



WINDOWS VISTA

ONE YEAR VULNERABILITY REPORT

By Jeffrey R. Jones
Security Guy
(and Microsoft Director)



Table of Contents

Executive Summary.....2

About the Author.....3

Overview4

 Interpreting the Analysis4

 The Security Researcher Ecosystem5

Windows Vista vs Windows XP6

 Windows Vista – Year One..... 6

 Windows XP – Year One..... 8

 Side-by-Side Comparison 10

Windows Vista vs Other Operating Systems ..12

 Red Hat Enterprise Linux 12

 Ubuntu 6.06 LTS 14

 Apple Mac OS X v10.4 16

 Side-by-Side Comparison 17

Final Observations19

APPENDIX A: FREQUENTLY ASKED QUESTIONS20

Appendix B: Sources and Methodology23

 Discovering Unfixed Vulnerabilities 23

EXECUTIVE SUMMARY

Windows Vista shipped to business customers on the last day of November 2006, so the end of November 2007 marks the one year anniversary for supported production use of the product.

This paper analyzes the vulnerability disclosures and security updates for the first year of Windows Vista and looks at it in the context of its predecessor, Windows XP, along with other modern workstation operating systems Red Hat, Ubuntu and Apple products.

The results of the analysis show that Windows Vista has an improved security vulnerability profile over its predecessor. Analysis of security updates also shows that Microsoft improvements to the security update process and development process have reduced the impact of security updates to Windows administrators significantly compared to its predecessor, Windows XP.

Note that this report is an update to the previously published [Windows Vista 90-Day Vulnerability Report](#) and [Windows Vista 6-Month Vulnerability Report](#). However, since one year is a more informative time frame, this report contains the results of a deeper level of analysis.



ABOUT THE AUTHOR

Jeff Jones is a Security Strategy Director in Microsoft's Trustworthy Computing group. In this role, Jeff draws upon his 20 years of security experience to work with enterprise CSOs and Microsoft's internal security teams to drive practical and measurable security improvements into Microsoft process and products. Prior to his strategic position at Microsoft, Jeff was the vice president of product management for security products at Network Associates where his responsibilities included PGP, Gauntlet and Cybercop products, and several improvements in the McAfee product line. These latest positions cap a 20 year hands on career in security, performing risk assessments, building custom firewalls and being involved in DARPA security research projects while part of Trusted Information Systems. Jeff is a frequent global speaker and writer on security topics ranging from the very technical to more high level, CxO-focused topics such as Security TCO and metrics.

Jeff actively encourages readers to challenge his assumptions, analysis and conclusions and provide critical feedback – but asks for equal (or better) rigor in methodology and analysis to support the challenges, as opposed to enthusiastic espousal of unsupported evangelistic fervor.



OVERVIEW

This is the 3rd vulnerability analysis and report I've done for Windows Vista since it shipped, the first two completed after 90 days ([go day report](#)) and six months ([six month report](#)) had passed.

In this report, I've updated the structure a bit to reflect the deeper analysis and the report will generally cover two broad sections:

- Windows Vista compared with its predecessor, Windows XP
- Windows Vista compared with other industry OS offerings

INTERPRETING THE ANALYSIS

I think it worth spending a moment to discuss what this analysis covers, why it might be useful to some people and, perhaps most importantly, what it does **not** say.

If it was possible to measure “security” in one metric, it would have to encompass a complex combination of factors including (but not limited to) the software quality, administrative controls, physical controls, and much more – and even then, it would all be in the context of whatever security policy was defined for the systems in question.

So, this is not an analysis of “the security”. I don't look at protective mechanisms and see how they might protect in certain scenarios. Nor do I look at security features and see how they might enable better privacy or help secure business process. And I certainly don't look at how easy it is to manage the security policy for these products.

Is there anything in this analysis which will prove one piece of software is “more secure” than another? No, that is not my intention.

This report is a vulnerability analysis, which may provide some elements that could be *part* of a broader security analysis. I fundamentally believe that security and non-security features need to be built upon a foundation of good engineering and solid security quality if they are to perform as we expect and not be misused to the detriment of security.

So, how are the metrics relevant then? Acknowledging that one factor can't measure the absolute “security”, we can still look at individual factors that contribute to improving security or making it easier to manage risk. Ask yourself:

- All other things being equal, is it easier to mediate risk on a system that has 10 vulnerabilities in a year or one that has 100 vulnerabilities in a year?
- Which has a more negative impact on your security team and risk management process – deploying 10 security updates per year or deploying 100 security updates per year?

Note that individual metrics can even be mutually exclusive. For example, vendor policy could mandate a single security update per year which would definitely decrease the number of patches to deploy. However, that same policy would almost certainly mean that the exposure time for publicized issues would increase.



My own context for vulnerability analysis is to measure against the goal of reducing customer risk. To me, all other things being equal, fewer vulnerabilities make it easier to manage risk. All other things being equal, fewer patches mean more time to spend on other security projects to reduce risk.

THE SECURITY RESEARCHER ECOSYSTEM

Before jumping into the analysis, let's look a bit at the ecosystem of security researchers and the science of finding security flaws in software. The recently published [Microsoft Security Intelligence Report](#) indicates that new vulnerability disclosures 1H2007 were over 3400, while vulnerabilities found for all of 2001 were only 1528.

This is one of many indications that the security researcher industry is maturing, growing and becoming more proficient at finding and disclosing software vulnerabilities. In recent years, tools have improved significantly, several professional code scanning tools have released as products and newer techniques such as Fuzz testing have been developed and expanded to further stress the boundaries of software security.

How much more scrutiny does a new operating system face today compared with the year 2001? I can't easily put a number on it, but in my opinion, it does seem like there are more researchers, better trained, and with better tools and techniques than ever before – creating an ecosystem better able to find and disclose security vulnerabilities.



WINDOWS VISTA VS WINDOWS XP

Windows Vista, the successor to Windows XP, released to business users on November 30, 2006. Since the release of Windows XP in 2001, the Microsoft approach to security has gone through some significant changes. In January 2002, only a few months after the release of Windows XP, Microsoft launched their Trustworthy Computing initiative and began to revise their entire product development process with the goal of long-term, ongoing, security improvement for customers.

How much impact has that commitment had for Windows Vista security in terms of reducing the number of vulnerabilities in the new Windows product? We should continue to monitor performance on an ongoing basis, but as of the end of November 30, 2007, the full release of Windows Vista has been in production use by business customers for a full year – a reasonable period for which I think we can look for indications of improvement.

WINDOWS VISTA – YEAR ONE

To get a complete view of the early vulnerability indicators, we will look at advisories and updates, vulnerability fixes and vulnerability disclosures in the first year for Windows Vista.

