

# Introduction to Windows Integrity Control

Tony Bradley, CISSP-ISSAP 2007-02-02

This article takes a look at the Windows Integrity Control (WIC) capabilities in Windows Vista by examining how it protects objects such as files and folders on Vista computers, the different levels of protection offered, and how administrators can control WIC using the ICACLS command-line tool. WIC is intended to protect a system from malware and user error by helping to establish different levels of trust on objects.

## System integrity - Who can you trust?

When the developers at Microsoft set out to create the latest version of their operating system, Windows Vista, they set out to ensure it was the most secure version of Windows yet. One of the functions that has been built in to Windows Vista which helps to make it more secure is Windows Integrity Control, or WIC.

The purpose of WIC is to protect objects, whether they are files, printers, named pipes, registry keys, and so on from attacks, malware or even innocent user error. The concept of WIC is based on establishing the trustworthiness of the various objects and controlling the interactions between objects based on their integrity, or level of trustworthiness.

The integrity levels of WIC are a mandatory control and override discretionary controls such as NTFS file and folder permissions which most administrators are familiar with. The primary objective of WIC is to ensure that only objects with an integrity level equal to or greater than the target object are allowed to interact with it. Essentially, if an object is less trustworthy, it is prohibited from acting on, or interacting with more trustworthy objects.

Again, WIC trumps normal permissions. That means that even if a file or process has Full Control permissions to another object, if the file or process has a lower integrity level than the object it is trying to interact with WIC will override the permissions and the interaction will be denied.

## Determining trustworthiness using WIC

In order to police the interactions between objects, Windows must first determine the trustworthiness, or integrity level of each object. WIC assigns one of the following six integrity levels to each object:

- **Untrusted** – processes that are logged on anonymously are automatically designated as Untrusted
- **Low** – The Low integrity level is the level used by default for interaction with the Internet. As long as Internet Explorer is run in its default state, Protected Mode, all files and processes associated with it are assigned the Low integrity level. Some folders, such as the Temporary Internet Folder, are also assigned the Low integrity level by default.
- **Medium** – Medium is the context that most objects will run in. Standard users

receive the Medium integrity level, and any object not explicitly designated with a lower or higher integrity level is Medium by default.

- **High** – Administrators are granted the High integrity level. This ensures that Administrators are capable of interacting with and modifying objects assigned Medium or Low integrity levels, but can also act on other objects with a High integrity level, which standard users can not do.
- **System** – As the name implies, the System integrity level is reserved for the system. The Windows kernel and core services are granted the System integrity level. Being even higher than the High integrity level of Administrators protects these core functions from being affected or compromised even by Administrators.
- **Installer** – The Installer integrity level is a special case and is the highest of all integrity levels. By virtue of being equal to or higher than all other WIC integrity levels, objects assigned the Installer integrity level are also able to uninstall all other objects.

In terms of the impact on Windows Vista security, these integrity levels and WIC protect objects from intentional or unintentional modification or deletion by less trusted objects. By setting the Medium integrity level as the default mode for standard users and for all unlabeled objects, Vista protects the majority of objects on the computer from being affected in any way by threats from the Internet, which run at the Low integrity level by default.

Similarly, although Administrators are more powerful than standard users and operate at the High integrity level, the operating system kernel and core functionality receive a higher System integrity level, ensuring that even an absent-minded Administrator or compromised Administrator account can not adversely impact the core system.

To reiterate, the WIC integrity levels and controls are very similar to normal NTFS file and folder permissions. The primary difference is that NTFS permissions are discretionary controls while WIC integrity levels are mandatory controls. Basically, file and folder access privileges and permissions are assigned by the object owner or an administrator, while WIC integrity levels are dictated by the operating system.

While the upper four levels receive little practical use, the differentiation between Low integrity and Medium integrity is where the majority of WIC's functionality lies. Implementing mandatory controls rather than relying only on the discretion of users or administrators certainly provides more security at all levels. But, the ability to segregate files and processes from the Internet and protect the computer from Internet-borne malware is one of the primary reasons for the existence of WIC.

## Protecting Vista from Internet threats

While standard users operate at a Medium integrity level and Administrators are designated as High integrity, WIC assumes that the Internet, and any associated files or processes, are completely untrustworthy and assigns them a Low integrity level by default.

When a user receives an email with a link to a malicious web site (the sort of email they have been told a thousand times to delete), and he clicks on it, the malicious web site may

attempt to install some sort of nasty malware. The malware will typically copy itself to some location on the hard drive and modify Registry keys to ensure its continued existence. It may also try to modify or delete other files or execute processes to initiate other malicious activity.

In Windows XP or older systems, whether or not the malware succeeds is more or less a function of the rights and privileges of the logged in user and whether or not the system and Registry have been hardened or protected in any way to block such attempts. With Vista, because everything related to the Internet runs at a Low integrity level, the malware will be unable to modify, delete or interact with virtually anything else on the system.

These protections will protect Vista systems from the vast majority of malware. Most of the time, users now become compromised or infected by malware through visiting malicious web sites, or opening email file attachments. The same protection does not apply when a user brings in files on CD, DVD, USB drive or other removable media however. These files will execute in the context of the integrity level of the logged in user.

## Using Protected Mode

The automatic designation of Low integrity relies on Internet Explorer running in Protected Mode. Protected Mode has been hyped as one of the significant security updates in Windows Vista and in Internet Explorer 7. As long as Protected Mode is on, everything that Internet Explorer does is assigned Low integrity by default.

Some sites may not function properly with the restrictions imposed by Protected Mode. It is possible, on the Security tab of the Internet Options configuration console, to uncheck the option to 'Enable Protected Mode'. Doing so removes most of the protection Vista provides against unauthorized or malicious activities via the Internet though, so it is highly recommended that you leave Protected Mode on.

