

Securing Your NuSphere Installation on Red Hat Linux

By Paul Adams, System Administrator

TABLE OF CONTENTS

Basic Ideas Behind
Security

Nuts and Bolts

Turning Off Services

System Configuration

About NuSphere

When you set up a new Red Hat Linux machine running NuSphere, one of the first things you should do is make the installation secure against intruders. (This is best done prior to setting up the network connection.)

The world of the Internet is not as benign as one might hope, and any machine sitting out there unprotected will find itself vulnerable to an array of attacks: both directed attacks from people who may have a grudge against your site or company and want to damage it or steal information; and random attacks from strangers who notice that your machine is vulnerable and seize the opportunity. You're not choosy, though—you want to prevent all unauthorized access, no matter from whom. There are a number of steps you can take to make your machine more secure.

There's no such thing as perfect, impenetrable security—in computer networks or in the real world—but a lot can be done to close up the big obvious holes in a system and make it a difficult and unappealing target. The higher your profile, and the more valuable the information on your system, the more rigorous a security policy and setup you want to have. If, for example, you are running a multinational bank on NuSphere, you may want to spend a while tightening your system.

Keep in mind that, in addition to the issues discussed here, there are a variety of safeguards you should put in place to make sure your MySQL server and data are safe. The NuSphere Tech Library has other papers that talk in detail about securing your operating environment.

Basic Ideas Behind Security

Before you start locking down your system, you should develop a general security policy—what you want to achieve, how much risk is acceptable, how much you trust the people you are allowing to use the system, and so forth. These guidelines will be helpful when you are making decisions about security in the future.

The fundamental principle of security is to permit as little access as possible. In practice, it's necessary to achieve a balance between maximizing security—allowing nobody to do anything on the system—and maximizing freedom, which comes close to allowing anyone to do anything. There are a number of standard procedures in place to help you achieve this balance. The safest way to proceed is to start with the tightest possible security and then loosen it only where necessary. This philosophy is often encapsulated with the phrase "that which is not expressly permitted is prohibited."

The first security measure that most people encounter on a Linux system is the username/password combination. Each person who uses a Linux machine is assigned a username and password. This has the dual purpose of keeping unauthorized users at bay and keeping authorized users separate from each other. It operates on the principle that it's hard to guess somebody's password, which is true to a finite extent. A password like "grandma" is much easier to guess than something like "ynZ8e5oD6". There are software tools, such as Crack (<http://www.users.dircon.co.uk/~crypto/download/c50-faq.html>), which are designed specifically to guess passwords, and they are remarkably effective. So making your passwords extremely difficult to guess is highly desirable.

You can use tools that come with Red Hat Linux to enforce good password policies among the users of your system: set passwords to expire after a certain amount of time, require a mix of letters and numbers in passwords, and so forth.

Each user is assigned certain permissions by default—files and directories that they can and can't access, and programs that they can and can't run. This is the nature of a multi-user operat-

ing system, and the system allows everybody to keep his or her files private. It also keeps users from modifying or damaging parts of the system they're not supposed to, either accidentally or intentionally. The majority of system-administration tasks are permitted only to the **root** user. For this reason, "getting **root**" access is a primary goal of computer system intruders.

Nuts and Bolts

When securing your machine, it's important to isolate what task or tasks the machine will be performing, and streamline it to perform those tasks and only those tasks. This is an offshoot of the general security principle: run as little as possible thereby minimizing possible entrances to the machine.

The first thing you should do after installing Red Hat is to check the Red Hat Errata Page (<http://www.redhat.com/support/errata/index.html>) for updates and security advisories that have been issued since the version you installed was created. The errata page for Red Hat Linux 7.1 is <http://redhat.com/support/errata/rh71-errata.html>. Make sure to download and install all of the errata for your system.

But even after you install all of the errata, it's important to keep on top of new changes that are added to the errata page. You can use the Red Hat Update Agent, **up2date**, to automate downloading and installation of these files.

Turning Off Services

Out of the box, Red Hat 7.1 is set up to run a number of services that are probably unnecessary. Unless you explicitly need a particular service, such as **telnet**, it should be turned off. Each service you run gives intruders additional opportunities to gain access to your system. During installation of the operating system, you have the option to choose what runs by default. Choose as little as possible. Some services, such as **exec** and **shell**, are generally considered insecure and should never be run.

