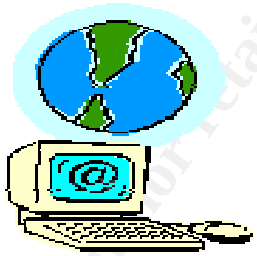
Interested in learning more about security?

SANS Institute InfoSec Reading Room

Can Microsoft .NET Deliver "Trustworthy Computing"?

The aim of this paper was to analyze the security framework of Microsoft .NET, and examine whether its components and features will deliver Microsoft chairman Bill Gates, his ambition of transforming Microsoft into the leading software provider of web services and "trustworthy computing". The initiative to deliver "Trustworthy Computing" is grounded in the strategic decision taken by Microsoft in positioning .NET as being their main platform and software development. Unlike the desktop environments of the 80's and 90's...

Can Microsoft .NET deliver "Trustworthy Computing"?



By: Nikhil Viswanathan

Table of Contents

Summary

The aim of this paper was to analyse the security framework of Microsoft .NET, and examine whether its components and features will deliver Microsoft chairman Bill Gates, his ambition of transforming Microsoft into the leading software provider of web services and "trustworthy computing". The initiative to deliver "Trustworthy Computing" is grounded in the strategic decision taken by Microsoft in positioning .NET as being their main platform and software development. Unlike the desktop environments of the 80's and 90's, .NET will be exposed to malicious threats globally, and weaknesses in the security architecture, will greatly damage its potential for success. Therefore, to mitigate these threats, Microsoft has invested substantial resources and shifted its strategy to ensure that core components such as stability, availability, integrity and privacy are delivered in its products and services. It is felt that by delivering these components, Microsoft will be able to regain the trust of customers and industry, which may have diminished in recent years due to viruses such as the "I love you virus" and "Code Red".

This paper will argue that .NET has the features and functionality that are needed to deliver "trustworthy computing". This functionality is provided by the managed code architecture that is the foundation on which developers will build the applications and services of the future. The managed code architecture provides developers with the tools and features to ensure applications contain secure and stable code even at a development stage. Additionally, .NET will also have security features such which include role based and evidence security. Microsoft will have to continue development of its products including the Passport service, which still poses many security risks. However, with the security initiative to building secure operating systems and applications having commenced, and Microsoft taking bold measures such as retraining staff on IT security issues and implementing strict privacy guidelines, should enable the software giant to strengthen its software credentials and achieve a greater level of trust in the industry and amongst its users.

This paper will also examine the major alternative to .NET, the J2EE (Java 2, Enterprise Edition) offered by Sun Microsystems. The analysis will focus on the similarities and differences between the two platforms, and argue that the Java system does incorporate a strong security architecture and has not had the publicity problems suffered by Microsoft, but will still face strong competition from Microsoft due to the improved security functionality and extensive support base and market coverage offered by Microsoft.

Introduction

The Oxford dictionary describes the phrase "trustworthy" as being the:

"Ability to be relied on as honest, truthful or reliable [1]*"*

In the past few years, the damage caused by cyber attacks to mainly Microsoft products has been so great, that many corporations have been forced to examine whether their Microsoft products can be relied upon, and whether Microsoft can be taken trusted on the issue of security. An indication of the seriousness of the problem was demonstrated in January this year, when Microsoft chairman Bill Gates in an internal memo to his staff, signalled a strategic shift in the approach taken by Microsoft in implementing and delivering security in its products. This strategy stressed the critical importance of ensuring security of all Microsoft products and services, including Microsoft's vision of computing and the Internet of the future, which is represented by the .NET platform[2].

Importance of Security in the Internet environment

In the 80s and 90s the desktop environment dominated the world of computing. Microsoft capitalised on this environment through the various flavours of the Windows operating systems, and subsequently went on to become one of the most powerful and influential corporations in the world. This success was based on its ability to produce software that offered users functionality and user friendly features, such as a graphical interface and multitasking. However, critics of Microsoft often argue that the functionality offered was often delivered at the cost of more critical features such as reliability, stability and security. In the traditional enterprise, these flaws were not seen as a major issue, as the problems could be isolated and controlled within the organisation.

However, the growth of the Internet and global networks increased the exposure of these vulnerabilities to hackers globally. Global viruses such as "Code red" and "Nimda" were specifically targeted against Microsoft products and led to considerable damage and recovery costs for enterprises and corporations globally. Microsoft's initial response to each security vulnerability that was to release a software patch to fix the exploit, after it had been identified. However, as this strategy was reactive in nature, it did not help corporations from incurring significant damage and cost.

---

[1] The Concise Oxford English Dictionary, 10th ed, p1540

[2] Gates, Bill *"Microsoft .NET today"* from http://www.microsoft.com/net/defined/net_today.pdf

The threats posed by the growth of the Internet also demonstrated that Microsoft's strategy in combating cyber threats like viruses was inadequate and obsolete. Bill Gates described the growth of the Internet as encompassing an online world, which would include the provision of a variety of web services such as online data storage, and email. It would also be a world where users had the ability to get online through a variety of platforms and devices[3]. In this environment, the company that was able to provide a framework that could provide platform independent distributed computing could expect to enjoy great success, and potentially dominate the market in a similar method to Microsoft in the late 1980's and 1990's. Therefore, to capitalise on this potential, Microsoft began development of the .NET framework, which it touted as being a language and device independent platform for the provision of web services.

