Table of Contents

Open Source Network Administration By James Kretchmar
ISBN: 0-13-046210-1 Publisher: Prentice Hall Print Publication Date: 9/22/2003

Chapter 1. Introduction

1.1. Network Administration

In the past two decades the number of networked computers in the world has grown at an astonishing rate. In the mid 1980s when the personal computer became the hot new item, most people had no concept of connecting these machines. Today, a mere 20 years later, it is hard to imagine an organization with more than a few computers that does not connect them in some fashion. More often than not, these networks are also connected to the global Internet, allowing connectivity to any other Internet-connected machine anywhere else in the world.

When networking technology was not so widely used, there was little need for network management. There simply wasn't much to manage. And when things did go wrong, it was usually the hardware at fault. Today the hardware is much more reliable, and the problems are often caused by bad software or malicious users. At the same time, the need for reliability has increased. More people now rely on the network in order to accomplish their jobs; some networks even support life safety equipment. These two influences, the greater need for reliability and the fact that problems are more diverse and unpredictable than ever before, have led to a greater need for network management.

But even though the need is great, increased managability comes at a cost. Networked devices such as switches come in both managed and unmanaged flavors, and the managed ones are much more expensive. You must decide how much management capability you are willing to pay for based on the number of machines you will be connecting and the context and environment in which you will be connecting them. If you are connecting just three computers in your home, you will have little need for an expensive piece of hardware that has every management bell and whistle available. If something does go wrong, you can always reach over and unplug a problematic host. If instead you are connecting dozens of machines in a situation where you cannot simply remove one from the network by walking across the room or where you need be able to monitor the traffic levels of the connected devices, you probably will need at least some management capability from your network hardware. And obviously, if you are connecting hundreds or thousands of machines, you will have a definitive need for manageable devices so that you can ensure the stability of the network should a problem arise.

On the Massachusetts Institute of Technology (MIT) network, which has over 30,000 hosts connected to it, each host is attached to a managed network port.[1] As far as the field has advanced in the past few years, it is still the case that a single misbehaving host can cause a very large problem. Thus the ability to locate and potentially disable a problematic host is crucial.

[1] This is a little bit of a lie. On all of the networks that are run by the Network Operations team, every host is on a managed port. There a few networks that are run by individual labs, and the labs may choose to do things differently.

1.2. Why Open Source?

Every piece of software described in this book is **open source** software. What is meant by the term open source? In short, it refers to software whose source code is available to the public without restriction. It also means the software can be modified by anyone for any use and that the modified program can also be redistributed as open source. There is a much more detailed definition available at http://www.opensource.org/, though this is only one interpretation, of course.

Though open source software is free, it still carries a license that governs its use. Usually, this license is there to ensure that the software continues to remain open. Additionally, open source software can be sold even though it is simultaneously available free. For example, you can buy copies of the Red Hat Linux distribution even though you can also download it at no cost. Why would anyone pay for something that could also be obtained free? In the case of Red Hat software, it is because you are also paying for support service from Red Hat, which does not come free.

Open Source Network Administration By James Kretchmar
ISBN: 0-13-046210-1 Publisher: Prentice Hall Print Publication Date: 9/22/2003

1.2.1. The Price Is Right

The most obvious reason for using open source software is the price. Whereas other network administration tools can cost tens of thousands of dollars, open source software is available free. This isn't to say that you shouldn't pay for quality. If a tool comes along that is exactly the right tool for you, does everything you need, and greatly improves your ability to manage the network for a price that you can afford, by all means buy it. But if there is a tool available that is just as good or better and costs nothing, which one makes more sense?

The fact that the program is free has two subtle side effects. One is that you get to take the ultimate test drive. Not sure if a tool is right for you? No problem. Try it for as long as you like, and if you're not fully satisfied, you haven't paid a dime. The only loss is the time you invested in learning about the tool and setting it up. But increasing the knowledge and abilities of your staff in this manner is often a better investment than the one made in another company anyway.

The other benefit to using a piece of free software is that there is less pressure to stick with the product if a better one comes along. If you spent $20,000 on a tool last year but find something better this year, you may be tempted or pressured to stick it out with the old tool as long as possible.

1.2.2. Eggs in Your Basket

Buying a piece of software for which you do not have access to the source code is, at heart, a gamble. Imagine that something goes wrong with the software: A serious security vulnerability is discovered, or an irreversible change to your environment trips a bug that causes the software to stop working. If the company that produces the software is no longer in business or is unwilling to help, you're out of luck.