A NuSphere installation includes, at a minimum, the Apache Web server, PHP, Perl, and MySQL. Depending on the exact NuSphere product you have chosen, additional applications may be included. If a service on your Linux machine doesn't directly contribute to one of the applications installed with your NuSphere package, ask yourself why you'd need it, and if you don't need it, turn it off. Typically, if you are running NuSphere on a machine dedicated to serving database-driven Web sites, you can turn off NIS and NFS, Sendmail, and BIND.

Services are started automatically in two ways. Most services are started when the system boots up, by files in the `/etc/rc.d/rc.d` directories. There is an `rcX.d` directory for each runlevel: `rc3.d` controls the services that are started when you boot into runlevel 3, `rc5.d` handles runlevel 5, and so forth. In these directories you will see something like `S20rusersd`. `rusersd` is the name of the service—in this case, a daemon that provides information about currently logged-in users. `S` means that the service should be started at boot time, and the number 20 tells when in the boot process this particular service should be started. To disable the service for this runlevel, rename the file so the `S` is lowercase: `s20rusersd`. That will take it out of the running. Do this for all the services you don't need, and do it for every runlevel you're likely to boot into—typically 3 and 5. You can also use the built-in tool **ntsysv** to turn services on and off.

In addition to the services launched at startup from the `/etc/rc.d` directories, a particular batch of services run under the auspices of **inetd**. **inetd** is the daemon that listens to Internet ports and handles network connections. Take a look at the file `/etc/inetd.conf`. This is the configuration file read by **inetd**. You will see various entries, each on its own line, telling **inetd** how to deal with different services, for example:

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

The first word on each line is the name of the service. To prevent unwanted services from running, comment out the relevant lines by adding a pound sign (`#`) to the beginning of the line. You should turn off **chargen**, **daytime**, **echo**, **exec**, **login**, **shell**, and **time**. Unless you explicitly

need them, you should also turn off **pop2** and **pop3**, **imap**, **ftp**, **telnet**, **swat**, **auth**, **bootp**, **talk**, **uucp**, **comsat**, **finger**, **systat**, and **netstat**.

Note: the changes you make to `inetd.conf` won't take effect until you restart **inetd**, by typing:

```
/etc/rc.d/init.d/inetd restart
```

For many services, you will find that there's a more secure alternative. **SSH** is a secure alternative to **telnet**; likewise, **SFTP** is a secure alternative to FTP. Check out <http://www.SSH.com/> and <http://www.OpenSSH.com/> for information on these protocols.

System Configuration

TCP Wrappers

Using Red Hat Linux's built-in TCP Wrappers system, it's easy to control what remote users have access to what services. This is done using two files: `/etc/hosts.allow` and `/etc/hosts.deny`. `hosts.deny` is a list of IP addresses that are not allowed access to services on your machine; `hosts.allow` is a list of those that are. These lists allow you to limit access to particular services as well as globally. The best policy is to have a `hosts.deny` file that reads as follows:

```
ALL: ALL
```

This denies access to all services, to anybody who tries to connect to them. This blanket denial can then be overridden on a case-by-case basis with the `hosts.allow` list. If you want anyone to be able to connect to your Web server, but only machines on your local network (that is, machines with IP addresses 192.168.x.x) or from your domain to have direct access to **SSH**, your `hosts.allow` file might include lines like this:

```
httpd: ALL
```

```
sshd: mydomain.com, 192.168.
```

TCP wrappers have other powerful features, including the ability to log access. Look at the **tcpd** man page for details.

Binary Permissions

Certain executable files are given an **SUID** or **SGID** bit—an instruction that allows them to run with enhanced privileges, even when a non-privileged user starts them. Sometimes this is necessary, but such executables are a target for people trying to exploit your system—they provide an easy avenue for running unwanted code at a high level. You should make an effort to deny these enhanced permissions in all except the most essential cases. To compile a list of executables on your system that run with **SUID** or **SGID** permissions, run the following command as root:

```
find / -type f -perm +6000 > ~/suidlist
```

That will take a few minutes to search your system, and then generate the list as a file called `suidlist` in root's home directory. To remove **SUID** or **SGID** permission from a particular file, run

```
chmod -s filename
```

If you're not going to be using a particular package, though, it's not a bad idea to remove it altogether, rather than just removing its permissions. The Red Hat Package Manager can be used to remove packages cleanly:

```
rpm -e packagename
```

Firewalling

You can greatly increase the security of your NuSphere environment by placing the machine running NuSphere behind a firewall. A firewall is a mechanism that separates the machines on your internal network from the Internet at large. Its goal is to permit only certain (safe) types of traffic, by establishing rules as to what is allowed to pass the firewall in each direction, and what

is blocked. Linux makes it easy to set up such a machine, as the creators have been kind enough to build packet filtering mechanisms into the kernel.

