Security Implementations of Modern Operating Systems

By

Leonardo J. Stroud

Abstract

This paper discusses the ways in which the Windows Server 2003, Windows Vista, and UNIX operating systems implement security. There are several areas of concern that every operating system has to be aware of and provide a defense against; some of these attacks on security come from the Internet, some from an Intranet, and others come about as a result of system failures or a previously unknown problem in the operating system. Denial of Service (DOS), Buffer overruns, User Authentication, and Certificates among others, are examined and discussed. There is also a section about how hardware failures can affect the security of the system, and how the operating system is designed to react to these types of problems when they occur. The paper also covers some of the problems that can arise with software, such as those that are found in the kernel, memory, and user authentication from a local standpoint (i.e., the user is logged on at the computer). When Windows Server 2003 was released, there were several major improvements in the areas of security for the Internet Information Server (IIS), as well as other areas of the operating system, which are covered in this paper also. In the Windows Vista operating system, there are some controversial new security features, that of Kernel Patch Protection, and User Account Control. These new features are causing many people to be upset with Microsoft, while others welcome them as a way to add more security into the operating system right out of the box; however some see it as a new annoyance to be dealt with. At the end of the paper there is a section that compares these two Windows operating systems with that of UNIX, in the areas of security, and then the paper ends with the conclusions of the author about these operating systems and the direction that security seems to be headed in.

Table of Contents

Abstract 2
Introduction 4
Microsoft Windows Server 2003 5
Security concerns 5
Internet 5
Denial of Service 5
Buffer overrun 5
Certificates 6
User Authentication 7
Hardware 8
System failures 8
Software 8
Trustworthy Computing 8
Common Language Runtime 9
User Authentication 9
Microsoft Windows Vista 12
Kernel Patch Protection 12
User Account Control 14
Comparisons with UNIX 16
Conclusion 16
References 18

Introduction

Security in operating systems has always been an issue for both business and consumers. Now, with the advent of the Internet, there comes a greater opportunity for a person or group to attack computer systems than ever before. Manufacturers of operating systems have responded to these threats by releasing patches and updates as soon as possible after a vulnerability is discovered, but in some cases these have come about too late, with losses amounting to millions of dollars and untold damage in the cases of identity theft.

Security had always been built into Windows and UNIX, but the security features had been felt to be stronger in UNIX than that of Windows. Prior to Windows Server 2003, Microsoft began to implement several new security mechanisms in order to prevent new attacks from succeeding and to remove the risks from earlier vulnerabilities in the operating system, starting with Windows NT and then having several major new improvements in Windows Server 2003. The massive dominance of Windows in both the business and consumer world on desktop computers, lead to most of the attacks focusing on that, and not as many attacks on UNIX, since it was felt to be a more secure system anyway.

However, with the advent of Microsoft into the server market, that created a new area for attackers to test their skills and to cause much more damage than before. Servers usually have more critical and sensitive data than home computers, and so this led to the introduction of all of the new security features in Windows Server 2003 that many faulted Microsoft for leaving out or not making as secure as they should have been.

Microsoft Windows Server 2003

Security concerns

Security concerns in Windows Server 2003 can be split into three groups; that of the Internet, hardware, and software. The Internet is listed as a separate group because of the fact that attacks from there can be on both hardware and software, so it seems fitting that it be dealt with this way.

Internet

Denial of Service

This attack is sometimes abbreviated DoS, not to be confused with MS-DOS, one of the early disk operating systems. It occurs when a server is overloaded with requests from the Internet, thus forcing the server to be shut down or for a technician to restart the server. According to Tanenbaum, “it may cripple a web server by eating up all of its CPU time” (2001). Thus one can see how a DOS attack can be used to disrupt communications between various machines, and cause delays that can sometimes be of a critical nature.

The attacks can come in various ways, such as SYN floods, ICMP floods, and UDP floods. There are also many other types of these DoS attacks, beyond the scope of this paper. The curious reader is advised to do further research on this subject if desired.

Buffer overrun

This type of attack occurs when an attacker enters an URL with some bit of false data after the URL in order to cause the web server to try to execute the code and then to either crash or to allow unauthorized entry into the network by the attacker. Since the amount of the data that an attacker can enter is almost unlimited, it is something that can occur when the program has not been written to check every piece of data that it receives for the proper format and type.

As this type of attack has become more common, programs have been rewritten in order to code “traps” for this, but some programs still are vulnerable to this type of attack, and systems that do not have all of the latest patches installed will also be susceptible to it.

