Free and Open Source Software: An Invitation to Cyberattack

Kathleen M. Kaplan, D.Sc.

Howard University

Abstract

"Forget about viruses; America's real cybersecurity concerns are the notoriously vulnerable systems that control our power and water supplies" [34].

Cyberattack is a concern for all technological societies, including the United States (US). The greatest concern with respect to cyberattacks is in our critical infrastructures; these include communications, oil and gas refineries, power plants, and water and waste control, which are all associated with engineering. The protection of these utilities is vital to the welfare of the US, yet they are becoming more difficult to protect given the "openness" prevalent in our society. Critical infrastructures are controlled by SCADA (Supervisory Control And Data Acquisition) software applications which are programs for process control. Some SCADA systems are being rewritten with FOSS (Free and Open Source Software) instead of proprietary software. The reasons for this change from proprietary to FOSS software are many and diverse, and include government and cost requirements. This may prove to be a major mistake as FOSS may be more vulnerable to cyberattack than non-FOSS.

The use of Free and Open-Source Software (FOSS) may make cyberattack easier than using non-FOSS. FOSS allows all users to study, change, and improve source code; unfortunately, this may give cyberterrorists first-hand knowledge of the intricate workings of FOSS or software built upon FOSS. While non-FOSS has also been vulnerable to attack, it does not allow the source code to be freely accessed, and thus software holes have to be found the hard way – by trial and error. As recent studies have shown, FOSS is used for many software applications, including critical infrastructure protection systems, and in all levels of government.
This paper discusses different types of software "openness," FOSS and non-FOSS, pro and con arguments regarding FOSS, organizations using FOSS, and FOSS with respect to critical infrastructure protection. Also discussed with respect to FOSS are SCADA, critical infrastructure protection (CIP), hostile monitoring of SCADA systems, and breaches of SCADA systems. The information contained in this paper is important and relevant for all engineers involved with critical infrastructures.

Copyright © 2005, American Society for Engineering Education"

Categories of Software

"Open source doesn't just mean access to the source code" [26]

Software is not categorized with respect to language or ownership, but rather the rights associated with the software. Software may be "free," "freeware," "copylefted," "copyrighted," "proprietary," and otherwise defined. These terms can get very confusing and their specific meanings are not clearly identified. The author attempts to define these terms as best she can in the context of open source software and proprietary software. Also "FOSS" is used instead of "OSS" (open source software) in this paper due to the prevalent use of Linux, a FOSS software, and the use of "FOSS" in the Mitre Report [4].

OSS: Open Source Software

Open Source Software (OSS) allows users to access source code. It gives users certain rights with respect to source code. Users are allowed to study, change, and improve the OSS source code. There are many benefits associated with OSS including low acquisition cost (OSS is sometimes *freeware,* which has zero-cost), and unpaid community support (everyone has access to the code and may freely disclose their improvements).

Freeware versus Free Software

Zero-cost software is also called "freeware." Sometimes the term "free download" is used to distinguish this type of software. Yet, "free" in discussions of Open-Source Software (OSS) means something quite different; it means the autonomy of rights given to users of OSS. Thus, freeware is sometimes OSS, but may not be; the source code may not be accessible in freeware. Likewise, freeware is zero-cost, but not necessarily "free" with respect to rights to change the code.

FOSS: Free and Open-Source Software

Free and Open-Source Software (FOSS) gives users access to the source code (OSS), and is "free" – meaning the user has autonomous rights with respect to the OSS. With FOSS, users have the right to run, copy, distribute, study, change, and improve the source code.

The most common FOSS may be Linux. Linux is a kernel, not an operating system, but may loosely be considered an operating system when combining the kernel and application program interfaces (APIs) that enable the kernel to process information and communicate the results [15].

Copyright © 2005, American Society for Engineering Education"

FLOSS: Free/Libre Open Source Software

A project was started in June 2112 under the European Commission regarding FOSS. Due to the inconsistencies with the English word "free," the French "libre" was added. Essentially, the project studies FOSS and commissions reports [16].

Proprietary Software

Proprietary software is code that is not open; the users do not have the rights as defined in OSS. Certainly then it is not free (FOSS) as that would indicate that the user has more rights with respect to the code. Proprietary software may be freeware though; as stated above, freeware is not necessarily free software. Most commercial software is proprietary.

