**May 6, 2004**

# A Comparative Analysis of Linux vs. Windows Security Capabilities

**Stacey Quandt**

## EXECUTIVE SUMMARY

Security is now center stage due to world events, government regulations, and legislation. IT vendors are responding to this trend with tools to manage and reduce security risk. For example, Microsoft and Linux developers are enhancing security at the operating system layer. However, looking at only point product releases is not sufficient. Customers need a framework to evaluate operating system security that includes an assessment of base security, network security and protocols, application security, deployment and operations, assurance, trusted computing, and open standards. The objective of this study is to compare Microsoft Windows and Linux security across seven categories. The overall findings of this qualitative assessment are that Linux provides superior to comparable security capabilities in comparison to Windows. The one category in which Windows surpassed Linux is assurance. However, the pace of innovation continues and SUSE will achieve EAL4 by year-end, effectively reaching assurance parity with Microsoft.

## INTRODUCTION

The security capabilities and differences in architectural design between Linux and Windows continues to fuel the debate on which is better -- an open source or closed source operating system. Industry logic is that an operating system based on open standards and open source enables interoperability, improves bug detection and fixes, and is superior to a model of security through obscurity. Open source also forces Linux distribution providers to be absolutely transparent in the production process. Every step can be re-run by users and this enables incremental security on a meta level. Microsoft Windows, in contrast, does not enable equivalent transparency.

Historically, price/performance derived through benchmarks, such as the Transaction Processing Performance Council (TPC) is more often the focus of the IT industry. The corresponding marketing of benchmark results is designed to target the sale of hardware and software. Only recently, with the growing number of attacks from viruses, worms, and the potential for operational risk and adverse financial impact on the business, has operating system security gained importance. For years, operating systems have received scrutiny based on factors of scalability, availability, manageability, and serviceability. Today the selection of an operating system is no longer limited to attributes of measuring workload performance.

In vertical market segments, such as financial services, health care and other businesses, security has heightened significance. There are a number of ways for end users to enhance overall security, by defining policies, education, firewalls, and disabling services. However, these tactics do not target operating system design and, therefore, are limited in enabling compliance with regulations and legislation. The magnitude of security attacks compels organizations to understand the key differences and distinctions in operating system security architecture.

The challenge in evaluating Windows and Linux on any criteria is that there is not a single version of each operating system. Indeed, Windows 98, Windows NT, Windows 2000, Windows 2003 Server, and Windows CE are just a subset of Microsoft's offerings. A Linux distribution, while available from many providers --Debian, Red Hat, SUSE, etc. -- is defined by the Linux kernel release it is based on (e.g., 2.2, 2.4, and 2.6) and the versions of all packages contained. Hence, this study evaluates operating system security according to the current technology available in the market rather than legacy solutions.

Users need to keep in mind that there are philosophical differences in the design of Linux and Windows;. The Windows operating system is designed to support application by moving more functionality into the operating system, and by more deeply integrating applications into the Windows kernel. In comparison, Linux differs from Windows with a clear separation between kernel space and user space. This matters because the ability to make either operating system more secure will vary depending on architectural design.

Relative to the differences in design of Windows and Linux is the process and complexity of patch management. The number of patches and time required to test and deploy a patch can increase operational costs. Other factors that can impact the ease of patching a system include determining if a patch is backward compatible and can be implemented without breaking an application. The magnitude of patching a Windows system is complicated by the tight integration of a Windows application runtime environment and operating system. In contrast, under Linux the application runtime environment is a user space process and is not part of the operating system. The tight integration of a Windows operating system increases the number of potential security exposures; in effect, this means a Windows server patch is not a feature but often a requirement. The sheer landscape of IT infrastructure needing Windows patches will continue to grow because of the non-trivial nature of exploits like Blaster, Code Red, Sasser and others. This is compounded by the complexity of assessing a variety of Microsoft partners and independent software vendors to provide patch management. For years, Microsoft security has been the equivalent of using a lawnmower to trim a hedge--if you were careful, you wouldn't lose any limbs.

