

OpenNMS – an Introduction to Network Management

Objective:

The objective of this lab is to gain an introduction to network management through the installation of an open-source program called OpenNMS. The program must be correctly installed and configured to function on the IT network.

References:

<http://opennms.org>
<http://www.postgresql.org>

Equipment/Programs Used:

Dell Inspiron 8600 Laptop
Linux RedHat 9
OpenNMS
Tomcat4
PostgreSQL
J2SDK

Procedures:

The procedures of this lab will be organized into the following sections:

1. Install OpenNMS on a Linux Box.
2. Document each installation difficulty and how it was overcome.
3. Configure OpenNMS to monitor the IT network.
4. Understand and explain how OpenNMS functions as a monitoring tool.
5. Experiment with the functionality of the software and describe its capabilities along with how you discovered them.
6. Describe how OpenNMS can be extended.

1. Install OpenNMS on a Linux Box.

For the installation of OpenNMS, RedHat 9 was chosen as the Linux distribution. Redhat was installed on a second partition of a Dell Inspiron 8600. Upon completing the installation, the instructions for installing OpenNMS were retrieved from the OpenNMS website (<http://www.opennms.org>).

The installation type chosen was the “quick-start install”. This install is a web-based install that runs the script by retrieving it through lynx, and then sending it to sh. The command used was:

```
lynx -source http://install.opennms.org | sh
```

(Please note that a few days after this installation was complete, other students reported that the index script for that link was missing and it then became necessary to insert the following url in place of the given url:
<http://install.opennms.org/stable/index.txt>)

Upon running this command, the script began to install, but failed based on the dependency of a Java JDK. Upon more closely examining the instruction on the OpenNMS site, there was a statement that Java 1.4 JDK must be installed before running the given command.

The next logical step was then to download the Java JDK. At the first attempt, the J2SEE was installed, but the script again failed. The J2SEE was uninstalled, and the J2SDK 1.4.2 was installed by using the ‘rpm’ command. Once it was verified that the package was installed, the previous quick-start install command was issued again. The options to not include extra documentation or RPMs was selected during the install process. The script automatically downloaded and installed tomcat4 and postgresql. However, it gave an error while executing the section on configuring postgresql saying that it was unable to authenticate. At the end of the install, the script suggested attempting to ‘start everything’ and the ‘yes’ option was chosen. The script terminated indicating that everything had started correctly.

In order to test the installation, a browser window was opened and an attempt was made to go to the url localhost/opennms. Upon putting in this url, a login screen appeared. The documentation for OpenNMS indicated that the default login is admin/admin. However, this login did not work correctly. After searching through the frequently asked questions on the site, the following information was discovered:

Check your /var/tomcat4/conf/server.xml file. There should be a <Context> tag for the /opennms context. If not, you can create that context in one of two ways:

1. Automagically, by running install.pl again. Or,
2. Manually, by adding the following to the server.xml file:

```
<Context path="/opennms" docBase="opennms" debug="0" reloadable="true">
<Logger className="org.apache.catalina.logger.FileLogger"
prefix="localhost_opennms_log." suffix=".txt" timestamp="true"/>
<Realm className="org.opennms.web.authenticate.OpenNMSTomcatRealm"
homeDir="/opt/OpenNMS/" />
</Context>
```

Either way, you need to then restart Tomcat, close down all the windows of whatever web browser you were using, open a new web browser window, and surf to the URL again. This time your username and password should be read from the users.xml file correctly.

The server.xml file was edited as suggested, the service restarted, and an attempt to login was again executed. The login worked correctly at this point. However, instead of opening OpenNMS, a tomcat error was produced. The error indicated that an 'opennms' database was not found. This can quickly be explained by remembering that an error was produced in the install script that indicated the postgresql database could not be authenticated. Using Google, a search was made for the exact error that tomcat was producing. The first hit produced a result from the PostgreSQL site that mentioned the need to edit the /var/lib/pgsqll/data/pg_hba.conf file in order to allow 'localhost' to access the postgresql service. There were two lines near the bottom of the file that simply needed to be uncommented so that postgresql would allow login from localhost. Upon editing this file, postgresql was restarted, and the install script was run again. However, because it detected that OpenNMS was already installed, it did not re-configure postgresql. Thus the OpenNMS service was uninstalled by issuing the following commands:

```
rpm -e opennms-webapp
```

```
rpm -e opennms
```

Once the service had been uninstalled, the install script was called again, and it was clear through the terminal output that the opennms database was being configured. Once the install script completed, the postgresql service was restarted, and the opennms service was started. After re-opening the browser and logging in, the OpenNMS main page was displayed showing that 'calculations' were being made on the network.

2. Document each installation difficulty and how it was overcome.

All installation difficulties were documented and outlined in the previous step of the document. However, to summarize the difficulties, it need only be said that the install script did not correctly configure postgresql or tomcat. Once these services were configured correctly, then OpenNMS functioned correctly.