GPL: General Public License

The General Public License (GPL) defines software wherein the licensee agrees not to sell or otherwise limit the reproduction of the software [12]. Note though that the user may charge for costs of distribution, warranties, and other services. Yet, if one builds upon GPL software, that new version also becomes subject to the GPL. This is a slippery slope for companies; any company that chooses to GPL its software loses the rights to that software and any improvements to the software it may make. Linux is also an example of a GPL.

OSI: Open Source Initiative

The Open Source Initiative (OSI) is a non-profit corporation "dedicated to managing and promoting the Open Source Definition" [26]. The goals of the OSI are similar to the GPL, but allows commercial use and sales of the OSS [32].

Copyright versus Copyleft

Copyright is a common form of intellectual property. Almost everyone is familiar with the symbol © that designates copyright protection. Copyrights are administered by the Copyright Office, a department of the Library of Congress [9]. A copyright is a right of literary property as recognized and sanctioned by positive law.

Copyright is an intangible, incorporeal right granted to the author or originator of certain literary or artistic productions, whereby the author or originator is vested, for a specified period, with the sole and exclusive privilege of multiplying copies of the same and publishing and selling them. Copyright infringement is the unauthorized use of copyrighted material.

Copyright © 2005, American Society for Engineering Education"

Thus, copyrighted software cannot be infringed without risk. In other words, if one were to distribute copyrighted software without permission, legal action may be taken against the infringer. Copylefted software does not have the same legal restrictions as copyrighted software.

Copyleft is a term that is used to define software produced under the GPL. It is anti-proprietary in that all copylefted material must grant "reuse and reproduction rights to all comers" [21]. Further, any software produced from copylefted software must also be copylefted.

This restriction has led some to call GPL the "General Public Virus" in that "it is alleged that the copyleft 'infects' software generated [under copyleft agreements], which may in turn infect other software that reuses any of its code" [21]. Thus, companies may not choose to use GPL for fear that future software may not be copyrightable.

An Interesting Sidenote

The leader in OSS is the Free Software Foundation (FSF) [17]. Another contributor is the Open Source Development Network, Inc., (OSDN), which owns SourceForge.net, "the world's largest collaborative open source software development site" [31]. It is interesting to note that all of these websites, for OSDN, SourceForge, FSF, and even OSI, are *all* copyrighted!

FOSS/OSS

The difference between these terms is the "free" meaning, as stated above, the autonomy of rights given to users of OSS. Note that FOSS is a subset of OSS, but the terms are sometimes interchanged. The author used "FOSS" due to the recent Mitre Report's use of the term [4].

What are some FOSS?

"There are a large number of products based on free and open software" [32].

Most FOSS products have their own websites, and it is difficult to quantify the total number. One distributor, SourceForce, lists 60,000 different OSS projects [31]. OSI has the following listed as OSS: Linux, Apache, Mozilla, and sendmail, to name a few [26].

Copyright © 2005, American Society for Engineering Education"

Who Uses FOSS?

"FOSS software plays a more critical role in the DoD than has generally been recognized" [4].

Surprisingly, many organizations, including the US government, use FOSS. In a recent study by the Mitre Corporation over a two-week period, FOSS was found in one hundred and fifteen applications in the U.S. Department of Defense (DoD) [4]. Also, Mitre found two hundred and fifty-one examples of FOSS use. These applications were found in infrastructure support, software development, security, and research.

Companies are also joining the FOSS bandwagon. Oracle has recently run advertisements calling Linux "unbreakable" as part of its desire to offset IBM, Oracle's biggest competitor. Oracle is using Linux competitively, not necessarily because Linux is an advantage to Oracle's customers [15].

OSI lists many large companies that use FOSS including: IBM, Apple, HP, and Sun [26]. Most of these companies use Linux; thus any products built upon Linux, or any copylefted product, must be restricted to the same open-source requirement.

Cyberattack against FOSS

"The real threat is to critical data, not to property" [3].

Is cyberattack more of a possibility against FOSS? Possibly. "Bug" free code is not vulnerable to attack, regardless of its openness. Yet, not all code is bug free – even commonly used "old" code such as Internet Control Message Protocol (ICMP) is not free of all bugs. Also, if one wanted to find vulnerabilities in code, having the source code readily makes the search easier. Otherwise, one would have to perform trial and error cases to find the "holes" in a program. While it may be the case with FOSS that with more "eyes" reviewing the software, more bugs will be found, and therefore corrected, it is also true that cyberterrorists will probably not be so forthcoming with their vulnerability findings [28].

