

Microsoft Celebrates Fifteen Years of Poor Security

Source: <http://www.derkeiler.com/Newsgroups/comp.security.misc/2003-08/0519.html>

From: leslie (*LESLIE_at_JRLVAX.HOUSTON.RR.COM*)

Date: 08/24/03

Date: Sun, 24 Aug 2003 08:10:11 GMT

Microsoft's (lack of) security has cost its users billions of dollars, and given it's track record to-date, it doesn't plan to change...

<http://www.theinquirer.net/?article=11108>

Microsoft celebrates fifteen years of poor security

"Microsoft celebrates fifteen years of poor security
Augmented by the Infernet

By EURuomole: Tuesday 19 August 2003, 11:53

THAT THE Blaster worm should spread as rapidly as it did was testament to one thing only, the poor security in Microsoft's software.

In the first few months of last year Microsoft spent about eight weeks in what was reportedly an intense effort to improve the security of their software. And what a joke that turned out to be, because within a just few months we were seeing security alerts about Microsoft products that had supposedly been thoroughly checked and corrected.

These statements of 2002 were not the first time that Microsoft has declared the problem solved and buffer overflow banished. Back in September 2001 Jim Allchin, a Microsoft vice president, declared that this problem had been stamped out in Windows XP. Supposedly Microsoft had made a complete code review of its operating system and removed all the buffers which could overflow.

Microsoft has had more than 15 years to get it right and it still cannot create a secure operating system. In fact in 2002 Windows had the dubious honour of accounting for 87% of all virus infections reported to the Australian office of the Sophos anti-virus group. This came on top of about 130 vulnerabilities that were reported for Windows during the year 2000, which is an average rate of more than one every three days.

Given this kind of track record from Microsoft I am quite surprised

that in jurisdictions with strong consumer laws there has never been a class action against Microsoft for selling poor quality software. Other operating systems have achieved far better security and have done so since their very early releases, so why is Microsoft unable to?

As for secure operating systems, ask IBM users about the security of their operating systems prior to AIX which itself introduced the usual Unix problems. Or ask OpenVMS users about its security. Its bug list is still in the low double digits after about 30 major and minor versions in its 25 years, which is a sharp contrast to Microsoft's 130 problems in year 2000 alone!

OpenVMS is even more relevant to Microsoft because about 1989 it acquired about 20 software engineers from Digital's cancelled Prism project which was developing an operating system called Mica. These engineers were the designers for Microsoft's NT and borrowed a large number of concepts from OpenVMS, but unfortunately the security concepts were not included. Was it a matter of meeting release deadlines, potential breakage of other code or keeping third party software houses happy? We will probably never know.

Microsoft relies on the users to apply the stream of patches for Windows but many users are unaware of the patches or where to find them, and they are often reluctant to download large patches which can take hours over a dialup line. The frequency can be overwhelming and some users just ignore any problems that do not directly affect them. Microsoft's attitude seems to be so what if the virus mail bombs other users, so long as no damage happens to my system.

And wrapped around all this is the quite reasonable argument that if Microsoft cannot produce secure product releases then its ability to produce secure patches just as suspect.

In recent years Microsoft has had the gall to receive an award for its security from the Department of Defense (perhaps the first award for "lowering the bar" in many years) and another reward for the manner in which it created tools to allow users the ability to automatically patch their software versions. It is simply beyond a joke.

In my opinion, the fundamental problem is that the basic architecture of Windows has two fatal flaws in its memory management and while these remain in the software the ad hoc patches will never be enough to make Windows a secure operating system.

[snip]

Some Solutions

The solution that Microsoft has been trying to apply involves the use of software packages to identify vulnerabilities. It also appear to be

experimenting with different languages, perhaps in the hope of finding one which offers its programmers a better chance of fixing the problem or avoiding it altogether. Both seem to be rather a waste of time and effort when all it really requires is to use correct concepts in the operating system and compilers.

On the matter of memory regions and their protection it is absolutely clear that this technique needs to be applied and done so in a very strict fashion with none of the stupidity of EXECUTE_READWRITE. I can do no better than suggest that Windows and Unix take a good look at how OpenVMS handles these matters because it has the most effective system that I am aware of.

The method used by OpenVMS is one of separating the virtual memory into regions, each with their own protection. At execution time the various program sections (PSECTs) are loaded into one of these regions into orderly and defined areas, applying the protections specified for each PSECT as it does so. Thus data is separated from executable code.

It is similar to the protection offered by Windows XP, which is not surprising since NT arrived on the scene but the important difference is that PSECT protections are set by default and the programmer must explicitly modify them for special circumstances.

Now this introduction of proper memory access controls is all that is required to prevent the introduction and execution of malicious code but it does not solve the problem of an overflowed buffer corrupting the call stack.

A reliable solution to this second problem can probably only come about by altering the manner in which the stack is used.

One possible option is to change the order in which the data items are written to the stack so that the return address pointer and the pointer to the previous call frame are written before the parameters. This means that any buffer overflow on the function arguments would not corrupt these data items but may simply attempt to write beyond the end of the stack.

Another option is to write the parameters to the heap, not the stack and on the stack simply write their data length and their address on the heap. This would enable them to be used and if they were returned to the calling routine, it would be a simple matter to copy the specified number of bytes from the specified heap address into a known location for the calling routine. Buffer overflow on the heap would still quite possible but these would be impossible to eradicate without fundamental changes to several programming languages.

Both options would require changes to compilers so that the stack was used in a way that made it immune to the vexatious problem of buffer overflow corrupting the stack. One would imagine though that these

comp.security.misc: Microsoft Celebrates Fifteen Years of Poor Security

changes and the very small amount of additional processing would be a small price to pay in order to avoid these problems. μ"

Communist China is abandoning Microsoft in its government computers...

<http://asia.cnet.com/newstech/applications/0.39001094.39146335.00.htm>

China blocks foreign software use in gov't

"China blocks foreign software use in gov't

By Staff, CNETAsia

Saturday, August 16 2003 10:49 AM

A new policy by China's governing body the State Council will rule that all ministries have to buy only locally-produced software at the next upgrade cycle.

The move, aimed at breaking the dominance of U.S.-based Microsoft on desktop computers, will eliminate Microsoft's Windows operating system and Office productivity suite from hundreds of thousands of Chinese government computers in a few years' time..."

—Jerry Leslie (my opinions are strictly my own)

Note: leslie@jrlvax.houston.rr.com is invalid for email