

OPINION

FIXING 'THE VIRUS PROBLEM'?

Andrew Lee
ESET, UK



Recently I was asked: 'Will the new security measures in *Windows Vista* fix the virus problem?' After I had recovered my seat, having almost fallen out of it with surprise, I attempted what I hope was a reasonable answer. However, having had more time to think about it, I have decided that the question really does deserve more than

the simple, rather obvious answer 'No'.

A second, also obvious, answer might be that *Microsoft* itself clearly does not believe that the new security controls will solve the problem, as it has invested large buckets of cash in developing its own anti-malware solutions (both anti-virus and anti-spyware). These are bundled with a number of other tools as *Microsoft One Care*, and will ship with *Vista*.

There are two distinct parts to the question that require investigation. Part one is: 'What is the virus problem?', and part two is: 'What are the implications of the new security measures in *Windows Vista*?' A third question then arises: 'Does the answer to part two have an impact on the virus problem as defined in part one?'

WHAT IS THE VIRUS PROBLEM?

'The virus problem' is defined in popular parlance as being that bunch of 'stuff' that 'causes problems on [my] PC'. Occasionally, 'the virus problem' is heard about in the various broadcast and other media. It includes (in the public's perception at least) all categories of malware and a few other things besides.

This fuzzy and wide definition aside, and accepting that it will not ever be possible to satisfactorily divide out the various categories of undesired software neatly and formally (at least not in public), there is a deeper and more fundamental misunderstanding here. Not only do the general public, and even many non-specialist security people, not understand what constitutes malware and countless possible ways in which it can affect systems, but they don't understand the application of various security controls either.

It seems that, in the minds of many, the security measures that exist do so in the main due to, and to solve, 'the virus problem' – they don't, and won't.

Access controls, user authentication, data integrity controls, cryptography, non repudiation, interception detection, service hardening and other security measures have been with us for longer than 'the virus problem', and do very little to address it in any meaningful way for one very good reason – they weren't designed to.

Any sufficiently advanced operating system that is able to 'run' independently developed software programs is susceptible to viruses and malicious exploitation, regardless of the other types of controls (although to what extent, and to what degree of 'usefulness' are separate questions). Where one program can run and, for instance, collect personal information for entry into a database, another can run with the same functionality – the fact that in one case that database may be intended for the use of an attacker has no effect on the function.

The virus problem isn't a software one, nor is it necessarily a security one, it's not even a technology one – the virus problem is, first and foremost, a social problem. Most modern malware does not even fall into the category of what could classically be defined as 'viruses' – indeed the proportion is something less than 20 per cent, even if email worms are included.

So, what is it the attacker wants? Ask yourself this question: is it possible to run services and open ports on the system? If the answer is 'yes' (and it wouldn't be too much use if it were 'no'), then you will be able to control the system remotely with a backdoor program, or subvert an existing program that is listening and serving information. Is the keyboard attached to the machine for the purposes of text entry, including the entry of sensitive personal material? Again, yes, therefore keyloggers will still be a threat. Is it possible to install files onto the system? Yes? Then 'the virus problem' is still with us.

Any operating system that is deployed or used in an unsafe manner is subject to malicious exploitation, either directly by individuals, or by malicious software, including viruses and worms.

WHAT ARE THE NEW SECURITY FEATURES IN WINDOWS VISTA?

A list of the main 'new' security features in *Vista* includes: User Account Control, Consent and Credentials, Code Integrity, Data Encryption, Application Isolation, Data Redirection, Cryptography, Credential Providers, Service Hardening, *Windows Defender* and Rights Management Services [1].

Some of these we can discard, as they have nothing to do with 'the virus problem', and we can focus on the ones that may have an impact: User Account Control, Consent and

Credentials, Code Integrity, Application Isolation, Service Hardening and *Windows Defender*.

USER ACCOUNT CONTROLS (UAC)

This feature (long a staple of *nix and Mac OS operating systems), is not really new; it has been present for a long time within the *NT* family of operating systems. The difference is that now the user is made more aware of it, and what before was the ‘run as’ function is more closely integrated, in that it does not require the user to use it explicitly (assuming they were not running as an administrator anyway). Users will still be able to do important things like entering WEP keys, installing printers and running programs they’ve downloaded from the Internet – the only difference will be that *Windows* will ask them for permission first.

Not allowing full administrative privileges to a standard user (at least by default), if nothing else, finally brings *Microsoft’s Windows* OS to a point where its configuration may alert the more wary user to the problems that are common to undesirable programs. However, it does not take it far beyond that, and it doesn’t solve the real problem – that people simply don’t understand how to distinguish between legitimate actions and ones likely to cause a problem. Tools that will try to do that for them exist already, and they are created and marketed by security professionals who already know that user account controls don’t fix the problem.

CONSENT AND CREDENTIALS

Because of the new way that UAC is implemented, consent will be required for certain operations, in the form of the system requiring the user to input an administrative password to complete the action.

A defining feature of many undesirable programs is that, in order to install themselves, they rely on people with privilege to do something. Here, ‘something’ could include ‘something that you did because a program asked you if you wanted to allow it and you didn’t know what to do, so you clicked “OK” and allowed it’.

The user is often also the administrator (especially in a large percentage of home systems), and forcing him to enter a password when the operating system asks if it can perform a function is not going to solve the problem. This is already the classical problem with some software firewalls, and it will be a problem with *Vista* for the same reason: click first, think later.

What the implementation of UAC and C&C may do is alert users to the fact that something unexpected has happened –

for instance, the fact that a program attempted to install itself at startup – which may cause them to question or prevent the action.

