LinuxFocus article number 260
http://linuxfocus.org

by Georges Tarbouriech
<georges.t(at)linuxfocus.org>

About the author:

Georges is a long time Unix user. He is very fond of the free versions of this great system.

FreeBSD, a real alternative

Abstract:

Yes, you are reading LinuxFocus! However, other free Unixes exist and they are of much interest and full of great features. This article is an overview of another free Unix: FreeBSD.

Why FreeBSD ?

When talking about free Unixes, people always "think Linux". Sure, Linux is widespread and everybody (or almost) has heard about it. The number of new users keeps growing and the medias (either "specialized" or not) make much "noise" around it. One of the main reason of this success comes from the fact that Linux especially runs on Intel machines, that is the most widespread computer architecture. Of course, it can run with many other processors, since one of the idea behind Unix was "we don't care about processors". Nevertheless, Linux would probably not have been as successful if developed for a more "confidential" processor, or, at least, it would have taken much longer.
But free Unix is not limited to Linux: the BSD "family" is there too. There are three main flavors: FreeBSD, NetBSD and OpenBSD. BSD stands for Berkeley Software Distribution. If you want some more information about Unix history, you can take a look at article 176 (Be advised that this article does not talk about the differences between BSD and System V: a book would be necessary for that.)
Each flavor has something specific:

1. NetBSD works under every platform on earth (or almost), including very old computers: let us say something like 30 or 40 different platforms. NetBSD provides the evidence of what was mentioned earlier: Unix does not care about processors. It is the right choice if you have a nice old "piece of junk" at home or at work (but not only, of course!). Personally, I used it a few years on Amiga or Intel machines and was very happy with it.
2. OpenBSD is well known as one of the most secure Unix. It is ideal for servers and , of course, it also runs with different processors. It is probably one of the most used system for servers on Internet. It is unlikely to be found at home, however, everyone does what he/she likes...
3. FreeBSD is a "mixture" of the previous ones. It is especially dedicated to Intel and Alpha processors and "inherits" a part of the security philosophy from OpenBSD. Like for Linux, the fact

it runs with Intel processors makes it one of the widespread BSD’s. It is a good choice for the home user. This does not mean you cannot use it at work!

There is something else concerning FreeBSD: these people "share" their work with Apple for the free software part of Mac OS X. And, since I am very fond of OS X... By the way, do not get confused: even if OS X is based on BSD, the kernel has nothing to do with the FreeBSD one. It is a Mach kernel, already used on NeXTstep, "father" of OS X: obviously, since NeXTstep was born at the end of the 80’s, this kernel evolved a bit.
Now we met... let us go a bit deeper.

Finding and installing FreeBSD

Like most of the free software, FreeBSD can be downloaded from the FreeBSD web site. Obviously, this requires a fast connection. Don’t worry: you can find CDs everywhere in the world.
The present release (at the time of this writing) is 4.6. The article talks about the i386 release.
Installing FreeBSD from CD-ROM (or other) is simple... even if there is no graphical interface to do so! You will use ncurses instead. Just follow the instructions and FreeBSD ends on your hard disk. As usual, you will have to partition your hard disk, configure a few things such as networking or even your kernel (optional). Next you will meet sysinstall, a very good tool doing much of the install work, as the name says. Whether you install the whole bunch of packages or only some of them is your choice. You can add them at any time, anyway.
The management of packages is also obvious. You can use the pkg commands, like under Solaris, for instance, or you may prefer the ports. In short, the pkg commands are like rpm’s or deb’s for Linux, while the ports are archives to be compiled with make and make install. In other words: nothing complex or difficult.
Since we compare to Linux, let us add there is a Linux compatibility package that you should install: it will allow you to compile and run Linux applications.
Again, everything is obvious and, just in case, a very good handbook is provided, explaining each step. Furthermore this last exists in various languages.
Accordingly, there is no need to spend too much time on this matter.

Configuring FreeBSD

Again, we will not be able to mention everything. The most important is to know where the essential files are. As usual, you will find them in the /etc directory. They are called rc.something and can manage many different parts: general configuration, firewall, network, sysctl, etc.
The file rc.conf is an important one, since, as its name says, the configuration can be tuned there. In this file you can tell the system whether or not start some daemons (sshd, sendmail, etc), the type of your firewall, if you want to enable kernel secure level or not, if you want to enable IPv6 or not, etc. There are lot of options and you can see all of them in the defaults file surprisingly called /etc/defaults/rc.conf.
Do not get confused: this file contains the default settings not examples. That is, the /etc/rc.conf overrides these default settings. In other words, do not copy the /etc/defaults/rc.conf file to /etc/rc.conf or do not try to change it. The changes must be done in /etc/rc.conf.

For information, the choices made from *sysinstall* are automatically included in /etc/rc.conf (network card configuration, hostname, secure level, etc).

