How to set up a chroot environment with RedHat Linux 6.2

Harald Welte laforge@gnumonks.org

1.2, 2000/10/14 20:32:22

This document describes how to set up a chroot() environment for a combined web- and ftp server using RedHat 6.2, Apache and ProFTPd. Even if You want to run something else inside of a chroot() it might be interesting.

Contents

1 Introduction

1.1 What is a chroot() environment?

All flavors of UNIX provide a systemcall, called chroot(). This changes the root point of the filesystem as it appears to the current process to another directory than "/".

For more information see the chroot(2) manpage of Linux

1.2 What do I need a chroot() enviroment for?

Basically, it is a security enhancement. The process(es) running inside the chroot system exist inside another environment than those outside. The only thing "inside" and "outside" share is the kernel. All libraries, utilities,

If You want to provide other users some kind of access, but hide most of the system from them, chroot() is Your friend.

1.3 How secure is a chroot() environment?

This depends on the setup. As long as a process is running as root, it can always leave the chroot().

A process running as any user other than "root" cannot leave the chroot() by any means. But if there is a local-root exploit (i.e. the process can gain superuser access due to a bug in the operating system),

So as long as Your server software is not too buggy and the operating system has no exploitable security holes, the chroot() environment is safe.

1.4 What do I need inside a chroot() environment?

You need everything the process (in case of this howto the apache webserver and the proftpd ftp server) needs beside the kernel.

Usually this inclues

- C library (libc/glibc)
- Resolver for nameservices (DNS, NIS, ...)
- Authentication information (/etc/passwd, group, shadow, ...)
- Additional libraries like libz, ...
- If You want to run CGI scripts, additionally
  - Shells (bash, tcsh, zsh)
  - Perl, Python, whatever scripting language the cgi's use

1.5 How do I get all this stuff into my chroot()

Well, the first and obvious thing is to copy all the neccessary files from the real-root to the chroot. This is what many people still do today. This has some inherent disadvantages:

- You don't really know all the files You need, so You have to do a trial-and-error approach
- You are not able to update the chroot() environment easily, because if You update Your "outside" system, the old files still remain in the chroot().

So the more clean approach is to install RPM itself inside the chroot() and afterwards install the whole system using the ready-made binary RPMs. This is the same like what the normal RedHat installation procedure does if You are installing on a new system.

2 Installing the chroot() environment using rpm

2.1 Pre-rpm preparations

At first You have to create the chroot directory (called $ROOT in this document)

In order to make RPM work, it'll need the rpm database directory beneath $ROOT, so You have to create it using

```
mkdir -p $ROOT/var/lib/rpm
```

2.2 Installation of the RPM packages

All the packages are to be installed using

```
rpm --root $ROOT [filename]
```

2.2.1 RPM packages from the RedHat distribution

The list of packages I suggest to install (in chronological order):

```
setup
filesystem
basesystem
ldconfig
glibc
mktemp
termcap
ncurses
info
fileutils
bash    && libtermcap
gdbm
tcsh
perl
chkconfig
textutils
mailcap
apache
mod_perl
cracklib
cracklib-dicts
pwdb
glib
gawk
pam --nodeps
```

```
shadow-utils
slang
util-linux --nodeps
sh-utils
zsh
sed
bzip2
popt
zlib
rpm
```

2.2.2 RPM packages from other sources

As ProFTPd is still not included in the RedHat distribution, I got the latest RPM's from the ProFTPd site http://www.proftpd.net and installed them

```
proftpd-core
proftpd-standalone
```

2.3 Post-RPM installation tasks

After the installation of all the packages we have to do some configuration.

2.3.1 Authentication Information

As the authentication and uid/gid resolving is done in userspace, we need a valid authentication source inside our chroot() environment. Traditionally this is information is stored in $ROOT/etc/passwd and $ROOT/etc/group, but You can configure any source using PAM and a MySQL / LDAP / whatever auth module.

This HOWTO covers just the traditional case.

So create all the users and groups You need in $ROOT/etc/group and $ROOT/etc/passwd :)

2.3.2 Name service

Apache and ProFTPd may want to resolve hostnames to ip addresses and vice-versa,so You have to configure at least one valid source for name service lookups. Usually DNS is used, so just enter the DNS information into $ROOT/etc/resolv.conf.

2.3.3 Webserver / FTP server config

Just configure apache / ProFTPd / ... the same way You would configure them if they were running on a normal system. Just always remember that the configfiles are paresd from a process running already inside Your $ROOT. So all absolute pathnames refer to $ROOT instead of the real-root.

2.3.4 Starting the server(s)

Server startup is done inside the chroot() environment.

If You've installed a shell inside Your chroot(), just run the following command as root

```
chroot $ROOT
```

You are now running a shell inside Your chroot(). Just try to "cd /" and look around. You are jailed to the chroot() directory. Start Your apache using the "httpd" command and proftpd using the "proftpd" command. Then exit Your chroot() using the "exit" command and look if the servers are running.

Attention: You cannot bind to a port which is already in use outside the chroot().