In an enterprise, the ability to enable or disable Protected Mode can, and should be, be removed using Group Policy. For individual users, rather than disabling Protected Mode to access sites that have issues, simply add those sites to the Trusted security zone in Internet Explorer. Each security zone in Internet Explorer has its own unique security configuration, and the Trusted zone runs with Protected Mode disabled by default.

## Using ICACLS to view integrity levels

One of the issues that Administrators typically run into when it comes to dealing with rights and permissions in a Windows environment is trying to figure out who has access to what? If a process fails, or a file won't execute, or a user can't write data to a folder, one of the troubleshooting methods might be to examine the WIC integrity level of the object in question and the object it is trying to act on to determine if perhaps WIC is behind the failure.

Windows Vista does not provide anything slick or pretty to let you view or alter the integrity level of an object. There is, however, a command line utility called ICACLS which will display

the contents of the discretionary ACL, as well as mandatory labels. As stated earlier, objects that are not explicitly assigned a label are automatically designated as Medium integrity, however the Medium integrity label won't show up using ICACLS because it is implied and not explicit.

To use the ICACLS utility, you first need to open a command prompt window. There are a number of switches and syntax possibilities to use with the ICACLS tool. You can get information and details on each of the options and examples of their uses by simply typing 'icacls' at the command prompt and hitting Enter. We will focus on two uses of ICACLS here.

First, viewing the integrity level. To view the integrity level, and other contents of the discretionary access list, type `icacls` followed by the path of the object you wish to examine. For example, if you wish to view the mandatory integrity level of the `explorer.exe` file, you would type `icacls c:\windows\explorer.exe`. The results will look like this:

```
C:\windows\explorer.exe NT SERVICE\TrustedInstaller:(F)
                        BUILTIN\Administrators:(RX)
                        NT AUTHORITY\SYSTEM:(RX)
                        BUILTIN\Users:(RX)
```

As mentioned above, the mandatory integrity level assigned to the `explorer.exe` file is implied by the fact that it does not have a specific mandatory integrity level assigned. If there were a mandatory integrity level, there would be an additional entry that would look like this:

```
Mandatory Label\Medium Mandatory Level
```

Just keep in mind that if you are using ICACLS to try and determine the mandatory integrity level being used to determine object interactions in WIC, no mandatory label entry means that it is a Medium by default.

It is also possible to change an object's integrity level using ICACLS. To do so, a user must be assigned the `SeRelabelPrivilege`. In order to change the integrity level of an object, the user needs the authority to "change permissions" as well as "take ownership" of the target object. As long as these privileges are in place, a user may modify or elevate the integrity level of an object. However, the user can never set the object to a higher integrity level than their own.

Assuming you have the proper permissions and privileges, you can modify the mandatory integrity level of an object with the ICACLS tool by typing `icacls /setintegritylevel H|M|L`. The label at the end, either H, M or L, assigns a mandatory integrity level of High, Medium or Low to the specified object respectively.

## Safer and more secure with WIC

When it comes to securing data on a Windows-based computer, one of the most unpredictable and uncontrollable variables is the human component. Organizations have started to realize that users can not be relied upon to properly classify and encrypt sensitive information, so there is a growing trend to implement whole disk encryption, especially on mobile computing devices, and remove that variable.

Windows Integrity Control operates from a similar perspective. Users may have the ability to own files and folders and assign rights and privileges regarding which groups or individuals should be allowed to view, modify, delete, or otherwise perform actions on them. However, the user's discretion can not always be relied upon, therefore discretionary access controls can not be relied upon to protect the objects in question.

With WIC, Microsoft has added the concept of mandatory access controls which are more or less set by the operating system, and trump, or override the discretionary access controls. Segregating all files and processes that originate from the Internet and prohibiting them from interacting with or modifying files or objects on the system help to make the system more secure when surfing the Web.

WIC is not a silver bullet. There are some improvements that could be made, for example providing a better management and configuration tool than the ICACLS command line tool. The security provided by WIC is not perfect, but it is superior to the current Windows security model and helps protect a Vista system from many threats which could impact Windows XP, Windows 2000 or older operating systems.

## Further reading

Mandatory Integrity Control in Windows Vista:

<http://blogs.technet.com/steriley/archive/2006/07/21/442870.aspx>

First Look: New Security Features in Windows Vista:

<http://www.microsoft.com/technet/technetmag/issues/2006/05/FirstLook/default.aspx>

Protected Mode in Vista IE7:

<http://blogs.msdn.com/ie/archive/2006/02/09/528963.aspx>

Why Vista? Mandatory Integrity Control (MIC) (Security, Stability, System Integrity):

[http://adopenstatic.com/cs/blogs/ken/archive/2006/08/18/Why-Vista\\_3F00\\_-Mandatory-Integrity-Control-\\_2800\\_MIC\\_2900\\_-\\_2800\\_Security\\_2C00\\_-Stability\\_2C00\\_-System-Integrity\\_2900\\_.aspx](http://adopenstatic.com/cs/blogs/ken/archive/2006/08/18/Why-Vista_3F00_-Mandatory-Integrity-Control-_2800_MIC_2900_-_2800_Security_2C00_-Stability_2C00_-System-Integrity_2900_.aspx)

## Reprints or translations

Reprint or translation requests require [prior approval](#) from SecurityFocus.

© 2007 SecurityFocus

## Comments?

Public comments for Infocus technical articles, as shown below, require technical merit to be published. General comments, article suggestions and feedback are encouraged but should be sent to the [editorial team](#) instead.

[Privacy Statement](#)

Copyright 2006, SecurityFocus