Certificates

There are times when a server will have to interact with a user that allows access to sensitive or confidential data. In order to facilitate this, certificates were invented to authenticate the user and also to allow the user to authenticate the server, thus creating a trusted and secure connection between the two machines by way of the Internet.

If this had not been developed, there would be no way without having to “hardwire” each and every ip address of every computer that could possibly connect to the server that had a legitimate right to do so and access any confidential or sensitive data. By knowing in advance the ip address of every machine, it would be able to develop a list of machines that were allowed to have access.

However, this has many limitations. First, all computers must have a static ip address in order for this to work, and years ago this was not the case. In fact, it is still not common for every computer to have a static ip address for when it connects to the Internet. Second, this would tie every user to that one computer that they had put on the list, so working from any other location would be out, unless the company was told about the new address and it was put on the list. Third, there are times when it is only an occasional use that is needed in order to have a secure connection to a server, so all of this trouble would be soon a costly item for the company and also aggravating for the user. Thus, we have the use of certificates to authenticate the server and to authenticate a client computer.

Certificates are created by means of algorithms that convert the document into a “hashed” document that is very hard to recreate without the proper keys. A thorough discussion of cryptography is beyond the scope of this paper, but it provides a way for documents to be transmitted by way of the Internet in a browser or for secure e-mail.

Windows Server 2003 comes with an updated certificate server built-in, which the administrator can use in order to create self-signed certificates, or can also use to import and utilize certificates from certificate authorities such as VeriSign and Thawte.

User Authentication

Any Internet user that attempts to access any secured areas on a web server that is running on Windows Server 2003 has to be authenticated. This is accomplished in several ways, from the operating system providing a dialog box at the web page that requires a user id and password to be entered before proceeding, to requiring a client certificate to be presented before allowing access, or determining the software restriction policy.

IIS now runs under a new account that operates at lower privileges that the normal System Account. This change immediately improves the security profile of the server if a malicious hacker compromises the service (Mullins, 2003). Incorporating all of these new security features for Internet authentication allows Windows Server 2003 to run more securely than its predecessors.

Hardware

System failures

In the event of a system failure due to a hardware problem, the system can be left in either an unstable state or totally inoperative, depending upon the severity of the hardware failure. In some ways it is better if the system becomes totally inoperative because of a system failure, because then there is no chance that a malicious attacker can compromise the system when it is in such a vulnerable state.

Thus, the main discussion of this section will concern what happens when the system is left in an unstable state, where it is possible that an attacker would have an easier job of gaining unauthorized entry. If the failure has damaged part of the hard disk where critical system files are stored, then if the time comes that the operating system will need to load a program into memory that program will not be found, or a corrupted version possibly be loaded.

If a corrupted version is loaded, then security protections that were in the original program might not be available to completely protect the system as designed, and thus allow an attacker entry into the system. Of course, if the file is destroyed so much that the operating system is unable to even find it or load it into memory, then an error will be generated and the system administrator will be alerted to the fact that something is wrong and take the proper steps to fix the situation.

Software

Trustworthy Computing

Viruses exist and software security is an ongoing challenge. To address these facts Microsoft has made Trustworthy Computing a key initiative for all its products. Trustworthy Computing is a framework for developing devices powered by computers and software that are as secure and trustworthy as the everyday devices and appliances you use at home. While no Trustworthy Computing platform exists today, the basic redesign of Windows Server 2003 is a solid step towards making this vision a reality (What's New in Windows Server 2003 Security, 2003).

Common Language Runtime

The Common Language Runtime software engine is a key element of Windows Server 2003 that improves reliability and helps ensure a safe computing environment. It reduces the number of bugs and security holes caused by common programming mistakes—as a result, there is less vulnerability for attackers to exploit.

The Common Language Runtime verifies that applications can run without error and checks for appropriate security permissions; making sure that code only performs appropriate operations. It does this by checking for things such as: where the code was downloaded or installed from; whether it has a digital signature from a trusted developer; and whether the code has been altered since it was digitally signed (What's New in Windows Server 2003 Security, 2003).

User Authentication

There is always a constant need to authenticate the user who logs onto the workstation or the server. It starts with a user id and password, but in many cases the length or the complexity of either one has not been enforced, or in many businesses, there is no policy defining the requirements for the user id or the password.