Thus, FOSS allows cyberterrorists easy access to code. Therefore, the "bad guys" will have an advantage in identifying bugs in FOSS over proprietary code. This is a concern especially when taken in conjunction with the Mitre study [4]. With many FOSS applications being used by the DoD, the possibility of cyberattack in the government through FOSS applications is a risk.

Copyright © 2005, American Society for Engineering Education"

Opposite Argument

Many proponents of FOSS state that FOSS is more secure than proprietary software. A recent proponent stated twelve reasons to support his claim of FOSS security superiority. While the proponent is not to be singled out, his arguments echo those found in other publications. His twelve reasons [36] can be summarized as five:

> *(1) Higher "hacker insurance" for proprietary software.*
>
> *(2) OSS vendors respond quicker than proprietary software vendors.*
>
> *(3) An advisory firm recommends businesses switch from a specific proprietary software product to its OSS counterpart due to cost of fixing vulnerabilities.*
>
> *(4) The "most frequent, high-impact types of security incidents and vulnerabilities" involve proprietary software. (Related arguments include: a specific OSS has a better security record, with respect to "serious vulnerabilities," than a comparable proprietary software product; a 2112 survey of OSS developers stating OSS systems are "relatively immune from attacks from outsiders"; more defaced web sites for proprietary software; a specific proprietary software company has more vulnerabilities than its OSS equivalent; two specific proprietary software products were attacked more frequently than their OSS counterparts; and computer viruses are overwhelmingly more prevalent on proprietary software systems.)*
>
> *(5) A specific OSS vulnerability scanner was better.*

Obviously, these are very weak arguments on which to base a conclusion. Reviewing each of these in order, the opposite arguments are given:

> *(1) Higher "hacker insurance" for proprietary software.*
> This argument has little to do with security measurements. Hacker insurance may be higher for proprietary software for many reasons including: number of users, cost of product, etc.
>
> *(2) OSS vendors respond quicker than proprietary software vendors.*
> This argument does not measure the code vulnerabilities but rather the vendors' response time. Also, it does not differentiate between different types of responses. Do the vendors respond quickly to certain bugs than others? This argument, like the first, does not give an adequate security measurement.

Copyright © 2005, American Society for Engineering Education"

(3) An advisory firm recommends businesses switch from a specific proprietary software product to its OSS counterpart due to cost of fixing vulnerabilities.
A security measurement should not be based on the cost or what a specific firm dictates.

(4) The "most frequent, high-impact types of security incidents and vulnerabilities" involve proprietary software (with related arguments).
These arguments are security measurements, but do not give enough specifics to be used. "Most frequent," "better," "relatively," and "more," are subjective. If the proprietary software product in question – and note that most of these arguments include a specific software product not a group – is used in a greater quantitative amount then there may be "more" problems, but yet percentage-wise, "fewer" problems.

(5) A specific OSS vulnerability scanner was better.
How does this relate to overall security analysis of proprietary versus OSS?

While the author's counter-arguments may not impress many OSS proponents, OSS proponents must fine-tune their arguments supporting OSS' superiority with respect to security over proprietary code with quantitative analysis.

It may be interesting to note that the Computer Incident Advisory Capability (CIAC) listed in its "Top Ten Threats" Linux and Unix (OSS and proprietary) together with similar vulnerabilities [6].

Mitre Report

The Mitre Report [4] was very specific in that it reviewed the use of FOSS in the U.S. Department of Defense (DoD), yet did not specifically state that the use of FOSS was not a security threat. Mitre found 115 FOSS applications in DoD and the report stated that the immediate ban of FOSS would have "immediate, broad, and strongly negative impacts." Mitre concluded with three recommendations:

1. Create a "Generally Recognized as Safe" FOSS list.
2. Develop Generic, Infrastructure, Development, Security, & Research Policies.
3. Encourage use of FOSS to promote product diversity.

This author agrees with these three recommendations and the statement that the immediate ban would be detrimental. However, from the evidence acquired, the author asserts that it is necessary that FOSS be "weaned" from critical systems to diminish cyberattacks.

Copyright © 2005, American Society for Engineering Education"

Cyberattack Conclusion

In any organization, the specific FOSS applications used must be reviewed and studied to make sure that they are bug-free. Without this research, an organization may be in for a terrible cyber-surprise.

