+ Virtual

Print Subscriptions

Michael Horowitz

Defensive Computing

More posts | Read bio

January 30, 2009 - 3:50 P.M.

The best way to disable Autorun for protection from infected USB flash drives

6 comments

- TAGS:Autoplay, autorun, Microsoft, security, Windows
- IT TOPICS:Cybercrime & Hacking, Security Hardware & Software, Software, Windows & Microsoft

My first posting on the topic of Autorun/Autoplay, Test your defenses against malicious USB flash drives, described three ways that bad guys trick people into running malicious software that resides on an infected USB flash drive.

Although, at times, flash drive resident software can run by itself (think U3 flash drives) without any action on the part of the computer user (other than inserting the USB flash drive) the more normal case is that a person has to be tricked. My first posting on this topic has a sample autorun.inf file that safely illustrates three of these tricks, and you can download this file to test how well your PC is defended from inadvertently running software off the flash drive. Odds are that any Windows computer will be susceptible to at least one of the tricks.

Have no fear though, after making the simple change described below the only way to get infected by an infected USB flash drive is to manually seek out and run the malware (run.me.to.see.naked.pix.of.some.celebrity.exe). You won't be tricked into running it.

DEFINING AUTORUN AND AUTOPLAY

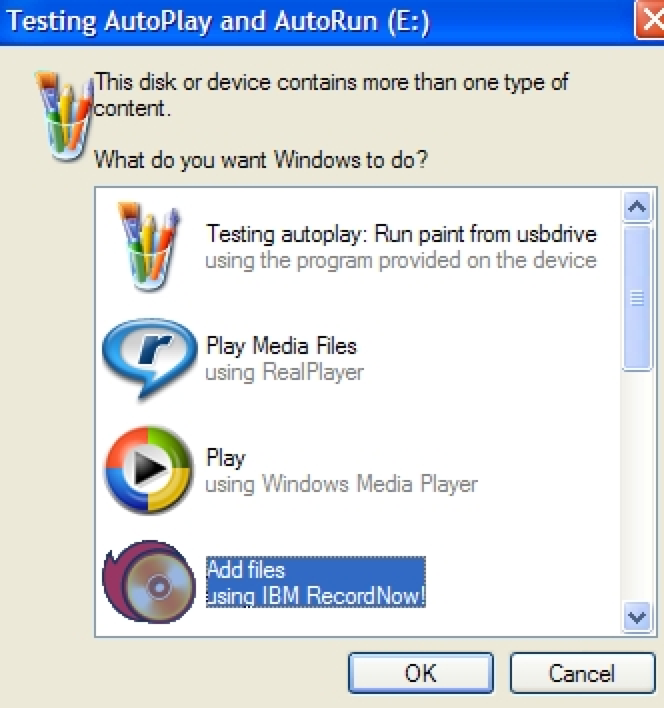
In my previous posting I griped about the terms Autorun and Autoplay. Microsoft uses these terms to mean different things at different times.

The classic/legacy definiton of Autorun is described by Dan McCloy as:

> AutoRun is the functionality that enables a CD-ROM drive or a fixed drive to specify a program or document to be started immediately upon the connection of the drive. Autorun has been designed not to work on removable drives such as USB flash drives, as these drives are much more readily infected and passed around to other computers.

Autorun was introduced back in Windows 95 and the living embodiment of it, is a configuration file in the root directory of removable media (CDs, USB flash drives, external hard drives, etc.) called autorun.inf. If you've ever inserted a CD into a Windows computer and had a program run automatically, autorun was behind it. My first posting on this subject offered a sample autorun.inf file.

Autoplay was introduced with Windows XP. Whereas autorun literally ran a program automatically, Autoplay puts the computer user in charge. A sample Autoplay window is shown below, no doubt you've seen it many times.



Testing AutoPlay and AutoRun (E:)

This disk or device contains more than one type of content.

What do you want Windows to do?

Testing autoplay: Run paint from usbdrive
using the program provided on the device

Play Media Files
using RealPlayer

Play
using Windows Media Player

Add files
using IBM RecordNow!

A sample Autoplay window

When Autoplay is enabled, Windows examines a newly inserted thingy (removable media) and offers a choice of programs to process the various types of files it finds. Autoplay looks for an autorun.inf file too. As we saw in my initial posting, the first Autoplay option can be controlled by directives in the autorun.inf file (this explains the first option in the screen shot above). However, Autoplay does not need or depend on the autorun.inf file (Autorun does).

Autoplay was a misnomer from the get-go, the only thing that happens automatically is the presentation of the Autoplay window. A better name would have been AutoDetect as the automatic processing is limited to detecting file types. No playing happens automatically.

TWO APPROACHES

There are two approaches to defending your computer from the Autorun and Autoplay tricks that I described in my first posting on the subject.

One approach, offered by Microsoft (which invented the problem in the first place) is to disable stuff.* This approach is, in my opinion, a complicated mess involving multiple registry keys, different behavior in different versions of Windows, different treatment for different types of devices, bugs in Windows, poor documentation, missing patches, yada yada yada. Life is too short.

Fortunately, two lone techies, Nick Brown and Emin Atac came up with a better solution that directly targets the source of the problem, the autorun.inf file. It deals with the disease rather than the symptoms. In fact, if you employ this solution, you can leave Autoplay (as defined above) enabled and let it, perhaps, serve a useful function.

What the Brown/Atac solution does is tell Windows to ignore any and all autorun.inf files. Period.

I tested Microsoft's approach with Windows XP SP2 and SP3 and will go into the details of where it works and where it falls down in the next posting. It *can* work, but, only if you twist like a pretzel when the moon is shining on a Tuesday. And, it is **not a comprehensive fix**, loopholes, such as U3 devices remain open. Nick Brown's approach, on the other hand, is ironclad, it always works, no IFs, ANDs or BUTs.

Just a few days ago, Dan Goodin in The Register wrote Disabling Windows Autorun - there's a right way and a wrong way. He was referring to two variations of the Microsoft approach. What Mr. Goodin considers the right way, I consider the wrong way.

To illustrate my point, the Microsoft Knowledge Base article that Goodin cites as the right way (How to correct "disable Autorun registry key" enforcement in Windows), offers no solution for Windows XP Home Edition users. Way to go Microsoft. In contrast, Nick Brown's solution does protect users of the Home Edition of Windows XP.

Goodin's article also illustrates the language problem that Microsoft has dumped on the world, forcing us to use two words to describe five different things. He wrote that Autorun is turned on by default, which is, strictly speaking, not true. For example, neither Windows XP SP2 nor SP3 automatically run a program on a USB flash drive.

IMPLEMENTING NICK BROWN'S SOLUTION

Nick Brown's solution requires updating the registry. Although the update can be removed, it can't hurt to make a full backup of the registry first. Windows XP and Vista users can do this using System Restore.

In XP: Start -> Programs -> Accessories -> System Tools -> System Restore. Click on the radio button to "Create a restore point" -> enter any description (a good example might be "before zapping the registry to disable autrun") then click on the Create button. It should take less than 10 seconds.

In Vista: Start -> enter "system restore" in the search box -> click the link to "Open system protection" -> this opens a new System Properties window -> click the Create button at the bottom -> enter any description (a good example might be "before zapping the registry to disable autrun") then click the Create button (not the same Create button as before, the new one).

Zapping the registry is simple, all you need is the three lines shown below in a .reg file. Then double click on the file.

You can either copy the three lines below from this web page or download the file using the link at the bottom of this posting. The file name is not important, other than it should end with ".reg". Computerworld does not allow attaching files ending with ".reg" to a blog posting, so the file type is ".txt" and you'll have to rename it to end with ".reg".

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\IniFileMapping\Autorun.inf]
@="@SYS:DoesNotExist"
```

Note that there are three lines in the file, the middle line may wrap when displayed by a web browser, but it needs to be a single line in the .reg file.

Nick Brown explains what this does:

> This hack tells Windows to treat AUTORUN.INF as if it were a configuration file from a pre-Windows 95 application ... it says "whenever you have to handle a file called AUTORUN.INF, don't use the values from the file. You'll find alternative values at HKEY_LOCAL_MACHINE\SOFTWARE\DoesNotExist." And since that key, er, does not exist, it's as if AUTORUN.INF is completely empty, and so nothing autoruns, and nothing is added to the Explorer double-click action. Result: worms cannot get in - unless you start double-clicking executables to see what they do ...

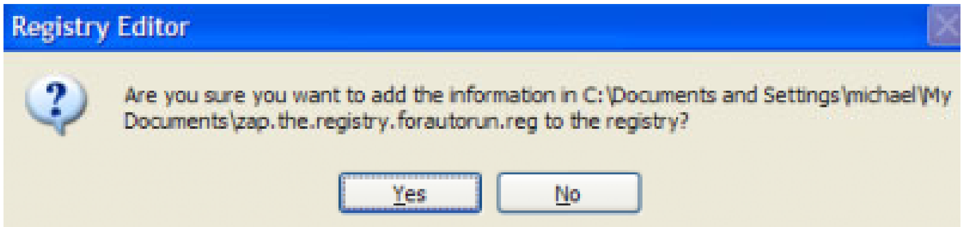
The text "DoesNotExist" in the third line is meant to be a place in the registry that does not exist. If this zap gets very popular, malware may look for it, so it can't hurt to change it just a bit. For example, I might use something like

```
   @="@SYS:DoesNotExistMichael"
```

on my computers. To be clear, this is not at all required. The zap as shown above works fine.

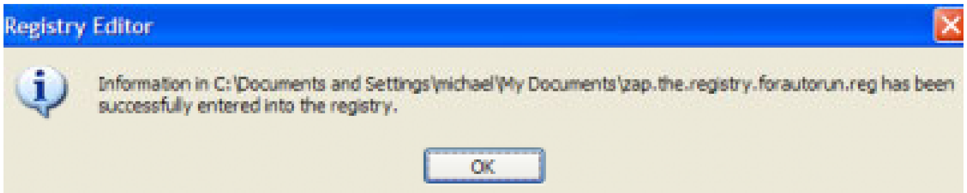
Once you have a dot reg file on your computer, just double click on it.

On a Windows XP computer, you'll see the following warning before the registry zap is actually made ...



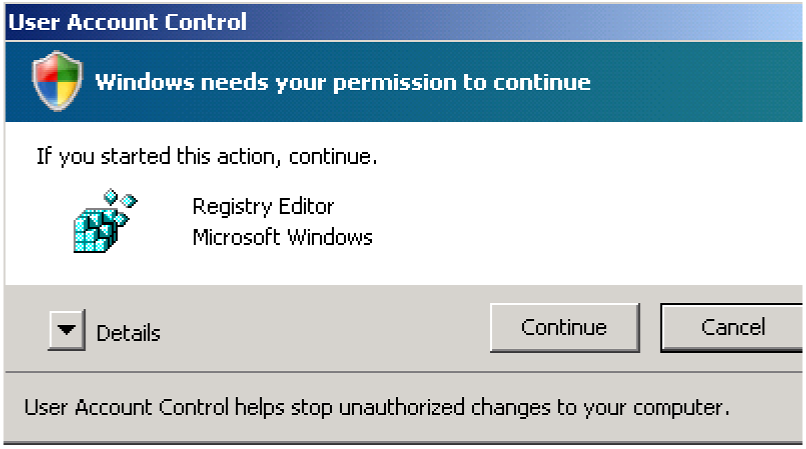
Are you sure you want to add the information in C:\Documents and Settings\michael\My Documents\zap.the.registry.forautorun.reg to the registry?

and the following confirmation, after it was made.**



Information in C:\Documents and Settings\michael\My Documents\zap.the.registry.forautorun.reg has been successfully entered into the registry.

On a Vista machine, you'll first have to deal with UAC as shown below.



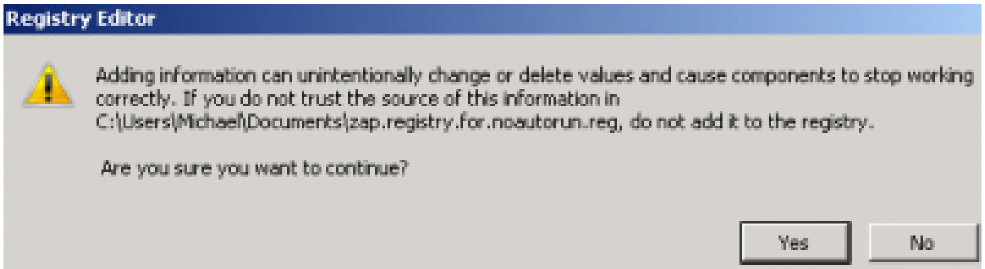
Windows needs your permission to continue

If you started this action, continue.

Registry Editor
Microsoft Windows

User Account Control helps stop unauthorized changes to your computer.

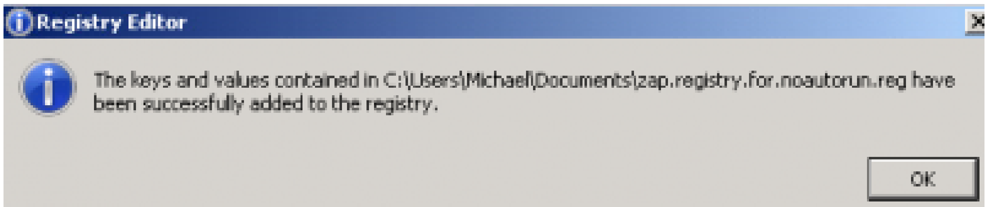
Then you'll see the following warning before the zap is actually made ...



Adding information can unintentionally change or delete values and cause components to stop working correctly. If you do not trust the source of this information in C:\Users\Michael\Documents\zap.registry.for.noautorun.reg, do not add it to the registry.

Are you sure you want to continue?

and the following confirmation afterwards.



The keys and values contained in C:\Users\Michael\Documents\zap.registry.for.noautorun.reg have been successfully added to the registry.

I've read that you may have to restart Windows for this registry zap to fully take effect. A reboot wasn't needed in my testing on XP SP2, XP SP3 and Vista machines.

A USB flash drive inserted in an XP machine when the zap was made, lives by the old rules until it is ejected, at which point the new sheriff takes control.

NEW SHERIFF IN TOWN

As Nick Brown explained, his zap tells Windows *never* to process an autorun.inf file. It has no effect on the Autoplay feature of Windows. If Autoplay was being invoked before the registry zap, it will still be invoked afterwards. If Autoplay was not in effect before the zap, it will remain off afterwards.

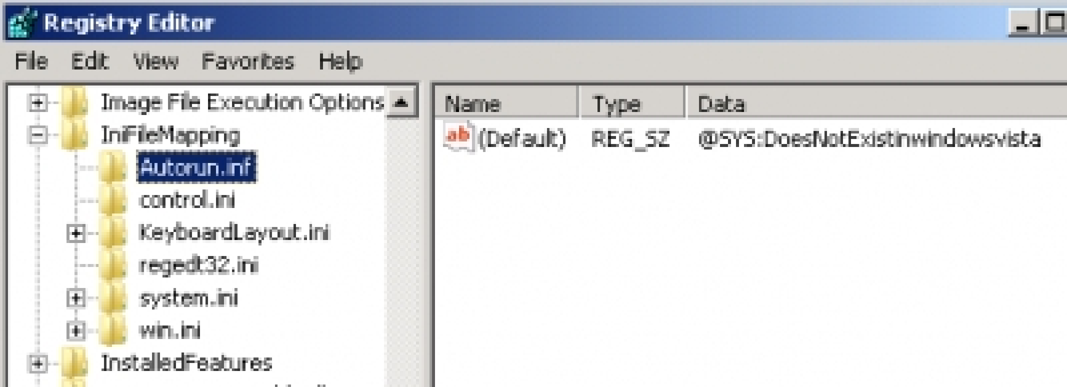
The zap will, however, disable the ability of an autorun.inf file to create an entry in the Autoplay window (for more on this, including examples, see my first posting on this subject, Test your defenses against malicious USB flash drives).

After the zap, double-clicking on **any** drive icon will *always* list the files/folders in Windows Explorer. This is true for USB flash drives, CDs, external hard drives, network shares, etc. The zap also insures that autorun.inf files can't modify the context menu any more. Again, you can see and test these changes using the sample autorun.inf file from my first posting.

Perhaps the most obvious change the zap makes is to the drive icon and name. My test autorun.inf file changes the drive name to "Testing AutoRun Stuff". After applying Nick Brown's registry zap, a test USB flash drive appears as "Removable Disk (x)", where x is the drive letter. My test file also makes the icon for Paint, the drive icon. After this zap, the drive icon appears as a normal Windows drive icon.

These simple changes serve as a great "tester". You can use them, in conjunction with my test autorun.inf file, to verify that Nick Brown's zap has been applied.

The only other way to know whether it has been applied is to look into the registry. If you do, you should see something like the below.



Name	Type	Data
(Default)	REG_SZ	@SYS:DoesNotExistinwindowsvista

On the other hand, it does no damage to make this registry change on a computer where it has already been made.

This registry zap does not interfere with booting the computer from a CD or from a bootable USB flash drive. The zap only comes into play after Windows is up and running.

RESTORING THE OLD SHERIFF

Should you ever want to undo this registry zap, Nick Brown suggests:

> open REGEDIT and navigate down to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping. There will be a subkey ("folder") called Autorun.inf. Delete it.

Regedit is the Windows registry editor. To run it in XP, click Start -> Run -> type "regedit" without the quotes -> then click the OK button. To run it in Vista, click Start and type "regedit" into the search box.

The look and feel of browsing the registry with regedit is very much like browsing files and folders with Windows Explorer. You delete a subkey in the registry by right clicking on it and selecting Delete from the popup menu.

After deleting the autorun.inf subkey from the registry, you have to reboot (at least in Windows XP) for the change to take effect.

As with making the zap initially, it's a good idea to backup the registry by creating a Restore Point beforehand.

FURTHER READING

For more on this see Microsoft Windows Does Not Disable AutoRun Properly (a.k.a. Technical Cyber Security Alert TA09-020A) from the U.S. Computer Emergency Readiness Team (US-CERT). It discusses the Nick Brown zap and some flaws in the Microsoft approach.

Another proponent of Nick Browns approach was Scott Dunn who wrote One quick trick prevents AutoRun attacks in November 2007, shortly after Nick Brown made his discovery public.

I can't help but note that two lone techies (Nick Brown and Emin Atac) came up with a great solution while Microsoft keeps twisting the problem like spaghetti, as if the whole thing was designed for job security.

I'll list all my gripes with the Microsoft approach in an upcoming posting.

Just for the record, I don't know Nick Brown. I have never spoken to him, emailed him or even seen his picture. But, I do know a good idea when I see it.

Update February 3, 2009: In the next posting on this subject, I test Nick Brown's registry zap on Vista SP1.

*I used the term "stuff" because Microsoft uses "autorun" and "autoplay" to mean any of five different things. Thus, most every usage of these terms is inaccurate. At least with "stuff" you know the term is fuzzy in its meaning.

**The text in the images is a bit fuzzy because they have to be scaled down to fit within the confines of this web page layout.

Attachment	Size
zap.the_.registry.to_.disable.autorun.inf_.files_.txt	125 bytes

Reply

Print

Email this

Digg this

Slashdot this

What People Are Saying

Add new comment

Why this long, complicated process....

Submitted by Anonymous on February 2, 2009 - 1:47 P.M.

of editing the registry when TweakUI from M$ does it very easily and simply?

NONE of my computers "autorun" anything! I want full control over what runs and when...not when M$ says so. When I build a new machine for myself, the first thing I do is install TweakUI and DISABLE autorun for *ALL* drive-letters.

Go here:

http://download.microsoft.com/download/f/c/a/fca6767b-9ed9-45a6-b352-839afb2a2679/TweakUiPowertoySetup.exe

Reply | Report this comment

more to come

Submitted by Michael Horowitz on February 2, 2009 - 5:21 P.M.

The TweakUI solution is not as ironclad as the solution presented here. It is vulnerable to USB devices that pretend to be CDs, such as U3. It is vulnerable to user mis-configuration on Vista. It's also more complicated and has been buggy. And more, that I'll write up soon.

You may want to test your machines as per my first posting to insure they're patched such that the MS TweakUI fix is working as expected.

Reply | Report this comment

Insanity

Submitted by Ron on February 1, 2009 - 10:58 P.M.

Editing the registry to disable autorun?

I really miss Windows . . . NOT!

Reply | Report this comment

editing the registry isnt too bad in this case

Submitted by Michael Horowitz on February 2, 2009 - 12:27 P.M.

In this case, editing the registry isn't too bad. All that's needed is downloading a very small file and double clicking on it.

In general though, I agree with you that editing the registry is not a great approach to problem solving. It's a sign of poor system design, as in this case.

Reply | Report this comment

Thanks!

Submitted by Nick Brown on February 1, 2009 - 3:40 P.M.

Michael, big thanks for this article. I hope that the step-by-step instructions, screen shots, etc will encourage more people to use our "zap" as you call it. (Emin conceived the idea - "wouldn't it be great if you could tell Windows to ignore the Autorun.inf file" - and I implemented it. Total time: 15 minutes.)

I particularly like your comment about treating the disease rather than the symptoms. But hey, treating the symptoms is the anti-virus industry's business model. :-)

Nick

Reply | Report this comment

more to come

Submitted by Michael Horowitz on February 2, 2009 - 12:31 P.M.

Nick: I'm working on documenting the flaws in Microsoft's approach. U3 devices are only one of a number of holes in their scheme.

Reply | Report this comment

Related Posts

Autorun and Autoplay: screwed by terminology

Michael Horowitz

Defensive Computing

More posts | Read bio

January 24, 2009 - 6:24 P.M.

Test your defenses against malicious USB flash drives

5 comments

- TAGS:autorun, Internet security, Microsoft, security, Windows
- IT TOPICS:Cybercrime & Hacking, Security, Security Hardware & Software, Software, Windows & Microsoft

The latest malicious software to spread to untold millions of computers goes by the names Downadup and Conficker.
Computerworld's Gregg Keizer calls its spread the "biggest attack in years". One way the software spreads is by infecting USB flash drives (a.k.a. thumb drives, pen drives, flash drives, memory sticks, etc).

This is certainly not the first malicious worm to spread by infecting flash drives.

A couple months ago, the Department of Defense dealt with a variant of the SillyFDC worm known as Agent.btz by banning the use of USB flash drives on government computers. In September 2008, a computer on board the International Space Station was infected with malicious software that spread via a flash drive. In December 2007, Randy Abrams at ESET, the company behind the NOD32 antivirus program, wrote that *"Trojans using autorun to infect computers have been one of the most prevalent threats that we have been seeing for several months now."* And I'll never forget this 2006 story, Social Engineering, the USB Way, about how a company was infected by malicious thumb drives dropped in the parking lot outside their office.

Here I'll show the tricks used by malicious software on USB flash drives and provide a safe sample file that can be used to test how well a computer is defended from the tricks that the bad guys use.

Autorun.inf is the key

An infected USB flash drive contains the malicious software paired with a malicious autorun.inf file. The autorun.inf file is used to trick the user into running the malware on the flash drive.

When a flash drive is inserted in a Windows computer, the operating system looks in the root directory for an autorun.inf file, and takes a number of actions, demonstrated below, based on the contents of the file.

The most dangerous action, of course, is running a program and there are four different mechanisms (that I know of) for running programs that reside on USB flash drives. The two most popular approaches are called AutoRun and AutoPlay. But, the terms can be vague, so I'll avoid using them as much as possible. In How to correct "disable Autorun registry key" enforcement in Windows Microsoft goes so far as to say *"Autorun is also known as AutoPlay"*.

Interestingly, the most dangerous of the three approaches is often overlooked. I myself, overlooked it back in March 2008 when I blogged about turning off autorun at CNET. In the worst case, the end user (you) has no visual clue that a program was run.

To get a bit ahead of myself, this happens when you double click on the drive letter for the USB flash drive in My Computer. All Windows users know that this brings up a list of the files and folders on the flash drive. But, a malicious AutoRun.inf file can tell Windows to run a program instead of listing the files/folders.

Below is a sample autorun.inf file. It safely illustrates the things that an autorun.inf file can control, including how to run a program rather than list files when the drive letter is double-clicked on.

I modeled this off the examples found in USB Drive AutoRun.inf Tweaking over a Daily Cup of Tech. You can copy/paste the text below or download the file directly using a link at the bottom of this posting.

This AutoRun.inf is designed to run a copy of the age-old Paint program, a copy that resides on a USB flash drive. If your computer is well defended, it can't run Paint. My guess is that the vast majority of you will find that Paint does, in fact, run. A Windows computer that lets this autorun.inf file execute Paint, is an accident waiting to happen.

On both Windows XP and Vista, Paint is file mspaint.exe and it resides in the C:\windows\system32 folder. Copy it to the root directory of a USB flash drive along with the sample autorun.inf file below. Other files can reside on the flash drive too, they are irrelevant to this testing.

