LibreNMS

Configuring LibreNMS

Goals

- Learn how to configure the LibreNMS Network Management System

Introduction

In this exercise, we will set up **LibreNMS** as our network monitoring package. The software will poll your routers and switches using SNMP and provide visibility of OSPF sessions and other useful information

Throughout these instructions we will use the names and addresses related to **campusX**. You should make sure you change any names and addresses in the examples to match your assigned group of machines.

SNMP Configuration on Network Devices

Before we configure our LibreNMS, our network monitoring software, we need to configure the SNMP community string on **all** the routers and the switches in our campus.

The IP address of the management system we will be setting up is 100.68.X.130 (replace **X** with your campus number) so we only allow that host in access-list 99. And then we will use the community called *NetManage*, and only allow members of access-list 99 access to the device. Finally, we tell the device to retain the interface index through reboots and the addition or removal of interfaces.

Here is a sample configuration:

```
access-list 99 permit 100.68.X.130
!
snmp-server community NetManage RO 99
snmp-server ifindex persist
```

If your switch or router doesn't take the `snmp-server ifindex` command, try this SNMP command instead:

```
snmp ifmib ifindex persist
```

Even though Cisco IOS is one operating system, the implementation details on different platforms can well be different. The latter format tends to be used on more modern devices, especially ethernet switches.

Before proceeding, check that all routers and switches in your campus have SNMP configured.

Setting up your NMS PC

The following sections go through the process of setting up LibreNMS on your Network Monitoring PC. The LibreNMS software has already been installed, but it has not yet been configured to monitor anything.

Login to your management PC

```
ssh sysadm@pc1-campusX.ws.nsrc.org
```

The password will have been given to you in class.

NOTE: These instructions assume you are the root user. If you are not, prepend `sudo` to the shell commands (the ones that aren't at `mysql>` prompts) or temporarily become a user with root privileges with `sudo -s`.

```
* Commands preceded with "#" imply that you should be working as root.
* Commands with more specific command lines (e.g. ''rtrX>'' or ''mysql>'')
imply that you are executing commands on remote equipment, or within another
program.
```

Basic configuration

We're going to make a few changes to the configuration file:

```
* Change the default SNMP community
* Tell LibreNMS which networks it's allowed to scan/discover
```

First be sure you are the root user:

```
$ sudo bash
```

Setting the SNMP community

First, let's change the SNMP community that LibreNMS will try when discovering and adding new devices.

Edit the file `/opt/librenms/config.php`,

```
# editor /opt/librenms/config.php
```

and find the line:

```
$config['snmp']['community'] = array("public");
```

And set it to:

```
$config['snmp']['community'] = array("NetManage");
```

Tell LibreNMS which subnets it's allowed to scan automatically

By default, LibreNMS will try ask for the list of “neighbors” that network devices “see” on the network. This is done using the Link Layer Discovery Protocol (LLDP) or Cisco's CDP (Cisco Discovery Protocol).

But to be on the safe side, and not scan networks outside your organization, LibreNMS needs to be told which subnets it's allowed to scan for new devices.

Still in the file `/opt/librenms/config.php`, find the line:

```
#$config['nets'][] = "10.0.0.0/8";
#$config['nets'][] = "172.16.0.0/12";
#$config['nets'][] = "192.168.0.0/16";
```

And replace this with the following to scan our specific subnets in use by our campus network and the workshop infrastructure. Where you see an X below replace that with your campus number:

```
$config['nets'][] = "10.10.0.0/16";
$config['nets'][] = "172.2X.0.0/19";
$config['nets'][] = "100.68.X.0/24";
```

We need to make one more change...

Tell LibreNMS not to add duplicate devices

A situation can happen where two devices have duplicate SNMP `sysName`. (that's `hostname` in IOS) They could be two different devices, so it would be a good idea to have LibreNMS automatically add and monitor them.

But it can also happen that the SAME device is seen multiple times by LibreNMS - once using LLDP/CDP, and another time via OSPF (for example).

In that case, it ends up added twice. For instance, you may suddenly see two devices called `rtr2-fa0-0.ws.nsrc.org` and `rtr2`, and this is not what we want.

Since “both” devices are in fact the same, their SNMP `sysName` will be identical, and we can tell LibreNMS to *NOT* add devices if one already exists with the same `sysName` - after all, this shouldn't happen in a well configured network! :)

Here's an example of this:

```
2016-07-06 20:16:47 rtr4 discovery Device rtr4 (10.10.0.224) (port
FastEthernet0/0) autodiscovered through CDP on rtr1.ws.nsrc.org
```

```
2016-07-06 20:09:45 rtr4-fa0-0 discovery Device rtr4-fa0-0.ws.nsrc.org
(10.10.0.224) (port ) autodiscovered through OSPF on rtr1-fa0-0.ws.nsrc.org
```

To avoid this, we will add the following setting:

```
$config['allow_duplicate_sysName'] = false;
```

... this will prevent LibreNMS from adding the device if it exists already with the same `sysName`. You will be able to see if there are duplicate devices deteced in the `Event Log` (Overview → Event Log).