However, this does put the onus on the user, and does require that they know what they are doing. Often inherent in assumptions about user account control is the idea that an administrative user won’t do something stupid. Long ago, in his book *A Short Course on Computer Viruses* [2], Fred Cohen demonstrated that controlling user privilege is no defence because it cannot prevent higher-privileged users from running code that a lower-privileged user could not. Just because you’re root, it doesn’t make you smart about malware.

CODE INTEGRITY

Code integrity in *Vista* means that unsigned drivers are prevented from running in kernel mode, and checks that system binaries have not been tampered with. This, at least, should go some way toward ensuring that system files don’t become infected by viruses, and that kernel mode rootkits have a hard time operating.

While this may improve stability in a system under attack, it does not prevent other files from becoming infected – nor does it prevent user mode rootkits, spyware objects, worms or trojans affecting the system.

APPLICATION ISOLATION

Application isolation assures that each process will run in its own privilege level, and a system called Mandatory Integrity Control (MIC) defines ‘integrity levels’, so that applications running as, say, a standard user, would run at a lower authentication level than a program running with full administrative rights, and should prevent escalation of privileges.

Very exposed programs, for instance *Internet Explorer*, will run at a low authentication level, and will not be allowed to modify users’ data, or any *Windows* binaries (although, this can be adjusted so that it is allowed).

Importantly, an application running at ‘low’ level can only write to areas of the system that are also marked ‘low’. In the case of *Internet Explorer*, this location would be the Temporary Internet Files folder. If (as is now popular with some spyware) a file is running from the Temporary Internet Files location, it should not be able to modify user data – however, if the file is moved from that location (not many users store downloaded files in that location deliberately), it will execute at the level at which the user is running (and potentially the admin level if that user has permitted it, as discussed previously).

Application Isolation is essentially a 'Good Thing', and may close down some of the vulnerabilities that are currently associated with the use of *Internet Explorer*. However, it is possible to adjust the level of integrity, so it is a likely focus of exploit finders to determine a way to do this adjustment covertly. Although it reduces a certain type of risk, it does not solve 'the virus problem' – mainly because, once again, it isn't designed to do so.

One thing that has been apparent in the lifetime of *Internet Explorer* is that vulnerabilities, when they are exploited here, hit hard. Any move toward securing *IE* further is a good one, though it seems to me that doing away with ActiveX would have been far more effective.

SERVICE HARDENING

Service hardening in the *Vista* context means that system services no longer all run with ultimate privilege. (I wonder who else has frequently run *taskmgr.exe* from an 'at' command to gain control over rogue system services, or in order to kill any other service.) Obviously, a scheduler does not need to be running at the highest privilege level, and the service hardening allows developers to assign services different levels of privilege based on their function.

Developers should also be able to 'write-restrict' services, so that they can only have write access to objects that allow it. The biggest problem here is that, while it is possible to harden a service, and thus reduce the impact of vulnerabilities, the effectiveness of service hardening will depend on the developer using this facility correctly.

WINDOWS DEFENDER

Windows Defender is *Microsoft's* anti-spyware program, which it purchased from *Giant* and re-badged. An examination of the effectiveness of this product is beyond the scope of this article. One excellent feature, however, is that *WD* (who else wishes they had called it *Windows Malware Defender* – *WMD*?) does tell the user in good detail every time a program (even a legitimate one) takes certain actions, such as writing to the registry. For the informed user, this is useful information.

What is most interesting, though, is that despite all of the other measures taken in *Vista* to preserve system integrity and reduce the attack surface for malicious exploiters, there is still a need for a standalone (albeit bundled) application which is dedicated exclusively to dealing with undesirable programs. This, more than any other indication, is tantamount to an admission that *Microsoft* does not believe that the new security controls in *Vista* are going to solve 'the virus problem'.

The fact that *Microsoft* is also now firmly in the anti-virus game with its repackaged version of *RAV*, is another tacit recognition of this fact.

SO FAR, SO GOOD ... SO WHAT?

So, what is the impact of the new security features on 'the virus problem'? *Windows Defender* will clearly have some impact – as will user access control. It may also be the case (as it was with *Windows 9x* and *Windows NT*) that, initially, a large tranche of older malware will be rendered useless on the *Vista* platform. Clearly that is a good thing, but history shows us that eventually the bad guys catch up, and soon it's business as usual in the malware creation world.

In recent years there has been a massive trend towards criminal exploitation of malware, and this has meant huge amounts of money being invested in malware development. Just as, in the laboratories of every anti-malware software vendor on the planet, there are many people scurrying around trying to get a product out that will work on *Vista*, there are as many people (maybe even more) out there who have the money to create their own infrastructure and hire malware authors with the express purpose of bringing *Vista* to its knees.

Recently we have seen direct malicious exploitations of zero-day vulnerabilities in *MS Word* and *MS Excel*, and there is no slowdown in the number of vulnerabilities being found. It is almost a certainty that in *Windows Vista* (as in any sufficiently large piece of code) there are vulnerabilities waiting to be found, or perhaps which have already been found, and are now waiting hungrily for a few bytes of exploit code.

If the end result of the laudable new measures in *Windows Vista* is that the user feels, like so many misguided GNU/Linux and Mac OS users, invulnerable to attack from either viruses or the plethora of other undesirable software attacks, particularly ones that employ social engineering techniques, then we will have moved backward rather than forward.

Users of any operating system have a responsibility to educate themselves about the dangers of using their systems, and the realistic possibility that, if they do not, at some point they will fall prey to an attack.

REFERENCES

- [1] <http://www.microsoft.com/technet/technetmag/issues/2006/05/FirstLook/default.aspx>.
- [2] Cohen, F. *A Short Course on Computer Viruses*. Wiley Professional Computing. 1994.