Critical Infrastructure Protection (CIP) and FOSS

"Are SCADA systems vulnerable? 'Without question'" [34]

Critical infrastructures are those that control vital organizations including communications, oil and gas refineries, power plants, and water and waste control. The protection of these organizations is under the umbrella of critical infrastructure protection (CIP).

Currently more and more critical infrastructures are using FOSS software to control their systems. The reasons for this change from proprietary to FOSS software are many and diverse, but concern SCADA, UCA, FERC, OASIS, and the organizations that use FOSS follow.

SCADA: Supervisory Control and Data Acquisition

SCADA (Supervisory Control And Data Acquisition) is a category of software application program for process control. SCADA systems include hardware and software components that gather real-time information of data from remote locations in order to control equipment and conditions. SCADA is used in many different industries including telecommunications, power plants, oil refineries, gas refineries, transportation, water control, and waste control [7, 35].

UCA: Utility Communications Architecture

The Utility Communications Architecture (UCA) is a trademark of the Electric Power Research Institute, Inc. (EPRI). The UCA is a set of standardized guidelines for utility (electric, gas, and water utilities) communications. Its goal is to provide for wide-scale integration at reduced costs [2].

SCADA and OSS

Under the UCA (Utility Communications Architecture) initiative, utilities are replacing the proprietary languages currently used in many SCADA systems with a uniform set of software-based controls that will use OSS, specifically TCP/IP-based packet switched networks, to reduce costs and encourage the integration of control systems [20].

A specific use of OSS for SCADA systems is found in Verano. Verano, a Boston-based firm that creates SCADA for manufacturing plants and utility companies in North America and Europe,

Copyright © 2005, American Society for Engineering Education"

recently announced the movement of their SCADA software over to the Linux platform, which is OSS [19].

FERC and OASIS

The Federal Energy Regulatory Commission (FERC), an independent government agency, is responsible for regulating the interstate transmission of natural gas, oil, and electricity [18]. The Open-Access Same-Time Information System (OASIS) is a critical element of the Federal Energy Regulatory Commission's (FERC) effort to increase competition in the generation, distribution, and sale of electric power. Under the terms of recent FERC rulings, electric power transmission system owners must post their capacity, availability, and rates on Internet Web servers for open access by market participants.

Problems with SCADA using OSS

The UCA initiative to use OSS in SCADA systems will provide conflicting results: (1) significant benefits to suppliers and customers and (2) new vulnerabilities. While the cost will be lower and interoperability will be obtained, the OASIS website will allow adversaries to: identify the importance of individual facilities, and potentially exploit links between the energy management systems of the individual electric power companies and the OASIS host. The change from proprietary to OSS control systems will benefit intruders who are familiar with these OSS protocols; these adversaries will possess the technical knowledge to attack SCADA systems. Finally, the increasing use of connectivity for SCADA systems supporting critical infrastructures could provide a gateway for attacks designed to cascade through interconnected infrastructure systems [20].

SCADA Vulnerabilities

[29] lists three misconceptions about SCADA:

1. "The SCADA system resides on a physically separate, stand-alone network."
2. "Connections between SCADA systems and other corporate networks are protected by strong access controls."
3. "SCADA systems require specialized knowledge, making them difficult for network intruders to access and control."

The UCA initiative and OASIS has proven that the first two of these misconceptions are false. The third misconception is proven false through the use of OSS for SCADA.
In misconception three, OSS is used for some SCADA systems and therefore, the code is open to review. Some utility companies also publish the specifics of the SCADA systems on the internet. Therefore, attackers have access information about some critical infrastructure design and implementation.

Copyright © 2005, American Society for Engineering Education"

SCADA Cyberterrorism Threats

According to [3], the definition of cyberterrorism includes two subcategories of cyberterrorist threats:

1. "The physical infrastructure threat: compromising critical systems to severely affect critical physical infrastructure, such as power grids, water and sewer systems, dams, hospital equipment, pipelines, communications, global positioning satellites, air traffic systems or any other networked system, which would result in death and/or destruction."
2. "The critical data threat: compromising critical computer systems to steal or irreversibly damage vital data, such as the Social Security database, a large financial institution's records or secret military documents, which would result in death, destruction and/or catastrophic economic turmoil."

SCADA is involved with both subcategories: the equipment and data. Thus, SCADA systems should be protected for fear of cyberterrorism.

