Order Code RL31627

Computer Software and Open Source Issues: A Primer

November 5, 2002

Jeffrey W. Seifert
Analyst in Information Science and Technology Policy
Resources, Science, and Industry Division

Computer Software and Open Source Issues: A Primer

Summary

The use of open source software by the federal government has been gaining attention as organizations continue to search for opportunities to enhance their information technology (IT) operations while containing costs. For the federal government and Congress, the debate over the use of open source software intersects several other issues, including, but not limited to, the development of homeland security and e-government initiatives, improving government information technology management practices, strengthening computer security, and protecting intellectual property rights. Currently, the debate over open source software often revolves primarily around information security and intellectual property rights. However, issues related to cost and quality are often raised as well.

Open source software refers to a computer program whose source code, or programming instructions, is made available to the general public to be improved or modified as the user wishes. Some examples of open source software include the Linux operating system and Apache Web server software. In contrast, *closed source*, or proprietary, programs are those whose source code is not made available and can only be altered by the software manufacturer. In the case of closed source software, updates to a program are usually distributed in the form of a patch or as a new version of the program that the user can install but not alter. Some examples of closed source software include Microsoft Word and Corel WordPerfect. The majority of software products most commonly used, such as operating systems, word processing programs, and databases, are closed source programs.

For proponents, open source software is often viewed as a means to reduce an organization's dependence on the software products of a few companies while possibly improving the security and stability of one's computing infrastructure. For critics, open source software is often viewed as a threat to intellectual property rights with unproven cost and quality benefits. So far there appear to be no systematic analyses available that have conclusively compared closed source to open source software on the issue of security. In practice, computer security is highly dependent on how an application is configured, maintained, and monitored. Similarly, the costs of implementing an open source solution are dependent upon factors such as the cost of acquiring the hardware/software, investments in training for IT personnel and end users, maintenance and support costs, and the resources required to convert data and applications to work in the new computing environment. Consequently, some computer experts suggest that it is not possible to conclude that either open source or closed source software is inherently more secure or more cost efficient.

At this time there appears to be no centralized accounting of open source software throughout the federal government. However, the growing emphasis on improved information security and critical infrastructure protection overall, will likely be an influential factor in future decisions to implement open source solutions. The rapidly changing computer environment may also foster the use of a combination of open source and closed source applications, rather than creating a need to choose one option at the exclusion of another. This report will be updated as events warrant.

Contents

Computer Software and Open Source Issues: A Primer

What is Open Source Software?

Open source software refers to a computer program whose source code[1] is made available to the general public to be improved or modified as the user wishes. Changes to such a computer program may be available freely through Web sites and users groups dedicated to that particular program. Some examples of open source software include the Linux operating system[2] and Apache Web server software.[3] In contrast, *closed source*, or proprietary, programs are those whose source code is not made available and can only be altered by the software manufacturer. In the case of closed source software, updates to a program are usually distributed in the form of a patch[4] or as a new version of the program that the user can install but not alter. Some examples of closed source software include Microsoft Word and Corel WordPerfect. The majority of software products most commonly used, such as operating systems, word processing programs, and databases, are closed source programs.

Although open source software has been attracting renewed attention recently, its origins date back to the development of ARPANET in the late 1960s. During this time, the individuals and universities that composed the small network of programmers often worked collaboratively, sharing source code as a means to build their knowledge base.[5] More recently, open source software has been seen by some observers as an alternative to the influence of a few large software companies in

---

[1]Source code is the set of programming instructions written by the software developer that allows a program to execute its functions. Source code is written at the keyboard and appears as a set of commands in the form of words, symbols, and numbers. After a programmer has finished writing the source code, it is compiled into a machine language that is recognized only by computers and is represented entirely as numbers. Proprietary software includes only the machine language code, which allows the computer to function but cannot be altered by the user. Open source software includes the source code (and sometimes the machine language code) so that the user can make changes to how the software program functions.

