GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Security Concerns in Using Open Source Software for Enterprise Requirements

SreenivasaRao Vadalasetty
GSEC Practical Assignment - Version 1.4b Option 1
October 15, 2003

Abstract

Information security is the biggest challenge for network and security administrators. The security of a given network highly depends on the software used and the administrative practices followed for operating systems, perimeter security, antivirus protection, intrusion detection, software development, systems and network monitoring, corporate mail, office productivity and so on. The rapid growth in Internet has resulted in several open source development communities. The collaborative effort of these communities has made it possible to have open source alternatives for almost all proprietary (also known as closed source) software. This paper highlights the security concerns of the end users in considering open source software for their enterprise requirements. This paper also highlights the risks pertaining to open source software and recommends certain guidelines following which these risks can be mitigated. These guidelines would help an end user to thoroughly evaluate open source software before they are considered for mission-critical functions.

Open source software

The words “Open Source” and “Open Source Software” refer to the software whose source code is available to the public and it can be used, modified and redistributed along with the original rights as defined by Open Source Initiative (OSI).[1] These two terms are interchangeably used in the rest of this document. It is always distributed under a license which allows the user to use it the way he wants either for customizing it for his specific needs or for designing a commercial solution based on it. GNU General Public License (GPL) is the most commonly used license for this purpose. The derived solutions based on open source software should be distributed along with the source code and the recipient should get the same rights with which the original source is distributed. The word ‘open source software’ is sometimes misused to refer to the software whose source code is available but there are restrictions on its usage, modification and redistribution. Most of the universities, educational institutions and non-profit organizations use open source software. Many enterprises also use open source software but most of them do not disclose this information for various political and security reasons. Open source software is in fact so ubiquitous that the running gears of Internet such as mail transports and web servers mostly run on open source software.

[1] “The Open Source Definition.” URL: http://www.opensource.org/docs/definition.php

© SANS Institute 2000 - 2005 Author retains full rights.

The usage of open source software offers several advantages such as vendor independence, the reduced total cost of ownership in designing a solution, the flexibility to customize the code for site specific needs and moreover the feasibility of reviewing the entire source code for potential bugs and vulnerabilities before deploying the product in an enterprise.

Security in open source software

Security has become an important aspect and an integral part of all the phases of any software development. The trustworthiness of any software, either open source or closed source, depends on certain key aspects of the product design and development. These include the expertise and dedication of the developers to develop a secure product, quality of tools used in development, the level of testing carried out before releasing the product and the matured practices followed throughout the development cycle.

Once the open source software is made available to public, anyone and everyone interested in the product could review the source code to assess its quality and reliability. As this allows more users and experts around the world to go through the source code, the bugs could be discovered and fixed early. However, open source software would be benefited by this peer review process only when the people reviewing the source code were qualified enough and they reviewed it with the intention of discovering vulnerabilities for the good of society. [2] Though the open source has potential to be more secure than its closed source counterpart, it should not be taken for granted that open source is more secure because there are some constraining factors. Despite the fact that the source code is available for everyone, several vulnerabilities in open source remain undiscovered for the following reasons. The source code of some popular open source products such as Linux, Apache web server would reasonably be peer reviewed by several users and security experts around the world. But, it should not be assumed that the source code of every open source product would be reviewed by security experts at this level. Most of the time the users of open source review the source code to customize the product to their needs in their environment. If they happen to come across any bugs in this process, they fix them if possible or share them with open source community so that a patch could be developed with a collaborative effort. Only a few experts review the open source software with an intention to figure out the potential vulnerabilities in the product. The complexity of the product and the limited documentation provided along with most of the open source products make it a tough job for the reviewers to properly analyze them. The several vulnerabilities in sendmail which were undiscovered for a long time stand as best examples for these facts.[3]

---

[2] Gene, Spafford. “Is Open Source More Secure?” 5 Dec 2002. URL: http://www.techtv.com/screensavers/linux/story/0,24330,3406300,00.html

[3] Elias, Levy. “Is open source more secure than closed?” 17 Apr 2000. URL: http://www.securityfocus.com/printable/news/19

© SANS Institute 2000 - 2005 Author retains full rights.

Risks in using open source software

The following are certain risks in using the open source. Some of the risks mentioned below are inherent while the other risks might arise due to poor software management practices.