That has all changed however, and much of the driving force has been the rise in attacks both from the Internet and from inside the company. Many companies have now created strict guidelines for what a user id and password must be; however there is still some doubt as to whether or not these guidelines are being enforced.

Windows Server 2003 has several features that enabled when a computer is a domain controller. As one can see from the list below, taken from Step-by-Step Guide to Enforcing Strong Password Policies (2004), they require the user to have to fulfill certain conditions for the user id and password:

- **Enforce password history** determines the number of unique new passwords a user must use before an old password can be reused. The value of this setting can be between 0 and 24; if it is set to 0, enforce password history is disabled. For most organizations, this value should be set to 24 passwords.
- **Maximum password age** determines how many days a password can be used before the user is required to change it. The value of this setting can be between 0 and 999; if it is set to 0, passwords never expire. Setting this value too low can cause users to be frustrated; setting it too high or disabling it gives potential hackers more time to determine passwords. For most organizations, this value should be set to 42 days.
- **Minimum password age** determines how many days a new password must be kept before the user can change it. This setting is designed to work with the **Enforce password history** setting so that users cannot quickly reset their passwords the required number of times, and then change back to their old passwords. The value of this setting can be between 0 and 999; if it is set to 0, users can immediately change new passwords. It is recommended that you set this value to 2 days.
- **Minimum password length** determines the minimum number of characters a password can have. Although Windows 2000, Windows XP, and Windows Server 2003 support passwords up to 28 characters in length, the value of this setting can only be between 0 and 14. If it is set to 0, users are allowed to have blank passwords, so you should not use a value of 0. It is recommended that you set this value to 8 characters.
- **Passwords must meet complexity requirements** determine whether password complexity is enforced. If this setting is enabled, user passwords meet the following requirements:
  - The password is at least six characters long.
  - The password contains characters from at least three of the following five categories:
    - English uppercase characters (A - Z)
    - English lowercase characters (a - z)
    - Base 10 digits (0 - 9)
    - Non-alphanumeric (for example: !, $, #, or %)
    - Unicode characters
  - The password does not contain three or more characters from the user's account name.

- If the account name is less than three characters long, this check is not performed because the rate at which passwords would be rejected is too high. When checking against the user's full name, several characters are treated as delimiters that separate the name into individual tokens: commas, periods, dashes/hyphens, underscores, spaces, pound-signs, and tabs. For each token that is three or more characters long, that token is searched for in the password; if it is present, the password change is rejected. For example, the name "Erin M. Hagens" would be split into three tokens: "Erin", "M", and "Hagens". Because the second token is only one character long, it would be ignored. Therefore, this user could not have a password that included either "erin" or "hagens" as a substring anywhere in the password. All of these checks are case-insensitive.
- These complexity requirements are enforced upon password change or creation of new passwords. It is recommended that you enable this setting.

It can be seen from the above that there are many facets to controlling user authentication, and that one of the keys to a secure system is to control the user access both before and after they logon. Many times an attacker can simply gain entry by looking over another person's shoulder as they logon, and if the userid and password is simple, it makes it easy to see what is being keyed in. there is also the manner of checking for the simple userids and password's that one would choose, and then playing a guessing game to see what is correct.

That gives the reasons for the complexity of the userid and password combination, as well as enforcing a time limit before they have to be changed and also setting a policy of not giving either one out to anyone. With policies such as these in place, and strictly enforced, the chances of an attacker gaining entry because of this grows smaller and smaller, which is the goal of the security team in the first place.

Microsoft Windows Vista

Kernel Patch Protection

First of all, what is the kernel? The kernel is the lowest-level, most central part of a computer operating system and one of the first pieces of code to load when the machine starts up. The kernel is what enables the software of the machine to talk to the hardware and is responsible for basic OS housekeeping tasks such as memory management, launching programs and processes, and managing the data on the disk. All applications and even the graphical interface of Windows run on a layer on top of the kernel. The performance, reliability, and security of the entire computer depend on the integrity of the kernel (Field, 2006).

Thus, it can be seen that protection of the kernel is of the highest security level that can be found in a system. People that distribute “malware”, are especially interested in being able to “patch” into the kernel to replace some part of the legitimate code with their own code that then will subject the system to either an unstable state, release sensitive data to the attacker, or turn the system into a “zombie” computer that will be used to further send out spam or other malicious code over the Internet or the Intranet of a company.