```
[autorun]
;--------------------------------------------------
; Test your defenses against infected a USB flash drives
;
; Note: Lines that start with a semicolon are comments
;   and are ignored by Windows
;
; Place this in a file in the root directory of a USB flash drive
; The file name must be autorun.inf
; Also in the root folder, place a copy of Paint (mspaint.exe)
;
; Created by Michael Horowitz January 2009
;--------------------------------------------------
; This shows up in the first line of the Autoplay menu
action=Testing autoplay: Run paint from usbdrive

; This causes the AutoPlay window to run Paint from the flash drive
open=mspaint.exe
```

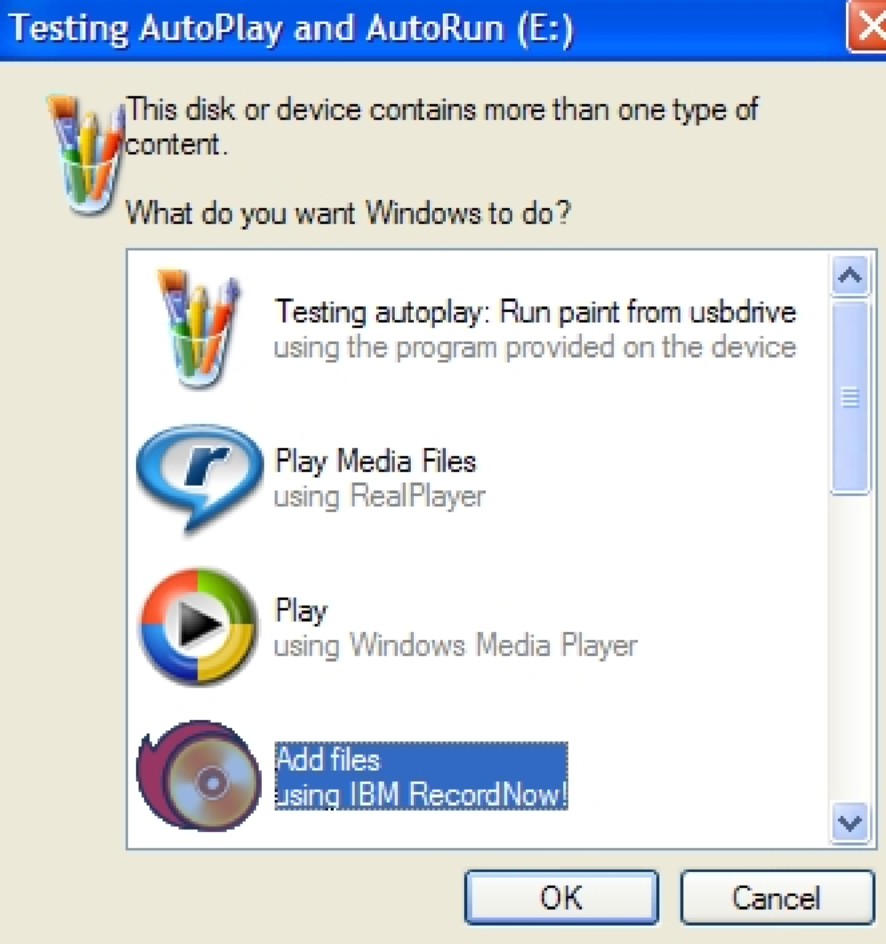
```
; Right click on the drive letter to see this
shell\FromFlash=Testing context: run paint from usbdrive
shell\FromFlash\command=mspaint.exe

; Run this when double click on drive letter in My Computer
; Because of the line above, this invokes Paint
shell=FromFlash

; The icon for the drive letter is taken from here
icon=mspaint.exe

; This is the volume name of the USB flash drive
label=Testing AutoRun Stuff
```

After putting these two files on a USB flash drive, eject it from the computer and then physically re-insert it. If Autoplay is enabled, a window like that shown below (from an XP Professional machine) will pop up in a couple seconds.



Testing AutoPlay and AutoRun (E:)

This disk or device contains more than one type of content.

What do you want Windows to do?

Testing autoplay: Run paint from usbdrive
using the program provided on the device

Play Media Files
using RealPlayer

Play
using Windows Media Player

Add files
using IBM RecordNow!

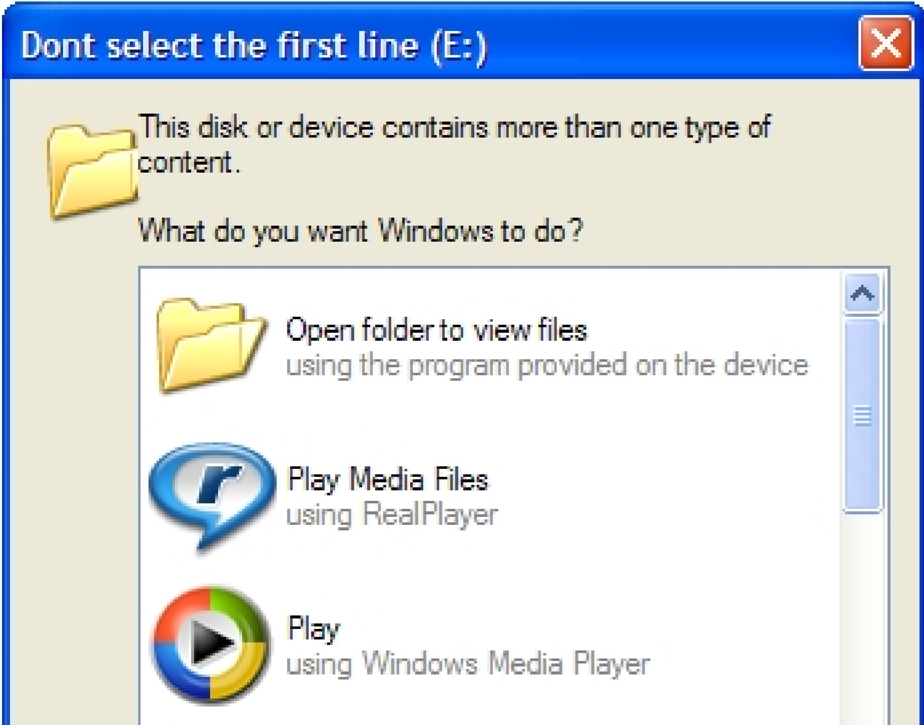
The sample autorun.inf file accounts for the first option "Testing autoplay: Run paint from usbdrive". The text comes from the "action=" line. The action that Windows takes, when you select this option, comes from the "open=" line. In this case, it runs Paint from the flash drive. The Paint icon (paintbrushes in the clear glass) comes from the "icon=" line.

Beware of AutoPlay Tricks

Bad guys abuse this to trick people into running malicious software that's on the removable drive.

Specifically, Downadup (a.k.a. Conficker) makes its new entry in the Autoplay menu look like the normally safe "Open folder to view files" entry.

Below is a Windows XP example of how this looks. Fortunately, there is an obvious clue that the first entry is running software on the flash drive, the second line says "using the program provided on the device". I don't know if this too can be changed. Still, even as it now stands, this will certainly fool some people.



Dont select the first line (E:)

This disk or device contains more than one type of content.

What do you want Windows to do?

Open folder to view files
using the program provided on the device
Play Media Files
using RealPlayer
Play
using Windows Media Player

The SANS Internet Storm Center has an example of this how this trick appears to Vista users (see Conficker's autorun and social engineering). They also describe the changes made by Downadup to an autorun.inf file. The malicious software isn't stored on the flash drive in an obvious place, such as an EXE file in the root folder, rather it's stored on the flash drive as a .vmx file in a phony copy of the Recycle Bin.

Fortunately, there are telltale giveaways in Vista too, if you know what to look for. (My only copy of Vista is virtual and it doesn't do USB drives at all. In fairness, this may be a VMware issue)

The real instance of "Open folder to view files" says "using Windows Explorer" on the second line, the phony copy says "Publisher not specified" on the second line. Also, the two copies are right underneath each other in Vista. In Windows XP the real "Open folder to view files" was buried near the bottom of the list of available options and not immediately visible.

Woody Leonhard also wrote about this, and his Vista screen shot also says "Publisher not specified" on the second line.

Not to spend too much time on Conficker/Downadup, but an F-Secure blog (the company specializes in anti-malware software) shows how the changes it makes to autorun.inf are obscured with random binary garbage in an attempt to fool antivirus programs.

F-Secure also has a Vista screen shot of the phony "Open folder to view files", but in their case the giveaway on the second line says "Published by Microsoft Windows". Again, **if the second line under "Open folder to view files" does not say "using Windows Explorer", don't click on it.**

F-Secure tried the same trick with Windows 7 and found it to be just as vulnerable to this type of malicious autorun.inf file as Windows XP and Vista.

Don't Double-Click That Drive Letter

Some Windows computers have Autoplay disabled. Even without it however, a malicious autorun.inf file can still trick people into running malware from a USB flash drive.

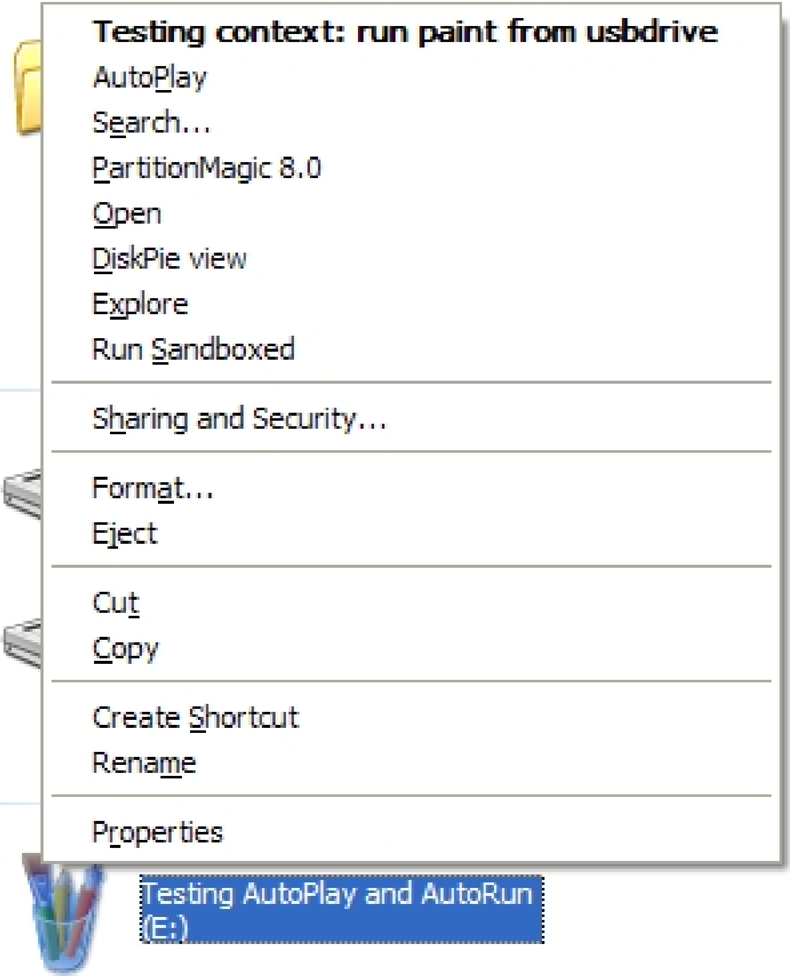
At least with AutoPlay you see a window and know that something is happening. Far more insidious is when an autorun.inf files takes advantage of double-clicking on the drive letter to run a program rather than (or in addition to) listing the files/folders.

If a computer is vulnerable to this type of attack, then my test autorun.inf file will run Paint when you double-click on the drive letter for the USB flash drive. This action comes from the "shell=" line. It will *not* list the files/folders on the drive, but you can see them by right clicking on the drive letter and selecting either Open or Explore.

Context menu

There's actually a third way that a malicious autorun.inf file can try to trick someone into running a program. It can modify the context menu, the menu that pops up when you right click on the drive letter in My Computer.

My test autorun.inf adds a new entry at the top of the context menu as shown below.


Testing context: run paint from usbdrive
AutoPlay
Search...
PartitionMagic 8.0
Open
DiskPie view
Explore
Run Sandboxed
Sharing and Security...
Format...
Eject
Cut
Copy
Create Shortcut
Rename
Properties
Testing AutoPlay and AutoRun (E:)

The option I added has the obvious name "Testing context: run paint from usbdrive". The text comes from the "shell\FromFlash=" line and the action that it invokes comes from the "shell\FromFlash\command=" line.

Rather than add a new entry to this menu, a bad guy would probably re-define an existing menu entry. For example, with the directives below, clicking on the "Open" menu entry would run Paint.

```
shell\Open\command=mspaint.exe
shell=Open
```

Immediately running a program

All my testing was done on Windows XP with ServicePack 2. According to Wikipedia, the rules for autorun and autoplay vary based on the drive type, the version of Windows, the service pack applied to Windows and how the operating system is configured. It's very complicated, but under certain conditions, Windows will run a program on a USB flash drive immediately as soon as the drive is inserted with no visual clue to the end user.

That is, there have been times when malware didn't need to wait for a double-click on the drive letter.

Jesper M. Johansson, formerly of Microsoft, explained in January 2008 how U3 flash drives trick Windows into immediately running software:

> In a nutshell, a U3-enabled flash drive lies about itself. It tells the OS that it is actually a USB hub with a flash drive and a CD plugged into it. Windows® versions prior to Windows Vista® will, by default, automatically run programs designated in the autorun.inf file on CDs, but not on USB drives. By lying about itself, the U3-enabled USB flash drive fools the OS into autorunning something called the U3 launcher. The U3 launcher, in turn, can start programs...

Immediately running software is probably rare nowadays, but if you are testing with this sample autorun.inf file and Paint runs as soon as the flash drive is inserted, your computer is *extremely* vulnerable.

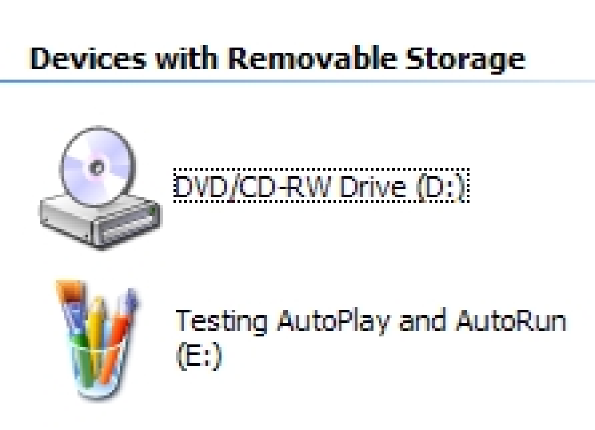
Othere Changes

Two other, relatively minor, changes that an autorun.inf file can inflict are changing the name and icon of the USB flash drive.

My test file changes the drive icon displayed in My Computer (see below) to be the Paint icon, paintbrushes in a glass.

It also changes the volume name to "Testing AutoPlay and AutoRun".

We also saw these changes in the first AutoPlay screen shot above. The volume name appeared in the blue stripe at the top of the Autoplay window.



Devices with Removable Storage

DVD/CD-RW Drive (D:)

Testing AutoPlay and AutoRun (E:)

A hidden autorun.inf file isn't hidden from Windows, so these changes can indicate the presence of an autorun.inf file even on machines configured not to display hidden files.

Jesper M. Johansson came up with an interesting wrinkle that combines methods to entice someone to click on an Autoplay window (see Figure 5). He posits that by making the drive letter name "My Secret Stuff" and by making the text in the Autoplay window say "My Porn Stash" that many users would probably click on it. Forewarned is forearmed.

Help is on the way

Soon, I'll go over a number of ways to defend a computer from malicious autorun.inf files and infected USB flash drives. Suffice it to say, it's much harder than it should be, which is why millions of Windows PCs are at risk.

Next up: Autorun and Autoplay: screwed by terminology where I explain that a large part of the problem here is not just poor design (though there's that), or software bugs (got that too) or even the old stand-by, poor documentation (got it). This quagmire suffers from yet another problem, poor terminology, that gets in the way of fully understanding the issues at hand.

January 26,2009: This was updated to reflect the fact that normal menu options on the context menu can be compromised. Also added the quote from Jesper M. Johansson.

January 27,2009: Updated to include Mr. Johansson's enticement using a combination of methods.

January 30, 2009: For the solution see The best way to disable Autorun for protection from infected USB flash drives

Attachment	Size
sample_autorun_dot_inf.txt	1.12 KB

Reply

Print

Email this

Digg this

Slashdot this

What People Are Saying

Add new comment

As (the primary I suppose)

Submitted by Carveone on January 28, 2009 - 11:54 A.M.

As (the primary I suppose) author of the wikipedia AutoRun article, I had a heck of a time sorting out from MS documentation when Windows would autorun and when it wouldn't. It was hard to be clear without being incorrect! I could try and clarify it.

With Windows defaults, only a U3 flash drive (the one with a fake CD-ROM) is ever a silent risk. One would have to have Windows 98SE, with AutoRun deliberately enabled for DRIVE_REMOVABLE to be at risk from that vector.

Additional points: 1) I believe that some USB HDDs are reported as being DRIVE_FIXED. This is very dangerous behaviour on XP before SP2. 2) I believe (but have not yet found a citable reference) that TweakUI ignores the Windows XP defaults of 0x95 for NoDriveTypeAutoRun and thus, by accident I suppose, enables AutoRun for Network drives. That is highly unexpected behaviour. 3) I would highly recommend the use File Mapping to kill any autorun.inf. This method is tested and proven and avoids user support calls from AutoPlay failing to pop up.

Reply | Report this comment

Apologies - the second

Submitted by Carveone on January 28, 2009 - 12:05 P.M.

Apologies - the second paragraph should say:

For a basic flash drive, one would have to have Windows 98SE, with AutoRun deliberately enabled for DRIVE_REMOVABLE to be at risk from that vector.

A U3 flash drive is a real problem for a default installation on Windows. Never allow anyone to put a flash drive in your computer.

Reply | Report this comment

wikipedia on autorun

Submitted by Michael Horowitz on January 28, 2009 - 12:02 P.M.

I read the Wikipedia article on Autorun and Autoplay and was, in general, impressed. Its a very very hard story to tell as the action taken by Windows varies greatly depending on the version of Windows. For example, XP SP1, SP2 and SP3 all treat this topic differently, and that's before users start changing defaults and Microsoft starts issuing patches that modify the OS behavior.

I agree that Nick Brown's FileMapping registry zap is the best way to deal with this. I'm working on that writeup now...

Reply | Report this comment

you missed the point

Submitted by Mitch on January 28, 2009 - 8:23 P.M.

the best to deal with this is simply not to use Microsoft Windows.

Reply | Report this comment

Great article - added to our removal steps

Submitted by Eddie Philips on January 25, 2009 - 12:09 P.M.

Thanks for this test - we've linked to it from the removal steps page at Downadup.com. We've received several emails from non-technical users trying to fix their PCs asking for this.

Reply | Report this comment

Related Posts

Is Microsoft getting ready to kill Windows?
US-CERT naysays Microsoft security advisory
It's time to start issuing PC licenses
MSNBC Spam-O-Rama

Today's Top Stories

Microsoft reveals 'My Phone' backup, sync service
Senate approves 'strict' rules on hiring H-1B workers

FAQ: How Google Latitude locates you
Elgan: Here comes the e-book revolution
IBM offers to shift workers losing jobs to lower-wage countries
Economy could slow enterprise adoption of Windows 7

Hot Posts

Bribing bloggers brings bad juju
Posted by Mark Everett Hall | 1 comment
Uh, oh. Looks like I was wrong about Amazon.com
Posted by Mike Elgan | 21 comments
What you don't know about the Windows Malicious Software Removal Tool
Posted by Michael Horowitz | 1 comments

Recent Comments

Easy Software Install or Update...14 min 30 sec ago
WHAT?...28 min 31 sec ago
ADD-ON...37 min 18 sec ago

Resource Alerts

to receive Security Resource Alerts

Webcasts

IBM Dynamic Infrastructure Forum: Reduce Costs, Manage Risk and Improve Service

Simpana® 8 Launch Webcast

Data Governance: How to Triumph over Bad Data

Whitepapers

Getting in Compliance With Government Data Regulations By Leveraging Online Security Technology

The Latest Advancements in SSL Technology

Web Threat Protection

Computerworld Reports

Computerworld Technology Briefing: Intelligent Users Use Business Intelligence

Trend Micro Gets Smart with a Hybrid Approach

Virtual Reality

Newsletters

Subscribe to our blog newsletters:

□ Computerworld Blogs □ Shark Bait □ TechGear (Personal Tech) Enter E-mail Address GO

Michael Horowitz's Archive

- February 2009
- January 2009

IT Topics

- Business Intelligence
- Careers
- Development
- Emerging Technology
- Government & Regulation
- Hardware
- Internet

- Management
- Mobile & Wireless
- Networking
- Security
- Servers & Data Center
- SOA & Web Services
- Software

- Storage

Sponsored Links

About Us Advertise Contacts Editorial Calendar Help Desk Jobs at IDG Privacy Policy Reprints Site Map

CIO Computerworld CSO DEMO GamePro Games.net IDC IDG IDG Connect IDG Knowledge Hub IDG TechNetwork IDG Ventures IDG.net InfoWorld ITwhitepapers ITworld JavaWorld LinuxWorld Macworld Network World PC World The Industry Standard

```
[autorun]
;----------------------------------------------------------
; Test your defenses against infected a USB flash drives
;
; Note: Lines that start with a semicolon are comments
;   and are ignored by Windows
;
; Place this in a file in the root directory of a USB flash drive
; The file name must be autorun.inf
; Also in the root folder, place a copy of Paint (mspaint.exe)
;
; Created by Michael Horowitz January 2009
;------------------------------------------------------------
;
; This shows up in the first line of the Autoplay menu
action=Testing autoplay: Run paint from usbdrive

; This causes the AutoPlay window to run Paint from the flash drive
open=mspaint.exe

; Right click on the drive letter and see this
shell\FromFlash=Testing context: run paint from usbdrive
shell\FromFlash\command=mspaint.exe

; Run this when double click on drive letter in My Computer
; Because of the line above, this invokes Paint on the USB drive
shell=FromFlash

