MARKET OVERVIEW

March 19, 2004

Is Linux More Secure Than Windows?

by Laura Koetzle

FORRESTER® Helping Business Thrive On Technology Change

MARKET OVERVIEW

March 19, 2004

Is Linux More Secure Than Windows?

by Laura Koetzle
with Charles Rutstein, Natalie Lambert, and Stephan Wenninger

EXECUTIVE SUMMARY

Microsoft gets a bad rap for security, while many believe that Linux is relatively secure. A fair assessment? Not really: After collecting a year's worth of vulnerability data, Forrester's analysis shows that both Windows and four key Linux distributions can be deployed securely. Key metrics include responsiveness to vulnerabilities, severity of vulnerabilities, and thoroughness in fixing flaws.

TABLE OF CONTENTS

NOTES & RESOURCES

Forrester interviewed 13 vendor and user companies, including: @stake, Debian, Hewlett-Packard, iDEFENSE, KDE, MandrakeSoft, Microsoft, Red Hat, SUSE Linux, and TruSecure.

Related Research Documents

"ASN.1 Flaws Aren't New Or Unique To Microsoft"
February 23, 2004, Quick Take

"Criteria for Selection: Operating System Security"
January 28, 2004, Planning Assumption

"Your Open Source Strategy"
September 23, 2003, Report

"How Much Security Is Enough?"
August 6, 2003, Report

"Can Microsoft Be Secure?"
March 20, 2003, Report

UNDERSTANDING THE VULNERABILITY LIFE CYCLE

When asked about the security of popular operating systems like Linux and Windows, many IT professionals have a reflexive reaction: Linux is relatively secure, Windows isn't. But are they right? To answer that question, Forrester collected and verified a vast amount of data on past security vulnerabilities — and the answers are a bit surprising. First, however, you must understand the timeline of a security vulnerability:

- **Real vulnerability to attack begins with disclosure.** Virtually every complex piece of code probably has some vulnerability in it. But users are unlikely to see attacks against their platforms until someone uncovers and discloses a vulnerability in a public forum like the bugtraq security mailing list (see Figure 1-1).

- **Software firms rush to issue patches for the vulnerable component.** Neither commercial independent software vendors (ISVs) nor open source component maintainers have the resources to address all security vulnerabilities instantly. They struggle to verify and prioritize all the flaws that surface and to build, test, and release stable fixes as quickly as they can.

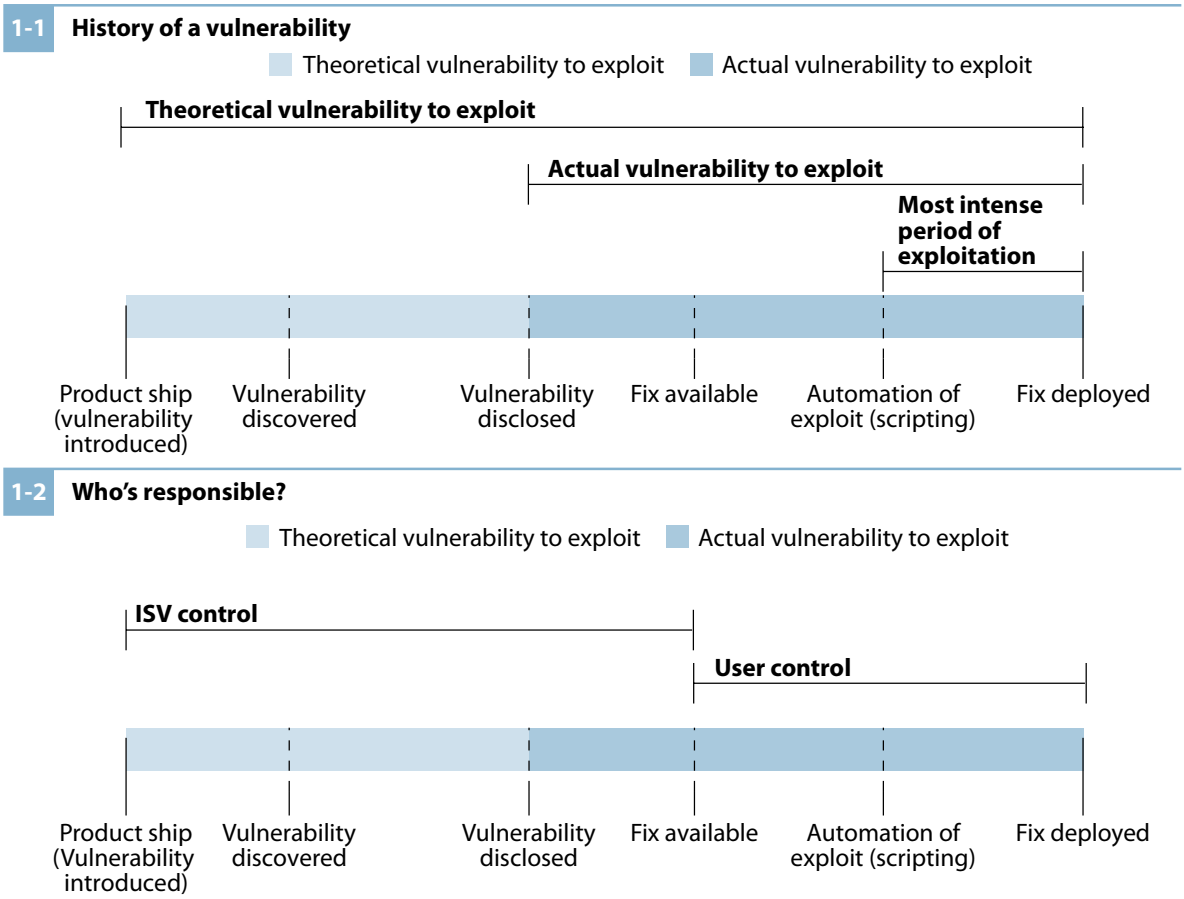
- **Exploit numbers explode only after unscrupulous hackers build scripted versions.** Although nearly all exploits start with public disclosure, the number of attacks doesn't really take off until a talented hacker provides an automated tool that relatively unskilled vandals can use.[1]