Patch management under Linux is often easier on account of the separation of kernel and user space, which reduces the number of potential significant security exploits. Although every Linux distribution comes with patch management tools the growth of Linux adoption, increases the opportunity for system vendors and independent software vendors to provide third-party tools. BMC, HP OpenView, IBM Tivoli, and Aduva all offer tools to distribute and deploy patches. Another benefit of patch management on a Linux system is that the process provides more transparency than Windows. Linux distributions provide all changes, which are applied to every package. Since Linux is open source, unlike Windows, there is unrestricted access to the history of all of the source code. Also, with Linux there is often more flexibility to use either a GUI or the command-line to patch a system. For example, Ximian Red Carpet's automated dependency and conflict resolution provides both a Web interface and command-line capabilities. Red Hat's system update tool, called up2date, and works with Red Hat Network to enable users to download and install new packages. SUSE uses a process called AutoBuild to enable quality assured patches and bug fixes.

Fundamental changes in the security capabilities of Windows and Linux are vital since they are positioned as the No. 1 and No. 2 operating systems based on new server shipments. However, advances in operating system security are only as good as the users who take advantage of them. How secure an IT infrastructure is will not only vary based on the Linux distribution and Microsoft product and service pack deployed, but also by what customers choose to implement.

## FUNDAMENTAL CHANGES IN LINUX AND WINDOWS SECURITY

For users of Linux and Windows, the evolution of these technologies has all the trappings of a muscle car drag race. Users may have their favorite but at the same time continue to assess the competition. Microsoft has shown a great willingness -- no doubt spurred on by industry cynicism and the growing adoption of Linux -- to dedicate massive resources to Windows security. Microsoft will make advances in Windows security within the next few months when it releases Service Pack 2 for Windows XP. This service pack

enhances Windows security by turning off some services by default and will also provide new patch management tools. For example, the Alterer and Messenger service has been turned off to reduce the amount of spam received. In many cases, turning off features is good since it makes a system more secure. However, the challenge is to enable to security without a tradeoff in key functionality or flexibility.

What is most outstanding is Microsoft's focus on enhancing security through improved usability. For example, a number of Microsoft security exploits in 2003 were the result of an email attachment launching as an executable (e.g., MyDoom). Service Pack 2 features an attachment execution service that will have a central place for attachments to be accessed by Outlook/Exchange, Windows Messenger, and Internet Explorer. This will reduce the risk of an end user enabling a virus or worm by launching an executable. Also, disabling execution of data pages will limit the potential for buffer-overflow exploits. Still, rather than actually fixing Windows' broken infrastructure and secure communications, the burden is carried by the user.

Microsoft's focus is clearly on shoring up application security. There are a number of Service Pack 2 enhancements that specifically target Outlook/Exchange and Internet Explorer. For instance, there will be an intelligent MIME-type review in Internet Explorer that will check the content type of an object and will let the user know if is a potentially harmful executable. This raises the question of whether a user's desktop will be able to distinguish a virus from a colleague's spreadsheet extension.

Another new feature in Service Pack 2 is the ability to uninstall additions to a browser, which potentially places more responsibility on the end user who may have to look at many of plug-ins and uninstall the right one. Outlook/Exchange will have the ability to preview email messages, so a user can delete a message without actually opening it. A further application security enhancement is a firewall that starts prior to the network stack. For software developers, the changes to remote procedure call permissions will make it a harder to write code that is not secure. Indeed, Service Pack 2 will offer many flashy new features for Windows users, but the question remains: Will these features burden system administrators, and possibility end users, with more complexity rather than addressing the security of Windows operating system code?

A purely philosophical difference between Linux and Windows is the approach to code transparency. Linux is licensed under the GNU General Public License, which means it is possible for users to copy, modify, and redistribute the source code. Windows is a closed source operating, which is why its security methodology is often characterized as "security through obscurity." In 2001, Microsoft responded to the demands of its customers (and perhaps its critics too) with the Shared Source Initiative. Today, the Shared Source Initiative has one million participants and source code is available for Windows 2000, Windows XP, Windows Server 2003, Windows CE 3.0, Windows CE .NET, the C#/CLI implementations, as well as components of ASP .NET and Visual Studio .NET. Shared Source Initiative licensees include corporate customers, governments, partners, academics, and individuals.

