A SECURITY COMPARISON OF OPEN-SOURCE AND CLOSED-SOURCE OPERATING SYSTEMS

Kishen Iyengar
Dept. of Information Systems and Operations Management, University of Texas at Arlington, Box 19437 University of Texas at Arlington, Arlington, TX-76019, kiyengar@uta.edu, Tel: 817-272-3584

Vishal Sachdev
Dept. of Information Systems and Operations Management, University of Texas at Arlington, Box 19437 University of Texas at Arlington, Arlington, TX-76019, vsachdev@uta.edu, Tel: 817-272-3584

M.K. Raja
Dept. of Information Systems and Operations Management, University of Texas at Arlington, Box 19437 University of Texas at Arlington, Arlington, TX-76019, raja@uta.edu, Tel: 817-272-3540

ABSTRACT

The argument whether open-source code is more secure than proprietary software has more or less been one of opinions and beliefs and not based on empirical tests and validation. In this paper, we empirically examine vulnerabilities data and compare closed-source and open-source operating systems. Our results indicate that although there are differences in the total number of vulnerabilities reported, there is no difference in the average risk per vulnerability between open-source and proprietary operating systems.
Keywords: Open-source, Security

Introduction

The argument between open-source code and closed-source code has existed for many decades now. In terms of security, proprietary software and open-source software both have their own support bases who intensely advocate their point of view (Hunter 2004, Schulz 2000). The discussion has more or less been one of opinions and beliefs and not based on empirical tests and validation (Chaudri 2004). This is probably due to the fact that it is difficult to ascertain the right metrics of measuring security and collect data.
The operating system is probably the most crucial piece of software that runs on any computer. Therefore it is important to examine security aspects of operating systems and compare closed-source and open-source operating systems. Bugs and Vulnerabilities in the operating system could make it at risk of attack and compromise. Such a comparison of open and closed-source operating systems could be very useful to conclude the debate on Open and Proprietary sources of software. This issue has been addressed by many researchers. (Balle 2004, Claasen 2004). Post (2003) conducted a survey of IT security professionals and found that they felt it difficult to keep up with patches for operating systems. Interdependencies of software on the operating systems have also been examined (Wang 2003.)
Specifically, Microsoft has received a lot of attention and focus for security issues related to its products. Schulz (2003, 2002) specifically looks at why Microsoft gets such unwanted attention

in terms of security issues. Indeed, security seems to be a pivotal factor in the emerging popularity of Linux (Goad 2000.)

Comparison of open-source versus proprietary OS security

So, are Microsoft's products inherently less secure than other products? Specifically, is the Windows Operating system less secure than other operating systems such as Unix and Linux. This forms the crucial step in a better understanding and final resolution for the debate over open-source and closed-source code. In order for such a comparison to be possible, the source of data needs to be neutral and unbiased. Measures for quality are often tricky to devise. The inherent subjectivity in measurement leads to questions regarding rigor. In this study, we endeavor to compare the Windows operating system with other operating systems in order to answer the question of quality differences pertaining to security between open-source and proprietary operating systems.

Data Source

The United States Computer Emergency Readiness Team, or US-CERT, publishes cyber security bulletins every week. Cyber Security Bulletins provide weekly summaries of security issues and new vulnerabilities. They also provide patches, workarounds, and other actions to help mitigate risk. These bulletins are available for viewing in their website at http://www.us-cert.gov/cas/bulletins/. Each of these weekly bulletins provides a summary of new or updated vulnerabilities, exploits, trends, viruses, and Trojans.
The bulletins provide information about vulnerabilities/bugs/holes that have been identified, even if they are not being exploited. The first part of the listing contains vulnerabilities identified only in the Windows Operating system. The second listing contains bugs identified only in the Unix/Linux operating systems. Vulnerabilities that affect both operating systems are tabulated under the third listing titled "Multiple operating systems." Each list contains information in five columns. These five columns contain data on fields which are Vendor and Software Name, Vulnerability, Common Name, Risk and Source