; The icon for the drive letter is taken from here
icon=mspaint.exe

; This is the volume name of the USB flash drive
label=Testing AutoRun Stuff
```

+ **Virtual**

Print Subscriptions

NEWSLETTERS

Subscribe to our e-mail newsletters
For more info on a specific newsletter, click the title. Details will be displayed in a new window.

□ Computerworld Daily News (First Look and Wrap-Up)

□ Computerworld Blogs Newsletter

□ The Weekly Top 10

More E-Mail Newsletters ▸

Enter E-mail Address

SIGN-UP

Michael Horowitz

Defensive Computing

More posts | Read bio

January 29, 2009 - 5:09 P.M.

Autorun and Autoplay: screwed by terminology

1 comment

- TAGS:Autoplay, autorun, Internet security, Microsoft, security, Windows
- IT TOPICS:Cybercrime & Hacking, Security, Security Hardware & Software, Windows & Microsoft

Many people are confused about Windows Autorun and Autoplay, including otherwise competent techies. You may be confused and not even realize it. I was.

In the good old days, neither Autorun nor Autoplay were all that important to Windows users. However, now that malicious software spreads by means of infected USB flash drives, it has become a very important topic.

A large part of the confusion stems from terminology.

I spent many years debugging software problems, and the hardest part was always understanding the issue at hand. The last thing anyone needs is for terminology to get in the way of understanding and communicating. Yet, that's what has happened.

Microsoft uses the words Autorun and Autoplay to mean different things at different times. No doubt this is driven by the fact that they have no terms for three important autorun related *things*.

As I noted in my previous posting, Test your defenses against malicious USB flash drives, there are four ways that malicious software on a USB flash drive (thumb drive, pen drive, USB key, memory stick, etc.) can execute and infect a Windows computer:

1. Run immediately and automatically
2. Run via the Autoplay popup window
3. Run when the user doubleclicks on the drive letter in My Computer
4. Run via a modification to the context menu (the pop-up menu displayed when you right click on a drive letter).

Four approaches, yet Microsoft has only two words to describe them, autorun and autoplay. Autorun describes the first approach and is typically used on CDs. Autoplay describes the second, it is a feature of Windows that was introduced with XP. Microsoft has no term for either the third and fourth approach.

Perhaps most importantly, Microsoft has no one term to describe the totality all four approaches. Thus, they can't even talk about what their customers care about, protecting their computers from infected USB flash drives. They have no term that encompasses all four potential attack vectors.

Just last week, the well regarded Woody Leonhard writing in the Windows Secrets newsletter discussed the latest worm (which goes by the names Downadup, Conficker, and Kido). At the end of the article, he linked to an older newsletter article, by Scott Dunn, for "...comprehensive instructions to disable

AutoPlay."

Scott Dunn's article describes an approach **which does not turn off autoplay**. It's a great approach, and one that I'll be writing on extensively next time, but it has nothing at all to do with Autoplay. A computer that's modified as per Scott Dunn's suggestion (which comes originally from Nick Brown) is perfectly protected from USB flash drives, yet the Autoplay feature of Windows is enabled.

Dan McCloy turned off Autoplay, yet his computer got infected from an infected USB flash drive using attack vector 3 or 4. To describe the problem he invented the term EDDC (Execution of the Drive's Default Command).

As is all too typical, Autorun/Autoplay has been victimized by poor design, software bugs and poor documentation. But perhaps the biggest issue is the terminology. You can't fix or understand a problem that you can't describe.

Microsoft has got to clear this up. They probably won't.

More on protecting your computer from infected USB flash drives next time.

See The best way to disable Autorun for protection from infected USB flash drives.

Reply
Print
Email this
Digg this
Slashdot this

What People Are Saying

Add new comment

Save yourself the bother

Submitted by Charles Norrie on January 29, 2009 - 8:53 P.M.

Although this is very detailed and helpful advice, there is a simple way of dealing with the whole malware problem.

Replace your OS.

Ubunu Linux (current version Intrepid Ibex) by design cannot be infected with malware, is free and has over 17000 software packages also free.

Reply | Report this comment

Related Posts

Test your defenses against malicious USB flash drives
Removing malware from an infected PC - battling antivirus programs
US-CERT naysays Microsoft security advisory
Is Microsoft getting ready to kill Windows?

Today's Top Stories

Microsoft reveals 'My Phone' backup, sync service
Senate approves 'strict' rules on hiring H-1B workers
FAQ: How Google Latitude locates you
Elgan: Here comes the e-book revolution
IBM offers to shift workers losing jobs to lower-wage countries
Economy could slow enterprise adoption of Windows 7

Hot Posts

Bribing bloggers brings bad juju
Posted by Mark Everett Hall | 1 comment
Uh, oh. Looks like I was wrong about Amazon.com
Posted by Mike Elgan | 21 comments
What you don't know about the Windows Malicious Software Removal Tool
Posted by Michael Horowitz | 1 comments

Recent Comments

Easy Software Install or Update...14 min 30 sec ago
WHAT?...28 min 31 sec ago
ADD-ON...37 min 18 sec ago

Resource Alerts

to receive Security Resource Alerts

Webcasts

IBM Dynamic Infrastructure Forum: Reduce Costs, Manage Risk and Improve Service

Simpana® 8 Launch Webcast

Data Governance: How to Triumph over Bad Data

Whitepapers

Getting in Compliance With Government Data Regulations By Leveraging Online Security Technology

The Latest Advancements in SSL Technology

Web Threat Protection

Computerworld Reports

Computerworld Technology Briefing: Intelligent Users Use Business Intelligence

Trend Micro Gets Smart with a Hybrid Approach

Virtual Reality

Newsletters

Subscribe to our blog newsletters:

□ Computerworld Blogs □ Shark Bait □ TechGear (Personal Tech) Enter E-mail Address GO

Michael Horowitz's Archive

- February 2009
- January 2009

IT Topics

- Business Intelligence
- Careers
- Development
- Emerging Technology
- Government & Regulation
- Hardware
- Internet

- Management
- Mobile & Wireless
- Networking
- Security
- Servers & Data Center
- SOA & Web Services
- Software
- Storage

Sponsored Links

About Us Advertise Contacts Editorial Calendar Help Desk Jobs at IDG Privacy Policy Reprints Site Map

CIO Computerworld CSO DEMO GamePro Games.net IDC IDG IDG Connect IDG Knowledge Hub IDG TechNetwork IDG Ventures
IDG.net InfoWorld ITwhitepapers ITworld JavaWorld LinuxWorld Macworld Network World PC World The Industry Standard

Dan McCloy's Autorun Reference Guide

Autorun Reference Guide

AutoRunGuard

Feedback

Why Disabling Autorun Only Helps The Viruses, and What You Should Actually Do to Protect Yourself.

The Internet is full of well-intentioned advice to disable AutoRun (or AutoPlay), so that you will be protected from getting infected from a worm on a USB stick. I took this advice seriously, and still ended up getting infected. This happened because I hadn’t understood some basic concepts, nor had I disabled the real culprit.

Basically, when an “Autorun worm” on a USB stick infects a PC, it is never by **AutoRun** and it is seldom by **AutoPlay**. Instead, they all employ a method that I’ll call **Execution of the Drive’s Default Command (EDDC).**

All three of these— **AutoRun** , **AutoPlay** , and **EDDC** —are separate functionalities that can run a program that has been specified in the **Autorun**.inf file . This is a file that is located (possibly invisibly) in the drive’s root (i.e. “top-level”) folder.

A quick explanation on the differences between AutoRun, AutoPlay, and EDDC may be helpful, and then we’ll look at a solution that actually works, one that every Windows user ought to employ.

AutoRun, AutoPlay, EDDC — What’s the difference?

AutoRun is the functionality that enables a CD-ROM drive or a fixed drive to specify a program or document to be started immediately upon the connection of the drive. AutoRun has been designed not to work on removable drives such as USB flash drives, as these drives are much more readily infected and passed around to other computers. (Note that while U3 flash drives do indeed employ AutoRun to automatically launch their own software, this is accomplished by hardware tricks built into the devices.) There is no virus that employs AutoRun from an ordinary USB stick.

AutoPlay is a more recent enhancement of AutoRun, and reflects an improvement in the security model. Instead of the external media deciding for itself what program to execute, the user makes the decision in response to the AutoPlay Menu. Which items appear on this menu is determined by the types of file found on the drive (such as pictures, music, and video), and by settings in the Autorun.inf file. A small minority of Autorun worms employ the AutoPlay menu, but only as a secondary strategy. (It is probably an ill-advised strategy, as stealth is lost and suspicions raised for even minimally alert users.) Because AutoPlay is user-controlled and thus “secure”, Windows enables AutoPlay for removable drives by default.

> Note: Vista will actually allow the user to configure his AutoPlay settings to automatically run whatever program is specified in the Autorun.inf file, which means that an infected USB stick will indeed effectively be permitted to "AutoRun". No one should never chose such a dangerous setting, with the possible exception of someone who lives alone in their own universe. There are much safer ways to accomplish whatever it is you'd be aiming for with that setting. More alarming, it's not hard to inadvertently end up with such settings for AutoPlay. When you insert a CD with a legitimate autorun program, and Vista's AutoPlay menu asks what you want to do, suppose you tell it to run the program, and you also check the box for "always do this." By default, that choice will also apply to removable drives. The next time you insert an infected USB stick, it will launch the virus without further interaction from you. Following the advice below for disabling AUTORUN.INF files will protect you from this problem, too.

Execution of the Drive's Default Command (EDDC) is a third distinct means of automatically starting a program specified in the drive's Autorun.inf file. The key difference from AutoRun functionality is that EDDC does not happen automatically upon drive connection. It is triggered when the user double-clicks on the drive icon under 'My Computer' (i.e. in an Explorer window), or selects the highlighted option from the drive's shortcut menu (accessed with a right-click). Autorun worms often hijack both the 'Open' and 'Explore' commands on the drive's shortcut menu, so that either of these will launch the viral executable.

Disabling AutoRun/AutoPlay

Both AutoRun and AutoPlay are disabled together via the same keys in the Windows Registry. (These keys are named NoDriveAutoRun and NoDriveTypeAutoRun .) These same keys can also be more conveniently (albeit less finely) configured via Group Policy or the TweakUI PowerToy for Windows XP .

Remember, however, that it is by EDDC—never by AutoRun and seldom by AutoPlay—that an Autorun worm on a flash drive infects a PC.

For current Autorun worms attempting to infect a PC from a thumb drive, the only effect of Disabling AutoPlay is probably in their favor: It means that the AutoPlay menu will never appear, and thus the clumsy minority of worms that would have advertised their presence in that menu will instead have to rely on the same stealth technique that the rest of them use: They wait for the the user to trigger the drive's default command.

Disabling AUTORUN.INF

What every Windows user ought to do is disable the Autorun.inf file. In some rare cases, some legitimate functionality will be lost, but this can be restored more securely in other ways we'll look at shortly.

The way to accomplish this was first suggested in a blog entry by Nick Brown . His explanation there is well worth reading, but to quote just his how-to instructions:

> All you do is to copy these three lines into a file called NOAUTRUN.REG (or anything.REG) and double-click it...
>
> ```
> REGEDIT4
> [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf]
> @="@SYS:DoesNotExist"
> ```

Alternatively, for your convenience, you can double-click the DisableAutorunINF.reg file found in DisableAutorunINF.zip

Note: On many computers, the change won't go into effect until you've restarted all instances of Explorer including the Windows Desktop. If you don't know how to do that, just log off Windows or restart the computer.

Inoculating Your USB Flash Drive

Before we move on, I should mention a trick to keep your USB flash drive from being a carrier for Autorun worms: (This has been suggested by Nick Brown and by others also.)

Create a folder on the drive named "Autorun.inf". Maybe even put a small file in there so that the folder is non-empty. That's all.

I've had several kinds of Autorun worms try to infect my USB stick, and all of them were foiled by this. They can write their executable to the drive, but not the Autorun.inf file necessary to launch that program, because they didn't anticipate that a folder by that name might already be present. Thus, using my USB stick in another computer after it's been in an infected computer won't spread the worm. Sooner or later, malware authors will catch onto this trick, but at least your drive won't be a carrier for any of the hundreds of Autorun worms that are currently wreaking havoc.

(Note: Typically a worm marks the Autorun.inf file it’s created with “Hidden” and “System” attributes, so that you won’t notice the file unless you’ve configured Windows to show you both hidden and system files.)

Replacing and Improving upon Autorun.inf Functionality

If you’ve disabled Autorun.inf as described above, there is no need to have AutoPlay disabled. AutoPlay offers a good deal of convenient functionality that it would be a pity to lose. e.g. When you insert your SD card from your camera, it may be handy to automatically launch Picasa’s import process.

However, there are other cases in which you really do want some kind of behavior to occur when a certain drive is connected:

- Maybe you have a U3 drive , and you want to start the **U3 LaunchPad** when the U3 drive is inserted.
- Maybe you always want an option for running a **virus scan** to pop up when any flash drive is inserted, or perhaps you'd like to be alerted if the drive just connected contains an autorun.inf file, or files/folders marked as hidden or system. (These are signs of a worm.)
- Maybe you want an option for **launching your backup program** to pop up whenever you connect that specific drive.
- Maybe you want to automatically **mount a TrueCrypt volume** any time a drive containing a .tc file is connected, or perhaps whenever a specific drive is connected.
- Maybe you have a portable application on your USB drive that you want to launch when the drive is inserted, but you also know that you should never trust any program on any drive that has been in a possibly-infected machine. (A virus on that machine may have injected itself into the program on your USB drive.) So you want to launch your portable app, *but only after verifying that it hasn't been changed* .

All of these are things that are best controlled by your computer. There are utilities that will enable you to control such behavior. Two options are Didier Stevens' USBVirusScan (freeware) and Uwe Sieber's USBDLM (free for private and educational use) . If you know of better options, please let me know. Despite the name, what USBVirusScan does is simply call a program of your choice any time a drive is connected. I have written a command script for USBVirusScan to call, and it is named...

AutoRunGuard™

I have written a small program called **AutoRunGuard** to be used with USBVirusScan. It can be used in tandem with AutoPlay or as an AutoPlay replacement.

Think of AutoRunGuard as a way of setting out rules --- as simple or complex as you like --- to govern what will happen as soon as a removable drive is connected or CD inserted. This may include launching a particular program immediately, or adding relevant options to a menu for you to choose from. And there's more...

For details, please see the AutoRunGuard page.

Stay safe!

Comments or feedback welcome.

Site created by SynthaSite | Design by Free CSS Templates

Cheap anti-virus tips, assorted Windows XP stuff, and other randomness.

23 October 2007

Memory stick worms

Lately, we've been seeing a lot of worms, and even some genuine viruses (*), coming into our network via USB memory sticks (aka "pen drives"). For those of us who remember the first MS-DOS viruses, which spread almost exclusively via diskette, it's rather nostalgic.

The culprit, of course, is Microsoft's desire to make things "simple" - meaning "automatic" - for Joe Sixpack; there's a fundamental incompatibility between a home entertainment system, which Windows has become, and an operating system for getting work done. (Here's a rule of thumb for you: any time you see stuff which starts without the user asking it to, look for malware to pop up in short order.)

These worms pretty much all reproduce the same way, at least in terms of how they jump to and from PCs. They have an AUTORUN.INF file and an executable of some kind. When you put the stick in the PC, Windows finds AUTORUN.INF "automagically". You can find documentation of some of the possible things which this file can do, but basically, the worm version will either run the executable immediately, or modify the Windows Explorer default behaviour so that the worm will run as soon as you open the stick by double-clicking on it. The executable will make a copy of itself and AUTORUN.INF on all the disk partitions and shared drive connections which it can see, and then open the root folder normally. (This takes a fraction of a second, but you won't notice it.) The executable will then sit around in memory and every time you insert a removable storage volume (such as another memory stick) or map a network drive, it will copy the worm "kit" to it.

Sometimes the executable will live in a fake \RECYCLED folder, which is quite clever because hardly anyone ever opens the recycle bin on a memory stick, and because the folder doesn't contain a real recycle bin structure, the worm will be safe, even if you empty the bin while the stick is in the drive.

Now, in theory you can prevent certain drive types from executing the contents of their AUTORUN.INF files using a registry value (NoDriveTypeAutoRun). But this is hard to do in practice. First, it's a per-user key, which in a corporate environment is harder to manipulate reliably than a per-PC key. Secondly, there are several bugs known for it. And thirdly, a little-known registry key called MountPoints2 contains cached information about every memory stick or other removable device which your PC has ever seen, and that overrides the NoDriveTypeAutoRun value if you insert a volume which the PC already knows about.

The only solution I could find from Microsoft is typically light and nimble: prevent all USB storage from running. This is fine if the aim is to stop people using memory sticks altogether, but didn't you just have a 4GB stick custom-printed for everyone in the company, and tell them to make their own backups on it?

Anyway, there seems to be a solution: a one-shot, quick way to prevent AUTORUN.INF files from being used on a PC, from any medium. My colleague and fellow low-budget Windows hacker Emin Atac thought up the idea, and I spent all of 15 minutes testing it. Now it's your turn (well, "the world is our beta site" works well enough as a corporate mantra for Microsoft).

All you do is to copy these three lines into a file called NOAUTRUN.REG (or anything.REG) and double-click it. Corporate network people can transform it into a script for their favourite command-line registry manipulator, or maybe make it a system policy thingy.

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf]
@="@SYS:DoesNotExist"
```

This hack tells Windows to treat AUTORUN.INF as if it were a configuration file from a pre-Windows 95 application. IniFileMapping is a key which tells Windows how to handle the .INI files which those applications typically used to store their configuration data (before the registry existed). In this case it says "whenever you have to handle a file called AUTORUN.INF, don't use the values from the file. You'll find alternative values at HKEY_LOCAL_MACHINE\SOFTWARE\DoesNotExist." And since that key, er, does not exist, it's as if AUTORUN.INF is completely empty, and so nothing autoruns, and nothing is added to the Explorer double-click action. Result: worms cannot get in - unless you start double-clicking executables to see what they do, in which case, you deserve to have your PC infected.

The only downside of this is that if you insert a CD with software on it, you have to explore it by hand to find the setup program. Of course, if that means your kids install less software, that could also be considered an upside.

If you want to check that the registry settings of this hack are in place, open Regedit, walk down to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping, and you should see something like this:

(*) If you don't know the difference, Wikipedia is (probably) your friend.

Labels: prevent virus USB memory stick worm conficker downadup kido disable autorun

# posted by Nick Brown @ 19:29

Comments:

This post has been removed by a blog administrator.

# posted by <a href="users2.titanichost.com/amalopra">JohnBraun</a> : 04 November 2007 19:28

This is exactly what I was looking for, I needed a way to disable just the AutoRun.inf functionalityThank you

# posted by Ahmad Hisham : 03 February 2008 09:44

Man, I'd been meaning to post that somewhere for ages! It took me hours of analyzing Procmon traces and Googling registry keys to figure out a way to accomplish this.The best part of disabling autorun in this manner is that CD audio and DVD Video will still autoplay when you insert the disc. Even those DVDs with silly autorun .exe files (or Sony's rootkitted CDs) will be properly recognized and opened up in your media player of choice! :)

# posted by Anonymous : 08 February 2008 02:02

I'm new at this, but does the second "@" sign inside the quote mark need to be there? When merged with the registry it leaves the @ sign there: @SYS:.....

# posted by Anonymous : 19 February 2008 01:00

Yes, you need everything.The first "@" means "the magic, unnamed default value for this registry key. It's just an instruction to the REGEDIT4 loader.The second @ is part of the final value which goes into the registry. It means "what follows is a pointer - follow it". We want Windows to follow the pointer and finish up at a dead end, where it will say, "OK, this entire INI file maps to nothing".

# posted by Nick Brown : 19 February 2008 09:11

Going the other way around, I put nullified (no commands to execute hostile programs) AUTORUN.INF files on my flash disks with the RHS attributes on them, figuring that most worms can't run a check whether their AUTORUN.INF is ok. XDThanks for this little piece of information. :)

# posted by Anonymous : 24 February 2008 10:55

Don't expect that to protect you too much when you put that stick in another PC. The worms I've seen are quite capable of deleting an "RHS" file. But so far I haven't seen one which also remembered to check that AUTORUN.INF wasn't a directory. So make a folder with that name and you might be protected somewhat (until a friend comes round and puts their stick in your PC...)

# posted by Nick Brown : 24 February 2008 19:54

Great tip. Unfortunately SanDisk U3 drives stop working after adding the register entry.As expected no software is autostarted, but manual launching of LaunchU3.Exe also fails. It looks like it uses AutoRun.Inf to store some settings. :-(

# posted by Krister : 27 February 2008 23:25

This is brilliant! Exactly what I want: The convenience of AutoPlay launching my own software, without the risks of AutoRun.inf launching something malicious.Incidentally, if there is something legitimate that you want to launch when media is inserted (such as for a U3 drive, or to run your backup program when you connect the external drive), I'd recommend using USBVirusScan to respond to drive insertions. Make a little batch file that effectively says, "If it's my U3 drive, start the U3 LaunchPad."

# posted by Dan : 20 March 2008 11:52

For fuller control of what happens when a USB stick is inserted (especially with Autorun.inf files disabled), I wrote a small, free program called **AutoRunGuard**. You can use this to make any USB stick behave like a U3 drive, even using MD5 hashes to verify that your portable apps weren't injected with a virus when you used that stick elsewhere. It can launch something appropriate to the drive or content immediately, or add relevant options to a menu to be presented to the user. It also recognizes when a drive has been infected by an autorun worm, and can inoculate the drive in the way you've suggested. Freely available at http://autorun.synthasite.com/AutoRunGuard.phpThat leaves only one remaining excuse for not disabling autorun.inf files: Some Vista users may employ autorun.inf files to instruct AutoPlay to restrict its content searches in certain ways, and that's pretty minor.Thank you Nick!Dan

# posted by Dan : 05 April 2008 08:05

I'm facing exactly the same difficulties with autoruns as well. I liked your approach very much but I found your instructions rather difficult to follow.Should I create a .REG file and double click it "execute it" and add the lines that will be in turn added to my main registry?I'm sorry, but I'm still a clutz to editing the registry. If you help me out I'll be eternally gratified.

# posted by Anonymous : 12 May 2008 02:39

Yes. Copy the three lines which I posted into a file with the extension .REG and double-click it, then accept whatever dire warnings it gives you.If you see more than three lines, make sure that everything from "[HKEY_LOCAL_MACHINE"... through to "Autorun.inf]" is on a single line.Good luckNick

# posted by Nick Brown : 12 May 2008 19:05

Hi,This is great tip, but could you write how to go back to previous state, because some CD`s don`t work after install.

# posted by Anonymous : 05 June 2008 11:07

The equivalent file to undo the changes would be:REGEDIT4[-HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf]Note the '-' character just after the left bracket.But I'm not sure what you mean by "some CDs don't work". They certainly won't autorun - that's the idea - but you can always explore the root directory of the CD, open Autorun.inf in Notepad, and see what it does.

# posted by Nick Brown : 05 June 2008 11:46

As soon as I find out, I`ll write it. It`s my college`s problem.Thanks for quick answer.Sorry for my English mistakes, I`ve just started learn.

# posted by Michael : 05 June 2008 15:01

Hello again,There is license key and product number in Autorun.inf file.It`s work now. Thanks.

# posted by Michael : 06 June 2008 08:24

Sorry to bring bad news but this technique is causing breakage on a new system.I used it on a previous system but now it's:a) disabling "Show desktop" and removing its icon from the shortcut b) making Outlook 2003 complain "The extension C:\Program Files\Microsoft Office\OFFICE11\ADDINS\*.ecf could not be installed. There is an error in the syntax or format of the file." for each .ecf filec) stopping Kaspersky Internet Security's firewall from fully starting.This is on Windows XP Pro SP2 + hotfixes. Applying the technique causes these problems; removing it and restarting fixes the problems.It should not make any difference but this WXP is running under VirtualBox 1.6.2 on unbuntu 8.04.For more details see http://forum.kaspersky.com/index.php?showtopic=76418

# posted by Charles : 17 July 2008 16:58

Hi Charles,We don't have Kaspersky on our site but we do have Outlook 2003 and we have not seen the first two problems which you mentioned.>>It should not make any difference>>but this WXP is running under>>VirtualBox 1.6.2 on unbuntu 8.04.Perhaps it "shouldn't" make any difference, but my strong suspicion would be that it is doing. ;-)Nick

# posted by Nick Brown : 17 July 2008 17:05

Charles, you're putting the **@="@SYS:DoesNotExist"** in the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping** key instead of the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf** key.That might cause your problem. You have to create the **Autorun.inf** key.