During the planning stages, Microsoft realised that the success of .NET was also linked to the critical issue of security. There was a realisation, that the mistakes of the 1980s and 90s could not be repeated, and software for the provision of online services could not contain vulnerable code, as this would erode the trust customers had in the ability of Microsoft to deliver stable and secure systems. Therefore, a strategic shift in the development of software was initiated, which involved incorporating a security architecture which had the capacity to provide users with products and services that delivered "Trustworthy Computing" or in Bill Gates words "computing that is available, reliable and secure"[4].

This paper will demonstrate that in addition to the components listed above, Microsoft will have to prove that its products and services perform with stability and also adhere to privacy guidelines.

The Microsoft .NET Framework

Microsoft officially announced its .NET platform in June of 2000. The .NET platform represents a major shift in the Microsoft application development environment, offering a transition away from desktop applications to web-based services and it is being positioned as the future platform of Microsoft, and the successor to the Windows generation of operating services and an improvement on the traditional client server relationship. The framework builds on concepts already in place in the Microsoft development context, such as Visual Basic, Active Server Pages (ASP), XML (Extensible Markup Language) and ActiveX Data Objects (ADO) to allow applications to interact with users and other applications, platforms and devices via the Internet. The concept behind the .NET framework is that it is meant to act as a language neutral environment, in which different programs and languages can interoperate[5].

---

[3] Gates, Bill *"Microsoft .NET today"* from http://www.microsoft.com/net/defined/net_today.pdf

[4] Gates, Bill *"Trustworthy Computing"* from, http://zdnet.com.com/2100-1104-817343.html

[5] DrPizza, *"Microsoft. NET"*, from http://arstechnica.com/paedia/n/net/net-1.html

The concepts mentioned above and various new software development tools are contained in Microsoft Visual Studio .NET, which is the new development suite released by Microsoft. This suite is meant to provide developers with the tools needed to create applications and web services (such as Passport services, Hailstorm, MSN, MS Office), for the .NET framework. These applications will be hosted and services on variety of Microsoft servers such as the Windows 2000 family in addition to .NET enterprise servers including specialised servers such as Application Server 2000, SQL server 2000, Mobile information server 2000 and the Internet Security and Acceleration Server 2000 for secure, fast Internet connectivity[6]. From a security perspective, the .NET platform will be an expensive failure for Microsoft if the development tools and the servers supporting them do not provide adequate protection against threats such as viruses, and -hacking[7].

. NET code

Like Java, code written for .NET is compiled into byte codes, which are executed in the Common Language Runtime (CLR) environment, which is similar to a Java Virtual Machine (JVM). The primary difference between the JVM and the CLR is that Microsoft maintains the CLR for the Windows platform only. There are projects underway that aim to provide CLR functionality on other operating systems such as Linux, but these are unsupported by Microsoft. An example of this is a version for Linux that is being developed by a company called Ximian[8]

The CLR concept allows .NET to present a solution for developers outside the traditional Microsoft arena. Unlike Java, which aims to eliminate the platform barrier, .NET aims to eliminate the language barrier by offering a translation layer that will compile several different languages into the Microsoft Intermediate Language (MSIL) of .NET. This means that not only can developers write .NET code in a variety of languages, but also that they can write one application in a variety of different languages. These languages are then translated to the MSIL format, prior to being complied in the CLR environment.

The primary language for development in .NET is C#, which is based on C/C++. C# builds on the technical strengths of C/C++ and also incorporates features from Visual Basic that help in the fast development of applications. C# is publicly available, and has been submitted to the European Computer Manufacturers' Association (ECMA) for approval as a standard. This has given Microsoft the chance to include features in .NET at the development stages, which have only been included in J2EE post development, like XML support. Visual Basic will also be updated for the .NET framework through

---

[6] http://www.microsoft.com/net/products/servers.asp

[7] Pescatore, John *"If Microsoft security fails, .NET fails"*, from http://zdnet.com.com/2100-1107-819752.html

[8] Mackenzie, Kate *"Developers caught in Micros oft's .NET"* from The Australian, Tuesday February 19th, 2002.

several additions and deletions aimed at improving the performance and flexibility of the language.

The Microsoft C# language as well as components such as CLR have been ratified by the communications standards body ECMA. Features of the .NET architecture show great promise for the creation of a highly integrated platform. . NET makes use of XML/SOAP as a messaging protocol, meaning that messaging should be robust and highly reliable (although not necessarily high-performance).

The .NET framework has been the subject of much media confusion since it's release, arguably due to its flexibility and adaptability to a number of different situations. Microsoft has marketed .NET as being the basis for web-based 'services' which will interact with each other to form the basis of billions of online transactions. However, .NET will also be used to provide services to consumers, delivering experiences such as Passport Single Sign In authentication and the Microsoft Network (MSN). It is likely that Microsoft's traditional desktop applications such as Microsoft Office will be migrated to .NET, perhaps even licensed to the user on a subscription basis. Recently, Microsoft released beta versions of its development framework called Visual Studio .NET.