Let's start off by looking at security updates for Windows Vista during the first year of availability. Microsoft released 17 Security Bulletins and corresponding patches in the first year affecting components of Windows Vista, grouped so that there were 9 days in the year when Windows Vista security updates were released.

I refer to those “days when at least one update is released” as a Patch Event. Looking at the number and frequency of Patch Events doesn't reflect on the security quality of the code, but it is likely of interest to security team that gathers when an update is released to assess applicability and decide on the deployment strategy.

Additionally, I break out the vulnerabilities fixed for each security update. Microsoft fixed a total of 36 vulnerabilities, encompassing 9 Patch Events, in Windows Vista during the first year. Table 1 details Security Bulletins, vulnerability identifiers and vendor severity ratings for each Patch Event.

Date	Security Bulletin	Vulnerabilities	Component	Vendor Severity
2/13/2007	MS07-010	CVE-2006-5270	Anti-malware engine	Critical
4/3/2007	MS07-017	CVE-2007-1212 CVE-2007-0038 CVE-2007-1215	EMF, Animated cursor, GDI	Important Critical Important
4/10/2007	MS07-021	CVE-2006-6696 CVE-2007-1209 CVE-2006-6797	CSRSS	Critical Important Low



5/8/2007	MS07-027	CVE-2007-0942 CVE-2007-0945 CVE-2007-0946 CVE-2007-0947 CVE-2007-2221	IE	Important Critical Important Important Critical
6/12/2007	MS07-032 MS07-033 MS07-034	CVE-2007-2229 CVE-2007-2222 CVE-2007-1751 CVE-2007-1499 CVE-2007-2227 CVE-2007-2225 CVE-2007-1658 CVE-2006-2111	User info. Store, IE, Windows Mail,	Moderate Critical Critical Moderate Moderate Important Critical Important
7/10/2007	MS07-038	CVE-2007-3038	Windows Firewall	Moderate
8/14/2007	MS07-042 MS07-045 MS07-047 MS07-048 MS07-050	CVE-2007-2223 CVE-2007-3041 CVE-2007-2216 CVE-2007-3891 CVE-2007-3033 CVE-2007-3032 CVE-2007-1749	XML Core, IE, WMP, Gadgets, VML	Critical Important Important Moderate Important Moderate Critical
9/11/2007	MS07-053	CVE-2007-3036	Subsystem for UNIX	Important
10/9/2007	MS07-056 MS07-057 MS07-058	CVE-2007-3897 CVE-2007-3893 CVE-2007-3892 CVE-2007-1091 CVE-2007-2228	Windows Mail, IE, RPC	Important Critical Moderate Low Important

Table 1: Security Update Details for Windows Vista

To get a better feel for the frequency and impact that these security updates had for administrators throughout the year, I've also charted a histogram of Patch Events on a graph of the first fifty-two weeks of availability. There were nine, and no week had more than one security update.

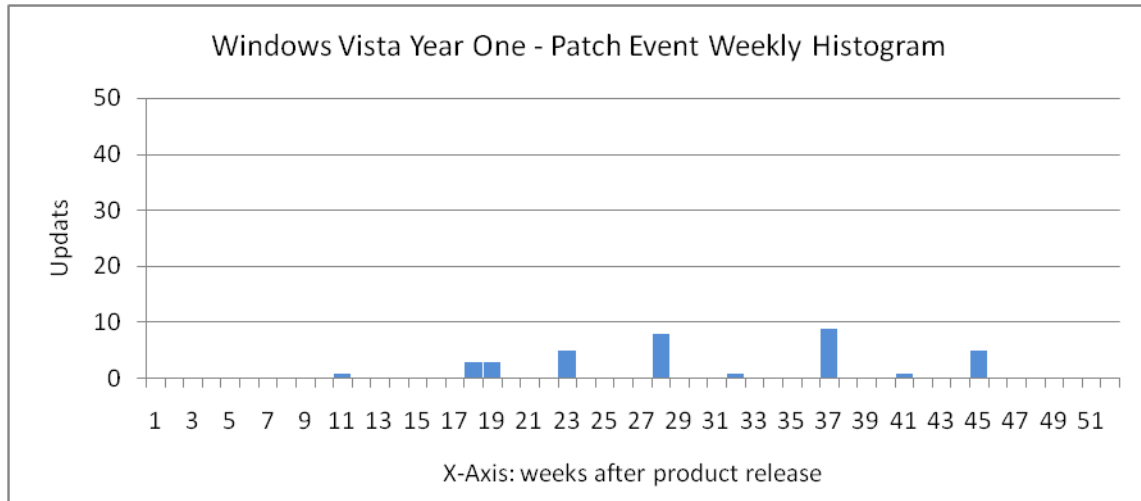


Figure 1: Windows Vista Year One - Patch Event Weekly Histogram

In addition to the vulnerability fixes outlined in the previous section, there were 30 vulnerability disclosures during Windows Vista’s first year that have not been addressed by a fix.

WINDOWS XP – YEAR ONE

Next, let’s take a similar look at the first year of Windows XP, which shipped on October 25, 2001, before Microsoft had initiated the Trustworthy Computing initiative.

Microsoft released 30 Security Bulletins and corresponding patches affecting Windows XP in the first year of availability. Also, because in 2001 Microsoft had not yet moved to a predictable monthly patch release policy, the 30 Security Bulletins were released on 26 different days throughout the year.

Extracting the vulnerabilities fixed for each security update, I find that Microsoft fixed a total of 65 vulnerabilities in Windows XP during the first year. See Table 2 below for details.

Date	Security Bulletin	Vulnerabilities	Component	Vendor Severity
10/21/2001	MS01-051	CVE-2001-0664 CVE-2001-0665 CVE-2001-0667	IE	Prior to Severity Ratings
11/1/2001	MS01-054	CVE-2001-0721	Plug-n-play	Low
11/8/2001	MS01-055	CVE-2001-0722 CVE-2001-0723 CVE-2001-0724	IE	Critical
11/20/2001	MS01-056	CVE-2001-0719	Windows Media Player	Critical
12/13/2001	MS01-058	CVE-2001-0727	IE	Critical