# posted by Anonymous : 21 July 2008 17:33

Ooops! [FX: embarrased] My mistake -- that was the most likely explanation all along. Thanks to Anonymous (Joop) for finding it and pointing it out.

# posted by Charles : 25 July 2008 11:37

Hello thx for the useful registery keybut i have a questionHow to Configure this key Using Group Policy ??

# posted by Yasser : 16 October 2008 18:08

>>How to Configure this key Using Group Policy ??I don't know... I don't use Group Policy. :-) I presume you can add registry keys to the cute graphic layout. But I don't know if the code which applies policy reaches as far as IniFileMapping.

# posted by Nick Brown : 16 October 2008 23:22

Microsoft finally offers an update to disable autorun completely, look here: http://support.microsoft.com/kb/953252

# posted by Anonymous : 26 October 2008 02:51

I'd seen that. But my version is (a) easier to understand, (b) immediately reversible, and (c) field-tested for over a year. :-))

# posted by Nick Brown : 26 October 2008 09:06

Hey - this looks like exactky what I need!But I am way behind on the required techy-ness...I currently have the autorun worm on my laptop. The source was either a malicious download, or USB stick from my friend (both opened yesterday)Before reading this article I just backed up my essential files - to a USB HDD (!?) and the USB stick is still in my computer. My virus scan (AVG) is finding the worm on C:\ but I can't see this in windows explorer. I believe that it is preventing me from accessing the internet even though I am showing as beiong connected local & internet on Vista. I can't collect e-mail either. How do I:get rid of the autorun worm, and ensure thatthe 2 USB drives are not infected. Your help is appreciated!!BTW: I am not a corporate user, I'm a small business, so it's just me.

# posted by Steve B : 27 October 2008 14:37

Steve,I'm afraid I don't have a simple solution for you. This post is about preventing this sort of thing from happening. (Mind you, so was Vista. I guess Microsoft got it wrong in your case. Perhaps they'll pay for your PC to be fixed by way of an apology.)If your a/v software can't remove it then you are going to need professional help. (Or rather, your PC is. I'm not suggesting that you need a doctor. :-)) If I were sitting in front of your PC I'm sure I could remove all traces of the virus in a few minutes, but there's no way I can teach you by mail all you'd need to know to do that.In theory all you need to do is to delete the AUTORUN.INF file from the root directory of every drive in your system. But in practice, 90% of memory stick worms will have given themselves other automatic start points, so that won't be enough. And clearing up all the automatic start points in Windows is a huge subject in itself.You should work on the assumption that every single USB memory device which you own is now infected. But then again, you should always assume that every USB device has a virus on it; that's why you need the preventive tips in this blog post.Once your PC is cleaned up, apply the fix shown in this post (or, if you can't work out how to do it, show it to someone who can, such as the person who is going to clean up the PC). Then, you can safely put your USB devices into your system one at a time and clean them up with your anti-virus software.Good luck,Nick

# posted by Nick Brown : 27 October 2008 15:24

Thanks Nick - good enough!Someone has already collected the computer and (apparently) are pretty clear on the issues as I was able to detail the extentof the damage thuis far. Will inform them of this fix also. BTW - great service. Not just fast, informative, convincing and believeable, but also funny! Especially given my rather dry post and the fact that I would expect responses to be court and clinical in this industry. Fantastic job!Obviously it would have been better if you WERE sitting in front of my computer, or even betterif you could improve your ability to teach by email (i'm a quick learner) ...but can't ask for everything ;-)

# posted by Steve B : 27 October 2008 17:27

Aw shucks. Can I give you my boss's e-mail address? :-)

# posted by Nick Brown : 27 October 2008 20:21

Hi Nick and everyone in the blog:I'm from a small town in Argentina, with my brother have set a cybercafe (six PCs) your technique to prevent autorun is excelent, I can see you know a lot about this so I'll ask you a favor.I want to put multimedia on shared folders but don't want people to *steal* those videos, images and audio.With steal I mean I want they download their files from Internet consuming time that I can charge later.At a first glance it will appear too mean to do this but I have to remember that it is a small town and for one hour the people in my cyber pay $ 1 (I only have 128 Kb connection -and, as you surely know, that's a nominal value, the little cents I do with the downloads are my incomes).I thought up this idea:the real files will be hidden at some folderthe files people will see in shared folders will be shortcutsthe shortcuts arrows will be erased from registry (so people will believe those are the originals)My problem is that obviously they'll found out that the files are only 1 KB, so in the right click - properties to the file they'll found the root folder.How can I disable this function from shared folders?Most important: can you send me addresses to learn at least a small fraction of all you know?You seam to be a technician with the registry, I'm reading an e-book about the registry from Jerry Honnycut (excellent, by the way) but your knowledge dazzles me.I hope you can help me.Thanks in advance...

# posted by Anonymous : 28 October 2008 20:28

I'm sure that most of what you want to do can be achieved with the registry, but I'm not really an expert. I'm a general "systems person" who's been programming and managing computers for over 30 years, but I've never really got into the whole Windows object model thing.Good luck in your searchNick

# posted by Nick Brown : 28 October 2008 21:34

Thanks Nick for this post. Really informative.Just wanna inquire though:Will this also work for 2003 server,XP,vista or 2008 server? and I noticed the syntax you gave uses Regedit 4. Do we just use the same syntax for regedit 5.Your response will be greatly appreciated. And would surely help lots of people. (Sorry had to use anonymous - don't have an account yet. will set it up soon).thanks!-Bliss (Philippines)

# posted by Anonymous : 20 November 2008 19:01

Hello Bliss,On most systems, Regedit should recognise the REGEDIT4 syntax. Otherwise you can change the first line to "Windows Registry Editor Version 5.00" (without the quotes). Or open Regedit, export a key, and look at the syntax and paste the patch data as appropriate.I'm assuming that this patch works on most versions of Windows. I use it on XP, and Server 2003 is basically the same code. Although why you would put a memory stick into a server is slightly beyond me. And of course there are no viruses or worms which affect Vista or later versions because Microsoft has made them totally secure (ha ha ha). That's why all the anti-virus companies have gone out of business (because nobody runs XP any more either, ha ha ha).Good luck,Nick

# posted by Nick Brown : 20 November 2008 19:53

Hello. I'm not saying they've nicked your code, but have a look at this page: http://www.avertlabs.com/research/blog/

# posted by snipsnip : 02 December 2008 11:09

Thanks for noticing that - someone else pointed it out to me last week, and I already send them a comment, which they haven't got around to publishing yet. :-/Still, the US-Cert article to which it links mentions this blog. I'll also ask US-Cert to add my name and Emin's to the "Credits" section.

# posted by Nick Brown : 02 December 2008 12:11

Just thought I'd tell you what the /?? prefix meant in the filename-It's used in the CreateFile() function, which is used to open any file, for the purpose of allowing filename lengths longer then the defined maximum previously used in Pre-NT operating systems.

# posted by BC_programming : 03 December 2008 01:30

Thanks for that - a mystery which has been bothering me (not very much...) for 12 years is now solved!

# posted by Nick Brown : 03 December 2008 12:38

Okay, so i made the fle, gave it a ".reg" suffix, and double-clicked it.Windows immediately asked me what program to use to open the file (suggesting EditPad Lite).So what do i do? Do i need to save it in a perticular directory?Running XP, SP3.

# posted by mike weber/fairportfan : 04 December 2008 06:08

Mike,When asked what program to use, browse to C:\Windows or C:\Windows\System32 and choose "Regedit".Nick

# posted by Nick Brown : 04 December 2008 08:39

How do I implement this hack using REGEDIT?.REG MERGE functionality is broken on my computer. And it would just be interesting to see what it looks like in REGEDIT view.

# posted by G Laverne Flambeau : 06 December 2008 22:33

Start Regedit. Go into HKEY_LOCAL_MACHINE, SOFTWARE, Microsoft, Windows NT, CurrentVersion. Right-click on "IniFileMapping" and choose "New"/"Key". Name the new key "Autorun.inf". Go into "IniFileMapping" and "Autorun.inf". Double-click "(Default)" (in the right-hand pane) and enter "@SYS:DoesNotExist". (Leave my quote marks out, of course.)

# posted by Nick Brown : 06 December 2008 23:00

After implementing this hack as explained and attaching an external USB drive, the little AutoPlay window and subsequently the AutoRun window appeared.Please correct me if wrong. This reg hack does NOT stop AutoRun / AutoPlay from working. It simply neuters any AutoRun.inf file on any attached device.So the appearance of the AutoPlay and AutoRun windows after implementing this hack, are still expected. Is this correct?Thanks

# posted by Anonymous : 11 December 2008 06:43

Yes, that's correct.However, since is a substantial number of cases the required action on inserting the media - for example, autostarting the Setup.exe procedure on a software CD - is described by the Autorun.inf file, there will, in such cases, be a change in overall behaviour (you'll have to run setup.exe manually, or whatever).Quite a lot of memory worms don't run anything directly when you insert the stick. Instead, they install (via Autorun.inf) a handler in Windows Explorer's "File" menu, which causes the worm to spread when you explore the drive.

# posted by Nick Brown : 11 December 2008 10:30

hi thank you for posting a very good idea. i wanted to know, if u can kindly tell me, that actually most of the autorun and assosciated worm files normally hide themselves...is there any tool or registry way to make them unhide....so that i can delete them?thanks for answering in advance....

# posted by Anonymous : 13 December 2008 07:34

There's no easy answer. If you have such a virus/worm, you'll probably find pointers to at least the first part of it in the Autorun.inf files which it will typically leave in the root directory of your hard disk partition(s). But many of these worms leave other things in other places. Look at programs like Silent Runners or Hijack This! for ideas on how to hunt down auto-running software.

# posted by Nick Brown : 13 December 2008 11:35

I am trying to show the boss the danger of AUTORUN.INFI created an AUTORUN.INF on a USB stick. I plug the stick into the USB port. The AUTOPLAY menu appears but executable does not execute:[Autorun]OPEN=AUTORUN.EXEICON=AUTORUN.ICOThe AUTORUN.EXE and AUTORUN.ICO files are on the stick. This is a clean install of XP HOME with SP3. It does have Windows Defender and AVG Free on it but they do not show any alerts or events in the logs.

# posted by G Laverne Flambeau : 14 December 2008 20:31

No idea... sounds like you're already protected somehow. :-)

# posted by Nick Brown : 14 December 2008 20:44

Disabling AutorunNumber: TR08-004Date: 22 December 2008 http://www.publicsafety.gc.ca/prg/em/ccirc/2008/tr08-004-eng.aspx"Sys:DoesNotExist" is the only Recommended Solution.Mike

# posted by Anonymous : 31 December 2008 22:40

Sweet! Go Canada!

# posted by Nick Brown : 01 January 2009 23:55

@G Laverne FlambeauContrary to on CD's/DVD's, the "Open=" command does not cause the app to start on insertion of the USB stick. Provided that NoDriveTypeAutoRun under HKCU has the default XP value of 0x91, and no value exists under HKLM, this should do the trick:[autorun]shellexecute=autorun.exeuseautoplay=1

# posted by Bitwiper : 04 January 2009 18:18

Great work Nick.Can you help with this scenario?Say a person's system is running XP and has a virus scanner installed. Said scanner has a "scan removable media" function, and it's enabled.Is that a reliable enough scanner to protect me and to not have to install your autorun hack?If not, and a user chose to hold down the Shift key when inserting the media to prevent autorun, in what danger is that user now of an infection from USB stick?You said, "Quite a lot of memory worms don't run anything directly when you insert the stick. Instead, they install (via Autorun.inf) a handler in Windows Explorer's "File" menu, which causes the worm to spread when you explore the drive." Will a right-click on the drive in WinExplorer cause the worm to activate?The concern is that if we bypass autorun successfully by some method to be safe, and then want to virus scan the USB device, will that "touch" of the device activate the nasty on there?Thanks for all your great help!

# posted by buddy lembeck : 09 January 2009 21:22

Hi Buddy,>Is that a reliable enough scanner>to protect me and to not have to>install your autorun hack?Only if your anti-virus scanner is updated to include every new memory stick worm before they arrive at your site. I haven't heard of any which claim to apply generic protection to removable drives.We remove 3 or 4 viruses a week from our network, which we detect using a variety of home-brewed means. When we find a worm or infected file, we always submit to the superb site virustotal.com, which runs it past about 35 antivirus products. It's very common for only 10 of the 35 to detect it. So to have an 80% chance of catching the virus, you'd need to have five scanners on your system, each updated daily. How much was that migration to a Linux desktop going to cost you? :-)>If not, and a user chose to hold>down the Shift key when inserting>the media to prevent autorun, in>what danger is that user now of>an infection from USB stick?My understanding is that holding down Shift does work to prevent any Autorun function. But realistically, your users are not going to remember to do this in 100.000000% of cases.>Will a right-click on the drive>in WinExplorer cause the worm to>activate?Yes, that's typically how these viruses work. At that point the worm will copy itself to the root of all the drives on the system and install an Autorun.inf on there too. With a network drive shared by 500 people, you can have 500 copies on your LAN by the end of business.Good luck!Nick

# posted by Nick Brown : 09 January 2009 23:28

I think I've found an easier way to prevent autorun. I just tried this on an XP Pro SP3 computer, and it worked when I tried it with a SanDisk. If there's anything wrong with this method, which is supposed to work on XP/Vista, please let me know.Go to the following Registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom Double-click on Autorun, and you'll see a default value of 1. Change it to 0 (zero). Then restart the computer. After you go to all this trouble, don't double-click that autorun.inf file on the disk—it can still run if you do. [Thanks to PC Magazine for this tip, which I found at:http://www.pcmag.com/article2/0,2817,2254345,00.asp].

# posted by Daveed : 10 January 2009 12:24

Interesting. I'd expect this to have no effect on a device which is not being handled by the CD-Rom driver. Some research would appear to be indicated. :)

# posted by Nick Brown : 11 January 2009 00:51

Nick,I also thought there would bea problem with any device other than a DVD/CD, but as I said, I tried it with a SanDisk disk-on-key. When the value is 1, the U3 software pops up. When the value is 0, the U3 software does not pop up, but all the partitions, including the one containing U3 are still visible. This method is easily reversible.I read that Vista SP2 is going to correct this problem, making autrun not work by default, by I haven't yet had the opportunity to check the beta version and see if and how it's been accomplished.

# posted by Daveed : 11 January 2009 19:35

I appreciate all the info-I'm a network chief cook/bottlewasher/genie/fire extinguisher so you've really cut short my work :)But there is one thing I haven't found any info on yet:How does the worm get into the pen drive in the first place???Thanks again!

# posted by Carole G : 17 January 2009 19:24

It gets onto the pen drive because the pen drive was put into an infected PC. The pen drive then infects other PCs.If the pen drive is the chicken then the PC is the egg. And as everyone knows, the egg came first (the chicken is just a device for making more eggs), which in this case means that the first PC "infected" in the world was the one belonging to the person who wrote the worm.

# posted by Nick Brown : 18 January 2009 00:31

Regarding use of Group Policy to apply registry changes, I'm attaching a sample script I wrote to enable PMTU Blackhole Detection and PMTU Discovery on Windows systems. The script was based upon a killbit script found through a link on the ISC - some residual comments remain. Saved as a command file in the netlogon directory the command file is executed as a startup script in Group Policy Computer settings. The script requires a parameter of "enable" or "disable" to set the values on or off. It could easily be modified to use the hack Nick posted above as well as the reversal hack should life go horribly wrong in your environmentScript follows:@echo offREM Print some debugging infodate /t >> %systemdrive%\PMTU.outtime /t >> %systemdrive%\PMTU.outREM Start writing the header to our registry script echo Windows Registry Editor Version 5.00 > PMTU.reg echo. >> PMTU.reg if "%1" == "" (echo ERROR: You must specify either "enable" or "disable" on the command line for this script to have any effect >> %systemdrive%\PMTU.out goto cleanup)if /i disable == %1 (echo Disabling PMTU Options >> %systemdrive%\PMTU.out REM Finish creating our reg file. This one sets the kill bit echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters] >> PMTU.reg echo "EnablePMTUBHDetect "=dword:00000000 >> PMTU.reg echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters] >> PMTU.reg echo "EnablePMTUDiscovery"=dword:00000000 >> PMTU.reg REM Call the reg file regedit /s PMTU.reg >> %systemdrive%\PMTU.out goto cleanup) else (if /i enable == %1 (echo Enabling PMTU Options >> %systemdrive%\PMTU.out REM Finish creating our reg file. This one sets the kill bit echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters] >> PMTU.reg echo "EnablePMTUBHDetect "=dword:00000001 >> PMTU.reg echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters] >> PMTU.reg echo "EnablePMTUDiscovery"=dword:00000001 >> PMTU.reg REM Call the reg file regedit /s PMTU.reg >> %systemdrive%\PMTU.out goto cleanup) else (echo ERROR: You must specify either "enable" or "disable" on the command line for this script to have any effect >> %systemdrive%\PMTU.out goto cleanup)) :cleanupREM Delete the registry scriptDEL PMTU.regecho Finished processing >> %systemdrive%\PMTU.outecho. >> %systemdrive%\PMTU.out echo == >> %systemdrive%\PMTU.out

# posted by Alan : 18 January 2009 03:35

> > Buddy Lembeck:> > If not, and a user chose to hold down the Shift key> > when inserting the media to prevent autorun, in what> > danger is that user now of an infection from USB stick?> Nick Brown:> My understanding is that holding down Shift does work> to prevent any Autorun function.XP: yes. However, under Vista this functionality was removed, see the bottom question at Microsoft's AutoPlay FAQ.

# posted by Bitwiper : 19 January 2009 23:05

Nick, thanks for the trick, it worked for me. However, I have some weird news: I caught Downadup from a Linux machine. I dunno how, but I've sticked my pendrive only in Linux boxes today at a fair and got infected when I sticked it up in my laptop back home. Have you heard about it before?

# posted by Fabio : 21 January 2009 07:03

Fabio, attempting to trace where you actually got it from will be impossible. My guess would be either that Wine is getting very good indeed, or that maybe the Linux machine created a share on the stick and Samba was running. Or maybe your laptop was already infected (there presumably has to be a Windows machine in the chain somewhere).Otherwise, I recommend this for light entertainment on a related matter: http://www.youtube.com/watch?v=xKZR3Bcj4jw :-)))

# posted by Nick Brown : 21 January 2009 08:42

>> The second @ is part of the final value which goes into the registry. It means "what follows is a pointer - follow it". >> We want Windows to follow the pointer and finish up at a dead end, where it will say, "OK, this entire INI file maps to nothing".Fair enough, however I am looking through a few dozen other "IniFileMapping" entries in the registry, and none of them starts with an "@". Some start with a hash (#), and some just say "SYS:...". Would you please provide a reference where you saw the "@" (at) symbol used before "SYS:" ?Thanks in advance!

# posted by Michael : 21 January 2009 16:21

The various prefix characters are explained here: http://technet.microsoft.com/en-us/library/cc722567.aspx.The @ is completely essential to this fix. If you don't have it, since the registry key doesn't exist, Windows will read the Autorun file, which is exactly what you're trying to avoid.The reason you haven't seen the @ character is because generally, IniFileMapping provides a way to set global values which override whatever is in the INI (etc; in our case, INF) file. If @ isn't provided and the file exists, then it will be read unless the corresponding value tag exists in the registry. Normally this is what you want, because normally you aren't trying to prevent stuff from happening. In the case which we're interested in here, without using @ you would have to provide registry values for every possible tag value that could appear in the Autorun.INF file.

# posted by Nick Brown : 21 January 2009 16:45

Nick, thank you for the great tip. I tried the undo solution you posted above but I can't seem to get it to work. I copied the two lines into a notepad file and saved it as undo.reg, and double-clicked it to enter it into the registry. The autorun.ini file is still not being executed. Am I missing a step somewhere? I also rebooted.

# posted by Anonymous : 21 January 2009 21:24

Hmmm. I suggest you open REGEDIT and navigate down to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping. You should not see a subkey ("folder") called Autorun.inf. If you see it, delete it. That's all the 2-line file does. And as far as I know, Autorun.inf should then work immediately, but certainly it should work after a reboot.

# posted by Nick Brown : 21 January 2009 23:51

Nick, thank you for the great work first. I have a question. What will happen if the **HKLM\SOFTWARE\DoesNotExist** key exists? Do you think there is a possibility for exploitation?

# posted by Vlad : 22 January 2009 00:23

Vlad, that's an excellent point. It would require an attacker to have previously had access to your PC to create that key. But if you want to be extra-secure, generate a random 32-letter sequence and use that in place of "DoesNotExist". Or, write a sentence which describes exactly what you think of Microsoft's attitude to security and use the last letter of each word of the sentence. :-)

# posted by Nick Brown : 22 January 2009 00:29

I suggest showing exactly how this hack is supposed to look using a screen shot pic, as one views it in regedit. The @SYS and even the rest may leave some in an unknown or even false sense of security. A screen shot of the regkey would be very useful.The Cdrom thing I think is one of the MS suggestions, no? I've seen it before. I don't have it set to 0 now so my guess is I used it before elsewhere and it worked but forgot about it (I am using an MS hack now I'm sure), or it didn't work well enough for me to continue using it. I always have been concerned about the USB drive - I know my discs won't autorun, but I wasn't sure about USB. I could have but an autorun.inf on it, right - coulda/woulda/shoulda -> didnta.

# posted by Anonymous : 22 January 2009 04:33

continued:P.S. I recall itunes having a problem recognizing changed CDs in the past when I had some sort of autorun hack in, and I think it was the above-mentioned Cdrom hack. I had to close/reopen itunes to get it to recognized a changed CD. That may be the reason I don't use the Cdrom hack anymore.

# posted by Anonymous : 22 January 2009 04:35

On further reflection, it would of course be even easier for an attacker to delete the IniFileMapping\Autorun.inf key - after all, there's a 2-line registry file to do it somewhere above this post. :-)So if you want to rely on this once the bad guys wake up to it, make it part of System Policy, or perhaps try and tie down the registry permissions so it can't be deleted (good luck with that).

# posted by Nick Brown : 22 January 2009 08:42

I'll look into posting a screen shot. One problem I see with that is that to verify it against your system, you have to open Regedit interactively, and if you're sufficiently well-informed to do that, you should know what you're looking for. But maybe I'm being too picky.A couple of people have suggested that I post the .REG file, ready to go. The problem there is that as soon as someone clicks it, I become responsible for every problem on their PC for the next year...

# posted by Nick Brown : 22 January 2009 13:16

Thanks Nick, good tech tool to have to help to keep the infections down. If you get an infection, this fellow has a good cure. I work at a college and we have been using this to keep our machines clean.http://www.en.mygeekside.com/?cat=8MAT

# posted by Anonymous : 22 January 2009 21:04

Nick, I played a little bit with the *Autorun.inf* key. First, I observed the permanent behavior change in the SanDisk Cruzer U3. After the key was applied the Cruzer started opening two Explorer windows, one for a CD-ROM emulation partition, another for a data partition. Double click on the U3 executable opened another Explorer window in the "...\Documents and Settings\{user-name}\Application Data\U3\{16-digit-hex-number}" folder. The behavior was very stable. Removing the *MountPoints2* did not help. After the *Autorun.inf* key was removed nothing changed until the system was rebooted. Even after reboot the behavior was different than initially. The U3 application started as it should but on top of that the Explorer window popped up with the U3 data partition. I have no problem with that but just wanted to document that the *Autorun.inf* key introduction affects the U3 application even after the key is removed from the registry and the OS was restarted. What I have a problem with is that U3 application stops working when the *Autorun.inf* key is present. Second, I tried playing with the *DoesNotExist* key and observed some very interesting changes in double-click on autorun.inf behavior. I think your suggestion to try the registry permissions is very valid. Should probably be done on both keys. As you said, "the world is our beta site" ;)

# posted by Vlad : 23 January 2009 01:51

"and if you're sufficiently well-informed to do that, you should know what you're looking for"Cut-n-paste only works if you cut and paste it all. I left off trailing quote. Try it. Unless you know what it's supposed to look like, you don't know what you're looking at. It took me more than once (okay, about four times) to notice why I wasn't seeing the Default with any value.