3. Configure OpenNMS to monitor the IT network.

After thus verifying that OpenNMS was installed, it was then necessary to configure it to work with the IT network. According to the documentation on the OpenNMS website, there were two files that particularly needed to be configured in order to ensure that the correct IP address range was discovered. The first file was titled 'discovery-configuration.xml. In this file there was a section titled "include-range" where the IP address beginning and ending needed to be altered. The beginning address was already at 192.168.0.1 by default, so this was left as is. The second address had to be altered to be 192.168.254.254. This will include the entire IP address range for the IT network.

The second file that needed to be configured was the poller-configuration.xml file. A line needed to be added to this file as follows:

```
<include-range begin="192.168.0.1" end="192.168.254.254"/>
```

Upon editing this file, the opennms service was restarted and then logged into through the browser. The page indicating that 'calculations' were being performed was displayed once again. After waiting for quite a long time, the calculations were finally replaced by percentages (initially all at 100%) showing what services were existing on the network.

4. Understand and explain how OpenNMS functions as a monitoring tool.

After exploring the many features that come with OpenNMS, it was clear that it offered a powerful way to monitor the network. OpenNMS uses various protocols such as ARP, ECHO, and SNMP in order to discover devices and the services they run. Upon discovering a device, it is then added to the list of devices to monitor. The program then continuously makes attempts to access these devices and services to show that they are still functioning. If a device fails to respond properly, then it is marked as having failed, and it is flagged as a warning in the various monitoring screens.

5. Experiment with the functionality of the software and describe its capabilities along with how you discovered them.

Perhaps the greatest feature of OpenNMS is the fact that it continuously monitors the devices without needing input from a human. It can be configured to send emails to an administrator if a specific device fails or has trouble. This would be particularly useful to an administrator who is not always able to be near the network. OpenNMS can send an email to a pager or cell phone in order to let the administrator know that a device has failed to respond.

OpenNMS also keeps ongoing statistics of the services and devices to show how frequently they are failing. This can be used to indicate weaknesses in the network, or places where redundancy may be more important than in other areas.

With all of the features in OpenNMS, it is also important to note that it is capable of generating automated reports of the status of the network in a variety of formats. These reports can range from simply showing overall network statistics, or can be specific to just one device or service. These reports can be especially beneficial in a business environment, where it is often necessary to show the status and progress of a network.

6. Describe how OpenNMS can be extended

Although OpenNMS has many wonderful features, it could easily benefit from having a few others. For instance, it was not clear that OpenNMS was capable of providing a connectivity map. Considering the constant monitoring of devices, it would also be convenient of having the program automatically show and update a connectivity map of all of the devices.

Another feature that OpenNMS could use would be an option to do an SNMP walk on a discovered device. That way it can be used for management and not just for monitoring purposes.

OpenNMS is excellent at monitoring the services and devices of a network, but it could also benefit from monitoring the actual traffic that is going through the network. For example, installing a network sniffer on the program that continuously monitors traffic and makes reports about where the heavy loads are being placed, and what services are being used the most. This could also be turned into an intrusion detection system that monitors for malformed packets, or known exploits through the network.

Results:

OpenNMS was successfully installed on a Redhat 9 distribution of Linux on a Dell Inspiron 8600. Several problems were encountered with the install script, including a lack of correctly configuring both Tomcat 4 and PostgreSQL. Using the OpenNMS documentation, troubleshooting guides, and some basic intuition, each of these problems were overcome and OpenNMS functioned as expected.

Some time was spent in the process of learning the basic functions of OpenNMS and how it can be used for management of a network. In particular, the reporting features and service monitoring features were explored.

Analysis:

Although there were several problems with the 'quick install' script for OpenNMS, the installation still can be considered to be a rather straight-forward task. The functionality of OpenNMS is quite unique and is comparable to many commercial network monitoring tools. The ability to automate the reporting process is especially useful in a corporate environment where reports are generated on a regular basis. Also, for administrative purposes, it is extremely useful to be able to send automated emails to the administrator (perhaps to a pager or other mobile device) so that the administrator will know when a device or service has difficulty.

Conclusions:

This lab concluded successfully and was very valuable in learning the process of installing and configuring a network monitoring utility. With the large-scale corporate networks that are currently becoming normal in our society, it is important to be able to quickly and efficiently monitor the services and devices on the network. This task is becoming increasingly difficult for network administrators. However, with the use of a tool such as OpenNMS, this task can be greatly simplified. Although this utility will by no means replace the necessity for the administrator to be watchful over a network, it does simplify the tasks of constant monitoring and automated several other tasks of the administrator.

This tool is extremely useful, but also has much room for growth and development. For example, it might benefit from having a way to not only monitor the network services and devices, but also monitor the actual traffic that is going over the network. For example, it might be beneficial to know that 75% of the traffic is being consumed by P2P downloads, so that the administrator can put a stop to such traffic and keep the network running smoothly. This task could be accomplished by adding on a module to sniff the packets going through the network, and analyze them to see what is actually happening. There are many other ways that this program could be enhanced, but overall, it is already a very powerful and time-saving utility.