Robert Skelton | CA04 | 11.16.2014

.NET Security Model

The main security features of .NET are Code Access Security and validation and verification. Code Access Security, or CAS, is used to prevent untrusted code from running in a privileged way, or having too much control. This is set by the system administrator, and is set in terms of different categories for different requests. CAS is based on evidence assigned to specific assembly. The evidence is the source of the assembly, which means the origin (whether it is installed on the local machine or has been downloaded). CAS uses this evidence to determine the permissions granted to the code. This means that code written locally could be assigned to have higher default permissions by the system administrator, where downloaded code will more likely have a lower allowance of permissions. Then again, an inexperienced system administrator might set not restrictions at all, so every program downloaded could have too high permissions, which could be an obvious security risk. Related sections of code can request that calling code be granted a specified permission, but it might not be granted the requested access. The demand causes Common Language Runtime to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if a single assembly is not allowed that permission, a security exception is thrown and the access is denied.

The new security related changes in .NET release 4.5 are Enhanced Strong Naming and a two-tier model for managed applications. This is a public-key digital signature that is used to verify the integrity of data being passed from an originator (signer) to a recipient (verifier). This uses SHA-2. Windows Store apps run in a Windows security container that limits access to resources. Within that container, managed applications run fully trusted. There is nothing a developer can do to elevate privileges. .NET was created as a “better” Java, and in some ways it is. .NET has a cleaner byte code setup and better checked exceptions.

Java Security Model

Java relies on both the compiler and virtual machine for security. The compiler enforces strong typing, scope definitions, static range checking, among other things. The virtual machine verifies bytecode, enforces automatic memory management, and performs dynamic range checking. Importantly, the virtual machine accesses the underlying operating system on behalf of the software it is executing. It does so through an access controller that mediates the actions it can perform. The access controller allows (or prevents) access from the core API to the underlying operating system based on permissions defined in security policies associated with a particular code source. Java runs in a sandboxed virtual machine, so your Java code can access resources from the host computer via the Java virtual machine (JVM). Your Java code cannot access system resources without going directly though the JVM. The JVM in Android is called Dalvik.

Comparison

Comparing the real-life security implications of .NET vs Java, Java seems to have more Zero-day attacks which Oracle is constantly patching, and .NET is more vulnerable to Cross-Site Scripting attacks (XSS). One useful statistic I found on the Gartner blog was this: “The vulnerability density (average flaws per MB of code scanned) for .NET was 27.2 and for Java the overall density was 30.0.” This stat is meant to show that .NET has less exploitable security flaws than Java, on average, per line of code. You could also interpret that to mean that .NET code takes up more space, or is so much wordier that the file size will be affected. Java has a noticeably higher number of major security vulnerabilities reported, but this could also be that Java is more widely used, and is available on non-Windows Operating Systems. Both .NET and Java are middle layers between your chosen programming language and the underlying OS.

Sources

javaworld.com/article/2078860/enterprise-java/why-microsoft--net-failed.html

veracode.com/blog/2010/06/which-tastes-better-for-security-java-or-net

computerweekly.com/news/2240181037/Enterprises-abandon-Java-due-to-security-holes

blogs.gartner.com/neil\_macdonald/2010/06/01/is-net-more-secure-than-java/

cs.odu.edu/~mukka/cs795sum14.net/Lecturenotes/day1/java\_vs\_net.ppt