Important Note: A firewall is an additional measure to, *not a substitute for*, the other security precautions in this article.

To set up a simple Linux firewall, you will need a machine with two network connections—one to the outside world, and one to the internal network that you want to wall off. Because the task it has to perform is so simple, the machine does not have to be particularly powerful. Secure the machine by shutting down all unnecessary services. There's no need for any user accounts on this machine other than root.

The firewalling code built into the Linux 2.2 kernel is called **ipchains**. This has been replaced in version 2.4 by **iptables**, which offers a number of improvements.

Good instructions on configuring a firewall using **ipchains** are provided in the **ipchains** HOWTO (<http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html>). An excellent tutorial for working with **iptables** can be found at <http://www.boingworld.com/workshops/linux/iptables-tutorial/>. And you can learn the basic ideas behind firewalling from the firewall HOWTO (<http://www.linuxdoc.org/HOWTO/Firewall-HOWTO/>).

MySQL operates on server port 3306 by default. This port should most definitely be blocked by your firewall to all but the most trusted hosts. Apache runs on port 80, and secure HTTP runs on port 443. You will want to create firewall rules to control those ports in particular, and any other ports your system may be using. A list of all registered ports and their uses can be found at <http://www.iana.org/assignments/port-numbers>. This is from the Internet Assigned Numbers Authority (<http://www.iana.org/>).

Keeping Abreast of Security

The final step, once you've got your system locked down, online, and operating, is maintenance. New ways to exploit existing security measures are constantly being developed. CERT (<http://www.cert.org/advisories/>), Bugtraq

(<http://www.securityfocus.com/frames/?content=/forums/bugtrac/intro.html>), and other groups provide forums where newly discovered holes and vulnerabilities are publicized. Anyone who is serious about keeping a system secure should subscribe to a mailing list or two and be sure to patch any holes as soon as they are exposed.

Depending on the NuSphere product you have installed, you will find a variety of security tools on your system. For example, the Access Manager for Apache, which is installed with all NuSphere products, offers an easy browser-based method of limiting access to your Web sites. The documentation for the Access Manager can be found by choosing the Documentation link after starting the NuSphere Administration Web Site. Since new tools are added on a regular basis you should check the NuSphere site frequently for updates.

For further information on Linux security, check out the Security HOWTO (<http://www.linuxdoc.org/HOWTO/Security-HOWTO.html>), or any of the numerous excellent books on the topic. You should also check on a list of security tips from the Apache group (http://httpd.apache.org/docs/misc/security_tips.html).

About NuSphere Corporation

NuSphere delivers the first Internet Application Platform (IAP) based on open source components, providing an integrated foundation that allows companies to deploy reliable, cost-effective, enterprise-class applications across Windows, UNIX and Linux environments. NuSphere® Advantage is an integrated software suite that pairs the reliability and cost-effectiveness of PHP, Apache, Perl and open source databases with new technology for building business-critical web applications and web services. Based in Bedford, Mass., the company's commercial software services include technical support, consulting and training. For more information, visit www.nusphere.com or call +1-781-280-4600.

NuSphere is a registered trademark in Australia, Norway, Hong Kong, Switzerland, and the European Community; NuSphere and PHPEd are trademarks of NuSphere Corporation in the U.S. and other countries. Any other trademarks or service marks contained herein are the property of their respective owners.

MySQL AB distributes the MySQL database pursuant to the applicable GNU General Public License that is available as of the date of this publication at <http://www.fsf.org/licenses/gpl.txt> and all of the terms and disclaimers contained therein. NuSphere Corporation is not affiliated with MySQL AB. The products and services of NuSphere Corporation are not sponsored or endorsed by MySQL AB.