



# Open Source Software



## Background

In the challenging world of security management today, system administrators are faced with tough choices—how to maximize services that users demand while balancing a tight budget and ensuring the security requirements are maintained. These constraints motivate administrators to seek alternatives to traditional commercially available software- in the form of open source software. Open source software (OSS) refers to software that is developed, tested, or improved through public collaboration as opposed to by an individual company. The software is usually free and unlike proprietary commercial software, the source code is available to the public. There are many different types of OSS, from operating systems like Linux, to security tools like the security scanner called Nessus.

Within the information technology arena, there is currently debate over the usefulness of open source software and there is not likely to be a unified agreement over whether open source software should be used in an organization or not. There are attractive features and drawbacks, and like with the consideration of any software, these pros and cons must be weighed before a final decision can be made.

The following section describes the pros and cons of open source software. System management can use this information to help them determine if using open source software is right for them. The research is based upon research for public school systems for usage of OSS, more information can be found at:

[http://www.netc.org/openoptions/pros\\_cons/comparing.html](http://www.netc.org/openoptions/pros_cons/comparing.html)

### Pros:

- Low Initial Cost

The price of an open source program is usually far less than a comparable proprietary program. Open source software doesn't have to be "no fee" but most programs are. Users can either download the software directly or pay a negligible fee to have a CD-ROM burned and shipped. Current users set up distribution networks using community Web sites and CD burners. Their motto is "share and share alike." Open source means anyone can try any program first for free. A user may eventually buy a formal copy (perhaps to get better service), but doesn't have to do so. The software will never expire or demand payment.

- Reliability/Stability/Security:

Open source software may be more reliable and secure than proprietary. It may not make as many errors or crash as often. (e.g. Linux is famous for not crashing.) Since any programmer can find and fix bugs, software may be repaired and improved more quickly. The initial program may not be more reliable than a proprietary alternative, but it may mature faster as hundreds or thousands of programmers correct mistakes and add



# Open Source Software



features. Some people think of this as permanent [beta testing](#). The open source community can endlessly troubleshoot and improve software as needed or desired.

However, this advantage depends on the participation of enough competent programmers. Just like proprietary software, the reliability of an open source program depends on clear feedback after rigorous use in a variety of environments. Without enduring, sufficient, talented interest, an open source project fails, and many do. In contrast, proprietary software companies may create and support necessary programs that no one would enjoy working on. Some companies are starting to blend the best of both models, by employing a core group of programmers while attracting volunteers from the open source community.

- Open source is more network friendly

Much of the popularity of open source comes from its performance on the Internet and lesser networks. Open source software is often very networkable. For example, more than half the World Wide Web runs on [Apache](#), an open source solution. Apple builds its OS X on [BSD](#), an open source [operating system](#). The Internet is a critical reason: Apple recognizes they can't privately innovate Internet functionality as well or as fast as the open source community. Most open source networking solutions are compatible with proprietary software. For example, a lab of Linux computers can be seamlessly nested inside a larger proprietary network using an open source intermediary (e.g. Samba).

- Open source makes license management easier

License management is much easier with open source. Users can install any number of copies, so open source companies don't bother with complicated licenses (and most of the possible licenses favor nonprofits like schools). There is no risk of illegal copies or license audits, and there are no anti-piracy measures (e.g. CD keys, product activation).

- Open source can be more customized

By its very nature, open source allows any user or organization with enough expertise to tailor software to their needs. The diversity of Linux distributions reflects this flexibility. Each distribution offers a customized operating system targeting a specific market. There are even two competing graphical desktop interfaces: KDE and GNOME. On the [backend](#), the open, modular nature of open source solutions allows advanced users extraordinary power to customize any aspect of a network (e.g. firewalls, spam filtering, email filtering).

Beyond customization, anyone can make significant changes to open source code.

Proprietary software may offer new features or flexibility in each new version. But open source will continue to offer greater ability to customize through the so-called right to fork: anyone can take the source code and develop it in a new direction.

- Open source means greater independence from companies/Does not commit organization to just one vendor

Open source users have more independence from software companies. Even if a software company goes bankrupt, the community still has the source code. This independence also



# Open Source Software



means "end of life" decisions or undesirable new features can't be forced on the users. Schools aren't [locked into](#) a vendor or their support department. The original software company may offer the best support. But since the software is open anyone can try to improve or support it. Schools can choose the best solution now with the freedom to change in the future. They don't have to rely on a single vendor for all aspects of the solution. A comprehensive solution may be easier, but schools can essentially purchase every component of a solution (including support) from different vendors.

## Cons:

- Software Compatibility

Though open source projects like Open Office have come a long way towards reading proprietary file formats and saving documents in formats that proprietary software can read, compatibility issues can still arise, especially in a networked environment where people are trying to collaborate across platforms.

- Hardware Compatibility

Many hardware vendors only provide proprietary drivers for their products, leaving it to the open source community to write their own Linux drivers for new hardware. Users without the wherewithal to whip up a hardware driver may be left waiting for someone else to get around to writing drivers for cutting edge hardware.

- De-centralized Support/ Potential Maintenance Costs

Without centralized support, users may be left to research solutions to their technical difficulties. Because the software was free, it comes with no warranty, and developers have no legal obligation to replace non-working products.

- Proprietary software offers more features

On the frontend, proprietary software offers more necessary and desirable features. For example, it may be easier to use peripherals like digital cameras with proprietary software. Solutions like Microsoft Office have matured through years of added features, interface improvements, and usability studies. Some open source projects are overtly cloning proprietary products (e.g. [OpenOffice.org](#) is clearly influenced by Microsoft Office). A cloned program may be just as user friendly as the original (such as it is). Of course, while most programs are flush with features, most users (including schools) only need a handful of features.

- Proprietary offers better service & support

Any software solution requires some service and support. For both open source and proprietary software, experts depend on email lists and community Web sites as well as contracted support. The quality and availability of help is proportional to interest and use, especially in open source. The support costs for niche solutions are usually high. On the backend, open source is common so the community is large and helpful. A variety of companies offer help for using open source on the backend. On the frontend, open source



## Open Source Software



is often still a niche. So it may be harder to find help (especially in rural areas) and contracted help may be more costly.

A proprietary company may have a longer, better reputation for service and support. For example, it's relatively easy to find hardware [drivers](#) for Microsoft Windows solutions. Since the open source community depends on volunteers, help may not be as certain or as timely.

### Conclusion

There are many issues surrounding the usage of open source software for an organization. For some agencies, usage of open source software may not be viable depending upon the ability of their user base to use untraditional products or their system administrators to maintain the software. However, open source software is a viable and well-recognized way for agencies to meet their user needs. Private and governmental organizations routinely use open source software. For example, National Institutes of Health uses SARA, a vulnerability tool, in their computing environment.

Open source software can be used at FSA with the following constraints. Appropriate research has to be done to verify that the program under consideration is from a legitimate source and that there is sufficient active support by the IT community to maintain the program. Two websites for reference information are The Open Source Initiative, <http://www.opensource.org/licenses/>, for general license information and for security tools, <http://www.insecure.org/tools.html>. The system must assess the potential risk of using the software and determine the amount of risk that the new software may be introducing. Maybe testing?