[2]For more information about the Linux operating system, see [http://www.linux.org/].

[3]For more information about Apache software, see [http://www.apache.org/].

[4] A software patch is a small piece of software that integrates itself into the larger program and is created to fix a specific problem, such as a particular security weakness or some other error or defect in the product.

[5]Jay Holander, "The Challenge of Open-Source Business Model," *Gigalaw,* April 2000, [http://www.gigalaw.com/articles/2000-all/hollander-2000-04-all.html].

some of the more popular user areas, such as operating systems and office productivity suites. One of the more well known open source programs is Linux, an operating system developed in the 1990s by Linus Torvalds, a Finnish programmer now working for an information technology company in the United States. While Linux holds a small share of the operating system market, compared to Microsoft Windows, and is not widely used in the federal government, some agencies have begun trial demonstrations in limited settings.[6] For example, the National Security Agency has been working with volunteer programmers to create a new version of Linux called Security-Enhanced Linux (SELinux) in an attempt to develop tools and applications that could be used to improve the security of government computer systems.[7]

Leading Organizations in the Open Source Software Community

Although the open source software community is loosely organized, two primary organizations are identified as leaders in advocating standards and definitions. One is the Free Software Foundation (FSF), founded in 1985, and dedicated to "promoting computer users' right to use, study, copy, modify, and redistribute computer programs."[8] By 'free software,' FSF means that the user should be allowed to alter, improve, and/or redistribute a version of a software program, either gratis or for a fee. Using this interpretation, FSF does not suggest that software should necessarily be cost-free. The second major organization is the Open Source Initiative (OSI).[9] OSI is self described as a "non-profit corporation dedicated to managing and promoting the Open Source Definition"[10] through a certification program it administers.[11]

---

[6]Declan McCullagh and Robert Zarate, "Super-Secure Linux, Inch by Inch," *Wired*, 11 June 2002, [http://www.wired.com/news/linux/0,1411,53004,00.html].

[7]For more information about NSA's support for SELinux, see [http://www.nsa.gov/selinux/index.html]

[8]For more information about the Free Software Foundation, see [http://www.gnu.org/fsf/fsf.html].

[9]For more information about the Open Source Initiative, see [http://www.opensource.org].

[10]The Open Source Definition is a set of nine criteria that a program must meet to be certified as open source software by OSI. These criteria address issues such as right of redistribution, availability of the source code, derived works, discrimination of use, and licensing. Details regarding these criteria can be found at [http://www.opensource.org/docs/definition.php].

[11]For more information about the OSI certification program, see [http://www.opensource.org/docs/certification_mark.php].

Open Source Software Issues

Open source software has been gaining attention as organizations continue to search for opportunities to enhance their information technology operations while containing costs. For the federal government and Congress, the debate over the use of open source software intersects several other issues, including, but not limited to, the development of homeland security and e-government initiatives, improving government information technology management practices, strengthening computer security, and protecting intellectual property rights. Currently, the debate over open source software often revolves primarily around security and intellectual property rights. However, issues related to cost and quality are often raised as well.

Information Security

Some critics of open source software suggest that it is less secure than proprietary or closed source software because it allows a potential hacker to search the source code to discover and exploit flaws. Some observers suggest that the 'security through obscurity' principle that accompanies closed source software enhances security by making it more difficult for potential flaws to be discovered and exploited.[12] Concerns have also been raised regarding the possibility of Trojan horse programs[13] being introduced into a computing environment through the downloading and use of open source software whose provenance may not be entirely clear.

In contrast, advocates for open source software suggest that it may be less prone to security flaws due to the peer-review nature of open source software development. This allows the source code to be scrutinized simultaneously by a wide audience of individuals who bring different perspectives and may test the software under a variety of conditions. Supporters of open source software suggest this approach can generate a faster response to security problems and minimize the potential of Trojan horse programs.[14]

So far there appear to be no systematic analyses available that have conclusively compared closed source to open source software on the issue of security. In practice, computer security is highly dependent on how the user and/or administrator configures, maintains, and monitors the application. Consequently, some computer security experts suggest that it is not possible to conclude that either open source or closed

---

[12]Although sometimes used as a derogatory term by critics of proprietary software, the philosophy behind the concept of "security through obscurity" is that security is enhanced if flaws are hidden from view and not publicized until a solution can be made available.