Here's an even more likely scenario: Imagine the company that produces the software is a very large company for which you are a very small customer. You have a support contract with the company, but if a problem is affecting *all* of the customers, the company's resources may be tapped out. Then who gets taken care of first? It's the big customers who spend lots of money. If you're a small customer, you will have to wait your turn, even if the problem is a critical one.

With an open source product, you have a fighting chance to deal with these problems on your own. This isn't to say it will necessarily be trivial. Some software problems are easy to solve and some are not. You may need a skilled programmer to help you out. But even if you do not have one on staff, you will still have the possibility of paying a consultant to help. Consider this option in contrast to having no recourse whatsoever.

One analogy is that buying closed source software is like buying a car with the hood welded shut, whereas with open source software, you can open the hood and poke around inside. Even if you can't fix anything on the car yourself, you can at least check the oil once in a while, and if an emergency does arise, you can have someone fix the car for you.

It is also true that because the popular open source packages are very widely used, a fixed version of the software will typically be available much faster than it will be for a proprietary product. The eyes of thousands of programmers tend to find bugs faster than the eyes of a dozen. Keep this in mind when choosing between different pieces of open source software. The more widely used the program is, the faster it will be repaired.

1.2.3. You Might Find You Get What You Need

Open source projects are often written by people who want to use the software themselves. Instead of relying on a marketing department to figure out what they want, the customers are writing exactly what they need. Since others also have access to the source, they can modify the software to fit their needs as well. And because the things you want to accomplish are probably also the goals of *someone* else out there, it is usually not hard find software that does what you want.

Open Source Network Administration By James Kretchmar
ISBN: 0-13-046210-1 Publisher: Prentice Hall Print Publication Date: 9/22/2003

Similarly, when a piece of open source software does not do what you want, it is often easy to modify it to meet your needs. Not only that, but you can modify it as quickly as is required. A case in point: We recently had a piece of open source software that had an unfortunate file size limitation in the code, but we needed to use the program right away to gather data about an ongoing operational problem. A programmer on staff was able to fix the problem overnight and it was ready to go the next morning. This was by no means an exceptional event; it has happened time and time again. But if this had been a commercial piece of software, it would have taken days at the very best, but more likely weeks or months to get a change like this implemented.

1.2.4. The Question of Quality

The most common fear about using open source software, especially in a critical context such as production network administration, is that it is somehow not as good as commercial software. If you're paying for it, it must be of higher quality, right? From experience, we say the answer is no. Many open source programs are just as good as or better than their commercial counterparts. Take Multi Router Traffic Grapher (MRTG), for example, which is described in Chapter 3. It is essentially the industry-standard tool for graphing bandwidth use on network links. When free software is used in an industry in which you can pay hundreds of thousands of dollars for a single piece of equipment, it must be doing something right.

Of course, just as there is both good and bad commercial software available, there is both good and bad open source software. Proponents of open source believe that the open source development model helps create better quality software. Common arguments include that the very large number of people working on these projects is beneficial as is the openness of the system, which prevents developers from hiding code that isn't really up to par. On the other hand, a greater number of developers does not necessarily lead to a higher quality product,[2] and some open source software, such as a device driver, is understood by so few people that the code goes mostly unread anyway.

[2] Read *The Mythical Man-Month* (Addison-Wesley, 1995) by Frederick P. Brooks, Jr.

It is worth pointing out that the people who write these programs are usually professional programmers. They either work in a context where their product does not need to be sold commercially, as is often the case for software that comes from universities, or work in the corporate world during the day and spend their off-hours working on these tools. The software quality only benefits from the fact that it is a labor of love.

1.2.5. Is It Secure?

One common criticism of open source software is that because anyone can read the source, it is easier for an attacker to find a vulnerability and exploit it. The expectation is that commercial software is more secure because the security holes are hidden. The truth, as has been proven repeatedly in the past few years, is that this is not the case. The Internet has seen very serious security problems in both open source and closed source software, some of which were responsible for very visible, Internet-wide problems. You probably read about a few of them in the papers. The only thing that differentiates the two is that the open source bugs tend to have fixes available much more quickly, which usually limits the amount of damage done.

This is where having the very large number of programmer eyes really pays off. Yes, the attackers can look for vulnerabilities in the software, but so can the good guys, and there are far more of them around. This means there's a better chance that a bug will be found and fixed than found and exploited. At the same time, the good guys will not devote much time to looking for vulnerabilities in proprietary programs, but the attackers will.