		CVE-2001-0874 CVE-2001-0875		
12/20/2001	MS01-059	CVE-2001-0876 CVE-2001-0877	Plug-n-play	Critical
2/11/2002	MS02-005	CVE-2002-0022 CVE-2002-0023 CVE-2002-0024 CVE-2002-0025 CVE-2002-0026 CVE-2002-0027	IE	Critical
2/12/2002	MS02-006	CVE-2002-0053	snmp	Moderate
2/21/2002	MS02-008 MS02-009	CVE-2002-0057 CVE-2002-0052	XML core, IE	Critical Critical
2/27/2002	MS02-012	CVE-2002-0055	SMTP	Low
3/4/2002	MS02-013	CVE-2002-0058 CVE-2002-0076	MS jvm	Critical
3/28/2002	MS02-015	CVE-2002-0077 CVE-2002-0078	IE	Critical
4/4/2002	MS02-017	CVE-2002-0151	UNC	Moderate
4/10/2002	MS02-018	CVE-2002-0072 CVE-2002-0073 CVE-2002-0074 CVE-2002-0075 CVE-2002-0147 CVE-2002-0148 CVE-2002-0149 CVE-2002-0150	IIS	Critical
5/15/2002	MS02-023	CVE-2002-0188 CVE-2002-0189 CVE-2002-0190 CVE-2002-0191 CVE-2002-0193 CVE-2002-1564	IE	Critical
6/11/2002	MS02-029	CVE-2002-0366	RAS	Critical
6/26/2002	MS02-032	CVE-2002-0372	Windows Media Player	Critical
7/30/2002	MS02-040	CVE-2002-0695	MDAC	Critical
8/21/2002	MS02-045 MS02-047	CVE-2002-0724 CVE-2002-0371 CVE-2002-0647 CVE-2002-0648 CVE-2002-0722 CVE-2002-0723	SMB, IE	Moderate Critical
8/28/2002	MS02-048	CVE-2002-0699	Cert enrollment	Critical
9/4/2002	MS02-050	CVE-2002-0862	Cert validation	Important
9/18/2002	MS02-051 MS02-052	CVE-2002-0863 CVE-2002-0864 CVE-2002-0865 CVE-2002-0866 CVE-2002-0867	RDP, MS jvm	Moderate Critical
9/24/2002	MS02-053	CVE-2002-0692	Frontpage Server	Critical



			Extensions	
10/3/2002	MS02-054 MS02-055	CVE-2002-0370 CVE-2002-1139 CVE-2002-0693 CVE-2002-0694	Zip decompression, Windows help	Moderate Critical
10/10/2002	MS02-058	CVE-2002-1179	OE, IE	Critical
10/16/2002	MS02-060	CVE-2002-0974	Help	Moderate

Table 2: Bulletins and Vulnerabilities During the 1st Year of Windows XP

Charted out as Patch Events, Figure 2 shows what the first year of Windows XP looked like for security administrators.

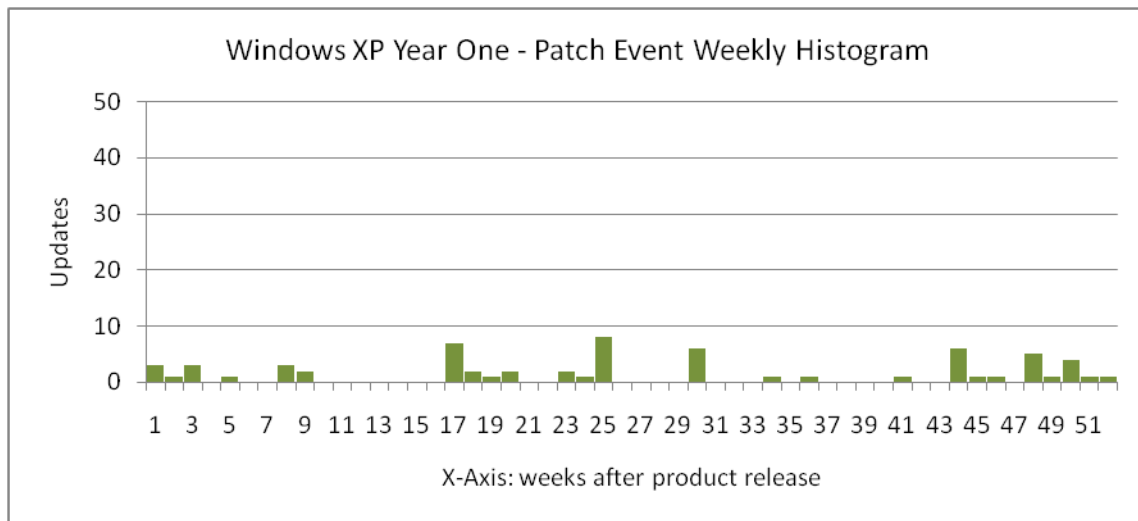


Figure 2: Windows XP Year One - Patch Event Weekly Histogram

In addition to the vulnerability fixes outlined in the previous section for Windows XP, there were 54 vulnerability disclosures during Windows XP's first year that had not been addressed by a fix.

SIDE-BY-SIDE COMPARISON

With the basic analysis completed for Windows Vista and Windows XP, we now have enough information to compare them. First let's look at a chart that shows fixed and unfixed vulnerabilities side-by-side.

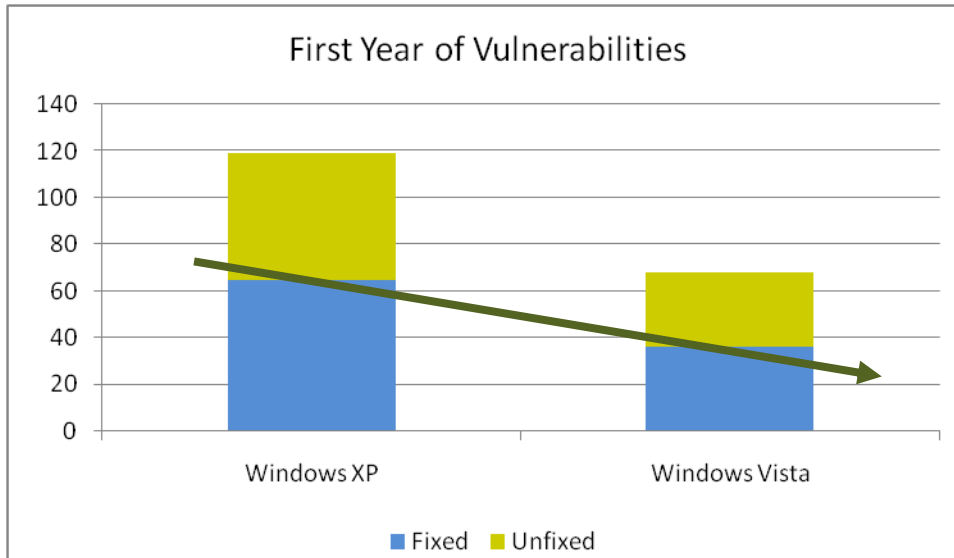


Figure 3: Side-by-side First Year Vulnerabilities for Windows Vista and Windows XP

The reduction in vulnerabilities from Windows XP to Windows Vista is clear to see in Figure 3, with security researchers having found significantly fewer vulnerabilities in Windows Vista.

Next, let's examine the impact that security updates had for administrators by looking at Patch Events in Figure 4. Windows Vista's 17 security updates occurred across 9 Patch Events in 9 different weeks in the year. Windows XP's 30 security updates occurred across 26 Patch Events in 25 different weeks in the year.