A very important note: FreeBSD comes with no active services in /etc/inetd.conf. This means, at first start, the daemons managed from inetd are off by defaults. In my humble opinion, this should be compulsory for every Unix system. Good point, anyway!

The other rc files will allow you to configure either the firewall or the sysctl and many others. This, of course, leads us to FreeBSD tuning.

Tuning FreeBSD

One of the finest tools for tuning is called sysctl (under Linux too!). You can use it from the command line to define a specific value or you can write a sysctl.conf file (recommended unless you use the command line for testing purposes).

For instance, if your machine is used as a gateway, sysctl can allow ip forwarding with the command *sysctl -w net.inet.ip.forwarding=1*. For better security you can check that the packets coming to an interface have the IP address corresponding to this interface with the command *sysctl -w net.inet.ip.check_interface=1*. You can control most of the behavior of your system using sysctl: check the man page. Obviously, if you want to write these into a sysctl.conf file, then just add the arguments without the sysctl command it self. The file uses the *variable = value* format. The above examples then become:

```
net.inet.ip.forwarding=1
net.inet.ip.check_interface=1
```

Of course, your own sysctl.conf file is supposed to hold more than two lines since it can control everything on kernel level, well, almost!

At the beginning of this article, we talked about secure levels. There are four different levels, from *-1* to *3*. *-1* represents the insecure level and *3* represents the highest security level. It is recommended to understand what every level does before selecting one. Using levels other than -1 or 0 can lead to a system unable to work as you expect. The first drawback appears at level 1 since it can prevent you from properly using the X server. Neither can you load or unload kernel modules, for instance. You have been warned!

However, these high secure levels can be useful for a specific server needing high security. As a home user you should keep level 0, but it is up to you...

The *chflags* command is related to secure levels. Investigate this command, reading the man page: it is a very powerful one.

To continue with this would make a "securing FreeBSD" article, accordingly, let us go to the next chapter.

Upgrading FreeBSD

Let us assume that you just bought a FreeBSD 4.5 CD set. A few days later, FreeBSD 4.6 is released. Bad luck!

Do not worry: FreeBSD comes with an upgrade system via CVS. You can use either anonymous CVS or CVSup. The first one is easier to use while the second one is more efficient. With these tools, the job

consists of getting the new source tree and synchronize it.
Next, you will have to compile the whole new stuff using *make buildworld*. How to do this is quite well explained in the provided handbook. The important point is that this allows you to have a brand new system always up-to-date.
However, this is not the end of it. As always, as far as free software is concerned, vulnerabilities are found and very quickly solved. Unlike many proprietaries system where a vulnerability can stay unsolved for ever (and one day), people involved in free software react at the speed of light and provide you with patches. Obviously, it is your job to check what is going on. The FreeBSD web site has a security section holding advisories by release. There you can download .asc files (text files) telling you what you have to know: a description of the problem and its solution. Other web sites provide you with this information. Let us mention, for instance, linuxsecurity, CERT, etc.
This of course means you have to update your system.

Updating FreeBSD

You do not want to keep well known vulnerabilities in your system, do you ? FreeBSD, like other free Unixes, provides you with patches. Go and get these patches and apply them. For Linux users, the main difference comes from the fact that you do not get a corrected package but a source patch instead. It means, that you will have to recompile the kernel after having patched the source code, this of course, implying the source code is installed on your machine. This is true under Linux as well, but people often download the modified rpm (for instance) at least for corrected applications, but also for patched kernel.
One of the most important difference between Linux and FreeBSD comes from the kernel. FreeBSD is based on BSD 4.4 and does not have various kernel versions like the Linux world. This means the kernel is a stable one but of course it becomes bigger through the time, being rather monolithic even if it can benefit from modules. The drawback is that you will have to compile your kernel more often than under Linux if you want to keep your system as secure as possible.
Anyway, I much prefer this philosophy than the Linux one but this is only my opinion. I do not think that having, let us say, 3 different kernels "on the work" can lead to stability. Not even to mention the applications that have to be "ported" from one version to the other, especially the one focusing on security. Another example could be the packet filtering system: 3 kernel versions, 3 different packet filtering software! And so on.
I have very much respect for the people doing this great job, but is it the right way to do things. Improvement is always a need but is it coherent to changing everything all the time? Never mind!
Obviously, you also can compile a new kernel to make it match your needs... and that time it is like under Linux. The way to do this is very well detailed in the handbook, so let us go to something else.