[13]A Trojan horse program is a destructive software program that appears as a benign application. One example is a program that is described as an upgrade or a service pack for a current version of a program, but is in fact designed to disable a computer's virus scanner and introduces new viruses to the computer.

[14]Drew Clark, "Defense, Cybersecurity Officials Praise 'Open Source' Software," *Government Executive Magazine*, 29 October 2002, [http://207.27.3.29/dailyfed/1002/102902td2.htm].

source software is inherently more secure.[15] However, this viewpoint may change as additional research is carried out.

Intellectual Property Rights

The implications of open source software for intellectual property rights continue to evolve. While open source software generally provides users with greater freedoms than closed source software,[16] open source software is usually distributed with some form of licensing agreement[17] that details the conditions under which a user may use, make changes, and redistribute the source code. Critics argue that open source software threatens intellectual property rights because any software that incorporates open source code must be freely redistributed at no cost. While one of the underlying principles of open source software is the unrestricted redistribution of source code, a distinction is made between software that incorporates open source code *into* its program, and programs that work *with* open source software. For example, the General Public License (GPL), drafted by the Free Software Foundation, is one open source agreement. The GPL requires that entities using open source code must, upon further distributing that code or subsequent modifications of that code, either provide a copy of the source code or offer to give any third party a copy of the source code. However, the GPL allows a user to include an open source program with a closed source program without providing the source code for the closed source program, provided the two programs are functionally separate,[18] such as in the case of an editor program that works with a shell program.[19]

In contrast to the concerns raised regarding the potential threats to intellectual property rights, some observers suggest that the increased use of patents (as compared to copyrights or trade secrets) by technology companies to protect online business methods such as one-click shopping, customer referral affiliate programs, and buyer-driven e-commerce, could hinder the future development of open source software.

---

[15]Jonathan Krim, "Open-Source Fight Flares at Pentagon," *The Washington Post*, 23 May 2002, p. E1; Michelle Delio, "Did MS Pay for Open-Source Scare?," *Wired*, 5 June 2002, [http://www.wired.com/news/linux/0,1411,52973,00.html]; Dennis Fisher, "Open Source: A False Sense of Security?," *eWeek*, 30 September 2002, p. 20;

[16]Most notably, the ability to make changes to the source code.

[17]There are many open source licenses currently in use. Some are more general in nature, designed to be easily adopted by anyone developing open source software, while others are more specific, created by a particular company or organization. A collection of some of the most well known licenses can be found at [http://www.opensource.org/licenses/].

[18]For a more complete explanation of this concept, see [http://www.gnu.org/licenses/gpl-faq.html#GPLInProprietarySystem].

[19]In this example, the editor program could be one that executes text commands to complete tasks, while the shell program provides a graphical interface with menus to allow the user to complete these same tasks without knowing the text commands. So, the editor program could be a closed source program used by people with a strong knowledge of text commands. The shell program could be an open source program developed later to help less knowledgeable users to complete the same tasks. The editor can work independently of the shell, and as such the editor can be distributed with the shell program without having its source code included.

Under this scenario, technology companies could choose to license patented business methods and technologies only to organizations for use in closed source software. This, in turn, could affect the type of software that could effectively be developed as an open source product as compared to a closed source product.

