A quick and dirty way to make sure the connection runs on startup and respawns on failure is to add it to /etc/inittab and have the init process (the, uh, kernel) keep it going.

Add the following to /etc/inittab on each client:

```
vi /etc/inittab
```

> sm:345:respawn:/usr/bin/ssh -Ng -L 3307:127.0.0.1:3306 mylockdown@example.com

And that should be all you need to do. Send init the HUP signal, kill -HUP 1 , to make it reload the configuration. To turn it off, comment out the line and HUP init again.

Setup the mail node

This server was covered in Issue #178 so consult that guide for more help. We are going to install the necessary packages for the mail server and some prerequisites for ISPConfig:

```
apt-get install postfix postfix-doc openssl getmail4 binutils dovecot-imapd dovecot-pop3d dovecot-
sieve amavisd-new spamassassin zoo unzip bzip2 arj nomarch lzop cabextract apt-listchanges libnet-
ldap-perl libauthen-sasl-perl daemon libio-string-perl libio-socket-ssl-perl libnet-ident-perl zip
 libnet-dns-perl postfix-mysql dovecot-mysql fail2ban php5-cli php5-mysql php5-mcrypt mcrypt
```

Now download, install and setup ISPConfig:
cd /tmp
wget http://prdownloads.sourceforge.net/ispconfig/ISPConfig-3.0.5.2.tar.gz
tar xzf ISPConfig-3.0.5.2.tar.gz
cd ispconfig_install/install
php -q install.php

When installing, I used the following options:

- Select language (en,de) [en]:
- Installation mode (standard,expert) [standard]: **expert**
- Full qualified hostname (FQDN) [localhost]: **mail1.example.com**
- MySQL server hostname [localhost]:
- MySQL root username [root]:
- MySQL root password []: **LocalSecretPassword**
- MySQL database to create [dbispconfig]:
- MySQL charset [utf8]:
- ISPConfig mysql database username [ispconfig]:
- ISPConfig mysql database password [5913543194e2a3e2a5043cd21e61ce86]:
- Shall this server join an existing ISPConfig multiserver setup (y,n) [n]: **y**
- MySQL master server hostname []: **127.0.0.1:3307**
- MySQL master server root username [root]:
- MySQL master server root password []: **SuperSecretPassword**
- MySQL master server database name [dbispconfig]:
- Configure Mail (y,n) [y]: **y**
- Configure Jailkit (y,n) [y]: **n**
- Configure FTP Server (y,n) [y]: **n**
- Configure DNS Server (y,n) [y]: **n**
- Configure Firewall Server (y,n) [y]: **y**
- Install ISPConfig Web Interface (y,n) [y]: **n**

NOTE: I made sure to use the SSH tunnel created earlier.

History

#1 - 04/16/2014 05:38 PM - Daniel Curtis

- Status changed from In Progress to Closed

#2 - 02/16/2015 11:17 AM - Daniel Curtis

- Project changed from 84 to GNU/Linux Administration

- Category set to Mail Server