Applications

As already said, a vast majority of free software applications (if not all) works under FreeBSD. You do not need the above mentioned ports or packages. You can compile most of the available archives, whatever the software type.
For the desktop environments addicts, Gnome and KDE are available... and are included in the distribution. Same for the window managers. The great GNUstep framework also works very well under

FreeBSD. To compile it you need a newer gcc version than the one included in the distribution: no problem! Just get the corresponding archive and compile the new compiler. Obviously, the GNUstep applications such as GNUMail.app, Gorm.app or ProjectBuider.app work fine too.
Really, all the software you run under Linux is available for FreeBSD... and there are many others!
For example, FreeBSD comes with a bunch of tools regarding security. Monitoring tools, managing tools, etc, are part of the distribution.
Firewalls, proxies, port scanners, IDS, etc, are also available. For instance, you can choose between IPsec or ipfw. You can use nessus, nmap, portsentry, etc. Again, most of them are part of the distribution and nothing prevents you from getting and compiling the latest versions.
We already talked about desktop environments or window managers but we did not say a word about what they require to be able to run: the X Window System. XFree 4.* is now the default. For information, till FreeBSD 4.5, XFree 3.* was the default. That is, you had XFree 4.* in the distribution but the install process used to select XFree 3.*. Now, XFree 4.* is the default but you can still use version 3.* if you prefer.

Hardware

Since we are talking about the i386 release, let us say most of the hardware available works under FreeBSD, even when it is a bit old. For instance an old 3Com combo network card will work fine as soon as you define the media type in the ifconfig command. If you use a BNC connection, enough to add *media 10base2/BNC* at the end of the ifconfig line in /etc/rc.conf.
Same for some old graphic cards: using XFree 4.* will solve most of the problems. This is particularly true for rather old ATI cards.
Anyway, to be sure you can consult the Hardware notes on the FreeBSD website.

Security

We already mentioned a few tools such as firewalls, port scanners, etc. FreeBSD comes with many other security tools. OpenSSH, OpenSSL, Kerberos, MD5, etc, are part of the distribution. As usual, you cannot live without those tools, and of course, again, you have to check for eventual vulnerabilities.
A recent example shows the importance of being very careful. Let us say that you are advised that a vulnerability has been discovered in OpenSSH, for example. The right behavior is to go to the home web site and download the corrected version.
Yes, but it is not enough! Please, do control the checksum of the package.
The example of OpenSSH has not be taken arbitrarily: a few weeks ago, someone replaced the original archive with a corrupt one. As usual, people at OpenSSH reacted quite fast. However, users having downloaded the package in this short lapse of time without verifying the checksum have installed a "bad" version. Hence, the need of using checksum... and the need of controlling it.
Even if it is out of the scope of this article, let us insist on the point that security, is before all a behavior.

Communications

Networking under FreeBSD is quite evolved. The IPv6 implementation is very well done. The new means of communication are not left aside. If you use ADSL or cable modems, configuration is very easy.
Many ISP use PPPoE today. FreeBSD manages this in a very simple way: a few lines in the ppp.conf file will do the trick. For PPPoA (ATM), only the Alcatel modem is supported at the moment. This should change in a very near future (perhaps, there are already other supported modems).

Documentation

The provided handbook is not the only documentation available. You can also find a FAQ, an install guide, etc, on the FreeBSD website. Another important source of information can be found at http://www.freebsd-howto.com.

What’s next ?

This is a very short overview of FreeBSD. This system, like the other free Unixes is a very interesting one. It has many great features and really deserves a try. Unfortunately this article cannot be a full review and lots of important features have not even been mentioned.
FreeBSD will require some more Unix knowledge than some "Windowish" Linux distributions but it remains quite easy to use.
Of course, someone with no Unix knowledge at all may get into trouble, since one will often have to recompile the kernel after updating the system. However, this is a good way to learn, even if it can take "some" time. People today seem to have forgotten that, at the beginning, Unix was made by developers for developers, not for the home user or the normal user.
Luckily free and proprietary Unixes changed the original approach. Thanks to the many developers who made this possible. In the 80’s, most of the today Linux users would have run away from Unix, only because of the installation process. The distributions were coming on tapes and the install phase could last a whole working day, the configuration process requiring at least another day. Not to mention the configuration of applications! And today, you can use Unix without even knowing it, with systems like Mac OS X. This was already true at the beginning of the 90’s with OS X "father", NeXTstep.
This may seem off-topic, nevertheless NeXTstep lead the way for free and proprietary Unixes... and a few others! Bringing Unix to the home user was a challenge and at least, it happened. Congratulations to all the people involved in this challenge, whether they worked for money or for free. The ones working for free deserve a special mention and many more thanks ;-)
The fact that FreeBSD "shares" with Apple looks very good to me. Each one benefits from the other in the right way. Proprietary software always used free software... without "shouting" it too loud. Apple says openly that they use free software and so does SGI. Some others seem to do the same... but with ulterior motive. Anyway, I keep thinking that there is "room" for all... well, except for those who want the whole "room".
The "noise" about Linux keeps other free Unixes in the dark: so far, so good. This can have a bright side... and this is the main reason why I use FreeBSD. Of course, I keep using Linux (and many others): I like diversity!

If you feel like giving FreeBSD a try, this article will not be useless. Test it by yourself and you will discover the bunch of things not mentioned here.
I told you: we are living in a great time!

2005-01-14, generated by lfparser_pdf version 2.51