Quality

Related to the issue of security, some supporters of open source software argue that the potentially large number of programmers contributing to the development of an open source program can contribute to a higher quality product with fewer 'bugs' because it is more likely an error will be discovered before it becomes a major problem. In addition, some observers suggest that open source organizations generally react quickly when a problem is discovered, and use small software 'patches' to fix a specific problem, potentially limiting unanticipated side effects. These observers contrast this approach with that of software companies, who may wish to wait and release multiple patches together in a single service pack.[20] While the use of service packs can be more convenient and efficient than having to install numerous patches individually, service packs can sometimes cause new problems due to the simultaneous introduction of several uncoordinated changes to the software program.[21]

In contrast, some critics cite the lack of formal vendor or technical support for open source software that is not commercially distributed. Since open source software is developed by a community of users, it often does not have a dedicated technical support team that will respond to troubleshooting inquiries on a fixed schedule. Where a company or agency relies on the proper functioning of the software to carry out mission critical tasks, vendor support can play an important role in the event of a mishap. Related to the technical support concerns, some observers suggest that open source software is not as reliable as the closed source or commercial alternatives because there is no identifiable company or organization whose profits and/or reputation is dependent upon the proper functioning of the software.[22] As with security, there does not seem to be any conclusive study comparing quality. Moreover, with both types of software continually evolving, and quality being a somewhat subjective measure, no firm conclusion may be possible.

Cost

The costs associated with using open source software compared to proprietary software is dependent upon a number of factors, including the cost of acquiring the

---

[20]A service pack is an update to a software version that fixes an existing problem, such as a bug, or provides enhancements that will appear in the next version of the product.

[21]Jim Rapoza, "eWeek Labs: Open Source Quicker at Fixing Flaws," *eWeek*, 30 September 2002, [http://www.eweek.com/article2/0,3959,562226,00.asp].

[22]In recent years as interest in open source software has become more widespread, some software vendors have begun to sell commercial versions of popular open source programs, such as Linux, which can include access to vendor provided support and service. Two examples of such companies include Red Hat [http://www/redhat.com] and The SCO Group (formerly known as Caldera International) [http://www.sco.com].

necessary hardware and software,[23] investments in training for information technology personnel and end users, maintenance and support costs, and the resources required to convert data and applications to work in the new computing environment. Calculating these costs, also referred to as the total cost of ownership (TCO), is unique to each organization and application.

Some observers suggest that by utilizing the community of unpaid programmers and users that grows around a particular product, open source software carries lower costs than closed source software by potentially decreasing the number of in-house information technology professionals needed to support the software and by eliminating the need for costly service contracts offered by proprietary software developers. The availability of community support may also offer the flexibility of allowing the adopting organization to decide if and when it will upgrade to a new version of a particular software program. In contrast, commercial software vendors usually discontinue support of older programs some time after new versions are introduced, which leaves organizations to decide whether or not to continue using unsupported software, or to incur the costs of upgrading to a new version.

On the other hand, organizations that adopt open source software, in part due to the ability to customize the software for their particular needs, will still need to either maintain an adequate level of internal information technology personnel, or outsource these responsibilities on a contract basis. Similarly, many of the most popularly used programs in business and government, such as word processing, spreadsheets, databases, and e-mail, are designed to work with a particular operating system, such as Windows. So, if an organization decides to switch to an alternative operating system, such as Linux, it may also have to adopt other new programs, requiring additional employee training and support.

Implications for Government Use of Open Source Software

Although there is increasing interest in federal government use of open source software, the extent to which the use of these applications will continue to grow remains to be seen. In an October 2000 report on the use of open source software for high performance computing, the President's Information Technology Advisory Committee (PITAC) recommended that the federal government "should aggressively encourage the development of open source software for high end computing." The report also recommended that the federal government examine its procurement processes as they relate to open source software. In addition, the report suggested that there was a need to analyze open source licensing agreements, "with an ultimate goal of agreeing upon a single common licensing agreement for open source software applications."[24]

---

[23] While many open source software programs can be acquired for free, commercially distributed versions of these programs are also sometimes available.

[24] President's Information Technology Advisory Committee, Panel on Open Source (continued...)