Absence of meticulous evaluation

If a company was to buy a commercial closed source solution for an enterprise use, a formal procedure would be followed before it finalizes a specific product. It would include items like conducting a requirement analysis, defining acceptance criteria, evaluating the product, comparing the product with its competitive solutions available in the market for its functionality and security features and so on. But, an open source product might not undergo this kind of scrupulous evaluation, especially when administrators and users have the liberty to install the open source software without any approvals. This would pose a lot of business and security risk and lead to some unanticipated costs such as the company losing the credibility among its customers and eventually losing the business as well.

Spurious open source

As the open source makes the source code available to the public, even amateurs could easily design and distribute some malware by embedding malicious code into the original open source distribution. They could then show off some exciting features in their malware attracting some innocent end users. If an organization does not have a clear security policy on the usage of open source, its administrators (if they are not security conscious) may happen to download and install some spurious open source from some unreliable sources. The malware thus downloaded and installed could in fact be performing some undesired activities apart from offering the interesting services that the administrator actually downloaded and installed it for. For instance, if an administrator comes across a free application (developed based on some genuine open source) offering some attractive features like intensive monitoring of all the servers in a domain, an administrator might just be tempted to immediately download and install it (with administrator/super-user privileges in many cases) to have a better monitoring system in place. But, if such tool is not a reliable open source, it might leave a backdoor for the remote attacker, or upload some sensitive system or corporate information as designed and instructed by the attacker.

Lack of sponsorship

Most of the popular open source software is normally maintained by a consortium which consists of a group of individuals and/or organizations dedicated to further enhance and maintain them. The efforts of such a

© SANS Institute 2000 - 2005, Author retains full rights.

consortium would typically be supported by grants from generous sponsors which could be individuals or organizations. For instance, Internet Software Consortium (ISC) sponsored by various companies develops and maintains various commonly used Internet technologies such as BIND, DHCP, and NNN. Not all open source products receive a great sponsorship as the popular open source products such as Linux, BIND, Apache and so on. If the open source product in question is not very popular (not widely deployed) or it is not well sponsored, it may become difficult for getting patches for the discovered vulnerabilities. The organizations using such an open source might not always have enough expertise available in-house to fix the bugs and develop patches for themselves.

Guidelines for deploying open source software in an enterprise environment

Here are some guidelines that could be considered before deploying any open source software in an enterprise environment.

Security policy

The first and foremost thing that any enterprise should do to maintain a secure network is to come up with a well documented security policy. An enterprise would realize the real benefits of open source only when the security policy contains clear guidelines about the installation and maintenance of open source. The policy should be explained and available to everyone in the organization by means of member handbooks, security awareness programs and so on. A well defined policy should clearly explain the scope, the basic guiding principle (a policy statement) and define the roles and responsibilities without any ambiguities. There should be a dedicated team consisting of system, network and security administrators, which is responsible for implementing the policy, ensuring that the policy is strictly adhered to and revising the policy as the business needs change. It is very important to ensure that the members of this team are really committed to maintain a secure infrastructure and the enterprise should find ways to help them stay current on security tools and technologies.

Evaluation

Any open source considered for an enterprise use should be thoroughly evaluated. The most crucial information about the product should be gathered from some trusted sources to see if the risks of using such an open source product fall with in the acceptable risk by the organization. The security levels promised or published by somebody should not be taken for granted because the security requirements vary from configuration to configuration and from site to site. Let us just take a look at what an end user can do to evaluate a given open source product.

The ISO (International Organization for Standardization) standard

© SANS Institute 2000 - 2005 Author retains full rights.

"ISO/IEC 15408:1999", commonly known as "Common Criteria" (CC), is used to assess security and assurance of IT products. When IT products undergo CC evaluation, they would be evaluated by independent laboratories against strict standards for various features, such as the development environment, security functionality, the handling of security vulnerabilities, security related documentation and product testing. 'Common Criteria' (CC) certified products give customer an unbiased assessment of the respective product. Though most CC certification is not affordable by most of the open source developers, almost all the Linux vendors are going for CC certification for their implementations of open source based Linux. When a product is CC certified, its "Security Target" will be made available to the public. The security target refers to a document which gives information as to what security requirements the product was tested for and the configuration of machines it was tested on and so on. If the open source in question has been CC certified and the configuration described in its security target is similar to the user's configuration and environment, the user can use the evaluation results for security assurance. [4]

If a product is not CC certified and there are no security evaluation results available, the end user can himself do some miniature analysis based on CC to see if the product meets the basic security requirements. This process includes identifying the security environment and the security objectives of the user and verifying whether the product meets his security requirements (both assurance and functional) or not.[5]

- The security environment (the environment that the product is going to be used in) should be clearly analyzed to identify the possible threats, and assumptions. The potential threats include attacks from insiders (like inexperienced users, disenchanted employees) or outsiders (like unscrupulous competitors, terrorists). Identifying the security environment also includes understanding the various ways in which the systems could be attacked and classifying the data and protecting the data accordingly Note that the existing organizational security policy greatly influences the security environment by imposing the restrictions on what services are restricted and what are allowed for the external systems to communicate with the Internet connected systems of the organization.
- The key security objectives of the user would typically be to protect the confidentiality, integrity and availability of the system and expecting the product to have provisions for authentication and auditing.