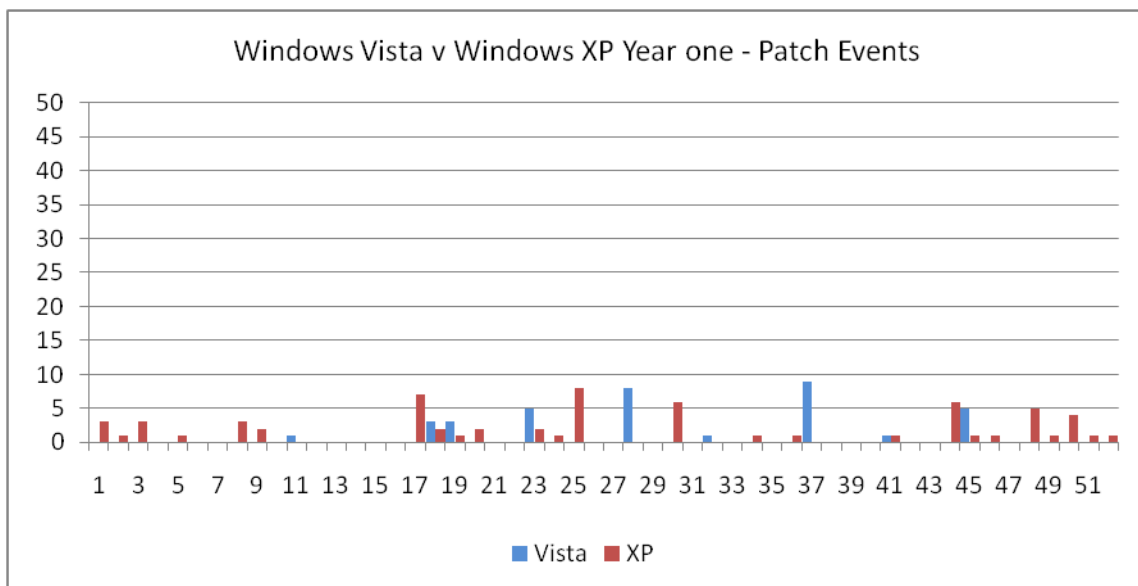


Figure 4: Side-by-Side Patch Event Histogram for Windows Vista and Windows XP

Graphed out visually, it is easy to see that the combination of a predictable monthly policy and fewer patches has had great impact in reducing the work necessary to manage security risk from 2001/2002 with Windows XP to 2007 for Windows Vista. It's a good illustration of progress that



Microsoft has made with the Trustworthy Computing initiative over time: implementing new processes that give customers a predictable schedule of updates, while at the same time reducing the number of serious vulnerabilities which cause customers the most work. Customers running Windows XP SP2 have also seen the benefits of the move to a monthly predictable update schedule, and those running Windows Vista should realize additional benefits over time. Here is a summary table of the key data I've discussed:

Metric	Windows Vista (year 1)	Windows XP (year 1)
Vulnerabilities fixed	36	65
Security Updates	17	30
Patch Events	9	26
Weeks with at least 1 Patch Event	9	25

Figure 5: Summary Table for Windows Vista and Windows XP

WINDOWS VISTA VS OTHER OPERATING SYSTEMS

While security improvement for Windows users is the key goal I am examining, it is also interesting to investigate how Windows Vista compares with other current operating systems. For this purpose, I've chosen to look at other workstation products that offer long-term support options – Red Hat, Ubuntu and Mac OS X 10. For each of these, I examined the most recent version of the operating systems that have been released to production for at least one year.

RED HAT ENTERPRISE LINUX

Red Hat is the most popular Enterprise Linux distribution, so their latest supported release that has been available for a full year, Red Hat Enterprise Linux 4¹ Workstation (rhel4ws), will be the first I examine².

- When rhel4ws shipped on February 15, 2005, there were 129 vulnerabilities already publicly disclosed in shipping components prior to general availability. On ship day, Red Hat issued 27 security advisories to address 64 of them.
- During the first year of availability, Red Hat issued 183 security advisories/updates for rhel4ws. If limited to just Critical and Important issues, there were 88 released on 57 different days.
- During the first year of availability, Red Hat fixed a total of 493 vulnerabilities in rhel4ws. If limited only to those vulnerabilities labeled Critical or Important by Red Hat, the number of vulnerabilities fixed is 214.
- At the end of the first year period, there were 82 vulnerabilities disclosed but without a patch (that would later be addressed with different fixes and security advisories). Adding

¹ Red Hat shipped Red Hat Enterprise Linux 5 in March 2007. It is not included since it does not yet have a one year track record.

² The primary source for this information is <http://rhn.redhat.com/errata>.



that to the fixed vulnerability count tells us that a total of 575 vulnerabilities were disclosed in RHEL4 components during the first year.

Of course, one school of thought is that is not “fair” to count the vulnerabilities for all of the components for the product that Red Hat ships and supports as Red Hat Enterprise Linux 4 WS (though I did count IIS Web Server vulnerabilities against Windows XP since it shipped with that component). To accommodate that idea, for comparison, I further analyzed a reduced set of rhel4ws components that deliver functionality comparable to Windows XP and excludes other optional components. I will only use the reduced set of components in comparisons.

RHEL4WS – REDUCED COMPONENT SET

Red Hat and other Linux distribution vendors add value to their workstation distributions by including and supporting many applications that don’t have a comparable component on a Microsoft Windows operating system. It is a common objection to any Windows and Linux comparison that counting the “optional” applications against the Linux distribution is unfair, so I’ve completed an extra level of analysis to exclude component vulnerabilities that do not have comparable functionality shipping with a Windows OS. In short, I install a rhel4ws computer and:

- I excluded any component that is not installed by default, which includes all optional “server” components that ship with rhel4ws.
- I additionally excluded *text-internet*, *graphics* (the gimp stuff) and *office* (OpenOffice) and *Development Tools* (gcc, etc) installation groups.
- I used the rpm command to list out all packages that get installed and used that package list to filter vulnerabilities for inclusion.

This process results in a Gnome-windows workstation that includes standard system management tools, Firefox for browsing, sound and video support, but excludes all server packages, as well as OpenOffice and other optional stuff that a Windows system wouldn’t have by default. This reduced rhel4ws build is then examined for comparison:

- During the first year, Red Hat issued 125 security advisories affecting the rhel4ws reduced set of components on 64 different days. If limited to only those security advisories containing issues rated Critical or Important by Red Hat, there were 62 advisories released on 41 different days during 30 different weeks.
- Red Hat fixed 360 vulnerabilities affecting the reduced (“workstation”) rhel4ws set of components. If limited just to those rated Critical or Important by Red Hat, the number was 154 vulnerabilities.
- At the end of the one year period, a total of 40 publicly disclosed vulnerabilities in the reduced set of components did not yet have a patch from Red Hat.