- **Vulnerability ends with the proper application of the patch.** Once the ISV or open source component maintainer releases the fix, it's up to the customer to apply it (see Figure 1-2). But doing so isn't a simple task: Because few firms stick to consistent platform configurations and most lack robust testing and deployment procedures, patch application can take months — or longer. For example, for the nine highest-profile Windows malicious code incidents as of March 2003, Microsoft's patches predated major outbreaks by an average of 305 days, yet most firms hadn't applied the patches.[2]

WHAT MATTERS: RESPONSIVENESS, RELATIVE SEVERITY, AND THOROUGHNESS

Of course, it's not enough for ISVs to simply release patches quickly — or for users to apply them quickly. Forrester believes there are three important criteria for assessing the relative vulnerability of platforms over time:

Figure 1 Vulnerabilities: History And Responsibility



Source: Forrester Research, Inc.

1. **Responsiveness: How quickly do you fix public security vulnerabilities?** All platforms contain security vulnerabilities — what matters is how quickly maintainers fix them once discoverers find and report them.

2. **Relative severity: How bad are your platforms' problems, relative to others'?** All vulnerabilities are not created equal — flaws that allow local users to modify the high scores for games, for example, are much less important than flaws that allow remote attackers to take complete control of the machine.

3. **Thoroughness: How close do you get to fixing 100% of public security flaws?** There's no credit for fixing 20% of vulnerabilities lightning-fast and ignoring the rest.

The Four Best Metrics For Quantifying Platform Security

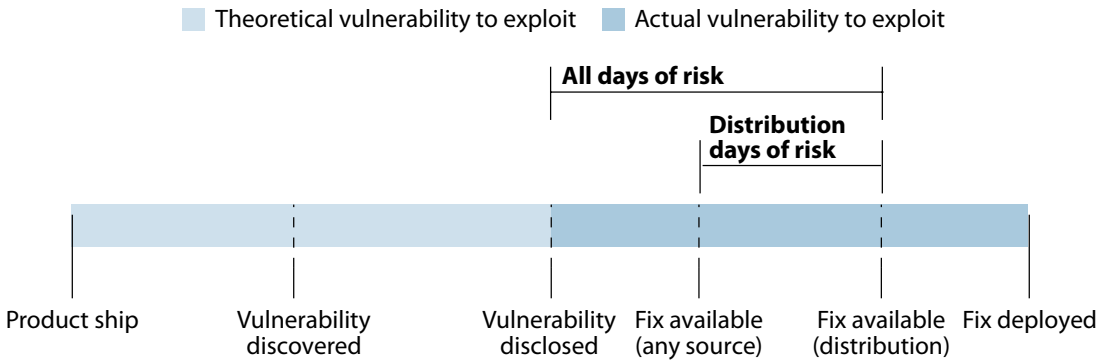
To get quantitative answers to these three questions, Forrester created two metrics for responsiveness, one for relative severity, and one for thoroughness:

- **"All days of risk" quantifies the platform's actual vulnerability to attack**. "All days of risk" measures the number of days between a security vulnerability's first public disclosure and the platform maintainer's first fix for the problem. We calculated "all days of risk" values for the platforms maintained by Microsoft and by Linux distributors Debian, MandrakeSoft, Red Hat, and SUSE Linux.[3]

- **"Distribution days of risk" compares the Linux distributors' responsiveness.** In the Linux world, distributors bundle together code from many sources — meaning there may be a lag between a patch being issued for a specific component and that patch being included in a new distribution. "Distribution days of risk" quantifies the elapsed time between the first fix for the security hole by the maintainer of the flawed component and the first fix for the flawed component issued by the platform maintainer (see Figure 2). We calculated separate values for "distribution days of risk" for Debian, MandrakeSoft, Red Hat, and SUSE, and used the "all days of risk" value for Microsoft.

- **"Flaws fixed" measures the platform maintainers' thoroughness.** "Flaws fixed" calculates the percentage of applicable public security issues that the platform maintainer addressed. We calculated "flaws fixed" for all five platform maintainers.

Figure 2 Days Of Risk



Source: Forrester Research, Inc.

- **Percentage of high-severity vulnerabilities.** To measure relative severity, we used the criteria applied by the US government's National Institutes for Standards and Technology's (NIST) ICAT project.[4] ICAT defines a vulnerability as high severity if an exploit: 1) allows a remote attacker to violate the security of a system (i.e., gain an account); 2) allows a local attacker to gain complete control of a system; or 3) the Computer Emergency Response Team Coordination Center (CERT/CC) issues an advisory.[5] We calculated the percentage of total applicable vulnerabilities that ICAT classified as high severity for all five platform maintainers — Debian, MandrakeSoft, Microsoft, Red Hat, and SUSE.

MICROSOFT, DEBIAN FIX FAST; RED HAT, MANDRAKESOFT MISS FEW FLAWS

To evaluate Debian, MandrakeSoft, Microsoft, Red Hat, and SUSE's platforms based on those four metrics, we collected security vulnerability data for the period between June 1, 2002 and May 31, 2003.[6] We used only public data sources, enabling others to verify our data — all of which is posted at Forrester.com. We considered the metrics in the following order: all days of risk, distribution days of risk, percentage of high-severity flaws, and percentage of flaws fixed. Here's what we found (see Figure 3):

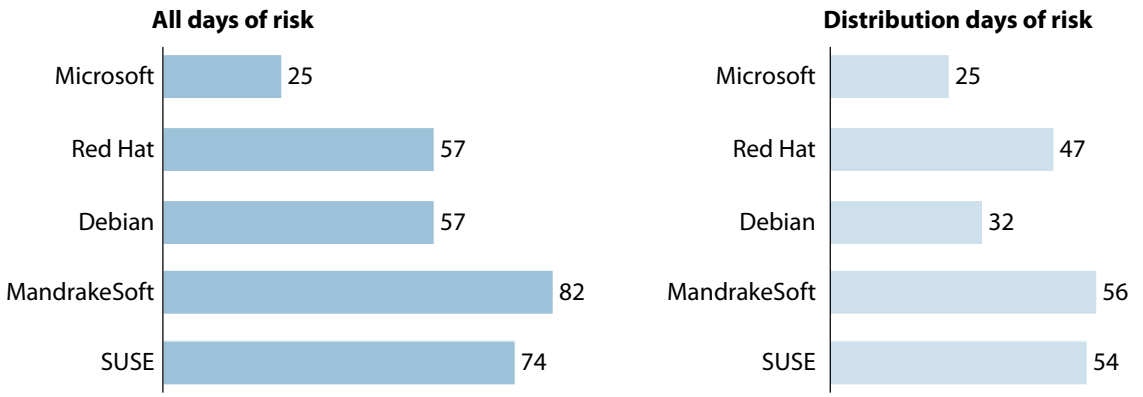
- **Microsoft demonstrated the lowest average "all days of risk."** Microsoft's average of 25 days between disclosure and release of a fix was the lowest of all the platform maintainers we evaluated. Microsoft also addressed all of the 128 publicly disclosed security flaws in Windows during our 12-month evaluation period. However, Microsoft needs to work on its percentage of high-severity vulnerabilities: ICAT classified 67% of Microsoft's vulnerabilities as high severity, placing Microsoft dead last among the platform maintainers by this metric.