---

[4] David. A. Wheeler. "How to evaluate Open Source Software / Free Software (OSS/FS) Programs" 28 Sep 2003. URL: http://www.dwheeler.com/oss_fs_eval.html

[5] David. A. Wheeler. "Secure Programmer: Developing secure programs" 21 Aug 2003. URL: http://www-106.ibm.com/developerworks/linux/library/l-sp1.html?ca=dgr-lnxw04SecureProgram

© SANS Institute 2000 - 2005 Author retains full rights.

- Having understood the security environment and security objectives clearly, the user should thoroughly verify the product to see if it is able to meet the assurance and functional requirements – the essential security requirements of any product in the given security environment. The product meets assurance requirements when it does not do anything else apart from what it is supposed to do and its behavior is in consistent with the documentation provided. The product meets its functional requirements when it is able to implement the security objectives of the user i.e., protecting confidentiality, integrity and availability of the system.
- If there is enough coding expertise available in-house, try to get the source code reviewed by the expert programmers. If this is not practical, at least use the source code scanners to identify the potential security problems in the source code. Flawfinder and RATS (Rough Auditing Tool for Security) are two source code scanners distributed under GPL. Flawfinder can be used to scan C and C++ code while RATS can audit C, C++, Perl, PHP and Python source code. It is important to note that these source code scanners do a mere pattern matching to highlight the areas of the code which make the products vulnerable. Any such weak areas of the source code could lead to security risks such as buffer overflows, racing conditions, shell meta character dangers and poor random number acquisition. Note that these scanners do not understand the semantics of the code. Nevertheless, the usage of source code scanners will never match an expert auditing the source manually for security vulnerabilities. [6]

If the organization does not have enough expertise or can not afford to carry out such a comprehensive analysis as explained above, the administrators could at least follow some simple steps to ensure that the open source in question is reliable, well developed and maintained. It should however be noted that it is worth spending enough time and money in thoroughly evaluating the product for security vulnerabilities before deploying it in an enterprise, especially when considering open source for some mission critical function. The user, apart from testing the product for the required functionality, must consider the following points related to security before finalizing the product installation.

- Go through the documentation of the product such as user guides to see if it explains how to configure and keep the product secure.
- Check if the project involved in developing the product has a process for reporting its users about the discovered vulnerabilities. Examine the respective developer mailing lists to see if the security issues and the ways to maintain the product security are discussed and well addressed.
- Examine the security vulnerability databases to see the vulnerabilities discovered in the product. The user may find the vulnerability information maintained by CVE (Common Vulnerabilities and Exposures) and “the

[6] Jose, Nazario. “Source Code Scanners for Better Code” 26 Jan 2002.
URL: http://www.linuxjournal.com//article.php?sid=5673

© SANS Institute 2000 - 2005 Author retains full rights.

CERT® Coordination Center" helpful for this purpose. It should be noted that CVE maintains only a list or dictionary that provides common or standardized names for publicly-known information security vulnerabilities and exposures. CVE is just a dictionary but not a comprehensive vulnerability database on its own. The standardized names given by CVE to information security vulnerabilities and exposures make it easier to share data across separate vulnerability databases and security tools.[7] The "CERT® Coordination Center" (CERT®/CC) is a major reporting center for Internet security problems. The CERT®/CC analyzes the security vulnerabilities, works with various security experts to find the solutions for various security problems and disseminates this information in a timely fashion to the Internet community by means of public mailing lists. [8]

Avoid ad-hoc installations

Administrator/super-user privileges given to users or system administrators should not imply that they can download and install any open source that they like. Any open source that is installed and used in an enterprise environment should properly be maintained. This becomes a reality only when open source installation goes through a formal installation process. There should be people identified to be responsible for keeping track of the open source products installed and maintaining the same secure and up-to-date by installing patches and upgrading to latest versions as and when they are released.