So, though the reduced component set of rhel4ws did have a better one year period than the full product, Red Hat customers did face a reasonably large number of vulnerabilities and security updates even with a reduced installation.

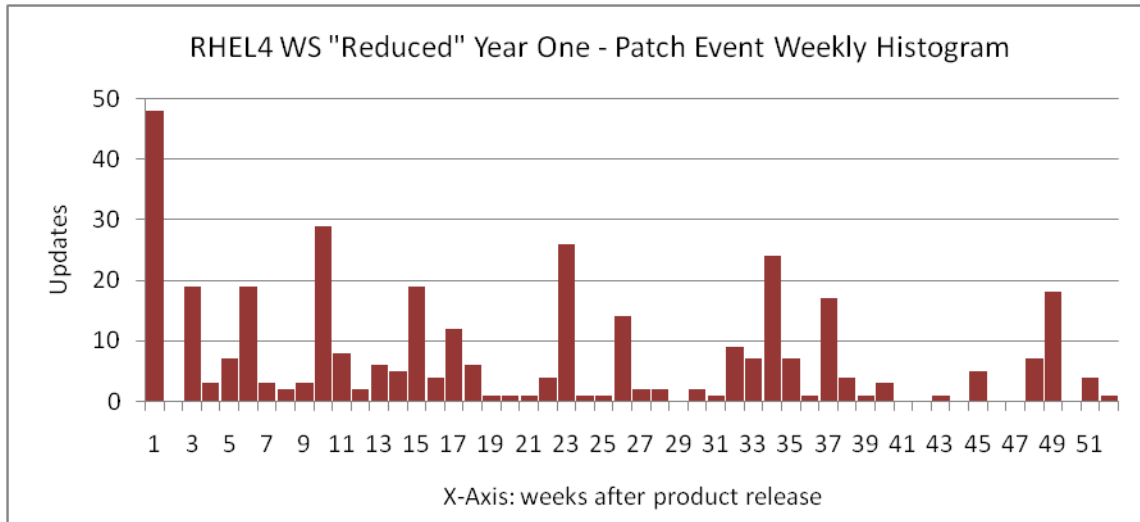


Figure 6: RHEL4WS Reduced Year One - Patch Event Weekly Histogram

In the charts, I used weekly breakdowns for charting Patch Events so not every single Patch Event is visible. In the case of Red Hat, several weeks actually had multiple patch events. As we can see in Figure 6, there were only 8 weeks in its first year that rhel4ws reduced did not have a patch event.

UBUNTU 6.06 LTS

Next up for comparison is Ubuntu 6.06 LTS. Ubuntu is considered by many to be the most popular up and coming Linux distribution and they committed to long-term support (LTS) for the Ubuntu 6.06 version³ released on June 1, 2006. Long-term support is a key requirement for a distribution to be considered for use within most businesses, so this makes the support commitment a strategic one for Ubuntu.

- Ubuntu 6.06 LTS had 53⁴ vulnerabilities already publicly disclosed prior to the June 1, 2006 availability date.
- During the first year, Ubuntu issued 181 security advisories for Ubuntu 6.06 LTS. The fixes were grouped so that Patch Events occurred on 119 days in the year.
- In the patches, Ubuntu fixed 406 vulnerabilities affecting Ubuntu 6.06 LTS. 160 of those fixed were rated High severity in the NVD.
- At the end of the one year period, there were at least 55 publicly disclosed vulnerabilities in Ubuntu 6.06 LTS did not yet have a patch from Ubuntu. Adding that to the 406 fixed, we get a total of 461 vulnerabilities.

³ Note that this also explains why I am not analyzing Ubuntu 6.10, 7.04 or later. So far, Ubuntu has only committed to long term support for 6.06 and not later releases.

⁴ Note that this number is quite a bit higher than the number I found for the six month report. This is because the methodology I use for the non-Microsoft products only counts vulnerabilities acknowledged and fixed by the vendor, so it is dependent on the time-to-fix for each vendor. To better understand why, see "Estimating Software Vulnerabilities," IEEE Security & Privacy, vol. 5, no. 4, 2007, pp. 28-32.



Ubuntu customers seem to have had a slightly better first year than Red Hat customers. Given that Ubuntu 6.06 shipped 16 months after rhel4ws, it may be that they benefitted from the open source contributions of Red Hat and other vendors.

UBUNTU 6.06 LTS – REDUCED COMPONENT SET

Similar to the component set reduction I did for RHEL4WS, I've completed an extra level of analysis for Ubuntu 6.06 LTS to exclude component vulnerabilities that do not have comparable functionality shipping with a Windows OS.

- The Ubuntu doesn't really give flexibility in terms of component selection at installation time. Instead they provide a separate installation CD for desktop and server installations. I downloaded the Ubuntu "desktop" iso and created a desktop installation disk.
- I ran the installation and afterwards, used 'dpkg -list' to generate a list of installed packages. I do note that none of the "optional server" packages are present, as they might be on a server installation.
- I manually excluded gimp and OpenOffice from the package list. I didn't exclude anything else because I felt that most users would not go to the effort to manually remove packages from the default desktop installation.
- I used the resulting package list to filter out vulnerabilities in packages that were not present.

Basically, this results in a Gnome-windows workstation that includes standard system management tools, Firefox for browsing, but excludes optional server packages, as well as OpenOffice and other optional stuff that a Windows system wouldn't have by default. This reduced Ubuntu build is then examined for comparison:

- During the first year of availability, Ubuntu issued 80 security advisories covering the reduced desktop build of Ubuntu 6.06. The fixes were released on 65 different days during the year in 39 different weeks.
- During the first year, Ubuntu fixed 224 vulnerabilities affecting the reduced Ubuntu desktop set of components.
- At the end of the period, a total of 18 publicly disclosed vulnerabilities in the reduced set of components did not yet have a patch from Ubuntu.

Again, we can observe that Ubuntu customers in a standard desktop installation experienced fewer vulnerabilities than users of Red Hat RHEL4WS. However, there was very little difference in terms of Patch Events, with Red Hat having 64 and Ubuntu having 65.