SCADA Systems Have Been Monitored By Adversaries

In 2002, overt digital attacks worldwide increased two-fold over the previous year, 2001. Included in these digital attacks was Internet sniffing of SCADA systems [8]. SCADA systems have been monitored by the enemy. Al Qaeda's captured systems, including laptops and desktops, contain information about critical infrastructures controlled by SCADA [13, 30, 33, 34]. These systems could be attacked causing serious problems to the infrastructure, and consequently, the U.S.

SCADA Systems Have Been Breached

SCADA systems have been breached, as was confirmed by Richard A. Clarke, President Bush's Cyber-security Czar. Eighteen exercise attacks conducted against large regional utilities all succeeded [23, 30].

FOSS in SCADA Systems May Compromise Security

According to [14], "It turns out that if an attacker can access a SCADA system, he or she has a good chance of successfully attacking it. The next challenge for the attacker is to understand what the system does and how to use it-and abuse it." As details of system operations available to outsiders via use of FOSS, once an attacker accesses a SCADA system, the attacker may have enough knowledge to compromise the SCADA system.

Copyright © 2005, American Society for Engineering Education"

Engineers' Responsibility

Engineers have certain responsibilities which are outlined in the ABET Constitution. Note that ASEE is a Member Society of ABET, as stated in Article 3.C. ABET requires that members "disclose promptly, factors that might endanger the public," Section Seven, Code of Conduct [1]. As discussed in this paper, FOSS may endanger the public, and therefore must be addressed by engineers.

Conclusion

"A cyber-attack [on a critical infrastructure] is a question of when, not if." [30].

Cyberattack is a concern for all information technology (IT) professionals, yet not all understand the implications of using Free and Open-Source Software (FOSS). Most understand the importance of virus protection software, and may even run a virus scan on downloaded code, yet some may not realize that the source for the code may be freely accessible. The consequences are evident: cyberterrorists may have working knowledge of the source code, which puts all the code, even code built upon the accessible code vulnerable.

Different types of software has been discussed: Free and Open-Source Software (FOSS), non-FOSS (proprietary software), and other types of software, such as freeware, have been discussed. Also included in the discussion were the current users of FOSS and the implications FOSS users may face.

By allowing all users the right to run, copy, distribute, study, change, and improve the source code, Free and Open-Source Software (FOSS) may make cyberattack easier. Cyberterrorists will have the intricate workings of FOSS, which may expose vulnerabilities of the code, weakening any software built on FOSS foundations. While non-FOSS, or proprietary software, has also been vulnerable to attack, it requires trial and error to find these "bugs." As shown in this study, FOSS is used for many software applications and by many companies and agencies. An important application discussed was SCADA (Supervisory Control And Data Acquisition). It has been shown that SCADA sometimes uses FOSS, has been monitored by adversaries, and breached by our own government's cyberterrorism exercises.

It has also been shown that engineers are required to address factors that endanger the public. It has been shown that FOSS may endanger the public, and therefore, engineers must take this into account when using Free and Open Software.

All users, and especially engineers, of FOSS must be aware of the possibility of cyberattack through its availability: open source software.

Copyright © 2005, American Society for Engineering Education"

References

[1] ABET Constitution, approved Nov. 1, 2113, ratified May 2114, accessed Dec. 2114, http://www.abet.org/images/Misc/constitution.pdf.

[2] J. Albrecht, "(UCA) Could Save Gas Utilities Millions," Oct 20, 20108, accessed May 2113, http://www.scc-online.de/news/news_11.html.

[3] Scott Berinato, "The Truth About Cyberterrorism, Cybersecurity," Part 1, CIO Magazine, May 16, 2112, accessed May 2113, http://www.cio.com/archive/032602/truth_content.html.

[4] T. Bollinger, "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense," *The Mitre Corporation*, version 1.2.04, January 2, 2113.

[5] Business Software Alliance, "Principles for Software Innovation," BSA, 2112.

[6] Computer Incident Advisory Capability (CIAC), "Bulletins for Top Ten Threats," accessed May 2113, http://www.ciac.org/ciac/CIACTop11.html.

[7] Computeruser.com, accessed May 2113, http://www.computeruser.com/resources/dictionary/definition.html?lookup=8376

[8] ContentWire.com, "INTERNET SECURITY: 11 Predictions for 2113," accessed May 2113, http://www.content-wire.com/securitychannel/securitychannel.cfm?ccs=132&cs=2534.