# posted by Anonymous : 23 January 2009 04:28

Wow, this is really great!!Can I translate it to Indonesian so I can give this information to my fellows Indonesian?Terimakasih banyak! (Thanks a lot! in Indonesian)

# posted by Rudra : 25 January 2009 06:42

Hi Rudra,It's no problem for me if you want to post a translation. Just please include a link to the original article.Thanks,Nick

# posted by Nick Brown : 25 January 2009 12:40

Nick, Your solution to the problem is great. To me, it's better, simpler and more ironclad than the assorted work-arounds from Microsoft. I wrote more about it on my Computerworld blog The best way to disable Autorun for protection from infected USB flash driveshttp://blogs.computerworld.com/the_best_way_to_disable_autorun_to_be_protected_from_infected_usb_flash_drivesAnd, I created a test autorun.inf file too so people can test whether their PCs are vulnerable to the assorted autorun.inf tricks. Test your defenses against malicious USB flash driveshttp://blogs.computerworld.com/test_your_defenses_against_malicious_usb_flash_drives

# posted by Michael Horowitz : 01 February 2009 18:50

Hi..This looks great but quick query.. G: for my laptop is always my USB mobile broadband toggle, which kicks off with autorun.inf (triggers autorun.exe), and obviously I still want that to happen..Without trying it first, will this overall fix generally then render the broadband toggle mute on start up..?? I don't want to have to physically run this each time..Alternatively, does the autorun program guard (haven't looked at it) allow more control in this direction (ie allowing g: to work but blocking the rest of the drives, for example?)Many thanks..Peter

# posted by Anonymous : 04 February 2009 20:05

Peter, I'm afraid you're going to lose that functionality if you apply this patch. Try creating a shortcut to G:\AUTORUN.EXE on your desktop and double-click that when you put the mobile broadband key in the PC.Alternatively, you could build a REG file which undoes the Autorun.inf hack before you insert the key and another which puts it back afterwards, but that's even more clicking.

# posted by Nick Brown : 04 February 2009 22:43

Hi Nick,That's interesting.. the desktop shortcut sounds like the obvious route with this, if keeping sound control. Actually it seems a lot more intuitive in any case; it reminds me of a time when everything was not in dumb down mode!!Many thanks for your help...Regards,Peter

# posted by Anonymous : 04 February 2009 23:34

Nick, please help!I have my USB infected with Win32.Virut.Q. I want to format my flash USB drive but I don't know how to do it without infecting my computer. Although, you showed me how to avoid automatic infection I will still infect my computer if I right-click the flash drive in order to format it. I tried to do it from DOS but DOS doesn't see the USB drive at all. Is there any way to format my hard drive, or should I start looking for a new one? Thanks in advance.

# posted by Anonymous : 07 February 2009 23:58

It sounds like your best bet would be to find another computer (with the @SYS:DoesNotExist patch installed) and clean up the USB stick on that. Then use a free anti-virus tool to clean up your PC.Something else which might just work is to open a command prompt, go into the memory stick (say E:) and run these commands:del /a /f \autorun.infmd \autorun.infecho x>\autorun.inf\a.txt(Most viruses won't delete a directory called autorun.inf in order to create the file with the same name.)Good luckNick

# posted by Nick Brown : 08 February 2009 01:11

Post a Comment

Links to this post:

Create a Link

<< Home

I Power Blogger

Subscribe to Posts [Atom]

Archives

- October 2007
- November 2007
- January 2008
- March 2008
- September 2008
- January 2009

HardwareSoftwareMusic & MediaNetworksSecurityPublic SectorBusinessScienceOdds & Sods

CrimeEnterprise SecurityAnti-VirusSpamIDSpyware
Cash 'n' CarrionNewslettersFeeds
JobsWhitepapers

Disabling Windows Autorun - there's a right way and a wrong way

Track this topic Print story Post comment

Redmond's Downadup protection

By Dan Goodin in San Francisco • Get more from this author

Posted in Security, 22nd January 2009 01:24 GMT

After some confusion about exactly how Windows users can protect themselves against a prolific computer worm called Downadup, Microsoft security watchers are once again reiterating the steps for disabling the Autorun feature.

Downadup has managed to infect an estimated 9 million machines at last count using multiple attack vectors. Two of those vectors are USB flash drives and mapped network drives, which are booby-trapped with files that compromise machines that are configured to automatically connect to CD and DVD drives and other types of media.

Disabling the feature has long been a good idea, as the 2005 fiasco involving the Sony rootkit made clear. Those unfortunate enough to have Autorun configured - and the feature is turned on by default - found their machines were secretly infected by digital rights management software after playing certain Sony BMG CDs on their PCs. With Downadup spreading like wildfire, disabling Autorun is an even better idea than ever.

For the low-down on exactly how that's done, head over Microsoft Knowledge Base article 953252. The May 2008 article revises an earlier Knowledge Base item that didn't completely remove the dangers inherent in Autorun.

It would appear that the US Computer Emergency Readiness Team referred to that older article when it warned the world on Wednesday that Microsoft's instructions "are not fully effective" and "could be considered a vulnerability."

The group went on to give labored instructions for changing registry settings that disable all Autorun features. We're sure they work, but we can't imagine Aunt Mildred having the slightest clue how to pull them off. ®

71 commentsPost a comment

Related stories

- Conficker Autoplay ruse gets teeth into Windows 7 (20 January 2009)
- US Army bans USB devices to contain worm (20 November 2008)
- Stealthy malware expands rootkit repertoire (30 September 2008)
- Malware hitches a ride on digital devices (11 January 2008)
- Chinese Trojan on Maxtor HDDs spooks Taiwan (12 November 2007)
- Kaspersky: Maxtor markets password-pilfering Dutch disk drives (19 September 2007)
- USB drives pose insider threat (27 June 2006)

Whitepapers

Article ID: 953252 - Last Review: February 2, 2009 - Revision: 4.0

How to correct "disable Autorun registry key" enforcement in Windows

SUMMARY

The registry key guidance that is offered in Technet article 91525 (http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/91525.mspx?mfr=true) did not correctly disable AutoRun features. After you set the registry keys to disable these features as described in this Technet article, the AutoRun capabilities, the Double Click feature, and the Contextual Menu feature continue to function as if they were not set. This article describes how to obtain updates that correct these registry key settings.

This article is applicable for all supported editions of the following products:

- Microsoft Windows 2000
- Windows XP Service Pack 2
- Windows Server 2003 Service Pack 1
- Windows Server 2003 Service Pack 2
- Windows Vista
- Windows Server 2008

MORE INFORMATION

The purpose of Autorun

The main purpose of Autorun is to provide a software response to hardware actions that you start on a computer. Autorun has the following features:

- Double Click
- Contextual Menu
- AutoPlay

These features are typically called from removable media or from network shares. During AutoPlay, the Autorun.inf file from the media is parsed. This file specifies which commands the system runs. Many companies use this functionality to start their installers.

Prerequisites to disable Autorun capabilities

To disable Autorun capabilities, you must install the following updates:

- Update for Windows XP (KB950582) http://www.microsoft.com/downloads/details.aspx?FamilyId=CC4FB38C-579B-40F7-89C4-1721D7B8DAA5 (http://www.microsoft.com/downloads/details.aspx?FamilyId=CC4FB38C-579B-40F7-89C4-1721D7B8DAA5)
- Update for Windows Server 2003 for Itanium-based Systems (KB950582) http://www.microsoft.com/downloads/details.aspx?FamilyId=5795F63E-1FD9-4A13-9650-1015E14B6D11 (http://www.microsoft.com/downloads/details.aspx?FamilyId=5795F63E-1FD9-4A13-9650-1015E14B6D11)
- Update for Windows Server 2003 x64 Edition (KB950582) http://www.microsoft.com/downloads/details.aspx?FamilyId=E8507286-CDF8-4BCB-AFC5-9734FE772C53 (http://www.microsoft.com/downloads/details.aspx?FamilyId=E8507286-CDF8-4BCB-AFC5-9734FE772C53)
- Update for Windows Server 2003 (KB950582) http://www.microsoft.com/downloads/details.aspx?FamilyId=705305E5-7060-4236-B5D2-40CA63A967FB (http://www.microsoft.com/downloads/details.aspx?FamilyId=705305E5-7060-4236-B5D2-40CA63A967FB)
- Update for Windows XP x64 Edition (KB950582) http://www.microsoft.com/downloads/details.aspx?FamilyId=21A0124C-6F50-4281-923E-E2B28068147A (http://www.microsoft.com/downloads/details.aspx?FamilyId=21A0124C-6F50-4281-923E-E2B28068147A)

- Update for Windows 2000 (KB950582) http://www.microsoft.com/downloads/details.aspx?FamilyId=C192EDCF-CA3D-44E3-8ECC-49C5F4DA5405 (http://www.microsoft.com/downloads/details.aspx?FamilyId=C192EDCF-CA3D-44E3-8ECC-49C5F4DA5405)

Note Windows Vista and Windows Server 2008-based systems must have update 950582 (Security bulletin MS08-038 (http://www.microsoft.com/technet/security/Bulletin/MS08-038.mspx)) installed to take advantage of the registry key settings that disable Autorun.

As soon as the prerequisites are installed, follow these steps to disable Autorun.

How to use Group Policy settings to disable all Autorun features

Windows Server 2008 or Windows Vista

1. Click **Start** , type **Gpedit.msc** in the **Start Search** box, and then press ENTER.

   If you are prompted for an administrator password or for a confirmation, type the password, or click **Allow**.
2. Under **Computer Configuration**, expand **Administrative Templates**, expand **Windows Components**, and then click **Autoplay Policies**.
3. In the **Details** pane, double-click **Turn off Autoplay**.
4. Click **Enabled**, and then select **All drives** in the **Turn off Autoplay** on box to disable Autorun on all drives.
5. Restart the computer.

Windows Server 2003, Windows XP, and Windows 2000

1. Click **Start**, click **Run**, type **Gpedit.msc** in the **Open** box, and then click **OK**.
2. Under **Computer Configuration**, expand **Administrative Templates**, and then click **System**.
3. In the **Settings** pane, right-click **Turn off Autoplay**, and then click **Properties**.

   Note In Windows 2000, the policy setting is named **Disable Autoplay**.
4. Click **Enabled**, and then select **All drives** in the **Turn off Autoplay** box to disable Autorun on all drives.
5. Click **OK** to close the **Turn off Autoplay Properties** dialog box.
6. Restart the computer.

How to selectively disable specific Autorun features

To selectively disable specific Autorun features, you must modify the **NoDriveTypeAutoRun** value under the following registry key subkey: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\ How you modify this subkey depends on the Autorun feature that you want to disable. For more information about Autorun registry key values, visit the following Microsoft TechNet Web page: http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/91525.mspx?mfr=true (http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/91525.mspx?mfr=true) Autorun is also known as AutoPlay. The following table shows the settings for the **NoDriveTypeAutoRun** registry value.

Value	Meaning
0x1	Disables AutoPlay on drives of unknown type
0x4	Disables AutoPlay on removable drives
0x8	Disables AutoPlay on fixed drives
0x10	Disables AutoPlay on network drives
0x20	Disables AutoPlay on CD-ROM drives
0x40	Disables AutoPlay on RAM disks
0x80	Disables AutoPlay on drives of unknown type

0xFF	Disables AutoPlay on all kinds of drives

The default value for **NoDriveTypeAutoRun** varies for different Windows-based operating systems. These default values are listed in the following table.

Operating system	Default value
Windows Server 2008 and Windows Vista	0x91
Windows Server 2003	0x95
Windows XP	0x91
Windows 2000	0x95

Registry key that is used to control the behavior of the current update

Important This section, method, or task contains steps that tell you how to modify the registry. However, serious problems might occur if you modify the registry incorrectly. Therefore, make sure that you follow these steps carefully. For added protection, back up the registry before you modify it. Then, you can restore the registry if a problem occurs. For more information about how to back up and restore the registry, click the following article number to view the article in the Microsoft Knowledge Base: 322756 (http://support.microsoft.com/kb/322756/) How to back up and restore the registry in Windows

All the fixes in the current update for Windows XP and for Windows Server 2003 have been included in the following two registry subkeys.

Note Values were concentrated in these subkeys so that you can revert to the previous configuration if it is required. Windows 2000 and Windows Vista do not use these registry subkeys.

HonorAutorunSetting registry subkeys

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\

Registry Value

Value	Data type	Range	Default value
HonorAutorunSetting	REG_DWORD	0x0–0xFF	0x01

When you install update 950582, the HonorAutorunSetting registry key is created only in the HKEY_LOCAL_MACHINE registry hive. The registry key has a default value of 0x1. This value enables the functionality that is present in the current update. Before you install the current update, this registry key is not present in the system. You can obtain prepackage installation Autorun behavior by manually setting the registry key to 0. (To do this, type **0** instead of **1**, in step 6 of the procedures to manually set the registry key.) If the registry key is present in both the HKEY_LOCAL_MACHINE registry hive and the HKEY_CURRENT_USER registry hive, the HKEY_LOCAL_MACHINE hive setting takes priority.

How to set the HonorAutorunSetting registry key manually

Windows Server 2003 and Windows XP

1. Click **Start**, and then click **Run**.
2. In the **Open** box, type **regedit**, and then click **OK**.
3. Locate and then click the following registry subkey: **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\Explorer\**
4. Right-click in the right pane, point to **New**, and then click **DWORD Value**.
5. Type **HonorAutorunSetting**, and then press ENTER.
6. In the **Value data** box, type **1**, click **Hexadecimal**, if it is not already selected, and then click **OK**.
7. Exit Registry Editor.
8. Restart the system for the new settings to take effect.

WORKAROUND

We have tested the following workarounds.

Workaround 1: To prevent creating Autorun.inf files on shares

To prevent the Autorun feature from being invoked and to keep any programs from writing Autoun.inf files to mapped network drives, follow these steps:

1. Delete any Autorun.inf files from the root of a mapped network drive.
2. Do not give anyone **Create** rights to the root of a mapped network drive.

Note After you implement this workaround, Autorun features will not be available from network drives.

Workaround 2: To disable the use of USB storage devices

The following Microsoft Knowledge Base article contains two methods to prevent users from connecting to a USB storage device: 823732 (http://support.microsoft.com/kb/823732/) How to disable the use of USB storage devices
Note After you implement this workaround, USB storage devices no longer function on systems in which these changes are applied.

APPLIES TO

- Windows Vista Enterprise 64-bit Edition
- Windows Vista Home Basic 64-bit Edition
- Windows Vista Home Premium 64-bit Edition
- Windows Vista Ultimate 64-bit Edition
- Windows Vista Business 64-bit Edition
- Windows Vista Business
- Windows Vista Enterprise
- Windows Vista Home Basic
- Windows Vista Home Premium
- Windows Vista Starter
- Windows Vista Ultimate
- Windows Vista Service Pack 1
- Microsoft Windows XP Professional x64 Edition
- Microsoft Windows Server 2003 Service Pack 2, when used with:
  Microsoft Windows Server 2003, Standard Edition (32-bit x86)
  Microsoft Windows Server 2003, Enterprise Edition (32-bit x86)
  Microsoft Windows Server 2003, Web Edition
  Microsoft Windows Server 2003, Datacenter Edition (32-bit x86)
  Microsoft Windows Server 2003, Standard x64 Edition
  Microsoft Windows Server 2003, Enterprise x64 Edition
  Microsoft Windows Server 2003, Datacenter x64 Edition
  Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems
  Microsoft Windows Server 2003, Datacenter Edition for Itanium-Based Systems
  Microsoft Windows XP Professional x64 Edition
- Microsoft Windows Server 2003 Service Pack 1, when used with:
  Microsoft Windows Server 2003, Standard Edition (32-bit x86)
  Microsoft Windows Server 2003, Enterprise Edition (32-bit x86)
  Microsoft Windows Server 2003, Web Edition
  Microsoft Windows Server 2003, Datacenter Edition (32-bit x86)
  Microsoft Windows Server 2003, Standard x64 Edition
  Microsoft Windows Server 2003, Enterprise x64 Edition
  Microsoft Windows Server 2003, Datacenter x64 Edition

Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems
Microsoft Windows Server 2003, Datacenter Edition for Itanium-Based Systems

- Microsoft Windows 2000 Service Pack 4, when used with:
  Microsoft Windows 2000 Professional Edition
  Microsoft Windows 2000 Server
  Microsoft Windows 2000 Advanced Server
  Microsoft Windows 2000 Datacenter Server
- Microsoft Windows XP Service Pack 2, when used with:
  Microsoft Windows XP Professional
  Microsoft Windows XP Home Edition
  Microsoft Windows XP Tablet PC Edition
- Windows Server 2008 Datacenter without Hyper-V
- Windows Server 2008 Enterprise without Hyper-V
- Windows Server 2008 for Itanium-Based Systems
- Windows Server 2008 Standard without Hyper-V
- Windows Server 2008 Datacenter
- Windows Server 2008 Enterprise
- Windows Server 2008 Standard
- Windows Web Server 2008

Keywords: atdownload kbbug kbexpertiseinter kbfix kbpubtypekc kbsecbulletin kbsecurity kbsecvulnerability KB953252

Skip to content

Home | FAQ | Contact | Privacy & Use

US-CERT: United States Computer Emergency Readiness Team

- Security Publications
- Alerts and Tips
- Related Resources
- About Us

Search US-CERT: GO customize

National Cyber Alert System

Technical Cyber Security Alert TA09-020A Archive

Microsoft Windows Does Not Disable AutoRun Properly

Original release date: January 20, 2009 Last revised: January 29, 2009
Source: US-CERT

Systems Affected

- Microsoft Windows

Overview

Disabling AutoRun on Microsoft Windows systems can help prevent the spread of malicious code. However, Microsoft's guidelines for disabling AutoRun are not fully effective, which could be considered a vulnerability.

I. Description

Microsoft Windows includes an AutoRun feature, which can automatically run code when removable devices are connected to the computer. AutoRun (and the closely related AutoPlay) can unexpectedly cause arbitrary code execution in the following situations:

1. A removable device is connected to a computer. This includes, but is not limited to, inserting a CD or DVD, connecting a USB or FireWire device, or mapping a network drive. This connection can result in code execution without any additional user interaction.

   Map Network Drive

2. A user clicks the drive icon for a removable device in Windows Explorer. Rather than exploring the drive's contents, this action can cause code execution.

   Windows Explorer Device Icon

3. The user selects an option from the AutoPlay dialog that is displayed when a removable device is connected.

   AutoPlay

Malicious software, such as W32.Downadup, is using AutoRun to spread. Disabling AutoRun, as specified in the CERT/CC Vulnerability Analysis blog, is an effective way of helping to prevent the spread of malicious code.

The Autorun and NoDriveTypeAutorun registry values are both ineffective for fully disabling AutoRun capabilities on Microsoft Windows systems. Setting the Autorun registry value to 0 will not prevent newly connected devices from automatically running code specified in the `Autorun.inf` file. It will, however, disable Media Change Notification (MCN) messages, which may prevent Windows from detecting when a CD or DVD is changed. According to Microsoft, setting the NoDriveTypeAutorun registry value to `0xFF` "disables Autoplay on all types of drives." Even with this value set, Windows may execute arbitrary code when the user clicks the icon for the device in Windows Explorer.

II. Impact

By placing an `Autorun.inf` file on a device, an attacker may be able to automatically execute arbitrary code when the device is connected to a Windows system. Code execution may also take place when the user attempts to browse to the software location with Windows Explorer.

III. Solution

Disable AutoRun in Microsoft Windows

To effectively disable AutoRun in Microsoft Windows, import the following registry value:

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\IniFileMapping\Autorun.inf]
@="@SYS:DoesNotExist"
```

To import this value, perform the following steps:

1. Copy the text

2. Paste the text into Windows Notepad
3. Save the file as `"autorun.reg"`
   Note: In certain circumstances, Notepad may automatically add a `.txt` extension to saved files. To ensure that the file is saved with the proper extension, select **All Files** in the "Save as type:" section of the "Save As" dialog.
4. Navigate to the file location
5. Double-click the file to import it into the Windows registry

Microsoft Windows can also cache the AutoRun information from mounted devices in the `MountPoints2` registry key. We recommend restarting Windows after making the registry change so that any cached mount points are reinitialized in a way that ignores the `Autorun.inf` file. Alternatively, the following registry key may be deleted:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2
```

Once these changes have been made, all of the AutoRun code execution scenarios described above will be mitigated because Windows will no longer parse `Autorun.inf` files to determine which actions to take. Further details are available in the CERT/CC Vulnerability Analysis blog. Thanks to Nick Brown and Emin Atac for providing the workaround and to Aryeh Goretsky for pointing out a possible issue with Notepad appending a .txt file extension.

Update:

Microsoft has provided support document KB953252, which describes how to correct the problem of NoDriveTypeAutoRun registry value enforcement. After the update is installed, Windows will obey the NoDriveTypeAutorun registry value. Note that this fix has been released via Microsoft Update to Windows Vista and Server 2008 systems as part of the MS08-038 Security Bulletin. Windows 2000, XP, and Server 2003 users must install the update manually. Our testing has shown that installing this update and setting the NoDriveTypeAutoRun registry value to `0xFF` will disable AutoRun as well as the workaround described above.

IV. References

- The Dangers of Windows AutoRun - <http://www.cert.org/blogs/vuls/2008/04/the_dangers_of_windows_autorun.html>
- US-CERT Vulnerability Note VU#889747 - <http://www.kb.cert.org/vuls/id/889747>
- Nick Browns blog: Memory stick worms - <http://nick.brown.free.fr/blog/2007/10/memory-stick-worms>
- TR08-004 Disabling Autorun - <http://www.publicsafety.gc.ca/prg/em/ccirc/2008/tr08-004-eng.aspx>
- How to correct disable Autorun registry key encforcement in Windows - <http://support.microsoft.com/kb/953252>
- Microsoft Security Bulletin MS08-038 - <http://www.microsoft.com/technet/security/Bulletin/MS08-038.mspx>
- How to Enable or Disable Automatically Running CD-ROMs - <http://support.microsoft.com/kb/155217>
- NoDriveTypeAutoRun - <http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/91525.mspx>
- Autorun.inf Entries - <http://msdn.microsoft.com/en-us/library/bb776823(VS.85).aspx>
- W32.Downadup - <http://www.symantec.com/security_response/writeup.jsp?docid=2008-112203-2408-99>
- MS08-067 Worm, Downadup/Conflicker - <http://www.f-secure.com/weblog/archives/00001576.html>
- Social Engineering Autoplay and Windows 7 - <http://www.f-secure.com/weblog/archives/00001586.html>

Feedback can be directed to US-CERT.

Revision History

January 20, 2009: Initial release
January 21, 2009: Added reference and details for Microsoft KB953252
January 29, 2009: Added information about Notepad and double file extensions

Last updated January 29, 2009

Print This Document

Home | FAQ | Contact | Privacy & Use | Get Adobe Reader US-CERT is part of the Department of Homeland Security

Top Story, November 8, 2007

One quick trick prevents AutoRun attacks

By Scott Dunn

The AutoRun function in Windows can launch installers and other programs automatically when you insert a CD or flash drive, but this convenience poses a serious security risk.

Unfortunately, simply turning off AutoPlay, a separate feature, isn't enough to prevent AutoRun from introducing a rogue program into your system.