- **Red Hat ties Microsoft with stellar relative severity and thoroughness records.** Best-known Linux distributor Red Hat tied with Debian with 57 days for "all days of risk," but lagged significantly behind Debian in average "distribution days of risk" with 47. However, Red Hat ties with Microsoft for first place overall, because: 1) Only 56% of Red Hat's Linux distribution's public vulnerabilities qualified as high severity, and 2) Red Hat fixed 99.6% — all but one — of the 229 applicable Linux vulnerabilities from the evaluation period.

Figure 3 Quantifying Platform Security

A spreadsheet with additional data is available online.

3-1 Days of risk



3-2 Percentage of high-severity flaws and of flaws fixed

Platform	Number of total flaws	Number of high-severity flaws	% of flaws with high severity	Number of flaws fixed	% of flaws fixed
Microsoft	128	86	67%	128	100.0%
Red Hat	229	128	56%	228	99.6%
Debian	286	162	57%	275	96.2%
MandrakeSoft	199	120	60%	197	99.0%
SUSE	176	111	63%	172	97.7%

Source: Forrester Research, Inc.

- **Debian's developer federation achieved the lowest "distribution days of risk."** With an average of 57 days for "all days of risk" and 32 days for "distribution days of risk," Debian excelled according to responsiveness metrics.[7] With only 57% of its 286 applicable vulnerabilities qualifying as high severity, Debian placed second among platform maintainers using relative severity metrics. However, the Debian security team needs to improve its thoroughness record — because Debian addressed only 96.2% of the vulnerabilities that surfaced during the evaluation period, Debian just slips into third place behind Red Hat.

- **MandrakeSoft ties with SUSE despite poor "days of risk" showing.** ICAT considered only 60% of the flaws found in MandrakeSoft's Linux distribution to be high-severity, and MandrakeSoft fixed all but two of its 199 applicable vulnerabilities (99%). Thus, MandrakeSoft's third-place finishes according to relative severity and thoroughness

metrics offset its weaker days of risk performance. With 82 days, MandrakeSoft had the worst "all days of risk" record of any of the platform maintainers, and the worst "distribution days of risk" record, with 56 days.

- **SUSE did well by "days of risk" measures, but trailed on other metrics**. SUSE bested MandrakeSoft on average "all days of risk" and "distribution days of risk," with 74 and 54 days respectively. However, because ICAT considered 63% of SUSE's 176 applicable vulnerabilities severe, and because SUSE only fixed 97.7% of those vulnerabilities, the firm finishes in a tie with MandrakeSoft.[8]