However, to a large degree Microsoft's Shared Source Initiative is a policy of "look but don't touch." The rare exception is the Windows CE Shared Source Premium Licensing Program available to companies, which brings Windows CE-based devices and solutions to market. This is the only Windows program under the Shared Source Initiative that provides original equipment manufacturers (OEMs), silicon vendors, and systems integrators full access to Windows CE source code. All licensees have complete access to the source code and the right to modify the code, however, only OEMs can commercially distribute those modifications in Windows CE-based devices. In contrast, all other shared source licensees have to make a trip to Microsoft in Redmond, Washington to access source code that is not available through the program. Although some users may find the program useful for debugging applications, the requirement to be physically at Microsoft headquarters to do a build is a significant limitation (with the exception of Windows CE licensees). This is an important distinction because despite Microsoft's efforts to add more transparency, the inability to do a build makes it difficult, if not impossible, to know whether the code will work when implemented within an actual IT environment. The restrictions to modify and recompile Windows source code reduce the incentive for people with access to the Windows Shared Source to look for security vulnerabilities. In addition, it is a smaller community than those who have access to the Linux source base. In this sense, Linux is open by design and Windows is open by decision.

**Linux Security Benefits in the Data Center and on the Desktop**

During the next 12 months, Linux will strengthen its hold in the data center and make significant inroads on Microsoft's desktop monopoly. To a large degree this will be the result of new features and functionality in the 2.6 version of the Linux kernel. With Linux v2.6, the security architecture is now modularized. Under this model, all aspects of the Linux kernel are designed for fine-grained user access instead of the prior capability of the superuser. The implication is that while Linux systems will still support root, which gives a user total access to a system, it will be possible to create Linux systems that do not follow this model. Another major change with Linux v2.6 is the addition of Linux Security Modules (LSM), which allows users to add additional security mechanisms to a Linux distribution without needing to patch the kernel. A variety of access control mechanisms have been built on top of LSM, including the United States' National Security Agency's Security Enhanced Linux (SE Linux). SE Linux, using a security scheme known as Domain Type Enforcement, can limit the impact of compromised applications or network services by separating applications from each other and from the base operating system. For example, Immunix offers a set of products, including StackGuard, and sub-domain LSM modules to configure a process to a specific system call. Red Hat has announced that SE Linux will play a major part in their security architecture in Red Hat Enterprise Server 4.0.

A benefit of Linux not requiring the support of a single vendor for its development is a diverse user base that can create new features and functionality. (The US National Security Agency (NSA) participates in the Linux community. The Security Enhanced Linux project (SELinux) grew out of the NSA's interest in operating system security and the value of mandatory access controls. The NSA researchers worked on Linux security modules to support type enforcement, role-based access controls, and multi-level security in the v2.6 kernel. SELinux's fine-grained Boolean labeling support has been added to v2.6. Today, Linux has a powerful, flexible mandatory access control architecture built into the major subsystems of the kernel. The system mandates the separation of data based on confidentiality and integrity requirements, therefore, any potential damage, even that of a superuser process, is confined on a Linux system.

Linux v2.6 also provides support for cryptographic security with the addition of a cryptographic API used by IPSec. This enables multiple algorithms (e.g., SHA-1, DES, Triple DES, MD4, HMAC, EDE, and Blowfish) to be used for network and storage encryption. The ability for Linux to support IPSec protocols for IPv4 and IPv6 is a significant advance. With security abstracted to the protocol level, applications are less vulnerable to a potential exploit. However, cryptographically signed modules are not yet a part of Linux; but if the issues about implementing such a feature can be resolved it will prove useful in preventing unsigned modules from being accessed by the kernel.