Download open source software only from trusted sites

The sites recommended by Open Source Initiative can be considered reliable. freshmeet.net, sourceforge.net, osdir.com, developer.berlios.de and bioinformatics.org are the sites recommended by Open Source Initiative.

Prefer source code to binaries wherever possible

Most of the open source products are provided in both source code and package formats. One should not blindly trust and use these binaries because they may not have been compiled with the source code with which they are associated and shown. It is always good to download the source code, verify against the MD5 checksums provided, analyze it for security vulnerabilities and compile it to the site specific needs.

Scan for vulnerabilities

Use vulnerability scanners to scan the network for vulnerabilities. Note that it is very important to get written approval from the concerned authorities in an organization for scanning the network for vulnerabilities. Scan one subnet at

[7] "About CVE". URL: http://www.cve.mitre.org/abou
[8] "The CERT® Coordination Center FAQ". URL: http://www.cve.mitre.org/about

© SANS Institute 2000 - 2005 Author retains full rights.

a time or any such small significant portion of network rather than an entire network because the scanners might bog down the network. Apart from various commercial products available for this purpose, SARA (Security Audit Research Assistant) and Nessus are two freely available vulnerability scanners. SARA scans for the top twenty vulnerabilities posted by SANS and FBI while Nessus can scans for all the vulnerabilities as defined by its plug-ins. Each plug-in of Nessus scanner tests for a specific vulnerability and plug-ins can be written in C or its own Nessus Attack Scripting Language (NASL).

Disable unwanted services

As true with any other software, configure the product in the most secure way possible by disabling the unwanted services and following 'deny by default if not explicitly permitted' model.

Have Defense-in-Depth strategy

The 'Defense-in-Depth' strategy should be followed and the strategy should include all the possible measures that need to be taken for all open source products along with other products in the network. Note that 'Defense-in-Depth' is a concept which helps to maintain a secure infrastructure by following a layered approach, i.e. defending against various threats at various levels right from application to the network. The idea is to make sure that we have a robust defense strategy in place.

Install and forget model is very dangerous

Open source software should be audited periodically and maintained like any commercial software. The people in charge of maintaining the open source software in an organization should subscribe themselves to respective open source security announcement mailing lists and they should religiously install patches and carry out the software upgrades.

Training and documentation are important

Administrators who are in charge of maintaining open source software should be trained on the relevant core technologies used in developing open source products and security aspects so that they can effectively manage them. Get a proper documentation done for all the practices and configurations to avoid the problems that might arise due to various reasons such as people leaving the organizations.

Consider open source software in DR and BC plans

Ensure that the disaster recovery and business continuity plans are thoroughly updated with the organization's dependence on the deployed open source products.

Is open source software really enterprise-ready?

© SANS Institute 2000 - 2005 Author retains full rights.

Despite the fact that there are several enterprises using open source to run mission-critical functions, the CIO's of some enterprises are still very concerned to prefer open source to the proprietary software for their enterprise requirements. Their major concern is about the support available for various open source products. Relying on voluntary help from someone over the Internet rather than on a vendor for fixing bugs and security vulnerabilities is what bothers them. The service received from support partners under so called service-level-agreements is what suits most of the business models. The websites of most of the open source products list the support partners who provide technical support and offer service agreements for the respective open source software. While it is true for most of the popular open source products such as Linux, BIND and Apache, all the open source products do not have vendors offering commercial support. Besides that, it is not very easy and practical to get some genuine vendors to migrate the customer's complex enterprise applications (like CRM and ERP applications, trading and settlement systems) to open source and offer an ongoing support for them. This situation and the lack of expertise in enterprises thus leave a scope for proprietary software vendors to still retain their presence in the market. Enterprises should make a wise decision after a meticulous risk analysis about choosing open source for their requirements. When an enterprise is considering open source for the first time, it is suggested to start with less critical functions such as web servers (like Apache on Linux), realize the benefits and then leverage open source for mission-critical functions after rigorous evaluation. [9]

Statistics pertaining to installations, market share, vulnerabilities and exploits could help us understand how confident and secure the users feel in using open source for their enterprise requirements. Linux has become popular and the commercial vendors like Red Hat and SuSe have been releasing enterprise-ready operating systems by assuring security and offering the support required for mission-critical functions. The famous websites google.com and yahoo.com are hosted on Linux and FreeBSD respectively. The Netcraft's web server survey in October 2003 reveals that about 65 percent of the Internet connected web servers run Apache web server.[10] Most of the sites prefer ISC's BIND for running DNS to any other proprietary software. OpenBSD operating system has been well known for being secure – mainly because of the focus and a rigorous testing carried out to keep the product secure. The initiatives taken by major vendors such as IBM, HP, Sun and Oracle to promote and support open source solutions are very promising for the end users considering open source for their enterprise needs.