Platform Security's Future: Responsible Disclosure And Reduced Attack Surfaces

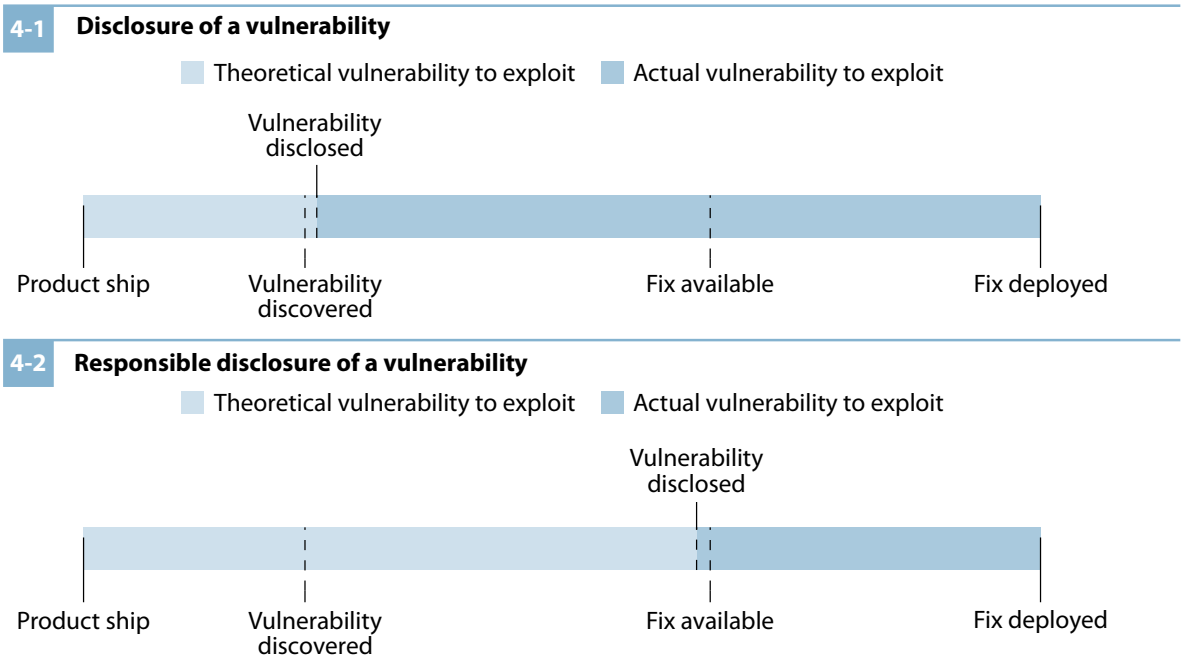
No platform maintainer will ever be able to eliminate all the vulnerabilities from its platform, be it commercial or open source. Does that mean that platform security will never get any better than it is today? Hardly.

- **Rewarding responsible disclosure creates a virtuous security circle . . .** A discoverer of a security bug has two options: 1) She can reveal her findings immediately to the public, or 2) she can inform the component's maintainer of the vulnerability privately and give him a chance to fix it first.[9] The second practice is called responsible disclosure.[10] If maintainers and discoverers work together to adhere to responsible disclosure procedures, they can announce the problem and its solution simultaneously. Coordinated releases mean everybody wins — they drastically reduce the period during which attackers are most likely to exploit customers' systems (see Figure 4).[11]

- **. . . which makes scheduled security update processes possible.** Security flaw discoveries are inherently unpredictable — but update processes don't have to be. As of late 2003, Microsoft had instituted a monthly (instead of weekly) release cycle for its security bulletins. Microsoft will make exceptions to the monthly release cycle only for emergency security updates. Similarly, SUSE groups security fixes together into omnibus advisories.[12] These practices trade possible increases in "all days of risk" for a schedule around which users can plan their patch testing and deployment processes. However, scheduled security releases will only ease users' patching workloads if discoverers continue to abide by responsible disclosure principles — otherwise, users will still get stuck with large numbers of emergency bulletins.

- **Future platforms will limit the damage caused by broad classes of security flaws.** Buffer overflows, stack overruns, and other exploits that rely on overwriting data with code and then executing that code are among the most common security holes.[13] To catch buffer overflows, platform maintainers must: 1) train developers to check input before assigning it to potentially vulnerable data structures, and 2) use compilation-time checks like those provided with Microsoft's Visual Studio .NET.[14] Fixing individual buffer overflows is good, but helping protect users from the whole class of problems — because a few will always sneak past code audits — would be far better. Innovations like exec-shield, which the Fedora Linux project includes in its latest kernel version, aim to do just that.[15] Next year, users should look for kernel upgrades that incorporate features like exec-shield.

Figure 4 Comparing The Risk: Full Disclosure Versus Responsible Disclosure



Source: Forrester Research, Inc.

RECOMMENDATIONS

HANDLING COMPETING PLATFORM REQUIREMENTS

Here's how customers should balance security with other pressing concerns:

- **If you want security updates as quickly as possible, think Debian or Microsoft.** Unix-leaning firms that want maximum security control and aren't afraid of command-line interfaces and community support should plump for Debian because of its low "distribution days of risk." Firms who find Windows a better fit for their environments shouldn't fly off the handle about every Microsoft security incident. Instead, those firms should: 1) focus on requiring every new deployment of Windows to conform to one of a few security-validated configurations, and 2) monitor Microsoft's new monthly security release policy closely to make sure that it doesn't cause an unacceptable increase in overall "days of risk."
- **Balance security with installation ease: go MandrakeSoft, Microsoft, or SUSE.** MandrakeSoft, Microsoft, and SUSE all hang their hats on the ease with which relatively unskilled users and administrators can install, configure, and patch their platforms. Except in markets like Germany, user companies do not yet consider any of the Linux distributions to have cleared the bar for enterprise-desktop-level ease of use. However, MandrakeSoft and SUSE both have already achieved notable success among end users with their desktop versions of their Linux distributions.
- **To maximize security and operator ease, look at Microsoft or Red Hat.** If you: 1) lack the manpower to validate and test security patches yourself, and 2) can't afford to subscribe to a vulnerability management service from vendors like SecureInfo or TruSecure, then you should subscribe all your machines directly to your vendor's auto-update service. Microsoft and Red Hat successfully handle large percentages of their customers through auto-update services, which can automatically download and apply security updates. MandrakeSoft and SUSE also offer fully automatic update services.