AutoRun starts Windows programs automatically

Every recent version of Windows has features known as AutoPlay and AutoRun. These functions are designed to launch applications automatically from a external device containing the necessary AutoRun information. This is what causes an installer window to pop up when you insert a software disc into your CD or DVD drive, for example, or makes a pop-up menu icon appear in the taskbar tray when you insert a USB flash drive. (In some cases, the action doesn't occur until you double-click the flash drive icon in Windows Explorer.)

When a disc is inserted or a drive is connected to your system, Windows looks in the root directory of the new disc or drive for a file named **autorun.inf.** If found, Windows executes the instructions in that file.

For example, an **autorun.inf** file on a CD might contain a line that reads **open=setup.exe.** This tells your computer to launch a setup program as soon as the CD is inserted into the drive.

However convenient this might be, unfortunately, AutoRun also opens a huge door for viruses, Trojan horses, and worms. All it takes is a USB flash drive with an **autorun.inf** file and an executable in its root. Once inserted, a worm launched in this manner can infect every disk partition it finds, jumping from computer to computer as network users connect to an infected drive.

Shutting down AutoPlay is not a fix

In both Windows XP and Vista, the default for USB flash drives is to prompt the user for a decision if **autorun.inf** tries to launch a program. Inserting a CD or DVD into a drive, however, defaults to running any **autorun.inf** file that may be present.

In XP, you can change the defaults for AutoPlay on a given drive by right-clicking the drive in Windows Explorer and choosing Properties. Click the AutoPlay tab and use the controls there to change the settings for different types of media. Making changes in this dialog box, however, has no effect in preventing **autorun.inf** from being executed.

In Vista, end users can choose one of several options, even for software programs that use **autorun.inf:** (1)

always launch the program, (2) always open a listing of the disc in a Windows Explorer window, (3) always prompt for a choice, or (4) take no action.

Unfortunately, none of the above steps can safeguard you against a malicious **autorun.inf** on removable media. I'm no hacker, but I was able in just a few minutes to make an AutoRun file that would run, even with AutoPlay disabled in XP and "take no action" selected in Vista.

The exploit involves creating an **autorun.inf** file that adds a new default command to a USB flash drive's context menu. If you have "take no action" selected in Vista, the flash drive doesn't automatically launch any programs when first inserted. But double-clicking the flash drive icon in My Computer, for example, is all it takes to launch whatever commands are in **autorun.inf** (which the attacker has made the default command, in place of Open). The steps are documented at Daily Cup of Tech.

A clever hacker could make a worm that (1) spreads itself to all your drives when launched in this manner and then (2) displays the drive contents in a window, as expected. This would make it appear that nothing unusual had happened.

Block AutoRun for all devices all the time

You might think that you could protect yourself from AutoRun by using two keys in the Registry known as NoDriveAutoRun and NoDriveTypeAutoRun.

However, self-described "low-budget hacker" Nick Brown points out that these keys can be overridden. A Registry key named MountPoints2 stores information about all USB flash drives and other removable media that have ever been connected to your computer. Brown says this cache overrides the Registry settings that turn off AutoRun.

The solution is to globally block **autorun.inf** files from executing, without trying to use the dialog boxes in XP and Vista to do this. Here's the procedure:

Step 1. Start Notepad or another text editor.

Step 2. Copy the following text from this page and paste it into your text editor (everything between the square brackets should be all on one line):

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\IniFileMapping\Autorun.inf]
@="@SYS:DoesNotExist"
```

Step 3. Save the file with a name like **NoAutoRun.reg,** taking care to include the **.reg** extension.

Step 4. Right-click your **.reg** file and choose Merge. Confirm any warning prompts to add the information to the Registry.

UPDATE 2009-01-21: As an extra precaution, it's a good idea to reboot your PC after Step 4, on the off chance that some old information was residing in cache memory.

The next time you insert a flash drive, CD, DVD, or other removable disc into your system, Windows will not execute the information in any **autorun.inf** file that may be present.

Naturally, taking these steps means that the next time you put a game or installer disc into your CD or DVD drive, its software won't launch automatically. You'll have to open a Windows Explorer window or use a command line to launch the desired executable.

The benefit is a big one: a rogue program that you never intended to launch won't silently take over your system if you happen to insert a Trojan-carrying disc into a drive.

Have a tip about Windows? Readers receive a gift certificate for a book, CD, or DVD of their choice for sending tips we print. Send us your tips via the Windows Secrets contact page.

Scott Dunn is associate editor of the Windows Secrets Newsletter. He has been a contributing editor of PC World since 1992 and currently writes for the magazine's Here's How section.

Help people find this article on the Web (

explain

):

- **Digg**
- **Del.icio.us**
- **Reddit**
- **StumbleUpon**
- **Other**
- **Permalink**

All Windows Secrets articles posted on 2007-11-08: ● = *Paid content*

Bonus Download	Get Woody's new e-book bonus
Top Story	**One quick trick prevents AutoRun attacks**
Wacky Web Week	Your life vest clashes with your oxygen mask!
Langalist Plus	Part seven: decluttering a PC frees up 6GB ●
Woody's Windows	Another batch of indispensable Windows utilities ●
Perimeter Scan	Apple's new Leopard OS shows Windows envy ●
	(Show all articles on a single page)

Get the latest on Windows.

Enter your e-mail address to receive the free Windows Secrets Newsletter weekly.

Go

For instance: jan@example.com

All subscribers are covered by our Ironclad Privacy Guarantee:

1. We will never sell, rent, or give away your address to any outside party, ever;

Michael Horowitz

Defensive Computing

More posts | Read bio

February 3, 2009 - 9:31 P.M.

Disabling Autorun and Autoplay on Windows Vista SP1 with Nick Brown's method

- TAGS:Autoplay, autorun, Microsoft, security, USB Flash Drive, Windows Vista
- IT TOPICS:Cybercrime & Hacking, Security Hardware & Software, Software, Windows & Microsoft

I'm a Windows XP person. My only copy of Vista, until recently, was virtual and one that, for whatever reason, doesn't support USB devices. Thus, when I wrote recently about disabling Autorun and Autoplay using Nick Brown's registry zap, all my testing was done with XP. Now that I got my hands on a real (non-virtual) copy of Vista, I tested the zap again.

To put this in context, I've written a few postings about disabling Autorun and Autoplay, **something that all Windows users should do** because malicious software, typically on USB flash drives, abuses these features to infect new PCs.

The first posting was about constructing a test USB flash drive to both illustrate the tricks the bad guys use and to test if a Windows machine is vulnerable to those tricks or not. The third posting was about, what I consider the best way to disable Autorun and Autoplay, a registry zap documented by Nick Brown on his blog but originated by Emin Atac. It is assumed here that you're familiar with the material in these two postings.

The test machine, a ThinkPad T41, was running Vista Home Premium with Service Pack 1. This was a cleanly installed, virgin copy of Vista, there was no other software installed on the system.

Nick Brown's zap basically tells Windows not to process any autorun.inf files. As noted previously, this is separate and independent from the Autoplay feature of Windows XP in Vista. The autorun.inf file is an optional part of Autoplay.

BEFORE REGISTRY ZAP

As described previously, there are three ways that a maliciously constructed autorun.inf file can trick an unsuspecting user into running software on a USB flash drive. The test Vista system was initially vulnerable to each trick.

Shown above is the Autoplay window Vista displayed when I inserted the tester USB flash drive. The ability to add the top option to the Autoplay window is the first trick. You see this here in the option to run Paint.

The second trick comes into play when someone double-clicks on the drive letter in Computer (a.k.a My Computer). Rather than listing the files and folders, a malicious autorun.inf file can run a malicious program. The third trick involves manipulation of the context menu (the menu displayed when you right click on a drive letter) by the autorun.inf file.

DOING THE ZAP

The exact same registry zap works on XP and Vista. This is one of the advantages of Nick Brown's approach to disabling autorun. In contrast, Microsoft's approach involves different software on XP Home, XP Professional and the Home editions of Vista. As described previously, installing the registry zap involves creating or downloading a very small ".reg" file and double-clicking on it.

AFTER THE REGISTRY ZAP

After applying Nick Brown's registry zap and re-inserting the test USB flash drive, Vista's Autoplay window was very different (see above).

Gone was the option to run Paint that had come from the autorun.inf file. In its place are four new options, no doubt due to the file types Vista found on the flash drive. Also, the flash drive is now referred to as "Removable Disk" rather than "Testing AutoPlayAutoRun", a name that came from the autorun.inf file. Instead of the Paint icon, the flash drive now has a standard Vista drive letter icon.

In a nutshell, Autoplay is totally ignoring the autorun.inf file.

Vista is now also immune to the other two tricks. Double clicking on the drive letter now lists the file and folders as it's supposed to. Before the zap, it ran Paint. And, the context menu no longer has the option to run Paint.

As with Windows XP, there was no need to reboot to get the effect of the registry zap.

Backing out the registry zap works exactly the same in Vista as in XP. It involves deleting a key from the registry and rebooting (see previous posting). In this case, the reboot is required.

I didn't get to test with a business edition of Vista but it should function exactly the same as the Home Edition.

Next time, I'll report on tests of how well Nick Brown's registry zap protects from Autorun and Autoplay abuse with CDs and shared network drives.

Reply
Print
Email this
Digg this
Slashdot this

Reply

Your name: * Anonymous

E-mail: *

The content of this field is kept private and will not be shown publicly.

Homepage:

Subject:

Comment: *

Input format

● Full HTML

- Lines and paragraphs break automatically.

○ commenters

Scans show 6% of PCs already hit with worm; figure may be as high as 30%

By Gregg Keizer

Comments
14
Recommended
75
Share
More
Related

Downadup roundup

- Downadup worm now infects 1 in every 16 PCs, says Panda
- US-CERT: Microsoft's advice on Downadup leaves users open to attack
- FAQ: How to protect your PC against the Downadup worm
- 'Amazing' worm attack infects 9 million PCs
- 1 in 3 Windows PCs vulnerable to worm attack
- Researcher: Worm infects 1.1M Windows PCs in 24 hours
- 'Huge increase' in worm attacks plagues unpatched Windows PCs
- Microsoft releases emergency Windows patch to head off worm attack

January 21, 2009 (Computerworld) The computer worm responsible for the biggest attack in years has infected at least one out of every 16 PCs worldwide, a security company said today, and it may have managed to compromise as many as nearly one in three.

According to Panda Security, almost 6% of the Windows systems scanned with its antivirus technology were found to be infected with "Downadup," a worm that began aggressive attacks just over a week ago. Panda was one of the first security firms to sound an alarm over Downadup when it raised its security threat level on Jan. 12 as reports of attacks mounted.

Using data from antivirus scans performed by its consumer-grade security software and by a free online scanning tool that it makes available on its Web site, Panda found 111,379 PCs infected with the worm out of a pool of 2 million machines.

"I'm pretty confident in this number," said Ryan Sherstobitoff, chief corporate evangelist at Panda Security, as he cautioned it was just a snapshot. "Conficker is still infecting high volumes of machines and is a fast-propagating worm."

Conficker is an alternate name for the Downadup worm.

In fact, Panda's estimate is probably very conservative, Sherstobitoff said, since the bulk of the infected computers were scanned when their owners took the time to steer their browsers to the company's online scanner.

"The 6% was of people coming to our site and opting in for the scans. That's somewhat scary," said Sherstobitoff. "If we were actually to look at the [general] population, all the people who don't have antivirus -- or if they do, who haven't updated definitions -- the infection rate might be in the range of 20% to 30%."

While there has been some disagreement among security researchers about Downadup's infection volume -- last week, for instance, some disputed F-Secure Corp.'s estimate of 8.9 million infected PCs -- there has been little argument about the relative size of the worm attack. Nearly every researcher has pegged it as the biggest in years.

Today, Panda joined the chorus. "This is the biggest in at least six years," said Sherstobitoff.

Luis Corrons, the technical director of Panda's research lab, put it in the same terms. "[This is] a phenomenon we haven't seen since the times of the great epidemics of

Kournikova or Blaster," said Corrons in a statement, referring to major worm attacks of 2001 and 2003, respectively.

And things will get worse before they get better, both Corrons and Sherstobitoff predicted. "This is an epidemic, and the worst may still be to come, as the worm could begin to download more malware onto computers or to spread through other channels," Corrons said.

"We're still getting lots of reports of infections," echoed Sherstobitoff. "It could be a week or a week and a half before it slows down."

On a related note, the U.S. Computer Emergency Readiness Team (US-CERT) today said that Microsoft's advice for disabling Windows' "Autorun" feature is flawed and leaves users open to attack from the worm.

Make a Comment Recommend Story Slashdot this Digg this
Print Story Send Feedback Email this Reprints

Related Content

Webcast: Preparing for PCI 1.2 Web Seminar
Whitepaper: An Overview of PAN Manager by Egenera

CW Report: Trend Micro Gets Smart with a Hybrid Approach
FAQ: How to protect your PC against the Downadup worm
Microsoft's advice on Downadup leaves users open to attack, says US-CERT
Trojans from windshield fliers
What you don't know about the Windows Malicious Software Removal Tool
A New Internet Attack: Parking Tickets
Windows 7 vuln. in weakened UAC
Microsoft changes Windows 7 UAC after new exploit code surfaces

Today's Top Stories

Microsoft reveals 'My Phone' backup, sync service
Senate approves 'strict' rules on hiring H-1B workers
FAQ: How Google Latitude locates you
Elgan: Here comes the e-book revolution
IBM offers to shift workers losing jobs to lower-wage countries
Economy could slow enterprise adoption of Windows 7

What People Are Saying

See comments | Add new
See comments | Add new

Resource Alerts

to receive Security Resource Alerts

Webcasts

Dynamic Data Center and Virtualization Drives Operational Excellence at Emory Healthcare

How to Future-proof for Mobility: An Integrated Management and Security Strategy

Preparing for PCI 1.2 Web Seminar

Whitepapers

Report: Independent Evaluation of Requirements Management Solutions

Your Solution for Delivering and Optimizing Applications Across the Enterprise

An Overview of PAN Manager by Egenera

Computerworld Reports

Home :: News

DOD bans the use of removable, flash-type drives on all government computers

By Jennifer H. Svan and David Allen, Stars and Stripes

Mideast edition, Friday, November 21, 2008

The Defense Department has banned the use of removable flash media and storage devices from all government computers, according to a series of notices put out by the services this week.

The action comes following reports that a worm virus known as "Agent.btz" was discovered infecting some DOD networks, according to Wired magazine.

LeAnne MacAllister, 5th Signal Command’s director of Strategic Communication, U.S. Army Europe, said this week that leadership directed her office to stop using thumb drives — portable memory devices used to store or transfer files.

A separate internal Army e-mail told some government computer users across Europe to turn in all removable media devices.

In an e-mail sent Thursday to all Navy European customers in Naples, officials said "effective immediately all USB Thumb drives, memory sticks/cards and camera flash cards are PROHIBITED from use on any Navy Network (NIPR or SIPR) until further notice."

A worldwide directive issued Thursday by the Marine Corps offered similar restrictions.

"The only authorized media for use on DOD networks is media purchased and provided by the government," the Marine announcement said. "Under no circumstances will personally owned removable media be considered mission essential or used on government networks."

DOD officials at the Pentagon would not confirm the ban.

For security reasons, DOD officials won’t discuss "specific measures commanders in the field may be taking to protect and defend our networks," said Air Force Lt. Col.

Eric Butterbaugh, a DOD spokesman.

But Wired magazine, citing an internal Army e-mail on its online edition Wednesday, said the ban comes from the commander of U.S. Strategic Command and applies to both the secret SIPR and unclassified NIPR nets.

The worm virus "Agent.btz" is a variation of an older worm that copies itself to removable USB drives from infected computers and then spreads itself to whatever new systems it is connected to through USB ports, Wired reported.

The worm seriously degrades computer performance by copying itself to multiple programs.

The ban includes memory sticks, thumb drives and camera flash memory cards, according to the Marine Corps directive. External hard disk drives are not included in the ban.

Butterbaugh said DOD's Global Information Grid includes more than 17,000 local- and regional-area networks and approximately 7 million individual computers.

Recent News Articles

- Parachute training returns to Misawa
- 13th Fighter Squadron S. Korea-bound
- Biden tells Europe U.S. will push diplomacy
- KBR awarded contract despite criminal probe
- Economic crisis could affect defense budget
- Uganda rebels kill 900 during AFRICOM-backed operation
- Sigonella seeking input on upgrading Net service
- Navy training Iraqis in bomb disposal techniques

Most Popular Stripes Headlines

- AF seeks volunteer first sergeants to fill vacancies
- FOB Delta not just enduring – it’s growing
- Naval officer convicted of child pornography charges
- Air Force seeks to remedy 'dire need' for first sergeants
- Senior British officer is suspected in leaks about civilian casualties in Afghanistan

Quick Links

- Get home delivery
- Download the E-Edition
- Sign up for our newsletter
- Contact the site administrator

Stripes Messages

- Submit an online Valentine's message

Specialty sites

- Stripes Military Moms
- Stripes Guam
- Pacific Advertising Weeklies
- Welcome To Europe Guide
- Stripes Gamer

Latest Guides

- 2008 Holiday Shopping Guide

- Combined Federal Campaign Online Guide
- U.S. Constitution Day Primer
- 2008 Insurance Guide
- 2008 Retirement Guide
- 2008 Internet Shopping Guide
- 2008 Financial Planning Guide

Home | Archives | Stripes Lite | Ombudsman | Advertising | Contest rules
Newsletters | About Us | Feedback | Contact Us | Job Openings | Press Releases

COMPUTERWORLD

Security

IDG

JUMP TO More Resources

SEARCH GO

- Home
- News
- E-mail Newsletters
- Blogs
  - IT Blogwatch
  - Shark Tank
  - Topics
    - Business Intelligence
  - Careers
- Development
- E-Business & Web 2.0
- Emerging Technology
- Government & Regulation
- Hardware
- Internet
- IT Management
- Mobile & Wireless
- Networking
- Security
- Servers & Data Center
- SOA & Web Services
- Software
- Storage

- Shark Bait

- Back In The Day
- Boss Ahoy!
- Floundering Users
- Miscellaneous Bait
- News Bait
- Office Politics
- Q&A
- Sinking Projects
- Tricks Of The Trade
- Video
- Suggest a Topic
- Submit a Bait
- Register
- Login
- FAQ

  - Top Baits & Big Fish
  - Invite a Friend
  - SharkTank

- Knowledge Centers
  - Operating Systems
    - Windows
    - Linux & Unix
    - Macintosh OS
  - Networking & Internet
    - LAN/WAN
    - Hardware & Devices
    - Protocols & Standards
    - Wireless Networking
    - Network Security
    - VPN
    - VoIP
    - Internet
      - Internet Business
      - Internet Applications
      - Web 2.0
      - SaaS
      - Broadband
  - Mobile & Wireless
    - Mobile Devices
    - Laptops
    - Mobile Access
    - Mobile Applications & RFID
    - Wireless Networking
    - Wireless Trends & Technologies
    - Personal Technology
  - Security
    - Cybercrime & Hacking
    - Spam, Malware & Vulnerabilities
    - Security Hardware & Software
    - Standards & Legal Issues
    - Privacy
    - Intellectual Property & DRM
    - Disaster Recovery
- Storage
  - SAN
  - NAS
  - Hardware
  - Software
  - Disaster Recovery
  - Compliance
  - Storage Security
- Business Intelligence
  - Analytics
  - Data Mining
  - Data Warehousing
  - Databases
- Servers & Data Center
  - Servers
  - NOSes & Server Software
  - Virtualization
  - Data Center

  - Infrastructure Management
  - Grid & Utility Computing
  - Mainframes & Supercomputers
  - Disaster Recovery

- Hardware

  - Processors
  - Windows & Linux PCs
  - Macintoshes
  - Laptops
  - Servers
  - Grid & Utility Computing
  - Mainframes & Supercomputers

- Software

  - Desktop Applications
  - Enterprise Applications
    - CRM
    - ERP/Supply Chain
  - Open Source
  - Saas
  - Databases

- Development

  - Web Services
  - Web Site Management
  - Software Development

- Careers

  - Education/Training
  - Hiring/Recruiting
  - Skills
  - Search Job Listings
  - Outsourcing

- Management

  - ROI
  - Project Management
  - Outsourcing

- Government

  - Compliance
  - Legislation/Regulation
  - IT in Government

- Opinion

  - Columnists
  - SharkTank

- Webcasts
- Video
- Podcasts
- White Papers
- IT Careers
- Computerworld Reports
- Zones

- Application Delivery
- Business Continuity
- Enterprise Search Zone
- HBA Advantage
- SAS
- Security Management
- The Security Zone

- Case Study Library
- RSS Feeds
- Events

- Face to Face
  - Leadership
  - Awards
  - Storage
  - Mobile
  - Data Center
  - BI
  - SaaS
  - Green IT
  - Honors
  - Sponsorship
- Virtual
  - Storage Directions - June 2009
  - Virtualization Directions - April 2009
  - Enterprise Architecture - March 2009
  - Security Directions - Dec 2008
  - Virtualization Directions - Oct 2008

- Print Subscriptions

Malware infects space station laptop

By Gregg Keizer
Comments
12
Recommended
17
Share
Comments
Related

Active Comments

Anonymous says: Are you telling me they use Norton on the space station? Who is in charge of their security? FEMA!?...
Read the rest | Reply
t20racerman says: Thanks for that Ryan - I'll just phone them up now shall I?...
Read the rest | Reply
All Comments (12) | Post New

September 1, 2008 (Computerworld) Malware has once again managed to get from Earth onto the International Space Station, a NASA spokesman confirmed last week.

The attack code infected at least one laptop used on the station, an international operation led by the U.S. and Russian space agencies.

The NASA spokesman declined to identify the malware, saying only that antivirus software had detected it on July 25. The SpaceRef.com news site last week identified the bug as W32.Gammima.AG.

The spokesman said the worm posed no threat to the station or its crew. "It was never a threat to any command-and-control or operations computer," he said.

The spokesman refused to disclose how the malware was installed on the computer, though an entry into the station's daily logs, posted on NASA's Web site, suggests that digital camera storage cards may be responsible.

The spokesman did acknowledge that "there have been other incidents" of malware discovered on space station computers. "I don't know when the first one was, but the station will have been in orbit for 10 years [come] November," he said.

The malware discovery was first disclosed in the daily log by space station Commander Sergey Volkov on Aug. 11. Volkov reported finding the malware after running "digital photo flash cards from stowage through a virus check with the Norton AntiVirus application."

A week later, on Aug. 21, Volkov's daily report noted the discovery of malware during a scan of the hard drives and a photo disk on another laptop computer.

Graham Cluley, a consultant at Sophos PLC, noted that "if there is any good news at all, it's that the [W32.Gammima.AG] malware was designed to steal usernames and passwords from computer game players," and orbiting astronauts aren't likely to be spending a lot of time playing games.

This version of the story originally appeared in Computerworld*'s print edition.*

Got something to add? Let us know in the article comments.

Make a Comment Recommend Story Slashdot this Digg this
Print Story Send Feedback Email this Reprints

Related Content

Webcast: Preparing for PCI 1.2 Web Seminar

Whitepaper: An Overview of PAN Manager by Egenera

CW Report: Trend Micro Gets Smart with a Hybrid Approach
Malware infects space station laptops
Microsoft hurries antispyware, delays Exchange updates
Sidebar: It's 11 p.m. -- Do You Know What Your Computer Is Doing?
University traps infected PCs in its web
Trojans from windshield fliers

Today's Top Stories

Microsoft reveals 'My Phone' backup, sync service
Senate approves 'strict' rules on hiring H-1B workers
FAQ: How Google Latitude locates you
Elgan: Here comes the e-book revolution
IBM offers to shift workers losing jobs to lower-wage countries
Economy could slow enterprise adoption of Windows 7

What People Are Saying

See comments | Add new
See comments | Add new

Resource Alerts

to receive Security Resource Alerts

Webcasts

Dynamic Data Center and Virtualization Drives Operational Excellence at Emory Healthcare

How to Future-proof for Mobility: An Integrated Management and Security Strategy

Preparing for PCI 1.2 Web Seminar

Whitepapers

Report: Independent Evaluation of Requirements Management Solutions

Your Solution for Delivering and Optimizing Applications Across the Enterprise

An Overview of PAN Manager by Egenera

Computerworld Reports

Trend Micro Gets Smart with a Hybrid Approach

Computerworld Technology Briefing: Intelligent Users Use Business Intelligence

Trend Micro Gets Smart with a Hybrid Approach

Editor's Picks

Microsoft reveals 'My Phone' backup, sync service

Senate approves 'strict' rules on hiring H-1B workers

FAQ: How Google Latitude locates you

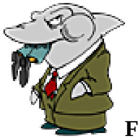
Elgan: Here comes the e-book revolution

IBM offers to shift workers losing jobs to lower-wage countries

Malware infects space station laptop

Economy could slow enterprise adoption of Windows 7

SHARKBAIT



Fired up about IT? Join Sharkbait and share your true tales of IT. SharkBait is the place for you to sound off about everything IT – the good, the bad, and the rest of the weird stuff you deal with every day.

New baits

SHARKBAIT

WHITE PAPER

Security and Trust: The Backbone of Doing Business over the Internet

Earning the trust of online customers is vital for the success of any company that requires sensitive data to be transferred over the internet. With VeriSign you can put technology in place to help your online business protect customer data and build consumer trust. Learn how with this white paper.

Download this white paper now! ▸

TODAY'S TOP BLOG

Dan Tynan: Is Google taking too much "Latitude"? Is Google Latitude yet another threat to your personal privacy? Well, yes and no. ... *[more]*

White Papers

Read up on the latest ideas and technologies from companies that sell hardware, software and services.

- The Value of Deduplication in Enterprise Data Management
- A Better Way to Manage Data in Virtualized Environments
- Backup and Recovery for Microsoft Exchange Server

View more whitepapers ▸

PARTNERED CONTENT

Juniper NETWORKS

Webcast: The Automation of IT Compliance Programs: Reducing Risk, Cost and Complexity of Corporate Compliance

To meet the growing number of industry and federal regulations, businesses spend significant time, effort, and budget determining how to best meet continuously evolving IT compliance requirements this new Forrester Research and Juniper Networks Webcast led by industry experts who examine global IT security and compliance trends, common IT compliance issues and challenges, and best practices for successful IT compliance programs.

View this webcast ▸

Whitepaper: Tackling the Top Five Network Access Control Challenges

The major challenge enterprises face today is how to create innovative business models and to increase productivity by opening the network to a dynamic workforce, while at the same time protecting critical assets from the vulnerabilities that openness and user mobility bring. In addition, to comply with industry and governmental regulations, enterprises must prove that they have stringent controls in place to restrict access to sensitive data. This paper describes the top five networking access control challenges that companies like yours are facing and solutions that they are deploying today.

Download this white paper ▸

Whitepaper: Addressing PCI Compliance with a Comprehensive Network Access Control Solution

The Payment Card Industry (PCI) is one of the most comprehensive data security standards in a cluster of regulations that have emerged over the past decade. Meeting its requirements is both complicated and expensive for many companies. Learn how a comprehensive access control solution allows retailers and consumer organizations adhere to the core tenets of PCI, and delivering the necessary information and reports needed for compliance audits.

Download this white paper ▸

Whitepaper: Control System Cyber Vulnerabilities and Mitigation of Risk for Utilities

Today's global industrial infrastructure includes thousands of electric utilities, water/wastewater management companies, oil and gas suppliers, chemical manufacturers and other facilities critical to daily functioning. Learn why relying on off-the-shelf operating systems and Internet-based remote access control to carry out production tasks, traditional control networks can leave today's global industrial infrastructures vulnerable to hackers, extortionists, worms, viruses and application-level attacks. Deploying network-based security can protect these at-risk systems–without requiring infrastructure replacement.

Download this white paper ▸

Sponsored Links

About Us Advertise Contacts Editorial Calendar Help Desk Jobs at IDG Privacy Policy Reprints Site Map

CIO Computerworld CSO DEMO GamePro Games.net IDC IDG IDG Connect IDG Knowledge Hub IDG TechNetwork IDG Ventures
IDG.net InfoWorld ITwhitepapers ITworld JavaWorld LinuxWorld Macworld Network World PC World The Industry Standard

sans.org
(Portal)
GIAC
My ISC
How To Submit Logs
port/ip lookup/search: GO

GREEN YELLOW ORANGE RED

Today's Internet Threat Level: GREEN
Handler on Duty: Mari Kirby Nichols

DiaryTrendsReportsAboutPresentationsTop 10ContactINFOConLinksXML

Handler's Diary: SPAM with a large Word file on the side;.gif Files Presenting a Not so ...

07:29 UTC

Diary

- previous
- next

Conficker's autorun and social engineering

Published: 2009-01-15,Last Updated: 2009-01-15 08:38:46 UTC by Bojan Zdrnja (Version: 2)
0 comment(s)

We wrote several diaries about Conficker (or Downadup, depending on the AV tool you are using). F-Secure posted some interesting information about the number of infections which is almost certainly in millions (and who knows how many machines will stay infected as the owners will not even notice anything).

One of the reasons for infecting so many machines is that Conficker uses multiple infection vectors:

1. It exploits the MS08-067 vulnerability,
2. It brute forces Administrator passwords on local networks and spreads through ADMIN$ shares and finally
3. It infects removable devices and network shares by creating a special autorun.inf file and dropping its own DLL on the device.

F-Secure also blogged about the autorun.inf file where they noticed that it contained a lot of garbage (about 60 kb of random binary data). This fooled some AV programs so they didn't scan the device properly (otherwise, they would have picked up the referenced DLL also stored on the device).

After removing garbage, one can see a nice autorun.inf file containing all important keywords. This grabbed my attention:

```
[Autorun]

Action=Open folder to view files
Icon=%systemroot%\system32\shell32.dll,4
Shellexecute=.\RECYCLER\S-5-3-42-
2819952290-8240758988-879315005-
3665\jwgkvsq.vmx,ahaezedrn
```

This is a typical autorun.inf file created by Conficker. The social engineering trick comes from the first two keywords (Action and Icon). When you put this in a Vista machine with default settings, an Autoplay window will pop up asking you what to do, as shown below:

Conficker's autoplay on Vista

So, as you can see, the first part, "Install or run program" is there because Vista detected an autorun.inf file containing the shellexecute keyword. However, the text comes from the Action keyword and the icon is extracted from shell32.dll (the 4th icon in the file) - and it's the standard folder icon!

This can easily fool a user in clicking this one and thinking it will open the USB stick in Windows Explorer instead of the second (the real one). The first option will run Conficker, of course.

Very smart. For administrators among you, I would suggest that you disable AutoPlay in your environments, unless it's really necessary. Depending on the environment you might even completely disable USB, if you don't need it. The following article explain nicely how the AutoPlay feature works and how to disable it (http://technet.microsoft.com/en-us/magazine/2008.01.securitywatch.aspx). Or check this article on the Autorun registry key (http://support.microsoft.com/kb/953252).

UPDATE

- fixed a typo in the vulnerability, it is MS08-067 (not MS08-068)
- Nick Brown sent a URL to his blog where he described another method for disabling Autorun by modifying the IniFileMapping registry key, see more at http://nick.brown.free.fr/blog/2007/10/memory-stick-worms.html

--
Bojan

Keywords: analysis autorun conficker downadup malware

0 comment(s)

- previous
- next

Comments

you need to log in to comment.

E-Mail:

Password:

login

Diary Archive

Share this Page

Tips

Manage Volume Shadow Copy Service from the Vssadmin Command-Line

Configure Remote Server Connections

Configure Parallel Processing in SQL Server 2008

Configure SQL Server 2008 to Automatically Manage File Size

How to Quickly Create a Copy of a Table using Transact-SQL

Related Articles

Security: Advances in BitLocker Drive Encryption

Security Watch: Malware Inspection at the Perimeter

Security: The Great Debate: Security by Obscurity

Security Watch: Passwords and Credit Cards, Part 3

Desktop Files: Mitigating Circumstances

Also by this Author

Security Watch: The Long-Term Impact of User Account Control

Security Watch: Principles of Quantum Security

Security: New ACLs Improve Security in Windows Vista

Security Watch: Passwords and Credit Cards, Part 2

Special Report: Security: Deconstructing Common Security Myths

Popular Articles

Hey, Scripting Guy!:

Security Watch Island Hopping: The Infectious Allure of Vendor Swag

Jesper M. Johansson

The technique of island hopping—penetrating a network through a weak link and then hopping around systems within that network—has been around for years. But it continues to take on new dimensions. In today's security-conscious IT environments, people are often the weakest link, and malicious users are finding ways to use this to their advantage (think phishing and other forms of social engineering). This combination of carbon and silicon can prove fatal to your network.

One of my favorite implementations of leveraging the human element was perpetrated by Steve Stasiukonis of Secure Network Technologies during a penetration test for a customer. He seeded th customer's parking lot with USB flash drives, each of which had a Trojan horse installed on it. When the employees arrived for work in the morning, they were quite excited to find the free gadgets laying around the parking lot. Employees eagerly collected the USB drives and plugged them into the first computers they came across: their own workstations.

Maybe some employees were wise enough to ignore these USB drives, and perhaps some of the USB drives were discarded, but it really only took one user with one drive to infect his own system and provide a gateway into the network. Stasiukonis did this exercise as a test, of course, but this technique has been used by real criminals to infiltrate large corporate networks.

USB flash drives are everywhere these days. At almost every conference, some vendor is giving them away like candy. Those drives may not have a lot of capacity, but you don't need a lot of storage space to take over an entire network. In this two-part series, I will investigate this type of attack and show ways you can help mitigate the risk. In this month's installment, I will discuss wha exactly USB flash drives can do and what you can do to control them. In next month's issue of *TechNet Magazine*, I will show you how to contain an attack to a single or small set of systems on the network.

The technical details of the attack are actually quite simple. It all starts with an infected USB flash drive being inserted into a single computer. What happens then depends on the payload on that drive and, of course, how gullible the user is.

In the Beginning

On the podium was an unattended laptop. The speaker was busily trying to ingratiate himself with the audience before beginning his presentation. The laptop itself was locked, but that hardly mattered. The attacker went up to the podium and lingered a bit, looking like he was just waiting for the speaker to return. Since those podiums are designed to hide ugly things like computers, the USB ports and the attacker's hands were shielded from view. Once the USB flash drive was insertec into the computer, it took only a few seconds to complete the attack.

There are some obvious variations on this particular theme. For example, when I used to travel the world doing presentations on a weekly basis, almost every time I finished, there were a few people asking for a copy of my slides. My answer was always to hand them a business card and ask them to send me an e-mail message. Why would I do that when every one of them was eagerly holding ; USB flash drive toward me? Because I know about tools like USB Hacksaw and Switchblade. (If you aren't familiar with these, look them up at wiki.hak5.org.)

Basically, these tools make it really easy for just about anyone to exploit people who leave their USB ports unprotected. For example, Switchblade can dump the following:

- System information
- All network services
- A list of ports that are listening

Learning to Excel

Windows Administration: Manage Printers with New Group Policy Features in Windows Server 2003 R2

The Desktop Files: PsTools Primer

Security Watch: Subjects and Security Principals

Interop: Authenticate Linux Clients with Active Directory

OUR BLOG

Hey, Scripting Guy! Browsing Active Directory

Inside SharePoint: Maintaining Security Account Credentials

A Month of Windows 7 beta 1 Tips

A Month of SQL Server Tips - Week 4

Deploying Microsoft Dynamics CRM 4.0

- All product keys for Microsoft products on the computer
- The local password database
- The password of any wireless networks the computer uses
- All network passwords the currently logged on user has stored on the computer
- Internet Explorer®, Messenger, Firefox, and e-mail passwords
- The Local Security Authority (LSA) secrets, which contain all service account passwords in clear text
- A list of installed patches
- A recent browsing history

All of this goes into a log file on the flash drive, and takes about 45 seconds.

Hacksaw is a slightly modified version; it installs a Trojan on the computer, which monitors all USB flash drive insertion events. It then e-mails all documents from all flash drives subsequently inserted into the computer to the attacker.

The tools I have discussed so far use U3 (u3.com), a technology designed to enable users to bring programs with them on a flash drive. In a nutshell, a U3-enabled flash drive lies about itself. It tells the OS that it is actually a USB hub with a flash drive and a CD plugged into it. Windows® versions prior to Windows Vista® will, by default, automatically run programs designated in the autorun.inf file on CDs, but not on USB drives. By lying about itself, the U3-enabled USB flash drive fools the OS into autorunning something called the U3 launcher. The U3 launcher, in turn, can start programs, give you a menu, or do pretty much anything that you could do with the computer yourself.

All the exploit tools do is replace the launcher with the exploit code. As soon as the flash drive is plugged into a Windows XP system, the exploit tool automatically runs. On Windows Vista, the AutoPlay decision flow works differently. AutoRun is actually enabled on removable devices, just lik on CDs. That means you will have some sort of action happen when the device is inserted. By default, the AutoPlay dialog will appear, as shown in **Figure 1**.

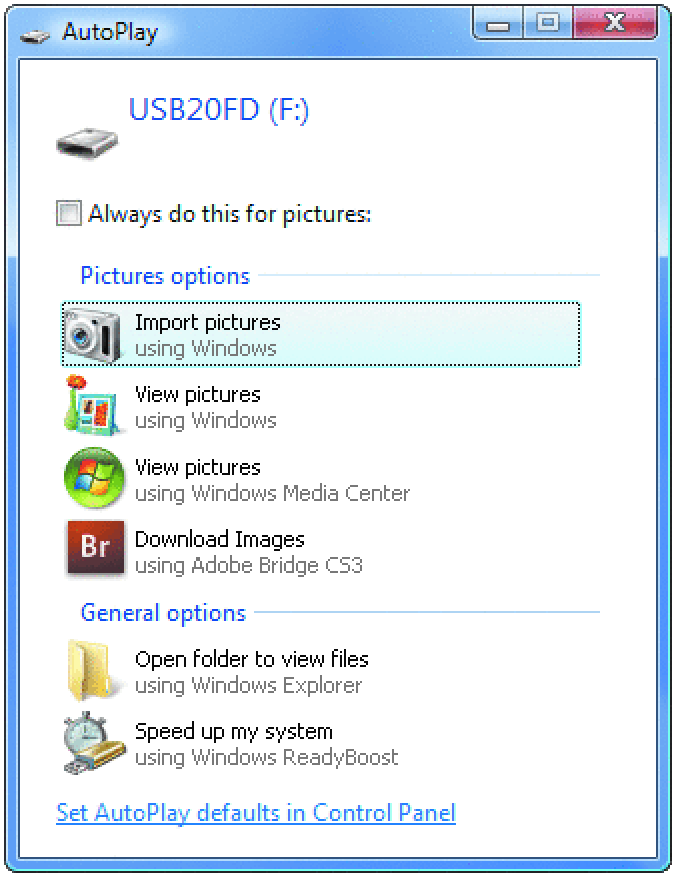
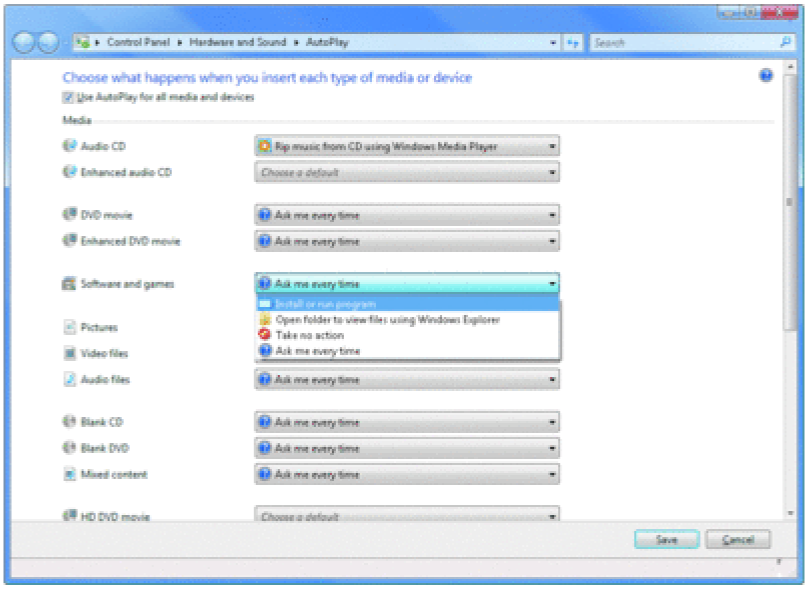


Figure 1 **Default AutoPlay dialog appears when a user inserts a removable drive with pictures**

Notice that the dialog in **Figure 1** has an option to "Always do this for pictures." This sets an AutoPlay option, which can be configured in the Control Panel, shown in **Figure 2**.



Choose what happens when you insert each type of media or device

☑ Use AutoPlay for all media and devices

Media

Audio CD	Rip music from CD using Windows Media Player
Enhanced audio CD	Choose a default
DVD movie	Ask me every time
Enhanced DVD movie	Ask me every time
Software and games	Ask me every time Install or run program Open folder to view files using Windows Explorer Take no action Ask me every time
Pictures	
Video files	
Audio files	Ask me every time
Blank CD	Ask me every time
Blank DVD	Ask me every time
Mixed content	Ask me every time
HD DVD movie	Choose a default

Figure 2 **AutoPlay configuration on Windows Vista** (Click the image for a larger view)

The AutoPlay configuration is particularly interesting for "Software and games." By definition, it means the removable drive has an autorun.inf file that specifies some program to be executed. The settings in **Figure 2** related to "Software and games" are stored in the following registry key:

Copy Code

```
Hive:    HKEY_CURRENT_USER
Key:    \Software\Microsoft\Windows\CurrentVersion\Explorer\AutoplayHandlers\
UserChosenExecuteHandlers\AutorunINFLegacyArrival
Value: (Default)
Data:
MSAutoRun – automatically executes the program specified in the autorun.inf file