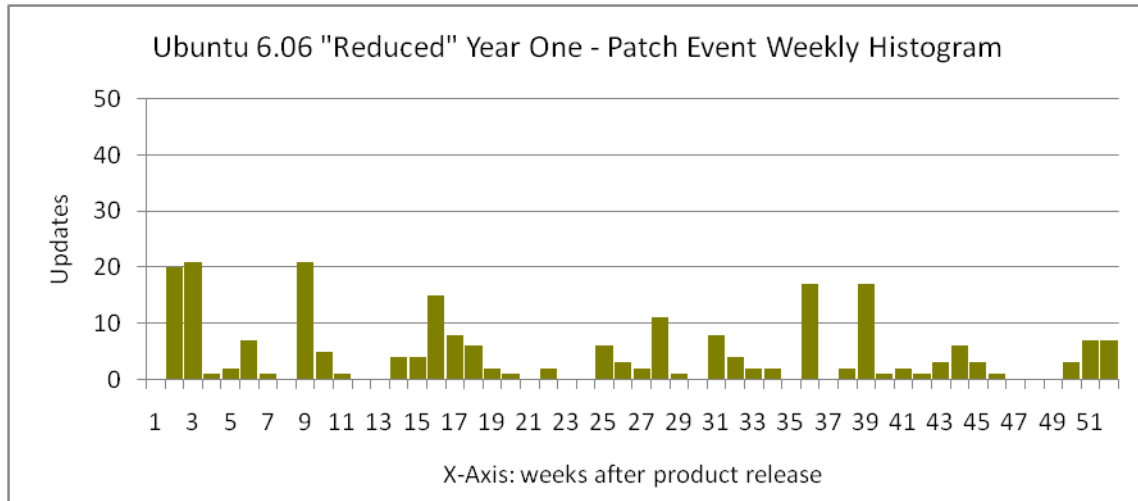


Figure 7: Ubuntu 6.06 Reduced Year One - Patch Event Weekly Histogram

There are only 13 weeks out of the year that Ubuntu 6.06 LTS did not have a Patch Event, so at least one security patch was released 75% of the weeks in the year.

APPLE MAC OS X V10.4

Apple⁵ shipped Mac OS X 10.4 on April 29, 2005.

- During the first year, Apple released 17 security updates affecting Mac OS X 10.4, each on a different day.
- Those updates fixed 116 vulnerabilities in shipping components of Mac OS X 10.4.
- At the end of the one year period, a total of 41 publicly disclosed vulnerabilities in the product did not yet have a patch from Apple, so the total vulnerabilities disclosed for the product including fixed and unfixed was 157 vulnerabilities.

Below is the Patch Events chart for the first year of Mac OS X 10.4.

⁵ Apple shipped Leopard on 10/26/2007. It is not included since it does not yet have a one-year track record.

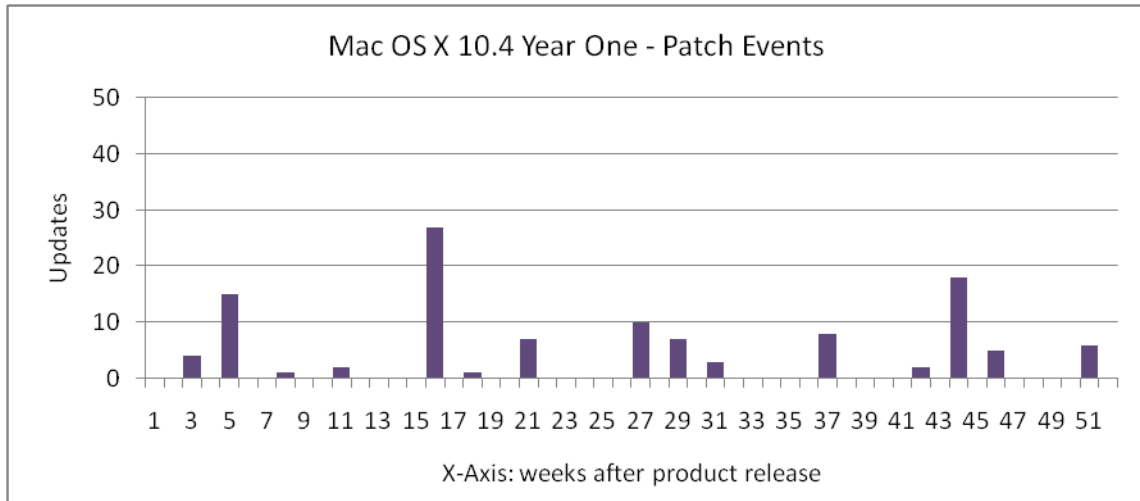


Figure 8: Mac OS X 10.4 Year One - Patch Events

Note that 17 Patch Events occurred in 15 weeks out of the year, with two of the weeks having updates on more than one day.

SIDE-BY-SIDE COMPARISON

With the basic analysis completed for Windows Vista and the other industry products, we now have enough information to compare them. First let's look at a chart that shows fixed and unfixed vulnerabilities side-by-side.

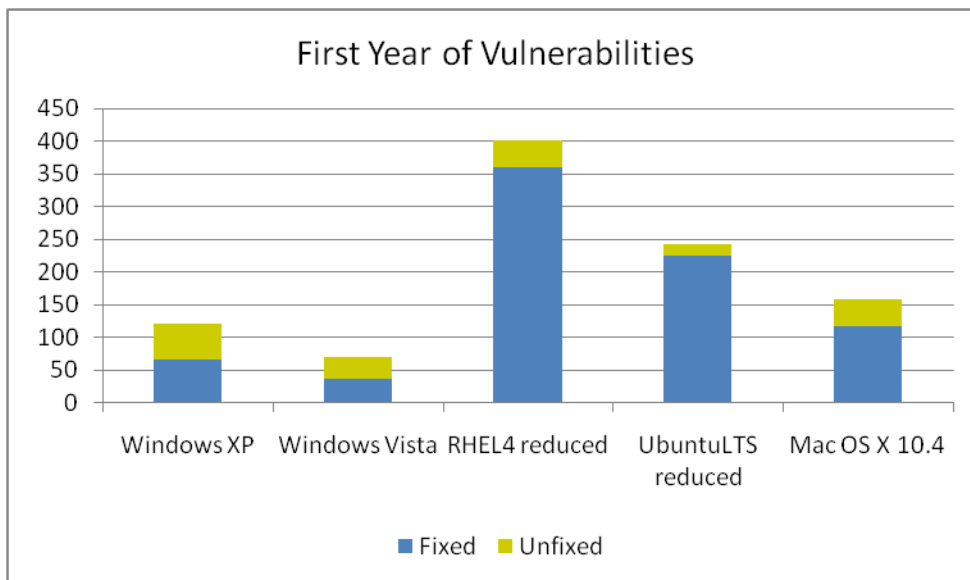


Figure 9: Side-by-Side Comparison of First Year Vulnerabilities for Windows Vista and Other OS Products

Figure 9 shows that the reduction in security vulnerabilities for Windows Vista is not just favorable as compared to its own predecessor, but is also favorable relative to other industry OS offerings.

Next, let's examine the impact that security updates had for administrators by looking at Patch Events. Windows Vista's 17 security updates occurred across 9 Patch Events in 9 different weeks in the year. I have charted that against the weeks in which the other OSes had patch events.