SUPPLEMENTAL MATERIAL

Online Resource

The spreadsheet that contains all of the data that we used to calculate the "all days of risk," "distribution days of risk," percentage of high-severity vulnerabilities, and percentage of flaws fixed metrics in Figure 3 is available in the online version of this document.

Methodology

Evaluation Period

We decided on an evaluation period from June 1, 2002 to May 31, 2003, inclusive. We included all of the vulnerabilities where a platform maintainer issued a fix during this period, as well as all vulnerabilities for which any component maintainer issued a security advisory during the period.

We deliberately chose to end our evaluation period more than six months prior to the publication of this document. By including only data on vulnerabilities that are already widely publicized and well-understood, we hoped to avoid providing any ammunition to malicious exploiters of vulnerabilities.

Platform Criteria

We used real-world data center conditions to determine what we included in the definition of platform for Debian, MandrakeSoft, Microsoft, Red Hat, and SUSE. Firms don't run bare operating systems. Thus, we included vulnerabilities for standard applications like mySQL, Microsoft SQL Server, the Apache Web server, and Microsoft IIS in our data set.

We also chose not to limit our data set to particular versions of the platforms in question. Thus, we included vulnerabilities for all of the versions of Microsoft's Windows and associated major applications that Microsoft supported during the evaluation period. The Linux distributors support multiple hardware architectures (Hewlett-Packard PA-RISC, Intel x86, Sun SPARC, etc), and maintain multiple versions of their platforms designed for different audiences (i.e., for home users, power developers, and enterprise servers). In order to make this data set useful to the greatest number of people, we decided to include vulnerabilities for all of the Linux distributors' supported software components and platforms, regardless of hardware architecture or target market segment.

To use the data set to evaluate your specific configuration of Windows or of one of the Linux distributions, use the "Vulnerable Component" and "Description of Vulnerability" columns included in the spreadsheet to determine which vulnerabilities apply.

Platform Maintainer Involvement

We gave Debian, MandrakeSoft, Microsoft, Red Hat, and SUSE ample opportunities to verify all of their data in order to help resolve any errors or conflicts prior to publication.

Data Collection Methods

To form our initial vulnerability data set, we catalogued all of the security advisory bulletins issued by Debian, MandrakeSoft, Microsoft, Red Hat, and SUSE during the evaluation period. We cross-referenced the security advisories with the entries in the Common Vulnerabilities and Exposures (CVE) dictionary, as well as with the secondary sources listed in the next section.

Additional Data Sources

We collected the security vulnerability data exclusively from public sources. Those public sources include: the various Full Disclosure mailing lists (i.e., Bugtraq, NTBugtraq, Vulnwatch, Vulndiscuss, etc.), the various SecurityFocus mailing lists (i.e., FOCUS-MS, FOCUS-Linux, Incidents, Vuln-dev, etc.), the Neohapsis mailing list archives, the MARC mailing list archives at marc.theaimsgroup.com, LWN.net, bugzilla.org, freshmeat.net, rpmfind.net, the CVE dictionary and CVE reference maps (available at http://cve.mitre.org), the ICAT project at NIST, (available at http://icat.nist.gov), CERT/CC at Carnegie Mellon University, the security bulletins released by Debian, MandrakeSoft, Microsoft, Red Hat, and SUSE, the security and bugs mailing lists run by Debian, MandrakeSoft, Red Hat, and SUSE, component maintainers' security bulletins (i.e., kde.org for the Linux KDE GUI components, the Internet Systems Consortium (ISC) for BIND, tcpdump.org for tcpdump, the Massachusetts Institute of Technology for Kerberos, etc), security research organizations' security bulletins (i.e., @stake, iDEFENSE, Internet Security Systems (ISS) X-Force, etc.), and the change logs for the software packages the platform maintainers released.

