SETTING UP CVS SERVER

http://www.kryptonians.net/cvs/server_setup.html

Get the source code for CVS. (http://ximbiot.com/cvs/cvshome/)

Configure using **./configure --prefix=/usr.** This sets the install directory. When you run *make,* it will create *bin, lib, man* and *info* directories using /usr as the root of its installation tree.

If you get the newest version of CVS, you won't need to patch it to correct the annoying ***.cvsignore*** bug as I had to (see the section below, "Fixing the Source Code". If you're running an older version, make the changes to the source code as describe at the bottom of this page. This will correct a couple of annoying bugs which only surface if you're setting up CVS as a server.

Modify */etc/services* to include the line:

services:cvspserver 2401/tcp # CVS protocol server

This identifies the port CVS server will use. In most Linux/Unix installations, this is already defined. Port 2401 is the commonly accepted default port for CVS use.

Modify *etc/inetd.conf* to include the following lines :

```
# Allow CVS to be accessed as server in a client/server relationship
cvspserver stream tcp nowait root /usr/bin/cvs cvs -f -l --allow-root=/raid/cvsroot
```

A couple of important notes here:

1. That second parameter is a lowercase 'L', not a 1 or an uppercase 'i'.

2. The last parameter, '*--allow-root=/raid/cvsroot*' defines the location of the repository to be used. There can be more than one, in which case each would have its own '*--allow-root*' parameter.

The following steps must all be done as root:

Using **linuxconf,** create a new group called 'cvsadmin'

Using **linuxconf,** create a new user called 'cvsuser'. This is the user account that CVS server will use to perform all requested actions on the repository/repositories.

Create your repository directory. In the above example, this would be */raid/cvsroot*.

Using **chmod,** set write permissions so that *cvsuser* can do anything he wants to.

chmod -R 775 /raid/cvsroot

should do the trick. Do the same thing with **chgrp,** changing group ownership to 'cvsadmin' for the repository and everything underneath.

In the repository directory, create a directory called *locks*. By the above example, this would be */raid/cvsroot/locks*.

Type the following command:

cvs -d /raid/cvsroot init [return]

This will initialize the repository, creating a directory called CVSROOT within it. Do this for each repository you wish to set up, substituted the desired pathnames as needed.

Go to your home directory and type the following command:

cvs -d /raid/cvsroot checkout CVSROOT [return]

This checks out the CVSROOT directory and all its contents. CVS uses itself to perform revision control on its own configuration files. Neat-o!

Using your favorite text editor, modify your copy of CVSROOT/config so that its contents look like this:

```
# Set this to "no" if pserver shouldn't check system users/passwords
SystemAuth=no

# Set `PreservePermissions' to `yes' to save file status information
# in the repository.
PreservePermissions=no

# Set `TopLevelAdmin' to `yes' to create a CVS directory at the top
# level of the new working directory when using the `cvs checkout'
# command.
TopLevelAdmin=yes
```

Note that first parameter. It tells CVS Server not to use the regular userbase for passwords, allowing us to define a separate userbase for CVS.

Create a file in /root/CVSROOT called 'passwd'. (The way I did it was to copy the one from /etc. If you're using shadowed passwords as most people do these days, you'd be copying the '/etc/shadow' file instead.) Using a text editor you absolutely can't stand, remove each entry that represents somebody (or something) that won't be accessing CVS server. What is left will be a list of CVS server users. (Do not include 'cvsuser' in this list). On each line, remove everything after the encrypted password for each user, so that the lines look something like this:

yadayada_username:WEOI$(@(@(#&JD(:

Don't forget that trailing colon! Then add the word 'cvsuser' to the end of each line, so that it looks something like this:

yadayada_username:WEOI$(@(@(#&JD(:cvsuser

This tells CVS Server to change this user to 'cvsuser' as soon as the login for a user is recognized.

Another way to create this file is to use **htpasswd,** which is normally used to generate Apache access password files. The format it creates is about the same as that which CVS requires, except for the **:cvsuser** appendage to each line. You'll still have to do that manually, but it allows you to create a passwd file with passwords different that those in the regular /etc/passwd file.