Open Source Network Administration By James Kretchmar
ISBN: 0-13-046210-1 Publisher: Prentice Hall Print Publication Date: 9/22/2003

1.2.6. Support

The downside to open source software is that it usually comes with no support service. If your open source tool fails, there is no one you can call to complain to and no one to take the blame. If it is important to you to have a scapegoat available when software fails, open source is not a good choice for you.

Instead, the first line of support for open source software is you and your staff. The more you have invested in learning about the software and its inner-workings, the better chance you have of solving a problem. After that, you can appeal to the software maintainers for a fix or consult online discussion forums related to the software. Should those fail, you can always resort to paying a consultant to help you.

Of course, if you can find open source software that *does* come with support service, you will have the best of both worlds.

1.3. Tools in This Book

The tools in this book cover many aspects of network administration, from traffic analysis to log monitoring. Some of the tools, such as MRTG and NetFlow, are very widely used and nearly industry standards. Other tools, such as Neo and Oak, cover areas of network administration in which there is not yet a single tool that most administrators use.

All of the tools described in this book are open source software, which means you may download and use them for free and you may modify them if you desire. The purpose of the book is to collect a good set of network administration tools in one place. Open source software developers are not known for spending serious time (or money) in self-advertising. So it can be a challenge to figure out which software is worthwhile and which is not. In addition to pointing out the good software, this book explains how the software works and how to install and use it.

The chapters are:

- **SNMP.** The Simple Network Management Protocol is the standard for remote administration of network devices. Chapter 2 includes a well-known set of tools for accessing information via SNMP.
- **MRTG.** The Multi Router Traffic Grapher is a very widely used tool for graphing bandwidth and other network statistics.
- **Neo.** This tool was written at MIT for high-level administration of switches, routers, and other devices that speak SNMP.
- **NetFlow.** NetFlow is a Cisco mechanism for collecting information about the internals of network traffic. Chapter 5 describes a well-known package called Flow-Tools that collects and processes NetFlow information.
- **Oak.** Oak is a tool written at MIT for collecting syslog messages from servers and network equipment, condensing the information as appropriate, and notifying operators of problem conditions when they arise.
- **Service Monitoring.** The Sysmon program, covered in detail in Chapter 7, tests network hardware and server software to ensure they are functioning, and if they are not, it notifies the appropriate administrators. The Nagios program, a more complex tool that serves the same purpose, is briefly discussed as well.
- **Tcpdump.** This is a standard program for directly analyzing network traffic at the packet level.
- **Basic Tools.** Chapter 9 covers the basic tools of network administration, including the ping, telnet, netcat, traceroute, MTR, and netstat programs.
- **Custom Tools.** In Chapter 10, a brief, working knowledge of the Bourne shell and Perl scripting languages is presented.

If you encounter a bug in any one of the programs described in this book or find that a particular feature would be be a real benefit, do your part and mail the maintainers of the software. Don't be shy; they want to hear your feedback. However, do not expect that your problems will be solved overnight.

Open Source Network Administration By James Kretchmar
ISBN: 0-13-046210-1 Publisher: Prentice Hall Print Publication Date: 9/22/2003

1.4. Environment

All of the examples given in this book were performed on machines running some flavor of the Unix operating system, in particular Solaris and Linux. All of the programs should work on other flavors of Unix as well, but the more exotic the variant, the more likely you will run into problems. A number of the programs advertise that they work under Windows too, and now that modern versions of MacOS run Unix under the hood, it should be possible to build the tools for those platforms as well.

1.5. Background

This book assumes you have an understanding of the basics of networking and does not go into detail about the Open System Interconnection (OSI) layered network model and other topics frequently included in books on networking. However, when an aspect of the underlying technology is particularly relevant, it is described in enough detail so that only a general familiarity with the material is necessary.

Each of the tools here is built from source, with explicit demonstrations in the text. The tools should build without much trouble, but a build process does occasionally fail. If it does, you may need some experience with building software in order to fix the problem. Often, it is simply a matter of instructing the build system to look for software in an unexpected location on the machine or of building a separate, required package that is not already installed on the machine. Sometimes the problem will be an error in the build system itself, in which case the maintainer of the software should be given a detailed report of the problem.

1.6. Terminology and Conventions

The terms *workstation*, *host*, *device*, and *node* are all used interchangeably in networking and in this text as well. The use of one instead of another is not meant to imply anything about the hardware in question, though the choice of word may reflect an expectation of common use.

Open Source Network Administration By James Kretchmar
ISBN: 0-13-046210-1 Publisher: Prentice Hall Print Publication Date: 9/22/2003