Availability and Stability

As discussed earlier in the paper, one of the key components of IT security is in ensuring that a service or product maintains high availability and stability. Historically, industry feelings have been that Microsoft products were not up to the challenge of being high-availability applications. The Microsoft server family has made significant inroads in recent releases however, with reported uptime for Microsoft Windows 2000 and XP Professional reflecting this in a very positive light[9].

Despite these improvements, industry analysts remain cautious. Forrester research warns that although improvements have been made to the reliability of Microsoft products of late, its track record is too weak to depend on for high availability[10].

During the development of the .NET framework, Microsoft has aimed to address the issue of the availability and stability of its new generation of products, including incorporating the concept of a managed code architecture in the .NET framework. The idea behind the managed code architecture is that it creates an environment in which the compilation of the code is managed, and where tasks such as memory allocation are performed by the compiler, and safe execution of code can occur enabling stability and availability. In addition, the managed code is seen a solution to common software weaknesses such as buffer overflows. The managed code architecture is made up of the

[9] Spanbauer, Scott "Windows XP Inside and Out" from http://www.pcworld.com/features/article/0,aid,63223,00.asp

[10] Johnson, Kyle et al (May 2001) *Making Microsoft Sites Work,* Forrester Research, Cambridge MA

MSIL (Microsoft Intermediate Language), CLR, class libraries, metadata and assemblies.

MSIL and CLR

The MSIL is an intermediate language, which is meant to offer increased compatibility with a variety of platforms. Code that is translated to the MSIL standard offers greater stability in execution than native code.

The CLR can be described as the execution area in which the MSIL code is executed. It is tasked with providing the environment in which the MSIL will execute stably[11]. The stability is achieved by enforcing the security rules and policies of the .NET framework on code while it executes, and as a result offers a greater level of security than could be achieved if the code was run in its native environment. The CLR contains components such as JIT (Just in Time) compilation, and AOT (ahead of Time) compilation, which are designed and tasked with executing code stably. Achieving stable code execution is an important security objective, as it ensures that the code executes in the way the designer intended, and it also prevents hackers from using instable code to disrupt the availability of an application. The CLR also contains features such as NGen (native code generation), (AOT) compilation and storage of code to increase processing speed and efficiency. In addition to the managing the execution of the code, the CLR also has the ability to manage data by controlling the method in which memory is allocated and tasked. The CLR has the ability to run both managed and unmanaged code and data, but by using managed code and data, developers are able to ensure certain conditions and stability in a .NET application are met. The importance of the CLR is that it provides developers with a tool that can be used to ensure that the code they create; will employ the security functionality and features offered by the .NET framework. This allows developers to create applications and services with secure code, and that by default conforms to the security standards set out by Microsoft.

The managed code environment of the .NET framework also contains a new class of assemblies. These assemblies function in a similar manner to an executable file or DLL, but are unique to the .NET framework. The assemblies also contain metadata that the CLR can use to manage data and perform a host of functions such as allocate memory, set runtime context boundaries and enforce security[12].

Buffer Overflows

One of the biggest threats to the stability and availability of a program is the presence of buffer overflows in the code. Overflows occur when the user input exceeds its allocated memory space, and as a result application code is overwritten. Hackers can use this

---

[11] Seltzer, Larry *". .NET code unsafe? . Not exactly",* from http://techupdate.z dnet.com/

[12] *"Security in the Microsoft.NET Framework"*, An analysis by Foundstone, Inc and CORE Se curity Technologies, from http://w ww.foundstone.com/compani es/dotnet.html

vulnerability to overwrite key bits of a users program's data or insert malicious code or data directly into an executing program[13].

In an effort to minimise the frequency of buffer overflows in applications developed in the .NET framework, Microsoft released new compilers for its programs such as Microsoft Visual C++. NET version 7 and Visual Studio. NET. These new compilers were designed to have security features such as the /GS switch and /RTSsu switch which are designed to prevent common software faults like buffer overflows[14]. Despite these features, industry experts still have concerns about the ability of the .NET framework to prevent buffer overflows. Recently "Cigital"a security consultancy firm, released a report, which claimed to discover a security flaw in Visual C++. NET version 7, which still left it vulnerable to buffer overflows[15].

Security and Privacy

The .NET framework was designed to incorporate evidence based security as well as role based security. Evidence based security refers to the allocation of rights and privileges based on the actual situation in the local environment, or within the assembly. The components of evidence-based security include policies, procedure and the verification process.

Policies

The .NET framework includes evidence-based security through the use policies and permissions. The policies in the .NET framework define which resources, code in executing assemblies can access. This prevents software from affecting the integrity and stability of the data. Policies are also installed by default for all users on all machines, and can also be deployed across a domain through a group policy[16].