Specific Notes

Debian, MandrakeSoft, and Red Hat all generally follow a policy of "one security bulletin, one security vulnerability" — except in cases where they're addressing a whole group of closely related vulnerabilities at the same time.

Microsoft used to group related security fixes together weekly and has started producing omnibus security updates monthly (with exceptions for emergencies).

SUSE often groups fixes for multiple security vulnerabilities into single security announcements. SUSE also includes previews of forthcoming security fixes in section two of its security advisory bulletins, and then sometimes declines to issue subsequent bulletins confirming that the fix has been made. Thus, in determining the dates for SUSE fixes, we followed the following order of precedence: security advisory date, distribution security update Web page date, and the change log for the software update package — SUSE includes the change log with the source code files bundled in the Red Hat Package Manager (RPM) format. We treated the few security flaws whose SUSE fix dates we couldn't determine using these resources as exceptions and noted them as such.

All the platform maintainers occasionally issue new versions of security bulletins with corrections or changes. The Linux distributors do this most often with kernel fixes. In the case where the platform maintainer issued more than one security advisory for a given vulnerability, we've used the date of the first security bulletin for the platform maintainer's fix date.

When Red Hat updates its existing security advisories, it issues a new version of the security advisory with a new ordinal number appended. For example, if the original advisory was labeled RHSA-2002:001, the updated advisory will be labeled RHSA-2002:001-05 or similar. In referring to Red Hat security advisories in general, we've omitted the ordinal number that follows the second dash (in the example above, this would be the "05") because Red Hat may issue subsequent, higher-numbered updates. Thus, we've used the date for the first public version (typically the one with the lowest ordinal number after the second dash) of the advisory.

To be as fair as possible in the determination of "all days of risk" and "distribution days of risk" values, we elected not to average in the data on the vulnerabilities that, to the best of our ability to determine, remain unfixed by the platform maintainers.

Vulnerability bulletins and fixes appeared simultaneously for 313 of the 480 total security vulnerabilities that we considered as part of our evaluation period. In all likelihood, discoverers made use of responsible disclosure procedures for these vulnerabilities.[16] This means that discoverers notified component maintainers privately and worked with the maintainers to produce fixes. Using the date of a discoverer's first private communication with a component maintainer about a vulnerability would have allowed us to calculate "mean time to fix" values for each of the five platform vendors. However, because responsible disclosure communications are private, their dates cannot be independently verified. Thus, we decided not to include this information in this study.

Researchers attempting to replicate this data set should note that we accepted variances in events like First Public Date and First Fix Date of up to three days because of the variety of time zones across which emails post to mailing lists and across which security advisories appear on the Web.

Acknowledgements

Forrester thanks Noah Meyerhans of Debian, Vincent Danen of MandrakeSoft, Allen Jones of Microsoft, Mark Cox of Red Hat, and Roman Drahtmüller of SUSE for the time that they so generously dedicated to this research.

Companies And Organizations Interviewed For This Document

@stake

Debian

Hewlett-Packard

iDEFENSE

KDE

MandrakeSoft

Microsoft

Red Hat

SUSE Linux

TruSecure

ENDNOTES

[1] We say "nearly all" exploits start with disclosure because we can't discount the possibility of a malicious discoverer who launches a zero-day attack without any public disclosure of his findings. Additionally, analyses of incident data from the Computer Emergency Response Team Coordination Center (CERT/CC) at Carnegie Mellon University indicate that the majority of exploits against security vulnerabilities correlate not with discovery, disclosure, or fix availability, but with the availability of scripts that automate the hacks. See William A. Arbaugh, William L. Fithen, and John McHugh. "Windows of Vulnerability: A Case Study Analysis." *Computer*, IEEE, December 2000, 52-59.

[2] As platform maintainer, Microsoft also has responsibilities to uphold. Microsoft must provide a single patch management utility and build security analysis wizards to make it easier for customers to deploy and operate Windows securely. See the March 20, 2003, Report "Can Microsoft Be Secure?"

[3] Novell announced its intent to acquire SUSE Linux in November 2003, and it completed the acquisition in January 2004.

[4] ICAT no longer stands for anything. According to NIST, the acronym became obsolete, but the project continues. See http://icat.nist.gov.

