Running xinc on OpenBSD's Apache Server

Asked 16 years, 4 months ago Modified 16 years, 3 months ago

Viewed 664 times Part of PHP and CI/CD Collectives

11

Has anyone been able to get xinc to run correctly under OpenBSD's chrooted default Apache? I'd like to keep our development server running fully chrooted just like our Production server so that we make sure our code runs just fine chrooted.

PHP CI/CD php continuous-integration openbsd xinc

Share

Improve this question

Follow

asked Aug 13, 2008 at 3:47

dragonmantank

15.5k ● 22 ● 85 ● 92

4 Answers

Sorted by: Highest score (default)

4

Have you posted the issue on the Xinc bug tracker? Xinc itself should run fine as it runs both as a daemon and as a web app. As you alluded to, the issue may be that the daemon is not running in a chroot'ed environment where as the web interface is, leading to either side not grabbing the files.

Share Improve this answer Follow

answered Sep 22, 2008 at 8:26

dcousineau
2,212 ● 18 ● 24

▲

2

▼

@dragonmantank

In Xinc's case, I hope you used PEAR to install it.

```
pear list-files xinc/Xinc
```

This should do it, and show you where your Xinc install put its files. So even though Xinc is "just" one big PHP script, it's still spread out into rc scripts and all those other things which are necessary to make an application run. I'm sure you don't need to add all paths listed there, but probably some in order to make it run.

Aside from Xinc itself, I think it also needs phpUnit and a bunch of other PEAR libs to run, so what I'd propose is this:

```
pear config-get php_dir
```

And then you need to add that path (like Henrik suggested) to the chroot environment.

Share Improve this answer

Follow

answered Aug 31, 2008 at 17:04

Till

22.4k ● 4 ● 60 ● 89

1

Having never used xinc myself, I can only hint as to how I usually get to chrooting apps.

First step would be to gather information on everything the app needs to run; this I usually accomplish by running systrace(1) and ldd(1) to find out what is needed to run the software.

Go through the output of

```
systrace -A -d. <app>
ldd <app>
```

and make sure that everything the app touches and needs (quite a lot of apps touch stuff it doesn't actually need) is available in the chroot environment. You might need to tweak configs and environment variables a bit. Also, if there is an option to have the app log to syslog, I usually do that and create a syslog socket (see the -a option of syslogd(8)) in order to decrease the places the app needs write access to.

What I just described is a generic way to make just about any program run in a chroot environment (however, if you need to import half the userland and some suid commands, you might want to just not do chroot :). For apps running under Apache (I'm sure you're aware that the OpenBSD httpd(8) is slightly different) you have the option (once the program has started; any dynamic libraries still needs to be present in the jail) of using apache to access the files, allowing the use of httpd.conf to import resources in the chroot environment without actually copying them.

Also useful (if slightly outdated) is this link, outlining some gotchas in chrooted PHP on OpenBSD.

Share Improve this answer Follow

answered Aug 24, 2008 at 11:57

Henrik Gustafsson
53.9k ● 9 ● 50 ● 60

1

> First step would be to gather information on everything the app needs to run; this I usually accomplish by running systrace(1) and ldd(1) to find out what is needed to run the software.

I'll give this a try. The big issue I've found with xinc is that while it is a PHP application, it wants to know application installation paths (yet it still spreads stuff into other folders) and runs some PHP scripts in daemon mode (those scripts being the hardest to get running). So, for example, I told it to install to /var/www/xinc and then made a symlink of

/var/www/var/www/xinc -> /var/www/xinc

and it partially worked. I got the GUI to come up bit it refused to recognize any projects that I had set up. I think the biggest problem is that part of it is running a chroot and the other half is running outside.

If all else fails I'm going to just have to build something as we program inside chrooted environments since our production is chrooted. We've run into issues where we code outside of a chroot and then have to back track to find what we need to make it work inside a chroot.

Share Improve this answer Follow

answered Aug 24, 2008 at 15:21

dragonmantank

15.5k ● 22 ● 85 ● 92