Permissions

Permissions grant users rights with relation to various resources and function as the tools of policy. In the .NET framework, developers have a greater control over the use and control of permissions when compared to other managed code platforms. For example, in the framework, permissions on objects can be customised to include application defined resources and methods for verifying access rights. Additionally, developers have the ability to insert permissions within assemblies. The danger of excessive permissions is negated by code access security (CAS), which ensures that the code that is being

---

[13] Walker, Joe and Sarah *"Maximum Server Security"*, from http://www .devx.com/premier/mgznarch/Javapro/2001/06jun 01/jsw0106/jsw0106 -1.asp

[14] Seltzer, Larry *". NET code unsafe? Not exactly"* from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2849190,00.html

[15] Ren, Chris Weber, Michael and McGraw, Gary *"Microsoft Compiler Flaw Technical Note"* from http://www .cigital.com/news/mscompil er-tech.html

[16] ibid

executed does not exceed the permissions granted to it. This gives developers the opportunity to create programs which only use the resources that they need to run, and are denied access to resources that they do not need[17].

The Verification process

Another feature aimed at providing stability to the .NET framework is the verification process. This process ensures that the risk of code executing and causing instability in the system or bypassing security defaults is mitigated. This process also ensures that memory is properly allocated to complete a task, and any errors such as buffer overflows are avoided. The verification process allows developers to develop applications and code in a managed environment, where variables such as stability and security can be controlled.

Role based security is used by the .NET framework to perform authentication and authorization to resources. The tools used in this process include the Passport Single Sign in service, Microsoft IIS server, Cryptography and application domains.

Authentication

As described earlier, one of the primary aims of the .NET framework is to provide the architecture for the provision of web services, and furthering e-commerce. The key element in the development of these services is authentication, which is needed to validate customer purchases and provide assurance to vendors and customers that transactions conducted online are valid and legal.

The .NET framework supports a wide variety of authentication methods including cookie authentication, but also contains new functionality in the form of Passport authentication and security features such NTLM and X509 certificates[18]. It also offers developers the opportunity to customise the authentication features in their programs.

Cryptography

In addition to providing support to industry standard cryptography algorithms, the .NET framework extends support to include other cryptography models such as the object inheritance model. Cryptographic models are also made easily available to developers through the design of the managed code environment, which includes features such as CryptoStream, which can be used by developers to access cryptographic algorithms and primitives in the development process. The object inheritance model also gives developers the ability to increase the type and diversity of algorithms available. In

---

[17] *"Security in the Microsoft .NET Framework"*, An analysis by Foundstone, Inc and CORE Security Technologies, from http://www.foundstone.com/companies/dotnet.html

[18] *"Security in the Microsoft .NET Framework"* An Analysis by Foundstone, Inc and CORE Security Technologies from http://www.foundstone.com/companies/dotnet.html

addition, the .NET framework also supports features such as digital certificates, message authentication codes/keyed hash and pseudo-random number generators[19].

Password service

Microsoft introduced the Passport authentication service in 1999. This service was mainly used on the Microsoft owned Hotmail site, and consisted of a single sign in process. However, it was soon expanded to most of the other web sites owned by Microsoft. The single sign-in service is considered to be unique, as it allows users to use authentication on a variety of web sites using the same username and password, and without having to use their own authentication services. The single sign-in service is supported at each Passport site by a Passport manager and a by a central user database, which is owned and administered by Microsoft. When a user attempts to log onto a Passport site, the Passport manager confirms whether the user has a valid entry ticket. If this cannot be established, then they are redirected by the single sign-in service to a Passport server that authenticates them. Once authentication is achieved, they are redirected back to the original Passport site they originally requested. The interaction between the Passport authentication server and the original Passport web site is facilitated by a combination of encrypted cookies and query strings[20].

Many privacy organisations and industry analysts have been critical of the Passport service and questioned its ability to provide privacy and a level of security[21]. In addition, the numerous security incidents faced by the Hotmail service have been seen as evidence of the inability of the Passport service to provide an adequate level of security. The main concerns with the service concern its reliance on protocols such as HTTP, and DNS as well as the MS IIS servers for their usual operation, as these protocols and systems have traditionally been exposed to numerous threats and vulnerabilities. It has been suggested that Microsoft need to base the Passport on more secure protocols and standards such as Ipsec and DNSSEC to strengthen the service[22]. Microsoft has responded by announcing that it intends to incorporate the use of the Kerberos security standard in the Passport service. From a privacy perspective it has also announced that organisations will be able to retain control on their data. However, this may not allay concerns surrounding the ability of hackers in accessing this data and using it for malicious purposes. The Passport

---

[19] ibid

[20] Rauschenberger, Jon *"Secure your web site with Passport"* from http://www.devx.com/premier/mgznarch/vbpj/2001/11nov01/jr0111/jr0111-1.asp