After you've added the above setting, save the file and exit - we're nearly done!

Add a host

Let's add localhost (i.e.: YOUR virtual server), using the following commands. Later you'll do this from the Web interface:

```
# cd /opt/librenms
# php addhost.php localhost NetManage v2c
```

You should see:

```
Trying community NetManage ...
Added device localhost (1)
```

Notice we explicitly tell LibreNMS which SNMP community to use. We also assume it's SNMP v2c. If you're using v3, there are additional steps which aren't provided here.

Add your routers

LibreNMS uses hostnames when it monitors devices. Normally we'd add our routers to the DNS but for this workshop we'll just add our ISP routers to the `/etc/hosts` file. Edit /etc/hosts and add entries for your devices.

```
# editor /etc/hosts
```

For example, if you have been allocated 100.68.X.0/24, then your entries would look like:

```
100.68.X.1   bdr1-campusX
100.68.X.2   core1-campusX
172.2X.10.2  dist1-b1-campusX
172.2X.10.3  edge1-b1-campusX
172.2X.10.4  edge2-b1-campusX
172.2X.20.2  dist1-b2-campusX
172.2X.20.3  edge1-b2-campusX
172.2X.20.4  edge2-b2-campusX
```

Make sure you change X to match your campus group

Save the changes and check that you can ping the routers using these names. For instance:

```
# ping bdr1-campusX
# ping edge1-b2-campusX
```

Now we can add the devices to LibreNMS:

```
# cd /opt/librenms
# php addhost.php bdr1-campusX NetManage v2c
# php addhost.php core1-campusX NetManage v2c
# php addhost.php dist1-b1-campusX NetManage v2c
# php addhost.php edge1-b1-campusX NetManage v2c
# php addhost.php edge2-b1-campusX NetManage v2c
# php addhost.php dist1-b2-campusX NetManage v2c
# php addhost.php edge1-b2-campusX NetManage v2c
# php addhost.php edge2-b2-campusX NetManage v2c
```

Do this to have all your local campus entries.

This assumes you haven't made SNMP community changes – if you have, replace `NetManage` with your community. It also assumes SNMP v2c.

Final configuration

Discover and Poll newly added hosts

LibreNMS first “discovers” each host that has been added. This means that it methodically examines each host you added and figures out what it should monitor. The `discover.php` script does not automatically scan your network to find new devices. To run this script do:

```
# cd /opt/librenms
# sudo -u librenms php discovery.php -h all
```

NOTE: This will very likely take quite some time.

Once this has finished you can now “poll” the hosts. This means LibreNMS now knows what it wishes to monitor for each host, but it has yet to populate its database with initial values for each item. To do this we do:

```
# sudo -u librenms php poller.php -h all
```

As you can see the `poller.php` script does quite a bit with just a few devices. When we add it to a cronjob below this helps explain why LibreNMS is a resource intensive tool.

Create cronjob

Create the cronjob which will run periodic tasks required by LibreNMS:

```
# cd /opt/librenms
# cp librenms.nonroot.cron /etc/cron.d/librenms
```

One last thing: edit the file `/etc/cron.d/librenms` ...

```
# editor /etc/cron.d/librenms
```

...and find the line:

```
    */5  * * * *  librenms  /opt/librenms/cronic /opt/librenms/poller-
wrapper.py 16
```

And change the 16 at the end to 4 (we have a single processor, and 4 threads is plenty)

```
    */5  * * * *  librenms  /opt/librenms/cronic /opt/librenms/poller-
wrapper.py 4
```

Save, and exit.

Install complete

That's it! You now should be able to log in to http://librenms.campusX.ws.nsrc.org/ and begin to explore the information being collected for your monitored devices.

You can add some additional devices via the LibreNMS web interface. Why not add:

- noc.ws.nsrc.org
- www.ws.nsrc.org
- s1.ws.nsrc.org

using the class snmp community. See if you can figure out how to do this on your own.

PLEASE NOTE: We have not covered HTTPS setup in this example, so your LibreNMS install is not secure by default. Please do not expose it to the public Internet unless you have configured HTTPS and taken appropriate web server hardening steps.

About Daily Updates

LibreNMS performs daily updates by default. At 00:15 system time every day, a `git pull –no-edit –quiet` is performed. If you don't want this, change the default by editing your `config.php` file. Remove the comment (the # mark) on the line:

```
#$config['update'] = 0;
```

so that it looks like this:

```
$config['update'] = 0;
```