Risk Levels

The Risk field consists of ranking the vulnerability as a High, Medium or a Low risk based on how the system may be impacted. The levels of risk are defined as follows:
High - A high-risk vulnerability is defined as one that will allow an intruder to immediately gain privileged access (e.g., sysadmin or root) to the system or allow an intruder to execute code or alter arbitrary system files. An example of a high-risk vulnerability is one that allows an unauthorized user to send a sequence of instructions to a machine and the machine responds with a command prompt with administrator privileges.
Medium - A medium-risk vulnerability is defined as one that will allow an intruder immediate access to a system with less than privileged access. Such vulnerability will allow the intruder the opportunity to continue the attempt to gain privileged access. An example of medium-risk vulnerability is a server configuration error that allows an intruder to capture the password file.
Low - A low-risk vulnerability is defined as one that will provide information to an intruder that could lead to further compromise attempts or a Denial of Service (DoS) attack. It should be noted that while the DoS attack is deemed low from a threat potential, the frequency of this type

Operating System	Number of Vulnerabilities (%)	Cumulative Risk (%)	Risk per Vulnerability

of attack is very high. DoS attacks against mission-critical nodes are not included in this rating and any attack of this nature should instead be considered to be a "High" threat.

Data Collection

From the CERT website (http://www.us-cert.gov/cas/bulletins/index.html), data was gathered on the total number of vulnerabilities for Microsoft, Unix/Linux, and Multiple Operating Systems. Each risk was ranked 1, 2 or 3 depending on their risk level for Low, Medium and High Risk respectively. This enabled us to quantify the risk for better power in explanation and analysis. The Risk Rank was then summed to arrive at a total risk index value for Windows, Unix/Linux and Multiple Operating systems. Although the CERT website contains information from January 21st, 2004, data was collected only from June 9th to November 2nd 2004 because of resource constraints.

Table 1 - Sample of Collected Data

Although the data in the CERT website is clearly well arranged and unambiguously defined, we encountered two problems in collecting the data. The first was that although most data was weekly, two-week data prior to August 31st was grouped into one sheet. However, this problem was solved by sorting this data into each of the two weeks based on the date provided in the

Week	Windows	Unix/Linux	Multiple OS	Windows Tot. Risk	Unix/Linux Tot. Risk	Multiple OS Risk
Oct 20 – Oct 26	30	51	31	68	115	68

'Source' field Column. When there were multiple sources of data, the 'Source' field had multiple dates listed. In such a situation, we used the earliest date to classify the vulnerability.

A potentially more serious problem occurred in the collection of the 'Risk level' data. Few vulnerabilities were listed in the database as "High/Low." We ran our analysis without such items and found that their influence was minimal to non-existent. Since such cases were very few, less than 10 from among more than a thousand cases, and the fact that they introduced no bias in the analysis, we decided to retain them in the analysis. Instead of resorting to list wise deletion of such vulnerabilities, they were averaged out and given a Risk rating of Medium.

Analysis

The tabulation below shows the distribution of vulnerabilities (or bugs/Holes) over a period of five months. A total of 1271 vulnerabilities were listed, with Unix/Linux (Open-source) Operating system the highest at 45.55%. Multiple OS and Windows Operating systems (Proprietary) were similar with 28% and 26% respectively.

The results for cumulative risk were similar to the number of bugs for each operating system. Unix/Linux was highest at 44%, followed by Multiple OS(29.51%) and Windows(26.44%.) The average risk per vulnerability was highest for Multiple OS with a score of 2.37, followed by Windows at 2.31. Unix/Linux was the lowest at 2.20.

Windows	331 (26.04%)	766 (26.44%)	2.31
Unix/Linux	579 (45.55%)	1276 (44.04%)	2.20
Multiple OS	361 (28.40%)	855 (29.51%)	2.37
Total	1271 (100%)	2897 (100%)	

Table 2 - Distribution of Vulnerabilities

The data was further reduced to three fields, type of software, No of Bugs and Risk, in order to run an Analysis of Variance. First, ANOVA was run with Type of Source as the independent grouping variable with number of bugs as the dependent continuous variable. Secondly, Another ANOVA procedure was run with number of Bugs as the independent variable and Risk levels as the dependent variable. Finally, the average risk per vulnerability was used in the third ANOVA procedure. The results are tabulated below.

Procedure	**Dependent Variable**	**F-Ratio(sig. level)**
ANOVA 1	Number of Vulnerabilities	6.93(p-value<.001)
ANOVA 2	Cumulative Risk	5.51(p-value<.001)
ANOVA 3	Risk per Vulnerability	0.29

Table 3 - ANOVA

A Tukey-Kramer multiple comparison procedure was ascertain the differences between the three groups in terms of the number of vulnerabilities and the cumulative risk. Results indicate that open-source OS have more vulnerabilities and risk associated with them than proprietary, and multiple OS systems. There were no differences between proprietary and multiple OS systems.