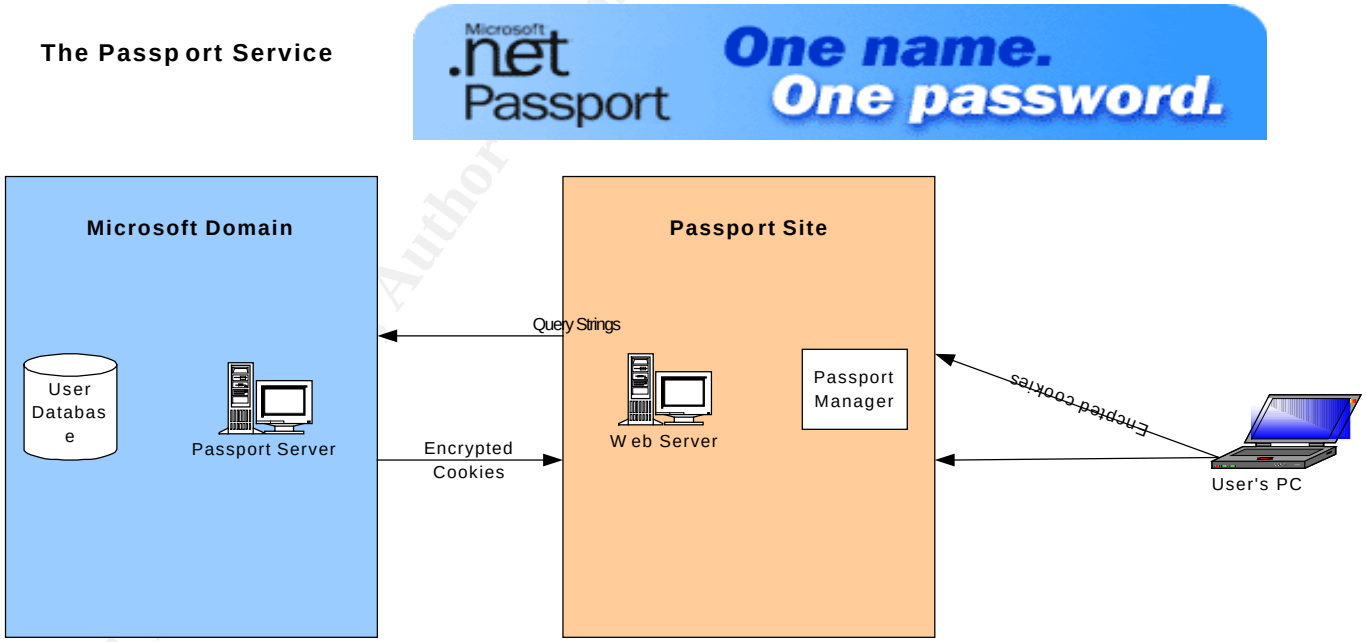
[21] Kanellos, Michael and Wong, Wylie *"Microsoft opens up on Passport"* from http://zdnet.com.com/2100-1106-273252.html

[22] Rubin, Avi and Kormann, Dave *"Risks of the Passport Single Signon Protocol"* from http://informit.com/content/articlex.asp?product_id={16FF6C44-8061-4889-9DDD-E205DEC45671}&element_id={3D605C1E-3023-4DDF-AB41-85AE84EE218A}&session_id={293D47C4-52BC-4A5B-AAE5-BE1F6A2D7A82}&st=E7DEAE6A-9C49-4669-BEA3-8F405E69BA3D

service continues to remain an area Microsoft will have to work hard to gain trust from the industry.

An example of how the Passport service functions is illustrated by the diagram illustrated in the following page:

The Passport Service



3. Passport server uses central user database to access user details, or redirects user to sign in page.

4. Once authentication is completed, user is redirected back to Password site

1. Passport Manager manages single sign on feature - ascertains whether authentication is required

2. If it is required, it then redirects person to a Passport server.

Communication between user, Passport manager and Passport sites occur through the use of encrypted cookies and query strings.

In the future, Microsoft has ambitious plans to strengthen the security of Passport, by including additional functionality such as Kerberos, as well as increased support for Public Key Infrastructure (PKI) and digital certificates.

Ensuring strong privacy is a method of establishing trust. Therefore, in addition to strengthening the security of its products, Microsoft has also been trying to ensure that it operates within the privacy guidelines. However, there is widespread concern in the community related to what Microsoft does with the customer data it collects or would collect from the Passport and .NET services and system. These fears have increased with the Windows XP Product Activation feature, as well as the scope involved with the .NET project. In an effort to allay fears related to the privacy, Microsoft has had to implement strict privacy guidelines to ensure that its staff are compliant with privacy regulations and laws and are aware of how to handle customer data. From a technical perspective, Microsoft has begun implementing privacy tools and guidelines, which meet industry standards. An example of this is the recent incorporation of the World Wide Web Consortium's Platform for Privacy Preferences Project (P3P) standards in the Platform architecture[23].

. NET Servers

In addition to revamping the security of the software tools used to create applications, Microsoft has also worked to harden the security build of the servers it uses to host the applications that have been created for the .NET framework. The servers that will be used in the .NET framework will include the Windows 2000 family, XP Professional and the new generation .NET servers.

One of the main criticisms with Windows NT Server 4.0 was the Internet Information server (IIS) web server that was included in the NT 4.0 server build. IIS was considered to be a very weak web server product, as it did not contain secure codes, and had numerous faulty dll files that were easily manipulated by various viruses such as "Nimda" and "Code red". In an effort to strengthen IIS, Microsoft in the beta 3 version of .NET server has implemented a number of security features with relation to IIS. In beta 3, IIS is not enabled by default, and the administrator has to manually start the IIS service. The .NET web server has also been configured with stability in mind. It can support upto 2GB of RAM and 2 CPU's per server, which should ensure that it provides high processing power and as a result provide high availability.