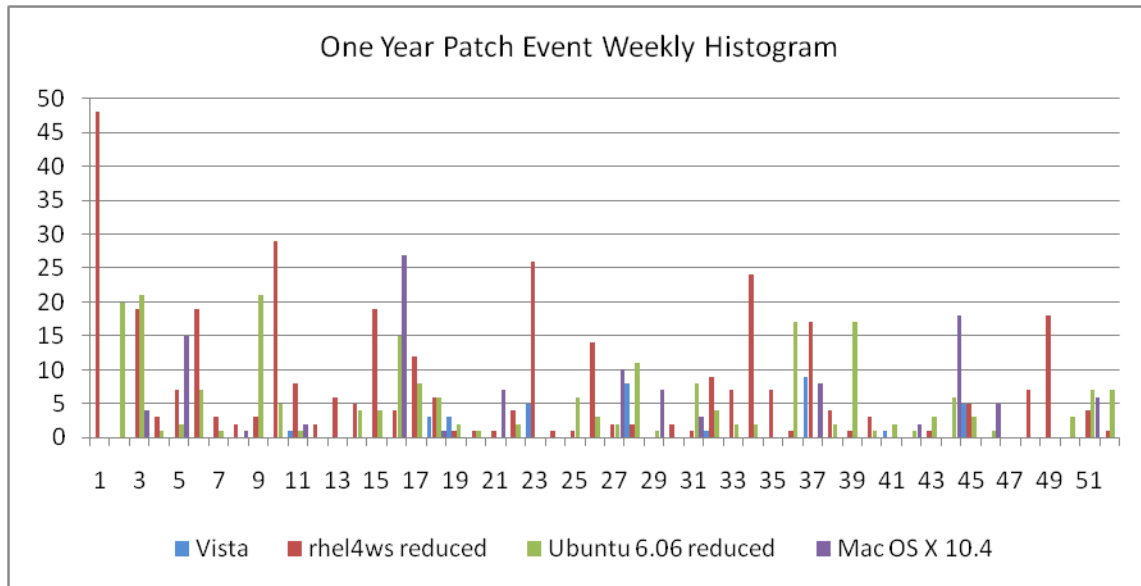


Figure 10: Patch Event Histogram for Windows Vista and Other Industry OSes

Graphed out visually in this case, it is hard to see anything, so Table 3 below is a bit more informative for comparison.

Metric	Windows Vista (year 1)	Windows XP (year 1)	Red Hat rhel4ws reduced (year 1)	Ubuntu 6.06 LTS reduced (year 1)	Mac OS X 10.4 (year 1)
Vulnerabilities fixed	36	65	360	224	116
Security Updates	17	30	125	80	17
Patch Events	9	26	64	65	17
Weeks with at least 1 Patch Event	9	25	44	39	15

Table 3: Summary Table for All Products Analyzed

As you can see from the table, Windows Vista again compares favorably against the other operating systems, having fewer vulnerabilities and having fewer weeks in which security updates potentially impacted security administrators.

FINAL OBSERVATIONS

Much has been said and written about Windows Vista security during its first year of availability, both positive and negative. In this report, I've attempted to take a methodical look at a few measurable elements that could be part of a broader security analysis and provide some insight into what level of improvement Windows Vista may represent in terms of security vulnerabilities and updates.

I examined vulnerabilities found and disclosed by researchers during the first year that products were released. This is one measurement that can be examined to see how effective the respective vendor development processes are at discovering and fixing security vulnerabilities before product release. My analysis found that researchers found and disclosed significantly fewer vulnerabilities in Windows Vista than either its predecessor product, Windows XP, or other operating systems such as Red Hat Enterprise Linux, Ubuntu, and Apple Mac OS X 10.4.

I additionally charted weekly histograms of the Patch Events - any day with at least one security update. Patch Events are an indirect measure of how the combination of product Security Quality and vendor update release policies and processes impact security administrators - specifically, how many days in the year did the administrators have to mobilize to assess the need to deploy one or more security updates. My analysis found that administrators were required to mobilize much less often for Windows Vista than any other product examined.

Finally, let me re-iterate that this is just part of the overall security picture, though I think security quality (as reflected by vulnerability reduction) is an important foundational part of product security. If you are utilizing this document as part of an overall security assessment effort, I encourage you to additionally examine:

- Security capabilities for your business scenario. For example, BitLocker full drive encryption is a capability useful for many laptop policies.
- Architectural and protective security mechanisms that can provide defense-in-depth layers of protection. Most current generation products, including Windows Vista, have implemented advances such as a firewall, stack protection and library randomization.
- Manageability in the context of your policy. How easily can you centrally make policy change? Lock down a port across the environment? Roll out new certificates? Verify that your environment is policy compliant? Simple misconfiguration can be a huge security issue, so manageability is an important consideration.



APPENDIX A: FREQUENTLY ASKED QUESTIONS

My experience with previous analyses and reports of this type shows that new readers frequently have some questions that I answer in blog comments or via email. I thought I would get in front of those a bit and include some of the questions I anticipate and answer them in the report itself.

Q: You work for Microsoft, so why should we believe anything you say?

The other variation on this is “would you have still published if the results for bad for Microsoft?”

People may be surprised to learn that I always like it when I get this question. The real answer is that I was not really worried about getting bad results. A better question might be why I was confident enough in the results to initiate the project.

Think about it – Microsoft has been investing heavily in security improvements for products for about 6 years now. The commitment to security is real – I made myself as sure of that as I could before I joined the company. Others may believe that or not, but I’ve been here for five years and observed the executive commitment and hard work first hand. I’ve had the pleasure of working with some great security people like Mike Howard and David Cross. I was here as the team grew and we attracted great industry experts like James Whittaker and more recently, Vinny Gullotto.

Because of that, I can say what I always so. Be skeptical! Assume I’m “spinning” things if you wish and try to go find out for yourself. That is ultimately my goal – to get people to actively question and dig into why the results turn out the way they do. All of my sources are identified in Appendix B: Sources and Methodology, so anyone can work to duplicate the analysis in this report. I am happy to discuss findings with them.

Q: Why did you look at the first year of availability for each product rather than November 30, 2006 to November 30, 2007 for all products?

I conduct my research for the first 12 months each product is in market which allows me to evaluate newly released products and ask “what sort of positive benefit did the product development methodology have?” If I took the same calendar timeframe to evaluate each product some would have been in-market for a few years (and gone through a stabilization period) and it would be more difficult to assess reported vulnerabilities back to the development process. Windows XP and Windows Vista had very different product development processes since Windows XP was released prior to the development of the Security Development Lifecycle (SDL).

Note that I will be doing a 2007 calendar year comparison as well, so stay tuned to <http://blogs.technet.com/security> for analysis if you are interested.

Q: Linux distros contain many more optional applications than Windows – that is Apples and Oranges - how can any comparison be valid?

