PREV NEXT

Use Defenses Added by the Compiler

The newer Microsoft compilers add defenses to compiled code. This defensive code is added automatically by the compiler, not by the developer. The major defensive options are the following:

- Buffer security check: */GS*
- Safe exception handling: */SAFESEH*
- Compatibility with Data Execution Prevention: */NXCOMPAT*

Buffer Security Check: */GS*

The */GS* flag is a great example of defensive code added by the compilerthe compiler injects code into the application to help detect some kinds of buffer overruns at run time. The latest Microsoft implementation of this defense in Microsoft Visual Studio 2005it was first available in Visual Studio .NET 2002performs the following steps when compiling native Win32 C/C++ code:

- A random "cookie" is placed on the stack before the return address. The cookie value is checked before the function returns to the caller. If the cookie has changed, the application aborts.
- The compiler rearranges the stack frame so that stack-based buffers are placed in higher memory addresses than other potentially attackable stack-based variables such as function pointers. This process reduces the chance that these other constructs will be overwritten by a buffer overrun.
- Code is added to protect against vulnerable parameters passed into a function. A vulnerable parameter is a pointer, C++ reference, or C structure that contains a pointer, string buffer, or C++ reference.

Best Practices

You must compile all C/C++ code with */GS*.

Safe Exception Handling: */SAFESEH*

The */SAFESEH* linker option adds only safe exceptions to the executable image. It does this by adding extra exception-handler information that is verified by the operating system at run time to make sure the code is calling a valid exception handler and not a hijacked (overwritten) exception handler.

Best Practices

You must link your code with */SAFESEH*.

Compatibility with Data Execution Prevention: */NXCOMPAT*

The */NXCOMPAT* linker option indicates that the executable file was tested to be compatible with the Data Execution Protection (DEP) feature in Microsoft Windows (Microsoft 2005).

Best Practices

You must test your application on a computer that uses a CPU that supports DEP, and you must link your code with */NXCOMPAT*.

The Microsoft Interface Definition Language (MIDL) compiler, used for building remote procedure call (RPC) and Component Object Model (COM) code, also adds stricter argument checking to the compiled code when you use the /robust switch.

As you can see, the extra defenses are cheap because the compiler automatically adds them. Also note that the execution time and code size overhead is tiny. In our analyses, the potential code size or performance degradation is balanced out by better compiler optimizations.

Best Practices

If your compiler does not add extra defenses to the code, you should consider upgrading the compiler to one that does. This is especially true for C/C++ compilers.

Important

Defenses added by a compiler do not fix security bugs; they are added purely as a speed bump to make attackers' work harder. Defenses are no replacement for good-quality code.