One of the issues that continue to plague Windows users is buffer overflow. Linux users will appreciate the addition of an exec-shield patch that enables protection against a variety of exploits that attempt to overwrite data structures or insert code within these structures. Since a recompile is not required for the exec-shield patch to work, this makes it easier to implement. Also, the addition of a preemptive kernel reduces latency, which is likely to drive the use of Linux not only in the data center, but also for applications that require a deterministic kernel with soft real-time capabilities.

Many Linux users depend on non-open source drivers and other binary modules from hardware manufacturers and systems providers. The problem is that although adding these drivers and modules is often useful, it is not necessarily beneficial to the operation of a Linux system. For example, a non-open source driver or binary module can overwhelm a system call and change the system call table. The Linux v2.6 kernel provides protection against these dangers by placing restrictions on the level of access a non-open source driver or module has to the kernel. This feature enables stability, but does not place any new restrictions from a security point of view to stop a determined hacker from writing a malicious module. Perhaps one of the most innovative developments for Linux users is User-mode Linux (UML). There are a number of advantages to UML but the more compelling attribute is the ability to use it as a virtual machine. Since processes within UML are not allowed access to the host system, it can be used as a sandbox to test software, run unstable distributions, and examine activities that could otherwise pose a risk. UML will eventually lead to a fully virtualized environment for security infrastructure.

**Key Findings: Linux vs. Windows Security Capabilities**

A qualitative assessment of operating system security is subjective and your "mileage may vary" based on present and past experience. The goal is to provide a framework for users to increase their understanding of Windows and Linux security capabilities. The following analysis is by no means comprehensive and is intended as a starting point for end-user evaluation (see Table 1). As the technical innovation of Linux and Windows continues, so will the discourse on which is more secure. The overall finding of this analysis is that Linux provides more secure capabilities than Windows.

Table 1: Key Linux and Windows Operating System Security Capabilities

| *Category* | *Capability* | *Linux* | *Windows* | *Qualitative Score* |
|---|---|---|---|---|
| Base security | Authentication, access control, cryptography, audit trail/logging | Pluggable Authentication Module, plug-in modules, Kerberos, PKI, Winbind, ACLs, LSM, SELinux, Controlled Access Protection Profile audit, kernel cryptography | Kerberos, PKI, Access Control lists, Controlled Access Protection Profile audit, Microsoft crypto application programming interface | **Linux is superior** |
| Network security and protocols | Authentication, layer, network layer | OpenSSL, Open SSH, OpenLDAP, IPSec. | SSL, SSH, LDAP, AD, IPSec | **Both are comparable** |
| Application security | Antivirus, firewalls, Intrusion detection software, Web servers, email, smart card support. | OpenAV, Panda, TrendMicro, firewall capability built into the kernel, Snort, Apache, sendmail, Postfix, PKCS 11, exec-shield | McAfee, Symantec, Checkpoint, IIS, Exchange/Outlook, PCKS 11 | **Linux is somewhat superior** |
| Deployment and operations | Installation, configuring, hardening, administration, vulnerability scanners | Install and configuration tools, Bastille, mostly admin through command line interface, Nessus, distribution specific Up2Date, YaST, Webmin | Install and configuration tools come with Windows, no specific hardening tool, admin GUI, security by default has been emphasized lately | **Both are comparable** |
| Assurance | Common Criteria Certification, flaw handling | Linux has achieved EAL3 and has good flaw handling | Windows has EAL4 and good flaw handling | **Windows is superior** |
| Trusted computing | Trusted Platform Module, Trusted Computing Software Stack, instrumentation, attestation | Trusted Platform Module device driver open sourced by IBM, Trusted Computing Group software stack is targeted for 2005 | Next-generation Secure Computing Base, possible availability with Longhorn 2006 | **Neither is superior** |
| Open standards | IPSec, POSIX, Transport Layer Security, Common Criteria | Linux meets all open standards | Microsoft participates in open standards but has some proprietary standards. | **Linux is superior** |

**Base Security**

Microsoft and Linux both provide support for authentication, access control, audit trail/logging, Controlled Access Protection Profile, and cryptography. However, Linux is superior due to Linux Security Modules, SELinux, and winbind. The user of a Linux system can decide to add additional security mechanisms to a Linux distribution without having to patch the kernel.