[5] ICAT defines a vulnerability as low severity if exploiting the vulnerability does not typically yield valuable information about or control over a system. A low-severity vulnerability may help an attacker find and exploit other vulnerabilities. Low-severity vulnerabilities are of minor importance for most organizations. Medium-severity vulnerabilities are all those that fall into neither the low-nor the high-severity categories. See http://icat.nist.gov. Additionally, the severity of vulnerabilities applicable to packages shipped by multiple vendors — such as the four Linux distributors discussed here — may vary from distribution to distribution. We use the ICAT severity definitions to apply a consistent standard across platforms in order to draw aggregate comparisons, but we recognize that the distributor's choices when building and configuring the package can and will impact the severity of these vulnerabilities.

[6] We collected the data on disclosures, first fixes, and platform maintainers' fixes using only public sources. See the Methodology section of this document for further details.

[7] Debian's responsiveness achievement is particularly noteworthy when we consider that Debian's default distribution includes a larger number of packages than the other Linux distributors' default distributions. For the evaluation period of June 1, 2002 to May 31, 2003, Debian's default distribution included approximately 960 packages, and its full distribution included more than 8,990 packages. Red Hat's default distribution for its Personal Desktop 8.0 included more than 500 packages, plus the 365 packages of PowerTools, for a total of approximately 865 packages, and Red Hat's full distribution included more than 1,460 packages. MandrakeSoft's default distribution for MandrakeSoft Linux 9.0 included more than 450 packages, with the total distribution comprising more than 2,220 packages. SUSE's 8.1 Pro distribution included about 395 default packages, and more than 2,840 total packages.

[8] Discoverers found 175 security vulnerabilities that applied to SUSE's distribution of Linux, the smallest number among any of the Linux distributors.

[9] We intend the term "component maintainer" to refer to any individual or entity responsible for the ongoing maintenance of the product in question. A maintainer can be a commercial vendor like Microsoft, a group of open source developers like KDE.org, or an individual software author.

[10] When using responsible disclosure procedures, the component maintainer must acknowledge a private communication from the discoverer about the security vulnerability within seven days (some responsible disclosure policies suggest five days). Post-acknowledgement, the discoverer must help the maintainer to reproduce the problem and must honor the maintainer's requests for further information. For more detail on a draft responsible disclosure policy followed by many security issue discoverers (also called originators) and component maintainers, see The Organization for Internet Safety, "Security Vulnerability Reporting and Response Process," www.oisafety.org/process.html. See also Rainforest Puppy, "Full Disclosure Policy (RFPolicy) v2.0," www.wiretrip.net/rfp/policy.html.

[11] All of the vulnerabilities in the spreadsheet with zero values for the "Component Fix" column represent examples of responsible disclosure — meaning that the announcement of the vulnerability and the fix occurred on the same day. Thus, 313 of the 480 total vulnerabilities we found during the evaluation period were responsibly disclosed. In the open source world, software authors, component maintainers, and the Linux distributors participate in the vendor-sec mailing list. The vendor-sec participants coordinate the release of security bulletins that both announce and fix vulnerabilities that discoverers have disclosed responsibly. The staff of Microsoft's Security Response Center handles the vendor's responses to all vulnerabilities.

[12] See the Methodology section of this document for further details on how SUSE structures its security bulletins.

[13] Two well-known examples of buffer-overflow exploits are the Microsoft SQL Slammer worm in January 2003 and the Linux Slapper worm in September 2002.

[14] See http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vccore/html/vclrfGSBufferSecurity.asp for details on the Visual Studio .NET/GS option, which can only detect direct buffer overflows that overwrite the return address.

[15] See http://people.redhat.com/mingo/exec-shield/ANNOUNCE-exec-shield for details on exec-shield.

[16] It's also possible that the component maintainer found the fix for the issue so trivial that he was able to make it the day of the vulnerability announcement despite not having any advance warning.

FORRESTER

Helping Business Thrive On Technology Change

Headquarters

Forrester Research, Inc.
400 Technology Square
Cambridge, MA 02139 USA
Tel: +1 617/613-6000
Fax: +1 617/613-5000
Email: forrester@forrester.com
Nasdaq symbol: FORR
www.forrester.com

Research and Sales Offices

Australia
Austria
Brazil
Canada
France
Germany
Hong Kong
India
Israel
Japan
Korea
The Netherlands
Poland
United Kingdom
United States
Spain
Sweden

For a complete list of worldwide locations visit www.forrester.com/about.

FORRESTER®