In the development process for Windows XP, Microsoft's main aim was to provide a platform that provided reliability, through providing improved code protection, increased file protection and side-by-side DLL's[24]. These features were aimed at

---

[23] Perkins, Earl *"Passing Passport"* from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2844846,00.html

[24] Hubley, Mary and Lubrano, Cynthia "Microsoft Windows XP Operating System" from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2852009-1,00.html

ensuring that the XP operating systems did not suffer the same type of system crashes experienced by NT 4.0, and that the notorious Windows "Blue Screen of Death" did not reoccur in the new systems. Despite these efforts, in December 2001 the "plug and play" functionality in XP, was identified as being a security flaw, as it had the potential to allow hackers to take control of a machine through this feature[25]. This incident demonstrates the need for Microsoft to conduct comprehensive testing of its products prior to releasing them to the market. Going forward, Microsoft has hired security consultancies such as Foundstone, and Cigital to provide independent security analysis of its products and also advise them on any security issues. This is a positive measure, and while Microsoft may never be able to eliminate all threats and vulnerabilities to its servers and operating systems, comprehensive testing and the use of independent consultants should ensure that there is a level of trust in the community that Microsoft products have been adequately tested and their capabilities and credentials have been independently verified.

Background on J2EE

The Java language was developed by Sun Microsystems. The goal was to create a language, which would be portable, giving programmers the ability to write code that could run on a variety of platforms. This initiative was encouraged by the growing popularity of the Internet, which resulted in the networking of different platforms and operating systems across the computing world.

Java was officially announced as being a real product in May 1995 – when it was revealed that Java would be incorporated into the next version of the Netscape Navigator web browser. The language itself is a development of C/C++ with the intention of improving the language for development in a modern, networked context.

To achieve it's goal of platform independence, Java employs a relatively unique execution model. Rather than compile code into the native machine language of a platform, Java code is compiled into bytecodes that are executed by a 'virtual' machine. This virtual machine is imported to the various different platforms that Java is compatible with. Therefore, any platform that has a Java Virtual Machine (JVM) is capable of executing Java code. The technology used by Java has developed to an extent whereby the Java code can be executed at almost the same speed as native C / C++ code.[26]

J2EE (Java 2, Enterprise Edition,) is a platform designed to meet enterprise-computing needs using the Java language. It incorporates a number of features from the Java 2

---

[25] Cha, Ariana Enjung "Security Flaws Compromise Windows XP" from http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=&contentId=A10033-2001Dec20&notFound=true

[26] Shiffman, Hank (retrieved August 2001)" *Boosting Java Performance: Native code and JIT Compilers"* . Silicon Graphics Inc.

Standard Edition platform and also adds components crucial to the success of an enterprise platform (such as XML support, Java servlets and JavaServer pages, all of which support the 3-tierd J2EE architecture). Like .NET, J2EE is not just a platform, but can also be described as a framework, which describes how a J2EE system is to be constructed and how its parts are to communicate in a networked environment. The J2EE specification is constantly undergoing development and is open and subject to the public who are allowed to suggest improvements.

A comparison between the two platforms will demonstrate that they share numerous similarities. From a security perspective the main similarity is the presence of a virtual environment (JVM in Java-CLR in .NET), in which the code is complied. This environment allows the respective platform to implement its rules and security features, and ensure certain conditions are met. Faced with a situation where both features share significant commonality, it could be argued that their performance should also be similar in areas such as reliability and security. Detailed testing and statistics on reliability and availability may not be possible until the .NET server operating system is released. If this is confirmed through detailed testing and industry experience, then Microsoft .NET will be able to deliver to its customers "trustworthy computing".

A comparison of the different features of .NET and Java are listed in the following page:

Feature comparison

The table below provides a comparison between some of the key features of J2EE and .NET

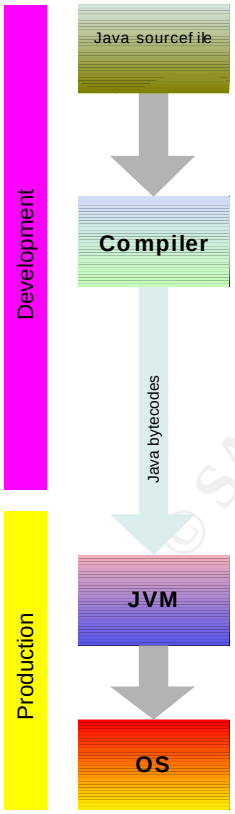
J2EE	. NET	Key differences
Java	C#	Both Java and C# are derived from C/C++ adding functionality such as garbage collection and hierarchical namespaces. Java code can be executed on any platform with a JVM. C# code can be executed in the Windows based CLR, and compilation for other environments is being developed.