Various access control mechanisms have been built on top of LSM; for example, building compartments that keep applications separate from each other and from the base operating system, which limits the impact of a security problem with an application. Linux base security is further enhanced by solutions, such as Tripwire, that enable System Integrity Check functionality to periodically verify the integrity of key system files and warn those responsible for system security whether a file's contents or properties have been changed.

A limitation of Windows base security is MSCAPI, which trusts multiple keys for code signing. Microsoft's model focuses on providing one build of a product that can enable weak or strong encryption simultaneously. Although modules are not all signed by one key, since MSCAPI trusts a large number of root certifying authorities, and trusts multiple keys for code signing, it only takes one key to be compromised to make the entire system vulnerable to attack. This can happen either by having an authorized code signer accidentally disclosing their private key, or by having a certifying authority issue a certificate in error. This has already happened once, when Verisign mistakenly signed two certificates in Microsoft's name and released control of these certificates to unauthorized individuals.

**Network Security and Protocols**

Linux and Windows support for network security and protocols are comparable. Both enable support for IPSec, an open standard for cryptography-based protection at the IP layer. IPSec verifies the identity of a host or end point and ascertains that no modifications were made to the data during transit across the network and encrypts data. OpenSSH, OpenSSL, and OpenLDAP are available on Linux and corresponding closed source implementations -- SSH, SSL, LDAP -- are available on Microsoft systems.

**Application Security**

Linux is somewhat superior due to continuing security issues with Microsoft IIS and Exchange/Outlook. Apache and Postfix are cross-platform applications and tend to be more secure than corresponding Microsoft products. Application security for Linux is also enhanced with firewalling built into the kernel. And Snort is an excellent intrusion detection system. One notable recent addition the Linux kernel for x86-based systems is Ingo Molnar's exec-shield, which provides protection against attacks from buffer or function pointer overflows and against other types of exploits that rely on overwriting data structures and/or putting code into those structures. The exec-shield patch also makes it more difficult to conduct a shell-code exploit. Since exec-shield operates transparently applications do not need to be recompiled.

Microsoft is taking strides to redesign the security of its products and provides patches for its installed base. Still, security issues in legacy Windows products persist and complicate this task. This leaves many Microsoft users exposed to security threats since patches must be well documented prior to deployment. Also, the tendency for Microsoft to mix data and program code in its applications, e.g., Active X, can allow untrusted data from outside the system and can cause the activation of arbitrary code with untrusted data. In some cases, Windows will even allow digitally signed code to be supplied from outside the system, which means a local systems administrator can't audit the code. Instead the system administrator is dependent on whoever signed the code to perform an appropriate code review.

Application security is improved for Microsoft-only applications on the .NET Framework. Of course, for IT shops with heterogeneous platforms, e.g., Linux, Windows, Unix, and especially for applications built on Java, application security for Microsoft-only products is limiting.

**Deployment and Operations**

With deployment and operations, Linux has a slight edge over Microsoft since most administration is done through a Command Line Interface. A variety of installation and configuration tools, e.g., up2date, YaST2, Webmin, are available from Linux distribution providers. Bastille Linux is a hardening tool and supports Red Hat, Debian, Mandrake, SUSE and Turbolinux Linux distributions. In contrast, most Microsoft system administrators use a GUI that can be easy to use but also allow mistakes in configurations easily. Despite the fact that some people believe that it is possible to train anyone to be a Windows system administrator in one week, the question is how much will they understand about administration? Therefore, the overall majority of Microsoft security problems are due to poor configuration during deployment and operations. Installation and configuration tools come with Windows, and Microsoft provides guidance in hardening domain controllers, infrastructure servers, file servers, print servers, IIS servers, IAS servers, certificate services, and bastion hosts. However, there is distinction between hardening infrastructure and hardening the operating system.