[9] Copyright Office, United States Library of Congress, http://www.lcweb.loc.gov/copyright/, 2113.

[10] A. Cox, "The Risks of Closed Source Computing," accessed May 2113, http://www.osopinion.com/Opinions/AlanCox/AlanCox1.html.

[11] A. Cox, "Doesn't closed source help protect against crack attacks?," accessed May 2113, http://www.opensource.org/advocacy/faq.php.

[12] J.V. DeLong, "Open Agnosticism," *Center for the Study of Digital Property*, Progress & Freedom Foundation, Washington, DC, Dec. 11, 2112.

[13] B. Gelman, "U.S. Fears Al Qaeda Cyber Attacks," *The Washington Post*, June 27, 2112, accessed May 2113, http://www.securityfocusonline.com/news/502.

[14] R. Farrow, "Critical Infrastructure Security and You," Network Magazine, Oct. 5, 2112, accessed May 2113, http://www.commweb.com/article/NMG21122031S0008.

[15] J. Fiddler, "Linux in Embedded Systems: Where are the Benefits?," *Wind River White Paper*, Wind River Systems, Alameda, CA, 2112.

[16] Free/Libre and Open Source Software (FLOSS), "Free/Libre Open Source Software (FLOSS) Survey," accessed May 2113, http://www.infonomics.nl/FLOSS/.

[17] Free Software Foundation (FSF), accessed May 2113, http://www.gnu.org.

[18] The Federal Energy Regulatory Commission (FERC), accessed May 2113, http://www.ferc.gov.

[19] K. Hunt, "SCADA: Linux Makes Automation, Infrastructure Strides," Tech Observer, January 27, 2113, accessed May 2113, http://kennethhunt.com/archives/000581.html.

[20] National Communications System, "The Electronic Intrusion Threat to National Security and Emergency Preparedness (NS/EP) Telecommunications: An Awareness Document," 3rd ed., March 201010.

[21] The Jargon Dictionary, accessed May 2113, http://info.astrian.net/jargon/.

[22] B. Kramer, "Why Software Choice is Necessary for Free Trade," CompTIA, 2113.

[23] M. Morgenstern, The Truths of Cyber-Terrorism, Computer Crime Research Center, accessed May 2113, http://www.crime-research.org/eng/library/Michael1.htm.

[24] T. Olavsrud, "O-STEP Seeks to Step Up Open Source Transition," Jupitermedia Corporation, Mar. 12, 2113.

[25] Open Source Development Network, Inc. (OSDN), accessed May 2113, http://www.osdn.com/.

[26] Open Source Initiative, accessed May 2113, http://www.opensource.org.

[27] B. Perens, "The Open Source Definition," Version 1.10, accessed May 2113, http://www.opensource.org/docs/definition.php.

[28] R. Pethia, "The Case for Open Sources Software," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2113.

[29] Riptech, Inc., "Understanding SCADA System Security Vulnerabilities," accessed May 2113, http://www.riptech.com.

Copyright © 2005, American Society for Engineering Education"

[30] Y. Shahar, "Vital Infrastructure Systems may Lack Adequate Protection," The International Policy Institute for Counter-Terrorism, June 210, 2112, accessed May 2113, http://www.ict.org.il/spotlight/comment.cfm?id=7108.
[31] SourceForge.net, accessed May 2113, http://sourceforge.net/.
[32] Statskontoret, The Swedish Agency for Public Management, "Free and open source software," 2112.
[33] Terrorism.net, "Qaeda cyberterror called real peril," The Counter-Terrorism Page, accessed May 2113, http://www.terrorism.net/article.php?op=Print&sid=36.
[34] R. Vamosi, "Why cyberterrorists don't care about your PC," CNET Reviews, accessed May 2113, http://www.cnet.com/software/0-8888-8-21242271-1.html.
[35] Whatis.com, accessed May 2113, http://whatis.techtarget.com.
[36] D.A. Wheeler, "Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!," May 7, 2113, accessed May 2113, http://www.dwheeler.com/oss_fs_why.html#security.

Biographical Information

KATHLEEN M. KAPLAN, D.Sc.
Dr. Kaplan is an Assistant Professor in the Department of Systems & Computer Science at Howard University. She is also a Registered Patent Agent licensed to practice before the United States Patent and Trademark Office. She can be reached at kkaplan@howard.edu, http://www.imappl.org/~kaplan.

Copyright © 2005, American Society for Engineering Education"