In a July 2001 report on the use of open source software for military applications, the MITRE Corporation suggested that a business case could be made for the implementation of open source software solutions for server and embedded systems, based on potential cost, reliability, and support advantages. However, the report emphasized the need for program mangers to consider several factors when selecting a strategy for a specific set of circumstances. These include assessing the size, talent, and organization of the supporting community, examining the market demand for the open source product in question, conducting a risk/benefit analysis, and comparing the long term costs of available options.[25]

More recently, an October 2002 MITRE report on the use of open source software in the Department of Defense (DoD) concluded that free and open source software "plays a more critical role in the DoD than has generally been recognized." It identified 115 open source applications and 251 examples of their use within the DoD. The report stated that open source software plays an especially significant role in the areas of infrastructure support, software development, security, and research. The report's authors made three recommendation regarding DoD policies toward open source software. They included: creating a "Generally Recognized As Safe" open source software list (as it regards information security and reliability), developing generic infrastructure, development, security, and research policies to promote the broader and effective use of open source software, and encouraging the use of open source software to promote product diversity (to reduce the dependence on a single software product).[26]

In addition to defense-related purposes, it has been widely reported that other agencies making some use of open source applications include National Aeronautics and Space Administration (NASA), the Department of Agriculture, the Federal Aviation Administration, and the Department of Energy.[27] Currently, one of more common applications is the use of Apache, one of the leading open source server programs, to run government Web sites.[28] However, at this time there appears to be no centralized accounting of open source software throughout the federal government.

---

[24](...continued)
Software for High End Computing. October 2000. *Developing Open Source Software to Advance High End Computing*. [http://www.ccic.gov/pubs/pitac/pres-oss-11sep00.pdf].

[25]Carolyn A. Kenwood. July 2001. *A Business Case Study of Open Source Software*. The MITRE Corporation.
[http://www.mitre.org/support/papers/tech_papers_01/kenwood_software/].

[26]Terry Bollinger. 28 October 2002. *Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense*. The MITRE Corporation.
[http://www.egovos.org/pdf/dodfoss.pdf].

[27]Peter Galli. "German Gov't Moves to Linux," *eWeek*, 3 June 2002,
[http://www.eweek.com/article2/0,3959,4279,00.asp].

[28]Drew Clark, "Defense, Cybersecurity Officials Praise 'Open Source' Software," *Government Executive Magazine*, 29 October 2002, [http://207.27.3.29/dailyfed/1002/102902td2.htm]; Jonathan Krim, "Open-Source Fight Flares at Pentagon," *The Washington Post*, 23 May 2002, p. E1.

The growing emphasis on improved information security and critical infrastructure protection overall, will likely be an influential factor in future decisions on implementing open source solutions. The rapidly changing computer environment may also foster the use of a combination of open source and closed source applications, rather than creating a need to choose one option at the exclusion of another. In addition, as the largest buyer of information technology products and services in the world, the choices made by the federal government could have a larger impact on the future growth or decline of open source software overall.[29]

For Further Reading

Bollinger, Terry. *Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense*. The MITRE Corporation. 28 October 2002. [http://www.egovos.org/pdf/dodfoss.pdf].

Brown, Kenneth. Opening the Open Source Debate: A White Paper. Alexis de Tocqueville Institution. June 2002. [http://www.adti.net/html_files/defense/opensource_debate.html].

Kenwood, Carolyn A. *A Business Case Study of Open Source Software*. The MITRE Corporation. July 2001. [http://www.mitre.org/support/papers/tech_papers_01/kenwood_software/].

President's Information Technology Advisory Committee, Panel on Open Source Software for High End Computing. *Developing Open Source Software to Advance High End Computing*. October 2000. [http://www.ccic.gov/pubs/pitac/pres-oss-11sep00.pdf].

---

[29]Patrick Thibodeau, "Could Feds Foil Microsoft with IT Spending?," *Computerworld*, 10 June 2002, [http://www.computerworld.com/governmenttopics/government/policy/story/0,10801,71851,00.html].