Actually, Windows Vista and Windows XP have different components too. Windows Vista Ultimate includes Media Center for example, which was not in Windows XP Professional. From a user perspective, I think it is Apples and Apples. Whichever OS is chosen, I believe most people



will install the default set of components and use that. If vulnerabilities are in those components, they will be exposed and need to take mitigating action.

I did, however, try to even the playing field as much as possible by excluding optional Linux-distro components and excluding even some default components for which there is no obvious counterpart. In contrast, on the Windows analysis, I included any component that shipped with the product. I think the comparison is valid and useful.

Q: What about (so called) “silent fixes”? Past analyses have been criticized saying that you don’t count issues that Microsoft finds internally and “silently” fixes, so comparisons are invalid.

This is an interesting line of thinking to me. It is true that I don’t know if *any* vendors’ product updates address more security issues than is documented. There’s no way to know things that haven’t been discussed publicly.

For example, I have no idea how many security vulnerabilities were found by the Apple Quality Assurance team during the release of Leopard and were simply fixed. Further, I don’t know how many “bugs” were found and fixed without anyone, even on their team, knowing their *might* have been security implications if it had not been found. This is equally true for Linux distributions. I don’t know how many “bugs” fixed during the development process for rhel4 Update 5 might have had a security implication.

In terms of enumerating vulnerabilities though, there are specific examples that I can point to that indicate that silent fixes sometimes happen. Take CVE-2007-5959, for example. It is a single vulnerability identifier, but the description says “multiple unspecified vulnerabilities”. I would count that only a single time in my analyses though, since there is only a single CVE identifier. Similarly, CVE-2004-1057, says that “multiple drivers in Linux kernel” do not properly mark memory and enable a denial of service. I would only count this as a single issue in any analysis, though technically there are an additional number of vulnerabilities silently fixed. These products are getting the “benefit” of the issues that are not detailed in any analysis.

On the other hand, I can say that in Microsoft security updates, the MSRC policy is to document any internally found vulnerabilities that change the risk assessment or severity of an externally found vulnerability, or ones where the mitigations and workarounds listed don’t apply. So, by counting the issues that get publicly disclosed for products I’m using an identifiable set of vulnerabilities that have an increased risk for customers.

More generally, if a theoretical “silent fix” (in any product) actually is easily rediscoverable and is proven to be so for any vendor’s product, then it will join the publicly disclosed set of vulnerabilities in due course and can be measured as well.

Ultimately, I see the so-called “silent fixes” criticism to be a bit of a Red Herring that distracts readers from the core results of the analysis of publicly disclosed vulnerabilities.

Q: Why did you not include Days-of-Risk (DoR) information?



January 15, 2008

I'm one of the people who do think Days-of-Risk (DoR) is a valuable metric at measuring a combination of factors that go into vendor response to publicly disclosed vulnerabilities – such as testing, responsiveness and so on. However, I also don't think there is much comparative value in seeing how Microsoft did in Vista in 2007 versus Red Hat in 2005 or Ubuntu in 2006.

A more interesting question might be how they all did in the most recent year. Along those lines, I will be completing a separate analysis and report covering Days-of-Risk for vendors in the calendar year 2007, so stay tuned to <http://blogs.technet.com/security> for analysis if you are interested.



APPENDIX B: SOURCES AND METHODOLOGY

The efforts to identify and fix vulnerabilities lacked a common naming mechanism until a consortium led by the Mitre Corporation began publishing the Common Vulnerabilities and Exposure (CVE) list, in an attempt to drive a common naming mechanism that could be leveraged by multiple vulnerability databases and security products. The CVE naming conventions and process has achieved success in being the most comprehensive list of vulnerabilities across software products of all types and worldwide. In this report, I use the CVE naming convention when identifying individual vulnerabilities.

The analysis in this report uses a set of data that has been compiled, customized and cross-checked using several sources of data available on the Internet:

- Microsoft Security Bulletins as published at <http://www.microsoft.com/technet/security/current.aspx> and associated web pages.
- Red Hat Security Advisories as published at <https://rhn.redhat.com/errata/rhel4ws-errata-security.html> and associated web pages.
- Ubuntu Security Notices as published at <http://www.ubuntu.com/usn> and associated web pages.
- The [National Vulnerability Database](http://cve.mitre.org) (NVD), a database superset of the Mitre CVE list (<http://cve.mitre.org>). The NVD is also sponsored by the US Department of Homeland Security and makes their data downloadable in an XML format at <http://nvd.nist.gov/download.cfm>.
- Many security websites were utilized for detailed verification and validation of vulnerability details, and especially dates for when the issue was first discussed publicly. Some of the most commonly utilized were: www.securityfocus.com, the Bugtraq mailing list, www.secunia.com, and www.securitytracker.com, but there were many others.

Leveraging these and many other sources, I compiled a database of vulnerabilities for the products analyzed. Note that more detail is also provided in the section entitled “Discovering Unfixed Vulnerabilities” with respect to compiling unfixed vulnerabilities.

Note that in this report, “disclosure” is used to mean broad and public disclosure and not any sort of private disclosure or disclosure to a limited number of people.

DISCOVERING UNFIXED VULNERABILITIES

I received many comments about the percentage of unfixed vulnerabilities after I published my 6 month report and analysis, so I thought I should provide some clarification and detail that fed perceived differences. In this report, I basically use two different methodologies to discover and count the disclosed but unfixed vulnerabilities.

Method 1: Thorough case-by-case examination. This is the method I used for Windows Vista and Windows XP in this report. Essentially, I scanned the NVD and other web sources for any reference attributing a vulnerability to a product that was not included in my fixed list and was disclosed prior to the end of the first year. The benefit of this method is that it is potentially more



accurate. The drawbacks are that it is more time consuming and is more difficult to apply definitively to snapshot distributions like most typical Linux-based operating systems.

***Example:** rhel4ws shipped with Firefox 1.0. I can identify 12 vulnerabilities in the NVD for Firefox 1.0 that were never addressed in a security advisory by Mozilla and 9 of those vulnerabilities were disclosed before the end of the rhel4ws' first year. Did they apply to rhel4ws? Probably, but I don't know. Perhaps Red Hat addressed these issues prior to ship in their own testing phase. I can't really know unless the vendor later fixes the issues and affirms them. So, I did not include these in the Red Hat "unfixed" count.*

Method 2 : Identify what the Vendor Fixed Later. This is the method that I used for Mac OS X and the Linux systems. Basically I take the list of all things ever confirmed and fixed by the vendor and then just filter for the ones that were disclosed prior to the end of the first year, but were not fixed until sometime later. The benefits of this method are two: it is easy to execute and the vulnerabilities identified have been confirmed by the vendor.

The drawbacks are that it isn't effective until a statistical time period has passed (so I couldn't use it at all for Windows Vista) and it does not identify any issues that the vendor never fixes.

As a result, one should look at the unfixed vulnerability counts identified by Method 2 as minimum count, rather than a full list.