Modify the file 'cvswrappers' so that it looks like this sample.

```
# This file affects handling of files based on their names.
#
# The -t/-f options allow one to treat directories of files
# as a single file, or to transform a file in other ways on
# its way in and out of CVS.
#
# The -m option specifies whether CVS attempts to merge files.
#
# The -k option specifies keyword expansion (e.g. -kb for binary).
#
```

```
# Format of wrapper file ($CVSROOT/CVSROOT/cvswrappers or .cvswrappers)
#
# wildcard [option value][option value]...
#
# where option is one of
# -f from cvs filter value: path to filter
# -t to cvs filter value: path to filter
# -m update methodology value: MERGE or COPY
# -k expansion mode value: b, o, kkv, &c
#
# and value is a single-quote delimited value.
# For example:
#*.gif -k 'b'
*.gif -k 'b'
*.tga -k 'b'
*.bmp -k 'b'
*.psd -k 'b'
*.tif -k 'b'
*.png -k 'b'
*.iff -k 'b'
*.aiff -k 'b'
*.obj -k 'b'
*.dat -k 'b'
*.exe -k 'b'
*.com -k 'b'
*.dll -k 'b'
*.dsw -k 'b'
*.dsp -k 'b'
*.lwo -k 'b'
*.lws -k 'b'
*.p -k 'b'
*.ico -k 'b'
*.frx -k 'b'
*.class -k 'b'
*.jar -k 'b'
*.zip -k 'b'
*.lzh -k 'b'
*.lha -k 'b'
*.rar -k 'b'
*.arj -k 'b'
*.arc -k 'b'
*.avi -k 'b'
*.mov -k 'b'
*.asf -k 'b'
*.smk -k 'b'
*.jpg -k 'b'
*.mpg -k 'b'
*.swf -k 'b'
*.frx -k 'b'
*.fli -k 'b'
*.flc -k 'b'
*.tiff -k 'b'
*.bin -k 'b'
```

```
*.dat -k 'b'
*.wad -k 'b'
*.ppt -k 'b'
*.pdf -k 'b'
*.3ds -k 'b'
*.max -k 'b'
```

Run the commands

cvs -d /raid/cvsroot add /root/CVSROOT/passwd [return]

cvs -d /raid/cvsroot commit /root/CVSROOT [return]

Be sure not to reverse the order of these commands, as that would lock you out of CVS!!!

Restart the inet daemon, using this command line:

/etc/rc.d/init.d/inet restart [return]

That's it. If you've done everything right and I haven't forgotten anything, CVS Server should now be large and in charge.

Now all you have to do is make sure WinCVS is working on all the client workstations. If WinCVS can't connect to the server or you get some odd message, chances are good you've screwed up /etc/inetd.conf, write permissions on the repository(ies), forgotten to restart inetd, failed to commit your changes to CVSROOT or botched the password file.

Fixing the Source Code

The source code for the version of CVS currently running as a CVS server on my server is located here and and has been modified slightly from the distributed version. These modifications specifically center on the use of a file called *.cvsignore*, which it expects to see in the user's home directory. When operating in server mode, CVS does not have direct access to the client's home directory and so attempts to access the file in /root, an act for which it does not have permission. This generates an annoying and misleading error message, when in fact nothing is wrong.

Another change allows the root account to check files in and out of a repository. Considering that initial setup for a CVS server requires that the installing user have root access or actually *be* root, this caused undue problems.

The first change is to ***ignore.c***, to a routine called ***ign_setup(),*** which attempts to access various lists of file types to ignore during normal operation. Here is the code to look for in the source file:

```
/* Then add entries found in home dir, (if user has one) and file exists */
/* Only do this, however, if we're not the server. Trying to do this as the
server makes no sense, and generates a useless, not to mention
confusing, error message. - GT 07/05/2000 */

home_dir = get_homedir ();

if (strcmp(command_name,"server")!=0) <-------- THIS IS THE PART I ADDED.
if (home_dir)
{
```

```
char *file = xmalloc (strlen (home_dir) + sizeof (CVSDOTIGNORE) + 10);
(void) sprintf (file, "%s/%s", home_dir, CVSDOTIGNORE);
ign_add_file (file, 0); free (file);
}
/* Then add entries found in CVSIGNORE environment variable. */
ign_add (getenv (IGNORE_ENV), 0); /* Later, add ignore entries found in -I arguments */
```

The other modification is to ***options.h*****.** Be careful with this, as ***options.h*** is recreated each time you run the ***configure*** script for the project. I turned off CVS_BADROOT, so that CVS would permit operations to the repository by *root.*

I understand that since this tutorial was written, this silly bug in ignore.c has been fixed, so you may or may not need to make this code change yourself.

SPECIAL NOTE ABOUT TYPES OF CHECKOUT:

CVS can use either *reserved checkouts,* or *unreserved checkouts.* A *reserved checkout* is what SourceSafe does by default. This situation is best for allowing only one programmer access to a source file at a time. In effect, files are 'locked' until they're 'unlocked' (and there are controls in WinCVS for this). CVS, on the other hand, defaults to *unreserved checkouts,* which is similar to SourceSafe's *multiple checkout.* It is possible to set a CVS repository so that simultaneous work on a single file by multiple programmers isn't possible, but the question of whether this is desirable or not is left as an exercise for the reader.