J2EE	. NET	Key differences
Java Server Pages (JSP)	Active Server Pages + (ASP+)	JSPs use Java code compiled into Java bytecodes. ASP+ will use Visual Basic, C# and probably code snippets from other languages, compiled into MSIL through the CLR.
JVM, CORBA IDL and ORB	MSIL / CLR	The JVM model allows Java code to be executed on virtually any platform. CORBA support allows code in multiple languages to use a shared set of objects (on any platform with an ORB available). This is not nearly as integrated into the J2EE framework as IL support in .NET. The Common Language Runtime (CLR) environment supports execution of code in multiple languages which are first converted into 'Microsoft intermediate language' (MSIL) format. Objects interact with each other just as if they were all coded in the same language (including features such as inheritance).
Java Swing	Web Forms / Win Forms	Swing is a set of components for defining graphical user interface elements. A generally accepted standard, Swing is supported in many Java IDEs and tools. Like Swing, Win Forms and Web Forms are used to define interface elements and are supported through Visual Basic. NET.
JDBC, EJB, JMS and Java XML Libraries.	ADO+ and SOAP based web services.	JDBC, EJB etc leave data interchange protocols are the discretion of the developer, and operate on top of either RMI or IIOP. ADO+ is based around the premise of XML data interchange (i.e. SOAP) over HTTP.

The two platforms also share similarities in the compilation methods as shown in the following page:

Compilation methods

Compilation Method 1



Compilation Method 2

Faster

The Java strategy involves platform independance. Therefore, a Java application should be able to run irrespective of the platform it is loaded onto.

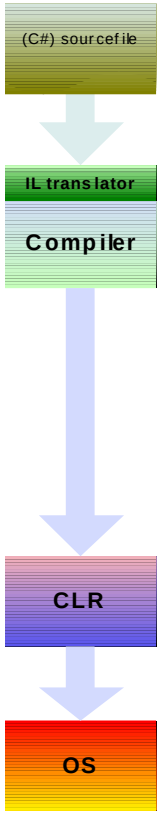
Java Compilation can occur in two ways: -

* The first method, a Java sourcefile is compiled into Java 'bytecodes'. The bytecode is then interpreted by Java Virtual Machine (JVM), and the microprocessor instructions are conducted one at a time.

* The seconf method involves the use of the JIT (just in time) compiler, which helps improve performace in executing and compiling code.

* In both methods, once the code is complied, it can then used by the operating system.

.NET Compilation



The .NET stratgey is based on language independance. Therefore, the type of language used by a developer does not prevent them from creating an application for the .NET framework.

* Like Java, a sourcefile is compiled into bytecodes. The bytecodes are then translated to the MS IL (Microsoft Intermediate Language). This format is preferrable to the native code, as it is designed with security and stability features.

* Once the code is converted into the MSIl language, it is then compiled in the CLR environment, which is simmilar to the JVM. The CLR ensures that the code is converted stabily and is not influenced by malicious code.

* Once the code is compiled it is then available to he operating system

Conclusion

The industry impression of Microsoft was that it did not handle the issue of security with much importance or success. This impression grew over the past few years as a result of the increased incidence of vulnerabilities in the form of viruses and hacking exploits. With the development of the .NET project, Microsoft knew that it had to address the issue of the security of its products and dispel the industry views and impressions that it was not focussed on security. Therefore, it has implement a new initiative, which Bill Gates has described as delivering "Trustworthy Computing" to its customers. The intention behind this initiative is to transform Microsoft into a company than can deliver software, products and services that have a high level of security. The initiative to improving security involves delivering certain security benchmarks including products and services, which deliver high availability, operate stably in most environments, consist of secure code, operate securely, and adhere to industry standards on privacy. To achieve these benchmarks, Microsoft has adopted a variety of practices and guidelines, which are aimed at ensuring that security is incorporated in all of its products, services and deliverables to its customers. The implementation of managed code architecture for the .NET platform is an example of Microsoft tackling the issue of security at the development stage. Additionally, Microsoft has worked to improve features in its previous products, which caused vulnerabilities and instability such as the IIS web server, problems cause by buffer overflows and unstable dll files.

The development of the .NET platform has provided Microsoft with an opportunity to show the industry that it's initiative is not just hype but instead is grounded in reality. It can successfully argue that at all stages in the development of .NET, the issue of security has been considered and where needed implemented, and constructive steps have been taken to strengthen the platforms against known threats, such as viruses, buffer overflows, unstable performance and poor privacy. It would be impossible for Microsoft to guarantee that the .NET framework will deliver faultless and completely secure computing, as irrespective of any security measures it adds to its products, security is not only achieved through the development of tools and features but is also determined by a variety of factors such as effective and diligent administration, monitoring, and enterprise planning. However, it can be argued that the steps taken by Microsoft to strengthen and focus on security during the .NET project; should help it establish an excellent foundation for the delivery of "Trustworthy Computing".

