

Summary of Microsoft compiler flaw discussions

Source: <http://www.derkeiler.com/Mailing-Lists/NT-Bugtraq/2002-02/0062.html>

From: Russ (Russ.Cooper@RC.ON.CA)

Date: 02/19/02

Date: Tue, 19 Feb 2002 15:41:24 -0500

From: Russ <Russ.Cooper@RC.ON.CA>

To: NTBUGTRAO@LISTSERV.NTBUGTRAO.COM

[Note: URLs are wrapped to more than one line, unwrap them before using]

Chris Ren of Cigital sent a message to SecurityFocus' Bugtraq mailing list;

<http://www.securityfocus.com/archive/1/256234>

in which he stated that Microsoft's latest release of both Visual C++.Net and Visual C++ v7 contained a flaw which made it a "vulnerability seeder". The implication was that code compiled using these compilers and a new switch intended to prevent some buffer overruns, /GS, would be vulnerable to "a very serious set of potential attacks" they otherwise wouldn't be vulnerable to.

Cigital implied that Microsoft touted this new switch as a panacea to thwart all buffer overflows, and Ren stated;

"The protection afforded by the new feature allows developers to continue to use vulnerable string functions such as strcpy() as usual and still be "protected" against some forms of stack smashing."

This assumption is probably derived from the sample code/documentation on the /GS switch provided with the compiler (and available online at MSDN);

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vccore/html/vclrfGSBufferSecurity.asp>

As you'll see, MS by no means suggests using such programming techniques with impunity, the sample is a simple way of demonstrating the potential capability of the /GS switch.

In fact the .NET documentation covers writing secure code extensively including how to avoid buffer overruns through coding.

NT–Bugtraq: Summary of Microsoft compiler flaw discussions

No "flaw" exists in Microsoft's new compiler.

The heart of Cigital's initial claim was;

"However, in its current form, the Microsoft feature leads to a false sense of security because it is easily defeated."

There's no arguing that it's possible for some coders to think that the /GS switch may prevent all buffer overruns, ergo, a false sense of security may be had. I would argue, however, that anyone who believes this is unlikely to be checking their code for any buffer overruns anyway, so at worst they will have added *some* protection to code which is likely insecurely written in the first place.

Unfortunately, seemingly in an effort to justify their position that the switch is "flawed", Cigital went on to state;

"This is a design–level flaw leading to a very serious set of potential attacks against code compiled with the new compiler. The Microsoft compiler is thus in some sense a "vulnerability seeder".

The key word there being "leading", Cigital says that using the switch will lead to a set of attacks only possible against code using the /GS switch. In a paper published on their website (and referenced in the Bugtraq article), Cigital says that attacks can be crafted which attack the "user_handler" variable, a variable used to determine what action is to be taken when a buffer is overrun.

This is true, it would be possible to target the "user_handler" variable, but using methods like those described by Cigital it would be possible to target anything that exists on the stack. The "user_handler" variable may or may not be the ripest target when everything is available, so one can hardly state that it represents a new and additional vulnerability.

Microsoft flatly denies Cigital's assertions, and Cigital's statements support the fact that there are no attacks possible against code compiled with the /GS switch that are not possible against code that has not employed the /GS switch.

Gary McGraw of Cigital stated in a subsequent Bugtraq post;

<http://www.securityfocus.com/archive/1/256643>

"We never made a claim that the use of the flawed /GS feature exposes code to "more attacks" as suggested in a bugtraq post. All we have done is point out that the /GS feature is itself susceptible to attack and should not be relied on to improve software security."

Contradicting himself it would seem. Firstly, Cigital did claim that the /GS switch exposes code to "more attacks", otherwise how could it be