There are many brand new security features in Windows Vista, but Kernel Patch Protection is actually not one of them. Kernel Patch Protection was first supported on x64 (AMD64 and Intel EMT64T) CPU architecture versions of Microsoft Windows including Microsoft Windows Server 2003 SP1 and Windows XP Microsoft Windows XP Professional x64 Edition. (Patch protection is currently not supported on x86 or ia64 architectures.) Though, as the use of 64-bit computers is increasing, Windows Vista users will end up benefiting most from this technology. Kernel Patch Protection monitors if key resources used by the kernel or kernel code itself has been modified. If the operating system detects an unauthorized patch of certain data structures or code it will initiate a shut down of the system.

Kernel Patch Protection does not prevent all viruses, rootkits, or other malware from attacking the operating system. It helps prevent one way to attack the system: patching kernel structures and code to manipulate kernel functionality. Protecting the integrity of the kernel is a fundamental step in protecting the entire system from malicious attacks and from inadvertent reliability problems that result from patching.

Kernel Patch Protection may impact compatibility of some legitimate software, on x64 systems, which were built using unsupported kernel patching techniques. Microsoft is sensitive to how application compatibility changes impact our customers and our partners. That is the reason that we have implemented this technology on x64 systems only. As customers adopt the x64 platform, and new native 64-bit software, we have the opportunity to build a more secure and reliable next generation platform that does not facilitate unsupported and unreliable practices such as kernel patching.

We have also been asked to provide a supported way for 'known good' vendors to continue hooking the kernel but prevent others from doing so. Unfortunately, there is no reliable mechanism for us to distinguish between 'known good' software and malicious software. Moreover, we cannot prevent a malicious software author from "bundling" purportedly good software in an attempt to thwart the system. Even if we could include such a mechanism, it's unclear if we could use this mechanism to selectively allow kernel hooking in a manner that provides an acceptable trade off between performance and reliability and security. Furthermore, creating such an exception would greatly hamper the ability to utilize hardware assisted security technology, such as a virtual machine hypervisor, to further improve the integrity of the operating system (Field, 2006).

The proceeding shows that Microsoft has been listening (somewhat) and is making serious attempts (sort of) to protect the core of the operating system more so than in the past, but it remains to be seen if the new implementation of protecting the kernel will work as planned or not.

User Account Control

Another new feature in Windows Vista is the User Account Control (UAC), which is designed to require all users run in standard user mode (User Account Control Overview, 2006). This is an attempt to plug one of the most well known security holes in Windows, which is to allow users to run as the administrator by default. Most of the time this feature is not changed on a computer, so that anything could be done while a user was logged in on a system. This allowed large amounts of malicious code to proliferate throughout businesses and consumer networks.

In the past, users had the ability to do what is known as "Run as", thus allowing a user with lesser security privileges to install programs when needed, as long as the administrator password was known. However, this was an option which history has shown did not always work as planned, thus the new feature of the UAC.

The article from User Account Control Overview (2006) goes on to state, "The goal of User Account Control is to allow users to run Windows with standard user privileges and decrease the number of tasks and applications that require administrator privilege. Any privilege elevation brings a potential risk to the system because the elevated software may be vulnerable to attack. If the user's computer has been exposed to malicious software (malware), the user could be tricked into allowing malicious software to run with administrator privileges when using the UAC consent dialogue or credentials entry. Before approving any request for permission to elevate a program, ensure that up-to-date anti-malware and anti-virus software is running on computer and no malware has been detected.

For the highest protection against code running with administrator privileges, we recommend organizations deploy PCs with standard user accounts and do not provide users access to administrator credentials. Computer administrators are advised to use a standard user account for most tasks, and when needed, log in to a separate administrator account in a separate user session that is only used for administrative tasks.

Administrator Approval Mode reduces the threat of some types of malware attacks by starting programs with standard user privileges by default and alerting the user if a program is attempting to run with administrator privileges. However, this mode does not provide the same level of protection as a standard user account and does not guarantee that the software will not attempt malicious actions once it is elevated.

User Account Control is part of Microsoft's defense-in-depth strategy to provide multiple levels of protection in Windows Vista. Notwithstanding the exceptions noted above, all levels of User Account Control offer greater protection than running a previous version of Windows with full administrator privileges, as most users do today. To further improve the security of Windows-based PCs, Microsoft continues to recommend using up-to-date anti-malware software, using a firewall, and keeping the PC up-to-date with the latest security updates”.

