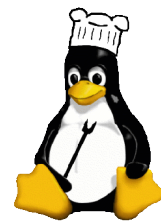


# The Open Pitt



What's cooking in Linux and Open Source in Western Pennsylvania

Issue 22

March 2006

www.wplug.org

## Security through Obscurity by Bill Moran

Most computer people with any networking experience will tell you that “security through obscurity is no security at all.”

What does this mean? On the surface, it seems rather silly. After all, the most common security mechanism is to password-protect the computer, and then hide (obscure) the password from anyone who doesn't need access.

Obviously, this isn't what the phrase means. It usually refers to any attempt to hide an insecure system as a substitute for securing it. Some examples include moving a telnet server to an unusual port and teaching your web server to lie about its version number. In both cases these are work-

arounds, and lousy ones. You're better off replacing telnet with the encryption-using ssh, and updating your web server instead of continuing to use one with vulnerabilities.

The fact remains, however, that these “obscurity” techniques do provide some measure of security. Why the saying, then?

Security works best when deployed in layers. Imagine someone is shooting at pistol at you. If you hold up a single sheet of paper in the bullet's path, it's not likely to be of much help. A phone book, however, might just save your life, yet the phone book is nothing more than hundreds of those flimsy sheets of paper.

Security is the same way. If moving telnet to an obscure port is your only method of security, you're doing the equivalent of holding up a sheet of paper to stop a bullet. On the other hand, moving your ssh server to an obscure port, if done in conjunction with other good security practices, is like adding another sheet of paper to an already thick phone book.

The rule against security through obscurity is the result of people trying to find a workaround that avoided fixing the real cause of the vulnerability. Perhaps a better phrasing would be “merely obscuring security holes does not improve security.”

*Bill Moran is WPLUG's Chair.*

## Book Review: Producing Open Source Software by Patrick Wagstrom

Author: Karl Fogel  
Publisher: O'Reilly Media  
ISBN: 0596007590  
\$24.95, 302 pages, 2005

The processes and best practices of developing Open Source software can seem obvious to seasoned veterans

while simultaneously being cryptic and confusing to newcomers. Traditionally, the best way to learn and understand them was to spend considerable time working on a project, slowly being brought up to date and more into the fold. While this works well for some projects, it presents diffi-

culties for others.

*Producing Open Source Software*, written by one of the primary authors of the Subversion version control system, attempts to unravel this process for both new contributors and old pros just the same. Each of its nine chapters examines a different issue that successful Open Source projects must address.

See **PRODUCING OSS**, p. 2

## February Roundup

*Feb. 4 General User Meeting:* This meeting was termed “Linux Demo Day” with the express purpose of introducing new users to the world of computers running on Linux. **Vance Kochenderfer** began with an overview of how a Linux system goes together. Office applications—including word processing, presentations, e-mail, calendaring, and instant messaging—were shown off by **Beth Lynn Eicher**. **Mike Hansell** followed up by firing up some Linux games. And **David Ostroske** rounded things out by demonstrating typical

home uses such as web browsing with Firefox (and its many feature-adding extensions), playing Internet streaming audio, and managing files.

*Feb. 11 Tutorial:* Continuing the new user theme, **Beth Lynn Eicher** presented her Linux Basics Tutorial. Packed into less than three hours were a wide range of subjects like selecting a Linux distribution, finding support, installing and updating software, basic system administration commands, and keeping your system secure. A PDF version of her slides is available.

Demo Day resources - <<http://www.wplug.org/demoday/>>  
Tutorial - <[http://www.wplug.org/meetings/one-meeting?wp\\_meeting\\_id=3216](http://www.wplug.org/meetings/one-meeting?wp_meeting_id=3216)>

## Coming Events

- Mar. 11:** General User Meeting,  
Topic: Database Administration.  
10AM to 2PM, 3002 Newell-Simon Hall, CMU
- Apr. 1:** General User Meeting,  
Topic: Linux on PowerPC.  
10AM to 2PM, 3002 Newell-Simon Hall, CMU
- Apr. 29:** Special Presentation,  
Topic: Virtualization with Bob Good of VMware. 10AM to 2PM,  
3305 Newell-Simon Hall, CMU

*The public is welcome at all events*

## Links of the Month *by Michael P. O'Connor*

This month I am going to look at a few Open Source news sites.