**Assurance**

The metric that defines operating system assurance is Common Criteria (CC), an ISO standard (ISO 15408). There is a hierarchy of evaluation assurance levels, for instance, EAL1 through EAL7. The Common Criteria evaluation is valid only for a specific system configuration of hardware and software. Windows is superior to Linux because it has achieved EAL4. Linux recently achieved EAL3 and there are plans to target EAL4. It is important to keep in mind that primarily government organizations require CC assurance. Even though assurance requirements started primarily with government accounts, and in particular the US Department of Defense, they are applicable in a commercial setting as well. However, most customers will not need to meet the same level of assurance as the Department of Defense.

**Trusted Computing**

Trusted Computing is an architecture that prevents the tampering of applications and enables secure communication with a vendor. A number of vendors, like Intel, Microsoft, and IBM, are embracing the potential of this emerging technology. At present, this capability is more vision than reality and neither Linux nor Windows is superior at this time. Microsoft's vision of Trusted Computing is related to digital rights management. There is a considerable amount of work that needs to take place before the open source community acknowledges value in Trusted Computing.

**Open Standards**

Linux is superior to Windows because it supports open standards. Although Microsoft also supports a number of the same open standards, like IPSec, IKE, Ipv6 and TCP/IP, it also embraces and extends standards. For organizations with heterogeneous systems and a requirement for interoperability, the support for standards, which have been extended with proprietary code, makes consistent flaw detection and bug fixing usually more time consuming and difficult. An example of this is Microsoft's extension of Kerberos, a standard protocol. Microsoft added an authorization capability to the Kerberos ticket and although Kerberos was initially defined for this specific purpose the functionality was never used. Moreover, Microsoft embraced and extended the Kerberos standard by specifying the process for other applications to share the authorization data field in the ticket. The impact of this is that Microsoft's version of Kerberos is not completely interoperable with the standard, therefore, IT managers who use Microsoft Kerberos will find it harder to deploy and manage Kerberos across a heterogeneous IT environment and will prefer an all-Windows IT infrastructure.

**Open Source**

If the criteria for a secure operating system is open source than Linux is clearly superior to Windows. Microsoft's Shared Source Initiative, especially the focus on governments, is an attempt to meet customer requirements for looking at source code. Yet, in large part, Shared Source subscribes to a "look, but don't touch" philosophy. The governments of Russia, the United Kingdom, China, and NATO participate in Microsoft's Government Security Program. Despite the pragmatism of this initiative to add transparency

and emphasize partnership, there are varying requirements on the access and use of Microsoft source code. For example, not all source code for Windows can be viewed online, so a user who wants to do a build and test an application must plan an on-site visit to Microsoft's Redmond, Washington headquarters.

**RECOMMENDATIONS**

Linux provides superior –to comparable security capabilities in comparison to Windows. Still, the security of a Linux system is largely dependent on the choice of a Linux distribution and the kernel it is based on and the skill of the IT staff to implement and support a Linux system. In selecting an operating system consider architectural design and the quality and feature/functionality of its components. Since your success in implementing and maintaining a secure operating system rests with your IT shops, make sure that they have the training and expertise to deploy, manage, and troubleshoot. Keep in mind the differences and distinctions between operating systems will remain relevant for the foreseeable future even with the potential of Web services and the use of abstraction layers to simplify application resource allocation and manageability.

For CIOs and CTOs security will continue to be a key area of focus due to business continuity and regulatory mandates. We recommend that users start with an analysis of their operating system security by becoming familiar with key security capabilities that are required to meet the organization's need for functionality, which will reduce risk and ensure compliance. If you are considering migration to a different operating system or upgrading your current product, select an operating system environment based on a qualitative analysis of security capabilities -- not point products. Formulate discipline on the part of the IT manager and system administrators who need to understand how to apply security best practices. If you are seeking a quantitative analysis of security vulnerabilities in Windows, Linux or other operating systems start with a quantification of remote exploits vs. writes application attacks. Looking at the security errata for a Linux distribution such as Red Hat or SUSE can do this. A list of operating system vulnerabilities with explanations can be found at www.securityfocus.com. Keep in mind that the severity of the attack and not just the number of attacks is also a key metric. However, when business needs are combined with an understanding of operating system security capabilities functional requirements can be fulfilled, risk reduced and compliance ensured.