Even with the new features there are still warnings as to what can happen if common, good sense security procedures are not followed, which illustrates the fact that one must be always thinking about security in order to guard against the malicious attacker that is lurking both inside and outside the company.

Comparisons with UNIX

It is difficult in a paper such as this to do a full comparison between Windows Server 2003 and UNIX, because there is only one company that manufacturers Windows while there are a large number of companies that manufacturer their own version of UNIX. Because of this, one has to make broad comparisons of the two operating systems in the area of security, so it will be kept to a few wide areas.

In the area of the user login, Windows keeps the user id and password in a secured file in the system directory, while UNIX had the password file in a directory open to everyone. Windows Server 2003 was designed from the start to incorporate many security features, and UNIX had many security features added in later on in the process.

Also, due to its monopoly on the desktop, Windows has received more attention when there is a problem than UNIX, and now that it has moved into the server market that has followed along. However, as versions of UNIX become more popular with users, that might change and there will be a proliferation of viruses and worms aimed at the UNIX community.

Conclusion

There are many new security features in Windows Server 2003 that have come about as a result of Microsoft learning from previous problems that developed once the operating system was out in the field. No person can foresee the myriad ways in which a system can be attacked, that is both the beauty and curse of human ingenuity. However, by taking some of the lessons learned and looking for ways to prevent future attacks, that is the role of the manufacturer.

Building on this with Windows Vista was one of the major items that Microsoft set out to do, and that result is seen in the new features that have been included in the new operating system. But, once again, it is impossible to build something that another person is not able to break, and so Vista will now become the target of the attackers, with new patches and updates to be expected in the future.

Having shown some of the differences in the ways that UNIX and Windows incorporate their various security features, it can be seen that UNIX provided a model for Microsoft to build on, although in many ways the paths that Microsoft took were not an improvement on what was in UNIX. It was to the advantage of UNIX to be perceived as being more secure than Windows because of the way that the security features were implemented from the start, since that helped to divert attention from the crackers of the world onto Windows and thus thrust Microsoft into the spotlight of having problems with the ability of the operating system to be secure.

References

Field, Scott. (August 11, 2006). An Introduction to Kernel Patch Protection. Retrieved October 20, 2006 from http://blogs.msdn.com/windowsvistasecurity/archive/2006/08/11/695993.aspx

Finnie, Scott. (October 5, 2006). Microsoft Places Its Vista Anti-Piracy Concerns Above User Security. Retrieved October 31, 2006 from http://www.computerworld.com/blogs/node/3665

Hedbom, Hans, Lindskog, Stefan, Axelsson, Stefan, & Jonsson, Erland. A Comparison of the Security of Windows NT and UNIX. (November, 1998). Retrieved November 24, 2006 from http://www.windowsecurity.com/uplarticle/18/nt-vs-unix.pdf

Introducing the Windows Server 2003 Operating Systems. (January 24, 2006). Retrieved October 20, 2006 from http://www.microsoft.com/windowsserver2003/evaluation/overview/family.mspx

Keizer, Gregg. (May 8, 2006). Vista's Security Will Be Pain In The Neck: Analyst. Retrieved October 31, 2006 from http://www.techweb.com/wire/security/187201321

Morimoto, Rand. (December 12, 2003). Integrating Smartcard and Secured Access Technologies in Windows Server 2003. Retrieved October 20, 2006 from http://www.samspublishing.com/articles/article.asp?p=102178&rl=1

Mullins, Michael CCNA, MCP. (October 2003). Get acquainted with Windows Server 2003 security features. Retrieved October 20, 2006 from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2914767,00.html

Schwartz, Mathew. (June 2006). How Vista's Arrival Will Affect the Security Market. Retrieved October 20, 2006 from http://esj.com/security/article.aspx?EditorialsID=1851

Step-by-Step Guide to Enforcing Strong Password Policies. (September 17, 2004). Retrieved November 24, 2006 from http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/directory/activedirectory/stepbystep/strngpw.mspx

Tanenbaum, Andrew S. (2001). *Modern Operating Systems Second Edition*. Upper Saddle River, NJ: Prentice Hall.

User Account Control Overview. (October 2, 2006). Retrieved October 31, 2006 from http://www.microsoft.com/technet/windowsvista/security/uacppr.mspx#EVH

What's New in Windows Server 2003 Security. (January 2003). Retrieved October 20, 2006 from http://www.microsoft.com/windowsserver2003/evaluation/overview/technologies/security.mspx