The software giant Microsoft itself genuinely perceived the open source

---

[9] Christopher, Koch. "Your Open Source Plan." CIO Magazine - 15 Mar 2003 Issue. URL: http://www.cio.com/archive/031503/opensource.html

[10] "October 2003 Web Server Survey." Netcraft. URL: http://news.netcraft.com/archives/2003/10/01/october_2003_web_server_survey.html

© SANS Institute 2000 - 2005 Author retains full rights.

movement as a threat to its business.[11] This clearly indicates the competency of open source software with which it is flourishing. Comparing the number of vulnerabilities of an open source product with its proprietary counterpart during a certain period of time may not always yield a good comparison report because various factors affect the meaning that is derived out of such comparison. The factors include the severity of vulnerabilities, impact of exploits and the difference in the subcomponents integrated within each product and so on. Most of the vulnerabilities in open source products were discovered much before they were exploited to affect the enterprises. This has been possible mainly because the source code is available to the public right from the development stage itself. At the same time open source can not be considered as a panacea for all the security problems.

Conclusion

The open source trend would keep turning the chunks of IT infrastructure into commodities by offering alternate solutions to proprietary software. As this trend continues, the enterprises would have equivalent or better open source alternatives available for their enterprise requirements. The end users would continue to look out for security assurance in open source products before considering them for mission-critical enterprise requirements. There are greater chances for most of the vendors to change to a support/service model from their ownership model by offering various support services for open source products. As part of this effort, the popular open source products would receive sponsorship from various vendors to undergo rigorous security evaluation and certification. The enterprises should do an extensive risk and security analysis before choosing open source solutions over their closed source counterparts. The analysis should consider various factors such as the expertise available in house and the support options available for the respective open source product. Well documented and implemented security policies and best practices help an enterprise to mitigate the risks and enjoy the real benefits of open source.

[11] David, Legard and Stacy, Cowley. “Is Microsoft Afraid of Open Source?” 5 Feb 2003. URL: http://www.pcworld.com/news/article/0,aid,109234,00.asp

© SANS Institute 2000 - 2005 Author retains full rights.

List of References

1. "The Open Source Definition." URL: http://www.opensource.org/docs/definition.php accessed on October 15, 2003.
2. Gene, Spafford. "Is Open Source More Secure?" 5 Dec 2002. URL: http://www.techtv.com/screensavers/linux/story/0,24330,3406300,00.html accessed on October 15, 2003.
3. Elias, Levy. "Is open source more secure than closed?" 17 Apr 2000. URL: http://www.securityfocus.com/printable/news/19 accessed on October 15, 2003.
4. David. A. Wheeler. "How to evaluate Open Source Software / Free Software (OSS/FS) Programs" 28 Sep 2003. URL: http://www.dwheeler.com/oss_fs_eval.html accessed on October 15, 2003.
5. David. A. Wheeler. "Secure Programmer: Developing secure programs" 21 Aug 2003. URL: http://www-106.ibm.com/developerworks/linux/library/l-sp1.html?ca=dgr-lnxw04SecureProgram accessed on October 15, 2003.
6. Jose, Nazario. "Source Code Scanners for Better Code" 26 Jan 2002. URL: http://www.linuxjournal.com//article.php?sid=5673 accessed on October 15, 2003.
7. "About CVE". URL: http://www.cve.mitre.org/about accessed on October 15, 2003.
8. "The CERT® Coordination Center FAQ". URL: http://www.cve.mitre.org/about accessed on October 15, 2003.
9. Christopher, Koch. "Your Open Source Plan." CIO Magazine - 15 Mar 2003 Issue. URL: http://www.cio.com/archive/031503/opensource.html accessed on October 15, 2003.
10. "October 2003 Web Server Survey." Netcraft. URL: http://news.netcraft.com/archives/2003/10/01/october_2003_web_server_survey.html accessed on October 15, 2003.
11. David, Legard and Stacy, Cowley. "Is Microsoft Afraid of Open Source?" 5 Feb 2003. URL: http://www.pcworld.com/news/article/0,aid,109234,00.asp accessed on

© SANS Institute 2000 - 2005 Author retains full rights.

October 15, 2003.

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.