Appendix 1 - Microsoft .NET Glossary

Term	Definition
SOAP	Simple object access protocol – a messaging format in XML which is used in the .NET framework to facilitate communication between applications across different systems and platforms via HTTP.
XML	The eXtensible markup language – a language allowing for flexible definition of data elements. For example, whereas a normal markup language may only contain data, XML would typically also contain information on what the data is and how to use interpret it.
CLR	Common language runtime – the environment in which .NET programs are executed. Programs from a variety of different languages can (theoretically) be compiled to run in the CLR. Microsoft will only support the CLR on Windows platforms, but other companies are working to bring CLR support to alternate platforms such as Linux.
IL	Interpreted / Internal language – this is the code which is compiled for the CLR. An IL translator is to be written for each language to be used in the .NET framework. IL translators already exist for C#.
C#	Pronounced 'c-sharp,' this new object oriented language was developed by Microsoft to be the main language of development in the .NET framework. C# is the language used in the VB.NET development environment. C# builds on C/C++ in the same way that Java did, also borrowing heavily from Visual Basic.
Service	In the context of the .NET framework, a service can be likened to a .NET application. For example, in an e-commerce store, there may be separate services responsible for providing a shopping cart, order confirmation, customer maintenance and shipping. Services are able to interact with each other both internally and externally through the use of UDDI and the SOAP protocol.
COM	Component object model – this framework provides an interface between components on a system. COM is superseded by, although not incompatible with, the .NET framework.

Term	Definition
DCOM	Distributed Component object model, roughly analogous to CORBA, DCOM allows communication between applications across systems. For example, an application may receive a request form a web page, and then retrieve data from a more specialised server on the network using the DCOM interface.
UDDI	Universal discovery, description and integration – a directory that allows a website to have it's XML based web services (.NET 'experiences') listed so that another web service may discover and communicate with them. For example, an application could look for suppliers of a certain product before making it's order every week, a new vendor would be listed in the UDDI with information on how the two applications should interact. This would allow the application to always obtain the best price from suppliers.
ASP	Active Server Page – ASP is actually a scripting language used to generate dynamic HTML pages, based on interactions from the user and/or data from external sources. ASP+ is the ASP implementation in .NET and builds on it's capabilities.
ADO	ActiveX Data Objects – an interface from Microsoft that lets programmers writing Windows applications get access to a relational or nonrelational database from both Microsoft and other database providers. ADO+ is the .NET implementation of ADO and builds on it's capabilities.

Bibliography

- Cha, Ariana Enjung "Security Flaws Compromise Windows XP" from http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=&contentId=A10033-2001Dec20&notFound=true
- DrPizza, *"Microsoft. NET"*, from http://arstechnica.com/paedia/n/net/net-1.html
- Gates, Bill *"Microsoft .NET today"* from http://www.microsoft.com/net/defined/net_today.pdf
- Gates, Bill *"Trustworthy Computing"* from, http://zdnet.com.com/2100-1104-817343.html
- Johnson, Kyle et al (May 2001) *Making Microsoft Sites Work*, Forrester Research, Cambridge MA
- Kanellos, Michael and Wong, Wylie *"Microsoft opens up on Passport"* from http://zdnet.com.com/2100-1106-273252.html
- Hubley, Mary and Lubrano, Cynthia "Microsoft Windows XP Operating System" from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2852009-1,00.html
- Mackenzie, Kate *"Developers caught in Microsoft's .NET"* from The Australian, Tuesday February 19th, 2002.
- Perkins, Earl *"Passing Passport"* from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2844846,00.html
- Pescatore, John *"If Microsoft security fails, .NET fails"*, from http://zdnet.com.com/2100-1107-819752.html
- *"Security in the Microsoft.NET Framework"*, An analysis by Foundstone, Inc and CORE Security Technologies, from http://www.foundstone.com/companies/dotnet.html
- Rauschenberger, Jon *"Secure your web site with Passport"* from http://www.devx.com/premier/mgznarch/vbpj/2001/11nov01/jr0111/jr0111-1.asp
- Ren, Chris Weber, Michael and McGraw, Gary *"Microsoft Compiler Flaw Technical Note"* from http://www.cigital.com/news/mscompiler-tech.html
- Seltzer, Larry *". NET code unsafe? . Not exactly"*, from http://techupdate.zdnet.com/

- Walker, Joe and Sarah *"Maximum Server Security"*, from http://www.devx.com/premier/mgznarch/Javapro/2001/06jun01/jsw0106/jsw0106-1.asp

Last Updated: January 16th, 2018

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS SEC455: SIEM Design Beta One 2018	Arlington, VAUS	Feb 12, 2018 - Feb 13, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS New York City Winter 2018	New York, NYUS	Feb 26, 2018 - Mar 03, 2018	Live Event
CyberThreat Summit 2018	London, GB	Feb 27, 2018 - Feb 28, 2018	Live Event
SANS London March 2018	London, GB	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Secure Singapore 2018	Singapore, SG	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CAUS	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Secure Osaka 2018	Osaka, JP	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Paris March 2018	Paris, FR	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VAUS	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, DE	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TXUS	Mar 19, 2018 - Mar 24, 2018	Live Event
ICS Security Summit & Training 2018	Orlando, FLUS	Mar 19, 2018 - Mar 26, 2018	Live Event
SANS Secure Canberra 2018	Canberra, AU	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Boston Spring 2018	Boston, MAUS	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS 2018	Orlando, FLUS	Apr 03, 2018 - Apr 10, 2018	Live Event
SANS Abu Dhabi 2018	Abu Dhabi, AE	Apr 07, 2018 - Apr 12, 2018	Live Event
Pre-RSA® Conference Training	San Francisco, CAUS	Apr 11, 2018 - Apr 16, 2018	Live Event
SANS Dubai 2018	OnlineAE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced