The Monthly GNU Column

Brave GNU World

Welcome to the Brave Gnu World, where we will be investigating one of the perennial problems of information technology: security.

Simple Security Policy Editor

The Simple Security Policy Editor (SSPE) [6] by Johannes Hubertz is a firewall solution for GNU/Linux systems based on Netfilter [7], FreeS/WAN [8], and OpenSSH [9]. It allows administrators to centralize the management of firewall rules for multiple hosts, distributing management tasks across multiple machines. SSPE's target group comprises geographically distributed enterprises with Internet attached branches, departments, or subsidiaries.

SSPE's developer, Johannes Huberts, started work on SSPE way back in December 2001. At the time, his employee was looking for a software application that would allow distributed management of multiple machines across a network. Non-transparent, proprietary systems were ruled out, and as none of the available systems fit the bill, the company decided to develop its own solution.

The developers opted for Perl and Bash shell scripts, as most administrators are familiar with them, and because they run well on any GNU/Linux distribution.

In March 2002, after only three months' development work, the project had already reached the stage where it

This column looks into projects and current affairs in the world of free software from the perspective of the GNU Project and the FSF. In this issue, we will be focusing on the Simple Security Policy Editor (SSPE), and the UK Free Software Network (UKFSN) Internet Service Provider.

BY GEORG C.F. GREVE

could be put to productive use. Today, two companies use the software at eight sites. A third enterprise plans to use it for several hundred field engineers. Despite the low version number, 0.1.7, SSPE is very stable – the developers have not needed to introduce any major changes since March 2003.

Not for everyone

SSPE is not suited for use on stand-alone machines. Also, users should have basic network security skills and be capable of compiling a Linux kernel.

SSPE is not restricted to any specific distribution. Administrators can manage the complete network infrastructure with a few text files, and that makes it extremely simple to integrate new branches. As Johannes also points out, flexibility, transparency and stability are just some of SSPE's major advantages.

In March 2003 Johannes succeeded in communicating the advantages of releasing this software under the GNU General Public License (GPL) to company management, thus removing any obstacles in the way of more widespread

THE AUTHOR

Georg C. F. Greve Dipl.-Phys. has been using free software for many years. He was an earlier adopter of GNU/ Linux. After becoming active in the GNU project he formed the Free Foundation Europe, of which he is the current president. More information can be found at http:// www.gnuhh.org.
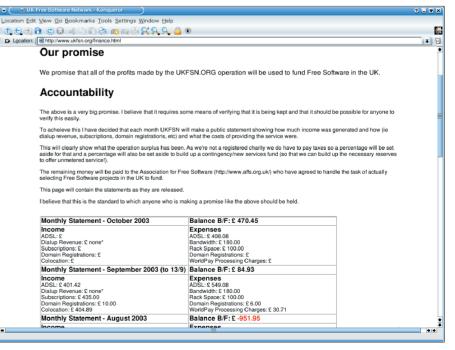
Figure 1: Each month, UKFSN, an ISP, discloses its financial situation on the Web to provide the community maximum transparency. After all, all their income is supposed to be dedicated to funding free software
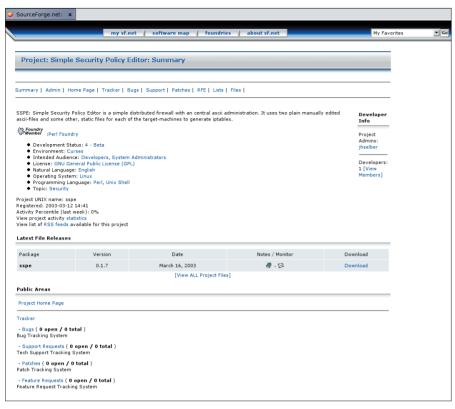
Figure 2: The Simple Security Policy Editor project site. The tool allows centralized administration of network and firewall configurations for multiple machines

use. Johannes would be happy to hear from developers interested in testing and expanding SSPE.

UK Free Software Network (UKFSN)

Jason Clifford's UK Free Software Network (UKFSN) [10] is a very special Internet Service Provider (ISP). Jason's aim is to provide premium Internet services based on free software and to channel profits into supporting and funding free software development.

Any profit generated by UKFSN, except for a contingency reserve, will go to the Association For Free Software (AFFS) [11] to help promote the development of free software. AFFS is an associate organization of the Free Software Foundation Europe (FSF Europe) [3].

UKFSN's broad range of services includes Internet access via ADSL and Web hosting. These services are based on the Apache Web server, Postfix and Tpop3d for email, the MySQL database system, and FreeRADIUS for dial-up authentication. The complete management system, from setting up new accounts to maintenance scripts, was written in Perl.

Customers that use UKFSN's Web hosting services can choose between Python, Perl and PHP for scripting. In addition, they receive a MySQL database, an unlimited number of POP3 mailboxes per domain, and simple account management based on the Web interface.

Not standing still

The icing on the cake is a Perl module that integrates World Pay, allowing site operators to handle payments by credit card, cellphone, or bank transfer. Incidentally, this is also a free software module released under the GNU GPL and available from CPAN. A PHP implementation is planned.

Jason has already started work on extending the range of services on offer. In future, customers will be able to run virtual Linux servers without deploying any hardware of their own by opting for "User Mode Linux". In addition, Jason intends to offer customer-configurable spam and virus filters for e-mail. Both of these extended services should be available by the time this issue goes to press.

UKFSN's anti-spam policy is worthy of note. Customers are required by the con-

ditions of use to pay a penalty of 150 GB pounds to the recipient of any spam they propagate. And this has caused quite a few people who were interested in opening up an account to look for a different provider.

In this way, UKFSN not only protects its own customers against spam, but also helps to reduce the number of spam messages on the Internet in general. Let's hope that other providers follow suit as even the best spam filters are incapable of recognizing 100 percent of all unsolicited mail.

There was another company that had achieved quite a degree of success with similar aims, however, as Jason found out, this company was unwilling to fund free software.

This is what prompted his decision to set up UKFSN in July 2002. By September, Jason had accumulated sufficient start-up capital to present UKFSN at the Linux-Expo in London, UK, on October 9 2002.

UKFSN took a major step forward in June 2003 with the introduction of broadband ADSL access. Hardware donated by Digital Networks UK in the form of several servers was another major milestone. And by August 2003 UKFSN finally started to return profits, which can be used to fund AFFS work.

UKFSN is living proof of the fact that an ISP can contribute to funding free software development. Jason is now set to show that it is possible to actually use these funds to support free software. To provide maximum transparency, Jason files monthly financial reports on the UKFSN website [12].

Let's hope that UKFSN continues to grow in future, attracting more British residents and visitors to Great Britain who require a temporary Internet connection. And it would be a good thing to emulate Jason's achievements in other countries.

That should provide enough motivation for the time being, but don't miss next month's issue for more.

Free Software for Scientific Applications

The principles of free software are very similar to scientific principles. Both thrive on co-operation between many individuals. Isaac Newton once expressed this as

follows, "If I have seen further, it is by standing on the shoulders of Giants."

Not only do the scientists or developers involved profit from this work, but society as a whole. This includes businesses, and people who have not contributed anything to, or even fought against, certain developments.

Many people understand the cybernetics involved, but the connection between free software and science is less clear:

A scientific approach means putting forward a theory and supporting it by experimentation. Demonstrating a theory can help to prove it to be true, but additional proof will not make it "truer than true".

The opposite case applies when an experiment proves a theory to be false. In this case, the theory needs to be revised or dropped completely.

A single falsification cancels out any number of verifications. Falsification is part of the scientific process – without it science cannot exist.

The Implementation is Decisive

Software is becoming an increasing part of science. This does not mean that scientists simply use free word processors to report on their results, for example.

The connection between software and science becomes more apparent when software forms the basis for experiments. In this case, software becomes both part of the scientific process and its results.

As every developer will be aware, simply publishing the algorithm used by a program will not suffice to provide a means of falsification. The implementation is decisive and also becomes a part of the result.

Proprietary software is like a black box. Imagine a black box with a switch and a light bulb. Imagine somebody telling you that an experiment is carried out in the box when you flip the switch, and this is why the bulb lights up. If another person with another black box

Isaac Newton: "On the shoulders of Giants". Science can only be successful when many individuals co-operate to create a platform for more research

maintains that the same experiment is carried out in her black box, without the bulb lighting up, nobody is any wiser. Neither case allows you to prove or disprove the results. In fact, the whole thing is reduced to a question of trust. From this, we conclude that proprietary software is incompatible with scientific methods!

But this is not the only difficulty. The course of development within a scientific discipline is part of humanity's cultural heritage and can point the way for future generations. Time should not be a factor that affects results. Someone wanting to recreate an experiment performed by Leonardo da Vinci will be able to do so without any problems.

The probability of being able to repeat an experiment based on proprietary software after a certain number of years have passed tends towards zero with increasing years. This is particularly true if the software used for the original experiment required a specific hardware platform. Free software that can be ported to any platform allows us to repeat experiments which were performed years ago.

In this respect, free software plays a major role in archiving the scientific and cultural development of mankind. It allows us to preserve the path that brought us to where we are today.

It soon becomes apparent that the relationship between free software and science is far more intensive than it would appear at first glance. In addition, there is evidence of the link between free software and the social and cultural aspects that are definitive of mankind.

That's All Folks...

My schedule for the coming month is quite strenuous with the UN conference on the information society in Geneva, talks in Madrid, Bern and Zurich. But I will be taking my laptop along, and of course I'm always interested in your comments. I am particularly interested in smaller projects that commonly go unnoticed and personal initiatives by free software supporters.

I would, as ever, like to encourage everyone to get in touch with any comments, questions and ideas to the usual address. [1]. ■

INFO

[1] Send ideas, comments and questions to: *column@brave-gnu-world.org*

[2] GNU Project homepage: *http://www.gnu.org/*

[3] Free Software Foundation Europe (FSF Europe) homepage: *http://www.fsfeurope.org*

[4] Georg's Brave GNU World homepage: *http://brave-gnu-world.org*

[5] "We run GNU" Initiative: *http://www.gnu.org/brave-gnu-world/rungnu/rungnu.html*

[6] Simple Security Policy Editor (SSPE): *http://sspe.sf.net*

[7] Netfilter: *http://www.netfilter.org/*

[8] FreeS/WAN: *http://www.freeswan.org/*

[9] OpenSSH: *http://www.openssh.com/*

[10] UK Free Software Network (UKFSN) homepage: *http://www.ukfsn.org*

[11] Association For Free Software (AFFS): *http://www.affs.org.uk*

[12] UKFSN financial statement: *http://www.ukfsn.org/finance.html*