Several sites collect together links to news stories on other sites. First up is <http://www.linuxtoday.com/> which as you might guess focuses mainly on Linux. So does Mad Penguin <http://madpenguin.org/>; it also contains some original reviews and interviews. Links to news about all operating systems, plus features, interviews, and editorials are found at <http://www.osnews.com/>.

To stay current on the new releases of all the Linux and BSD distributions, <http://distrowatch.com/> has you covered. The Distrowatch Weekly column is an easy way to keep up.

Two sites which feature a lot of their own original reporting on issues related to Open Source are <http://www.newsforge.com/> and <http://www.linuxplanet.com/>. To get news oriented toward the use of Linux as a desktop operating system, try <http://desktoplinux.com/>.

On the more technical side, Linux Weekly News <http://lwn.net/> gives a lot of information on development activity. If you want to focus specifically on Open Source kernels, <http://kerneltrap.org/> is a perfect choice.

Be sure to send in any suggestions to me at [wplug@mikeoconnor.net](mailto:wplug@mikeoconnor.net).

Till next month, enjoy these links!

## PRODUCING OSS, *from p. 1*

The first three chapters address the landscape of Open Source software and how to get started running an Open Source project. The author highlights the dual needs of Open Source software at the early stages: acquiring users and acquiring developers. Without these two elements, your project is doomed. To meet these goals, it is important to survey the landscape and ensure that your project will be useful and will not duplicate an existing Open Source project—a factor that could limit the number of users and developers willing to participate.

After this brief introduction, the book covers topics such as money, social infrastructure, communication, and licenses. Also included are the nuts and bolts of packaging and daily development. At the end, several useful appendices list different version control systems and bug trackers, amongst other things. The author tries hard throughout the book not to overly bias the reader in one direction or another. Even during the discussion of version control systems, where he has a vested interest, he objectively points out the advantages and disadvantages of each one.

One of the biggest issues that Fogel addresses is communication in Open Source projects. He makes it very

clear that Open Source development is not something that can be done in a cave. Rather, at all times you must be aware of how you are communicating. Beginning with several tips on creating useful web pages for people to download your project, and then moving to complex topics such as how to handle difficult people on mailing lists, the advice seems pointed and helpful. Despite the fact that it may be easier to assume knowledge on the part of users, or take a conversation private to avoid conflict, he stresses the need to take the time to accurately document decisions and discussions on a mailing list. This allows users to find the material much more easily in the future.

While *Producing Open Source Software* covers lots of ground and was thought-provoking regarding issues Open Source developers face, it still had a few shortcomings. The author tries to address everyone, from a college student looking to pick up some skills in his spare time, to corporate behemoths like IBM and Sun. In doing so, there are many parts of the book that may seem overly simplified or simply irrelevant to some. However, this attribute can also be a blessing because it results in a book that you can easily give to your company CIO, boss, friend, or even a relative who just started college.

The Open Pitt is published by the Western Pennsylvania Linux Users Group

<http://www.wplug.org/top/>

Editors:

Elwin Green  
Vance Kochenderfer

*What is Linux?*

Linux is a *kernel*, the core of a computer operating system, created by Linus Torvalds. It is typically packaged as a *distribution*, which includes the extra programs necessary to make a computer functional and useful. Since 1991, it has grown from a one-man project which ran on one computer to one with thousands of contributors running on everything from personal organizers to million-dollar supercomputers.

*What are Open Source and Free Software?*

Open Source and Free Software provide you, the user, with the opportunity to see the source code of the programs you use. You are free to use it, share it with others, and even make changes to it if you wish. While the Free Software and Open Source communities differ in their philosophical approach, in practical terms they share nearly identical goals. Learn more at <http://www.opensource.org/> and <http://www.gnu.org/>.

This newsletter was produced using Open Source and Free Software.

Copyright 2006 Western Pennsylvania Linux Users Group. Any article in this newsletter may be reprinted elsewhere in any medium, provided it is not changed and attribution is given to the author and WPLUG.

As an academic researching Open Source software, I've been to many conferences where people who claimed to be studying how it works just didn't get it—instead believing that the world operates just as Eric Raymond described it in *The Cathedral and the Bazaar*. Often I wished that I could point them to a book that described the actual process of Open Source development today, rather than a grandiose philosophical vision of it. While it's not perfect, *Producing Open Source Software* comes the closest to that goal of anything that I've found and is a welcome addition to my library. More information on the book can be found at the web site <http://www.producingoss.com/>.

*Patrick Wagstrom is a Ph.D. candidate at Carnegie Mellon University researching communication and collaboration in Open Source development. He has been using Linux since 1994.*