Conclusions and Limitations

From our analysis, there appears to be a difference between the number of vulnerabilities reported for Windows and Unix operating systems (0.01 level.) However these differences do not exist between Open source and multiple OS systems. Similarly, there is a difference between Windows OS and Unix OS in terms of cumulative risk, but not between Windows and Multiple OS. Any conclusions about the relative quality of open-source versus proprietary software should however be tempered by caution. The differences may just be an artifact of source code availability, and also the fact that there are more individuals examining open-source software, than those examining proprietary software.

The average risk per vulnerability seems to be the same between all types of operating systems. The ANOVA procedure detected no differences between groups. Since the average risk per vulnerability is considered, this measure may lead to more robust conclusions about the relative quality of software. Our analysis leads us to conclude that there are no inherent qualitative differences between Windows and other operating systems. That is, the type of vulnerabilities in open and proprietary systems, are similar to each other in terms of the average risk associated with them.

APPENDIX

Vulnerabilities Data Sample

Windows Operating Systems Only				
Vendor & Software Name	**Vulnerability - Impact Patches - Workarounds Attacks Scripts**	**Common Name**	**Risk**	**Source**
Abyss Abyss Web Server X1	An input validation vulnerability exists, which could allow a remote malicious user to crash the target service. It is reported that a remote user can submit an HTTP request for a URL containing a MS-DOS device name (e.g., CON, PRN, AUX) in the 'cgi-bin' directory to cause the web service to crash. No workaround or patch available at time of publishing. A Proof of Concept exploit has been published.	Abyss Web Server MS-DOS Device Names Processing	Low	SecurityTracker Alert ID, 1011812, October 20, 2004

UNIX / Linux Operating Systems Only				
Vendor & Software Name	**Vulnerability - Impact Patches - Workarounds Attacks Scripts**	**Common Name**	**Risk**	**Source**
Apple Safari 1.2.3	A cross-domain vulnerability exists when multiple windows are open, which could let a remote malicious user spoof web page functions. No workaround or patch available at time of publishing. There is no exploit code required; Proof of Concept exploit has been published.	Apple Safari Cross-Domain Dialog Box Spoofing	Medium	Secunia Advisory, SA12892, October 20, 2004
Multiple Operating Systems - Windows / UNIX / Linux / Other				
Vendor & Software Name	**Vulnerability - Impact Patches - Workarounds Attacks Scripts**	**Common Name**	**Risk**	**Source**
America OnLine America Online Webmail	A Cross-Site Scripting vulnerability exists in the 'msglist.adp' script due to insufficient input validation, which could let a remote malicious user execute arbitrary HTML and script code. No workaround or patch available at time of publishing. A Proof of Concept exploit has been published.	AOL Web Mail 'msglist.adp' Cross-Site Scripting	High	SecurityTracker Alert ID, 1011791, October 20, 2004

REFERENCES

Balle, Jakob. The big picture on big holes. *Network Security*, May2004, Vol. 2004 Issue 5, p18, 3p

Chaudri, Abida; Patja, Ville. Windows v Lindows – have Microsoft won the battle only to lose the war? *Computer Law & Security Report*, Jul2004, Vol. 20 Issue 4, p321, 3p

Claassen, Pieter. The state of the hack. *Network Security*, Apr2004, Vol. 2004 Issue 4, p12, 2p

Goad, G. Pierre; Holland, Lorien. China joins Linux bandwagon. Far *Eastern Economic Review*, 02/24/2000, Vol. 163 Issue 8, p8, 4p, 2c

Hunter, Philip. Linux security: separating myth from reality. *Network Security*, Aug2004, Vol. 2004 Issue 8, p8, 2p

Post, G.; Kagan, A Computer security and operating system updates. *Information & Software Technology*, Jun2003, Vol. 45 Issue 8, p461, 7p

Schultz, E. Eugene. Why can't Microsoft stay out of the InfoSec headlines? *Computers & Security*, May2003, Vol. 22 Issue 4, p270, 3p

Schultz, E. Eugene. Windows 2000 security: A postmortem analysis. *Network Security*, Jan2004, Vol. 2004 Issue 1, p6, 4p

Schultz, Eugene. Microsoft security issues again in the news. *Computers & Security*, 2002, Vol. 21 Issue 8, p679, 2p

Wang, Huaiqing; Chen Wang. Taxonomy of Security Considerations and Software Quality. *Communications of the ACM*, Jun2003, Vol. 46 Issue 6, p75, 4p