MSPromptEachTime – prompts the user, using strings specified in the autorun.inf file, if available
```

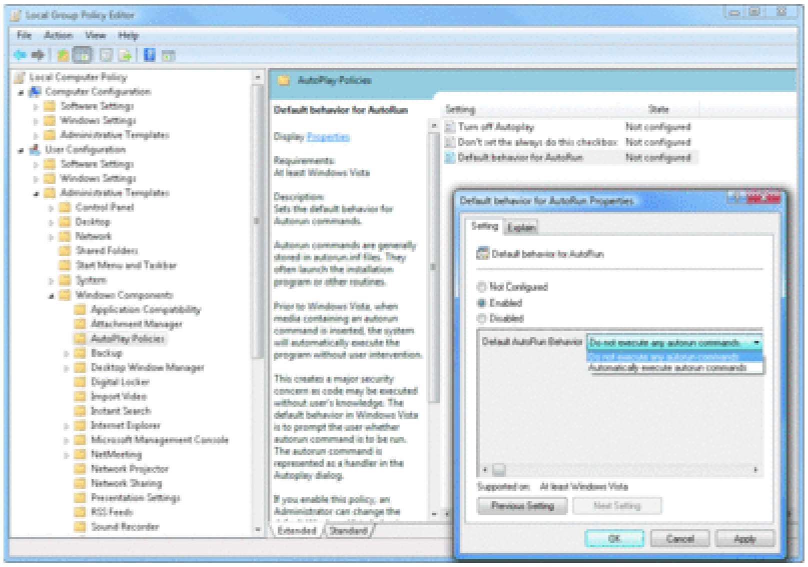
As you do not need to be an administrator to modify these settings, users can set them to whateve they want, including automatically running whatever malware is on the drive they found in the parking lot. In fact, they can do it by checking the appropriate box in the AutoPlay dialog.

Managing AutoPlay in Your Network

As an administrator, there are a couple of ways you can manage AutoPlay throughout your network First, you can prevent a user from enabling AutoPlay on removable media and CDs by modifying th NoDriveTypeAutoRun setting that controls which drive types AutoPlay is enabled for. Windows XP shipped with NoDriveTypeAutoRun set to 0x95, which disables AutoPlay on unknown drive types, network drives, and removable devices. Starting with Windows XP SP2, the NoDriveTypeAutoRun setting is configured to 0x91 by default. This enables AutoPlay on removable storage devices. Using Group Policy, there is a setting under User Configuration\Administrative Templates\Windows Components\AutoPlay Polices that allows you to manage the NoDriveTypeAutoRun setting. Enabling the "Turn off AutoPlay setting and select CD-ROM and Removable Drives" will disable AutoPlay on both types of drives.

As a result, though, the user does not get the AutoPlay dialog from **Figure 1**. In fact, nothing at all happens when the user inserts the drive. That may not be an ideal solution, however, as it can cause confusion for the user who isn't sure how to access the necessary information on the drive.

What you probably want to do is control which types of content can be played automatically. You can do this to some extent on Windows Vista, as this OS effectively has two levels of controls over automatically running software. Another control in Group Policy, shown in **Figure 3**, allows you to disable AutoPlay of software using autorun.inf files, while leaving the remainder of the Windows Vista AutoPlay behavior intact.



Default behavior for AutoRun

Display Properties

Requirements:
At least Windows Vista

Description:
Sets the default behavior for Autorun commands.

Autorun commands are generally stored in autorun.inf files. They often launch the installation program or other routines.

Prior to Windows Vista, when media containing an autorun command is inserted, the system will automatically execute the program without user intervention.

This creates a major security concern as code may be executed without user's knowledge. The default behavior in Windows Vista is to prompt the user whether autorun command is to be run. The autorun command is represented as a handler in the Autoplay dialog.

If you enable this policy, an Administrator can change the

Setting	State
Turn off Autoplay	Not configured
Don't set the always do this checkbox	Not configured
Default behavior for AutoRun	Not configured

Default behavior for AutoRun Properties

Default behavior for AutoRun

○ Not Configured
● Enabled
○ Disabled

Default AutoRun Behavior: Do not execute any autorun commands
Do not execute any autorun commands
Automatically execute autorun commands

Supported on: At least Windows Vista

Figure 3 **Disable autorun.inf files with Group Policy** (Click the image for a larger view)

The decision flow is a bit complicated when it comes to AutoPlay, and perhaps it is easiest to show with a flowchart. **Figure 4** depicts the decision flow used when determining whether to automatically execute a file on removable media.

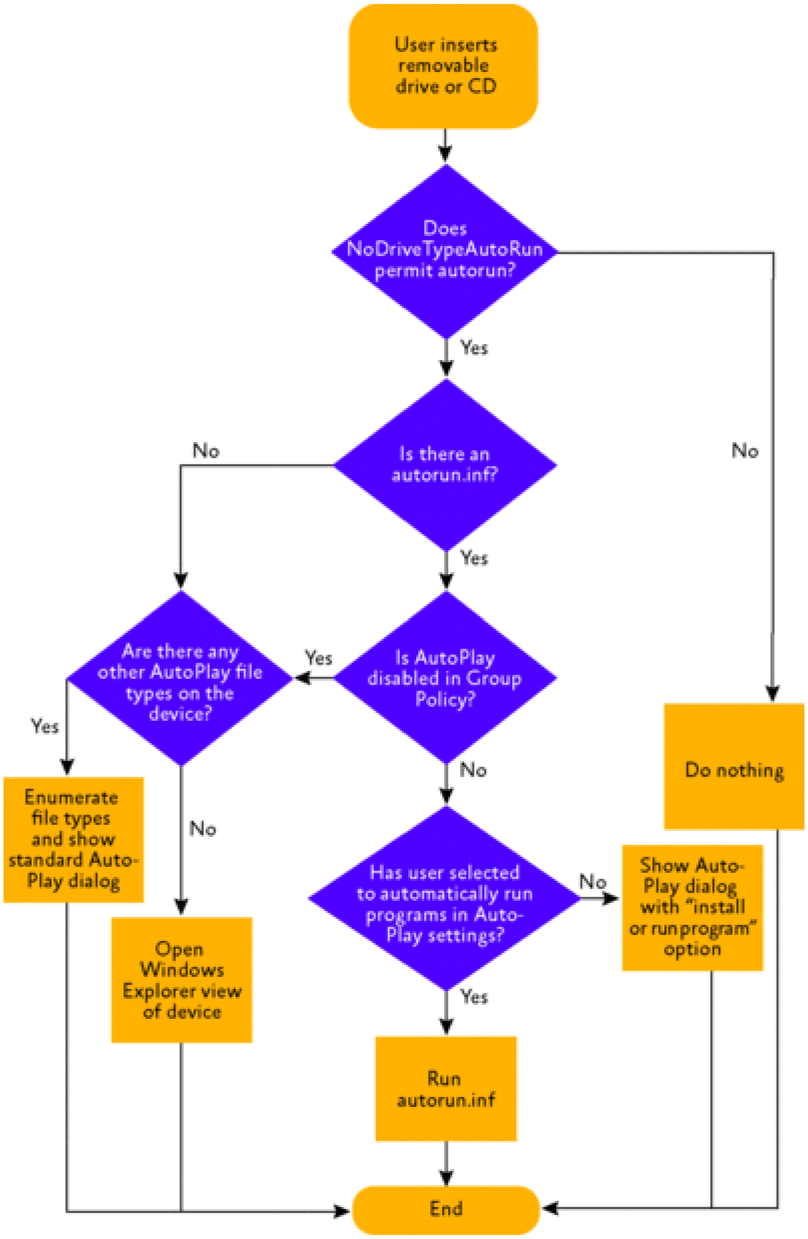

User inserts removable drive or CD
Does NoDriveTypeAutoRun permit autorun?
Yes
No
Is there an autorun.inf?
No
Yes
Are there any other AutoPlay file types on the device?
Yes
Is AutoPlay disabled in Group Policy?
Yes
No
Do nothing
Enumerate file types and show standard Auto-Play dialog
No
Open Windows Explorer view of device
Has user selected to automatically run programs in Auto-Play settings?
No
Show Auto-Play dialog with "install or run program" option
Yes
Run autorun.inf
End

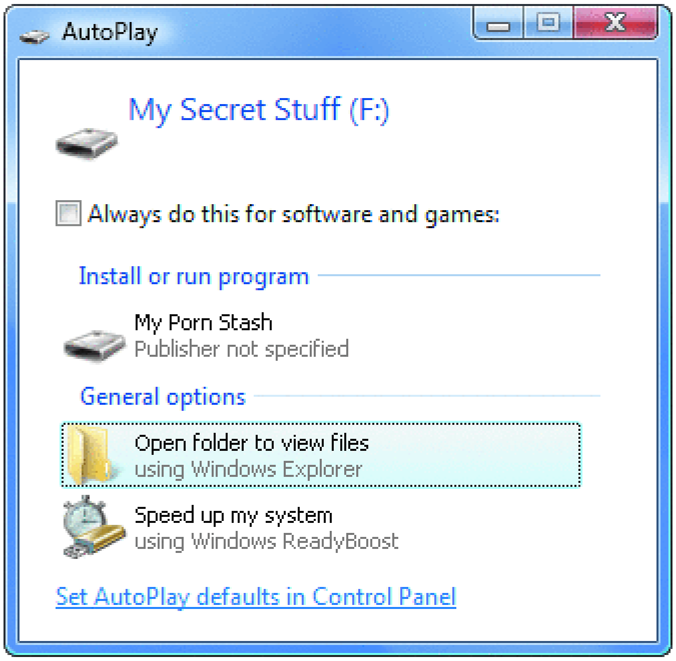
Figure 4 **The AutoPlay decision flow** (Click the image for a larger view)

Other Beginnings

Of course, these are just some of the tools that can be used to compromise a computer with a USB flash drive. All sorts of tools can be added to this scenario, such as a tool that dumps out the currently logged-on user's Windows NT® password hash (see microsoft.com/technet/community/columns/secmgmt/sm1005.mspx).

Many USB controllers are actually Direct Memory Access (DMA) devices. This means they can bypass the operating system and directly read and write memory on the computer. Bypass the OS and you bypass the security controls it provides—now you have complete and unfettered access to the hardware. This renders device control implemented by the OS completely ineffective. I am unaware of any hacking tools that currently use this technique, but I very much doubt that this has not already been done.

Another way to exploit the user/removable device combination is to simply present an enticing option. For example, how many users would click the dialog in **Figure 5**? Not all, but probably many. The point is, it's a fairly trivial task to concoct a dialog that will be too enticing for users to ignore.



AutoPlay

My Secret Stuff (F:)

□ Always do this for software and games:

Install or run program

My Porn Stash
Publisher not specified

General options

Open folder to view files
using Windows Explorer

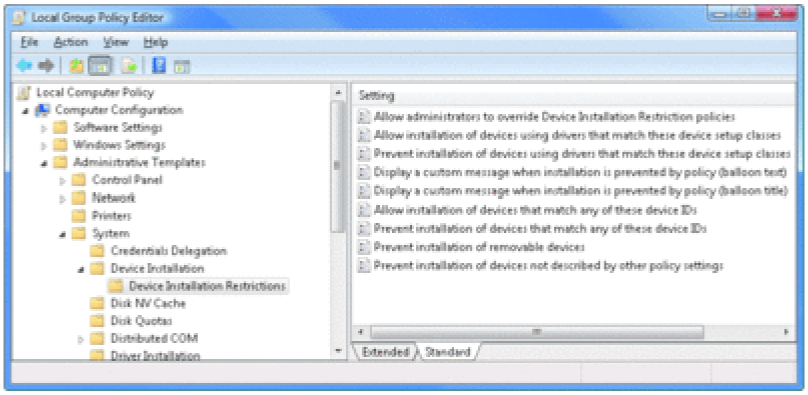
Speed up my system
using Windows ReadyBoost

Set AutoPlay defaults in Control Panel

Figure 5 **How many users would fall prey to this?**

Device Blocking

New in Windows Vista, Group Policy now has a set of policies to govern device installation (see **Figure 6**). As shown, the administrator can block all new removable devices from installing if the driver specifies that they are removable. If the driver states that they are not, however, the policy does not take effect. Thus, this policy can be circumvented using custom drivers.



Local Computer Policy
- Computer Configuration
  - Software Settings
  - Windows Settings
  - Administrative Templates
    - Control Panel
    - Network
    - Printers
    - System
      - Credentials Delegation
      - Device Installation
        - Device Installation Restrictions
      - Disk NV Cache
      - Disk Quotas
      - Distributed COM
      - Driver Installation

Setting
- Allow administrators to override Device Installation Restriction policies
- Allow installation of devices using drivers that match these device setup classes
- Prevent installation of devices using drivers that match these device setup classes
- Display a custom message when installation is prevented by policy (balloon text)
- Display a custom message when installation is prevented by policy (balloon title)
- Allow installation of devices that match any of these device IDs
- Prevent installation of devices that match any of these device IDs
- Prevent installation of removable devices
- Prevent installation of devices not described by other policy settings

Figure 6 **Windows Vista device installation policies** (Click the image for a larger view)

For more granular control, the administrator can use the policies to act on specific device setup classes. However, those require you to know the GUID for the specific device class you wish to bloc (or allow), making this approach considerably more difficult to manage.

Least Privilege Really Matters

Ignoring the DMA scenario for a moment, the success of the attacks I have discussed, as well as the success of the countermeasures, will depend on the privileges of the user using the computer. If the user is a standard user, the amount of damage the exploit can do is limited. It can still steal that user's data and anything that user has access to. However, the attack will likely not impact the network at large.

However, if the user being exploited is an administrator, the consequences can be a whole lot worse. In the worst-case scenario, the user is a Domain Administrator, and taking over the entire network is trivial. For instance, the attacker could extract the user's password hash from memory. The system caches the hash within the LSA process space at logon so that it can be used to transparently access network resources on behalf of the user. An attacker with administrative privileges can extract this hash and access any network resources with it. In other words, if the use is a Domain Administrator, the entire domain has been compromised.

This leads me to the topic of dependencies. So far, I have outlined a specific attack on workstations I have not, however, explored what the attack can lead to on a network scale. In next month's column, I will explore the concept of dependencies, showing you what they mean for the environment as a whole and, more importantly, what you can do to harden these dependencies.

Jesper M. Johansson is a Security Engineer working on software security issues and is a contributing editor to *TechNet Magazine*. He holds a PhD in MIS and has more than 20 years of experience in security.

y Watch: Island Hopping: The...

rity Watch Island Hopping: The Infectious Allure of Vend

hansson

nique of island hopping—penetrating a network through a weak link and ther —has been around for years. But it continues to take on new dimensions. In t re often the weakest link, and malicious users are

o use this to their advantage (think phishing and other forms of social engineering). This combination of carbon and silicon can p vorite implementations of leveraging the human element was perpetrated by Steve Stasiukonis of Secure Network Technologies c

lash drives, each of which had a Trojan horse installed on it. When the employees arrived for work in the morning, they were qui
ted the USB drives and plugged them into the first computers they came across: their own workstations.

employees were wise enough to ignore these USB drives, and perhaps some of the USB drives were discarded, but it really only t
ork. Stasiukonis did this exercise as a test, of course, but this technique has been used by real criminals to infiltrate large corpor

ves are everywhere these days. At almost every conference, some vendor is giving them away like candy. Those drives may not
work. In this two-part series, I will investigate this type of attack and show ways you can help mitigate the risk. In this month's i
trol them. In next month's issue of *TechNet Magazine*, I will show you how to contain an attack to a single or small set of system

details of the attack are actually quite simple. It all starts with an infected USB flash drive being inserted into a single computer.
he user is.

hing

n was an unattended laptop. The speaker was busily trying to ingratiate himself with the audience before beginning his presenta
e podium and lingered a bit, looking like he was just waiting for the speaker to return. Since those podiums are designed to hide
view. Once the USB flash drive was inserted into the computer, it took only a few seconds to complete the attack.

he obvious variations on this particular theme. For example, when I used to travel the world doing presentations on a weekly bas
My answer was always to hand them a business card and ask them to send me an e-mail message. Why would I do that when ev
tools like USB Hacksaw and Switchblade. (If you aren't familiar with these, look them up at wiki.hak5.org.)

se tools make it really easy for just about anyone to exploit people who leave their USB ports unprotected. For example, Switchb

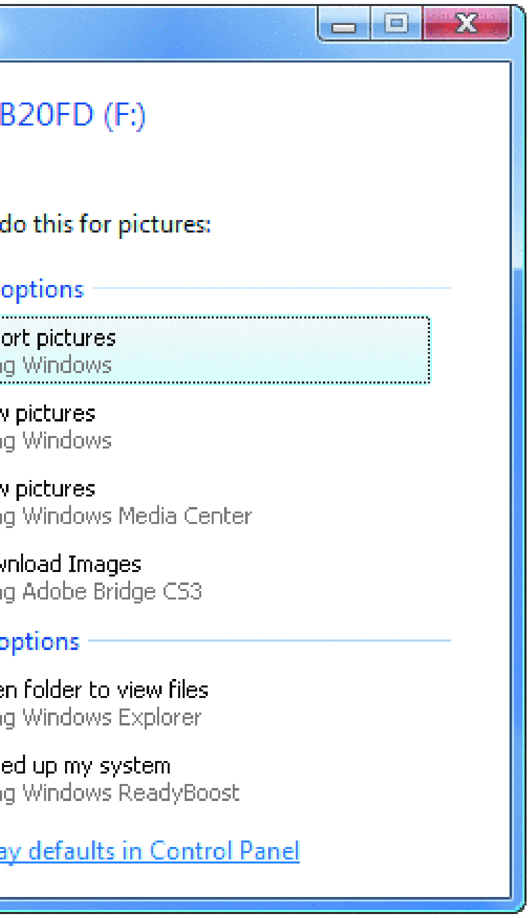
- System information
- All network services
- A list of ports that are listening
- All product keys for Microsoft products on the computer
- The local password database
- The password of any wireless networks the computer uses
- All network passwords the currently logged on user has stored on the computer
- Internet Explorer®, Messenger, Firefox, and e-mail passwords
- The Local Security Authority (LSA) secrets, which contain all service account passwords in clear text
- A list of installed patches
- A recent browsing history

s into a log file on the flash drive, and takes about 45 seconds.

slightly modified version; it installs a Trojan on the computer, which monitors all USB flash drive insertion events. It then e-mails
he attacker.

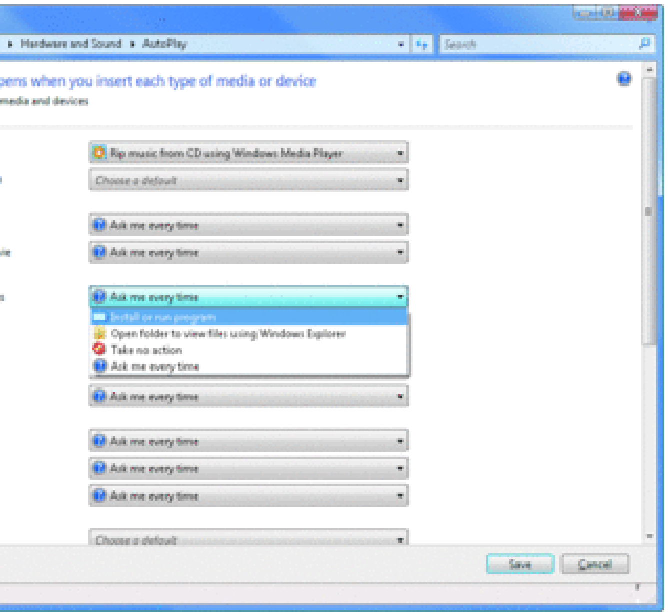
ve discussed so far use U3 (u3.com), a technology designed to enable users to bring programs with them on a flash drive. In a
3 hub with a flash drive and a CD plugged into it. Windows® versions prior to Windows Vista® will, by default, automatically run p
t itself, the U3-enabled USB flash drive fools the OS into autorunning something called the U3 launcher. The U3 launcher, in turn
the computer yourself.

tools do is replace the launcher with the exploit code. As soon as the flash drive is plugged into a Windows XP system, the explo
ntly. AutoRun is actually enabled on removable devices, just like on CDs. That means you will have some sort of action happen w
ure 1.



ault AutoPlay dialog appears when a user inserts a removable drive with pictures

e dialog in **Figure 1** has an option to "Always do this for pictures." This sets an AutoPlay option, which can be configured in the C


Hardware and Sound ▸ AutoPlay
Search
pens when you insert each type of media or device
media and devices
Rip music from CD using Windows Media Player
Choose a default
Ask me every time
Install or run program
Open folder to view files using Windows Explorer
Take no action
Save
Cancel

oPlay configuration on Windows Vista (Click the image for a larger view)

configuration is particularly interesting for "Software and games." By definition, it means the removable drive has an autorun.inf
ftware and games" are stored in the following registry key:

```
Y_CURRENT_USER
ware\Microsoft\Windows\CurrentVersion\Explorer\AutoplayHandlers\
xecuteHandlers\AutorunINFLegacyArrival
ult)
```

- automatically executes the program specified in the autorun.inf file

chTime – prompts the user, using strings specified in the autorun.inf file, if available

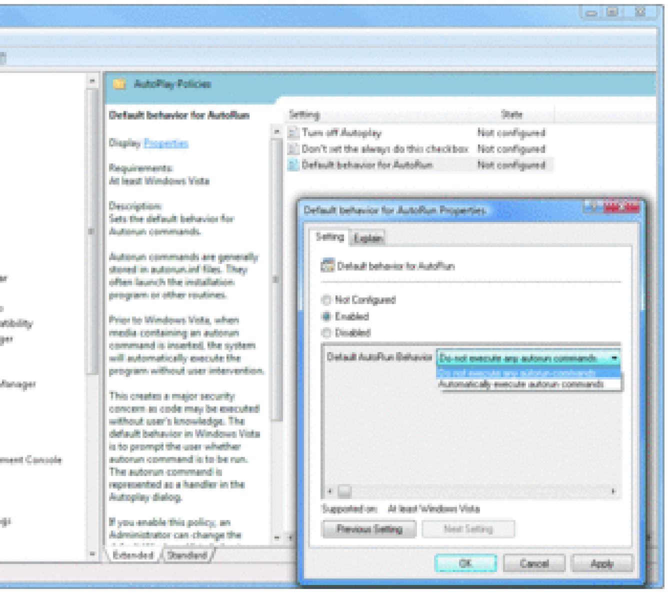
need to be an administrator to modify these settings, users can set them to whatever they want, including automatically runnin by checking the appropriate box in the AutoPlay dialog.

utoPlay in Your Network

strator, there are a couple of ways you can manage AutoPlay throughout your network. First, you can prevent a user from enabli AutoRun setting that controls which drive types AutoPlay is enabled for. Windows XP shipped with NoDriveTypeAutoRun set to 0x vices. Starting with Windows XP SP2, the NoDriveTypeAutoRun setting is configured to 0x91 by default. This enables AutoPlay or ation\Administrative Templates\Windows Components\AutoPlay Polices that allows you to manage the NoDriveTypeAutoRun sett rives" will disable AutoPlay on both types of drives.

ough, the user does not get the AutoPlay dialog from **Figure 1**. In fact, nothing at all happens when the user inserts the drive. T sure how to access the necessary information on the drive.

bably want to do is control which types of content can be played automatically. You can do this to some extent on Windows Vista ther control in Group Policy, shown in **Figure 3**, allows you to disable AutoPlay of software using autorun.inf files, while leaving t



Default behavior for AutoRun

Display Properties

Requirements:
At least Windows Vista

Description:
Sets the default behavior for Autorun commands.

Autorun commands are generally stored in autorun.inf files. They often launch the installation program or other routines.

Prior to Windows Vista, when media containing an autorun command is inserted, the system will automatically execute the program without user intervention.

This creates a major security concern as code may be executed without user's knowledge. The default behavior in Windows Vista is to prompt the user whether autorun command is to be run. The autorun command is represented as a handler in the Autoplay dialog.

If you enable this policy, an Administrator can change the

Setting	State
Turn off Autoplay	Not configured
Don't set the always do this checkbox	Not configured
Default behavior for AutoRun	Not configured

Default behavior for AutoRun Properties

Setting | Explain

Default behavior for AutoRun

○ Not Configured
● Enabled
○ Disabled

Default AutoRun Behavior: Do not execute any autorun commands
- Do not execute any autorun commands
- Automatically execute autorun commands

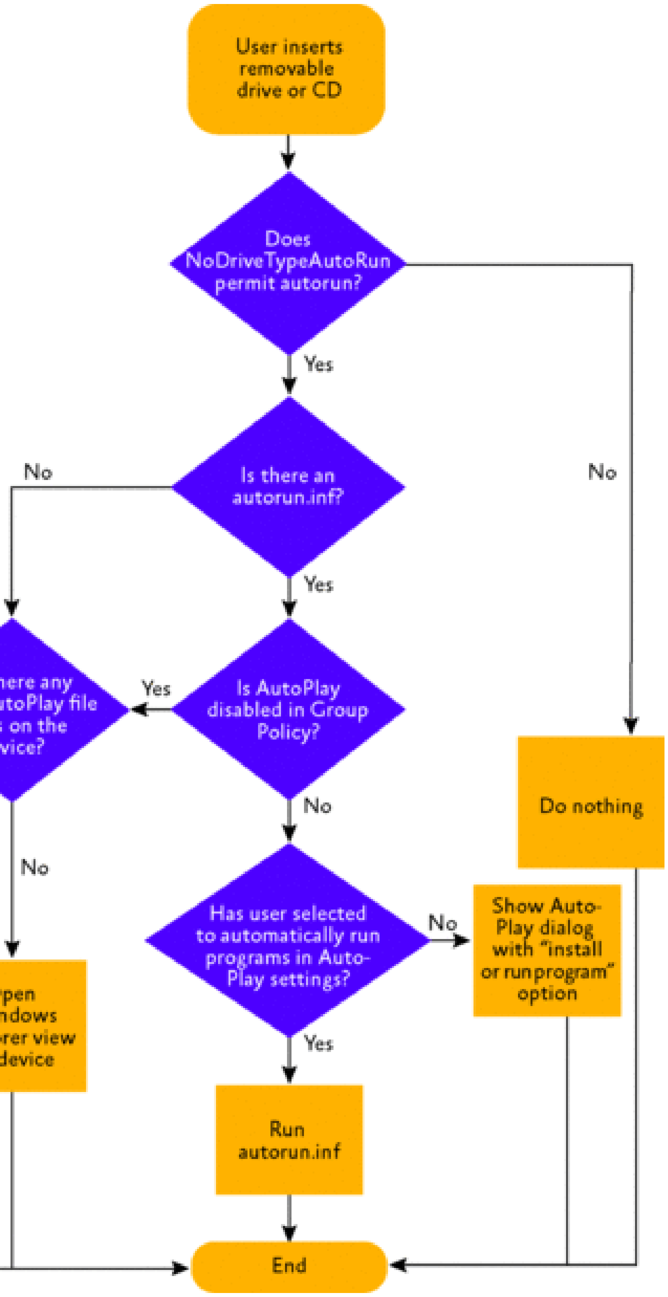
Supported on: At least Windows Vista

Previous Setting | Next Setting

OK | Cancel | Apply

able autorun.inf files with Group Policy (Click the image for a larger view)

low is a bit complicated when it comes to AutoPlay, and perhaps it is easiest to show with a flowchart. **Figure 4** depicts the deci edia.


User inserts removable drive or CD
Does NoDriveTypeAutoRun permit autorun?
Yes
No
Is there an autorun.inf?
No
Yes
here any utoPlay file s on the vice?
Yes
Is AutoPlay disabled in Group Policy?
No
Do nothing
No
Has user selected to automatically run programs in Auto-Play settings?
No
Show Auto-Play dialog with "install or run program" option
pen ndows rer view device
Yes
Run autorun.inf
End

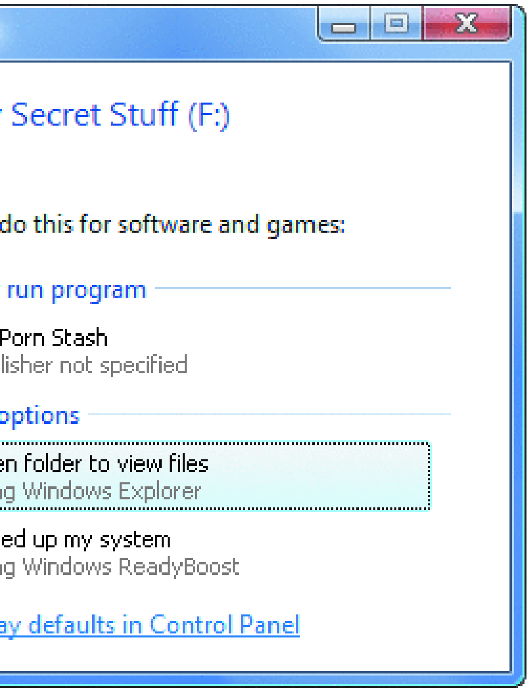
AutoPlay decision flow (Click the image for a larger view)

hings

ese are just some of the tools that can be used to compromise a computer with a USB flash drive. All sorts of tools can be added
vs NT® password hash (see microsoft.com/technet/community/columns/secmgmt/sm1005.mspx).

ntrollers are actually Direct Memory Access (DMA) devices. This means they can bypass the operating system and directly read a
ols it provides—now you have complete and unfettered access to the hardware. This renders device control implemented by the
ique, but I very much doubt that this has not already been done.

o exploit the user/removable device combination is to simply present an enticing option. For example, how many users would cli
sk to concoct a dialog that will be too enticing for users to ignore.

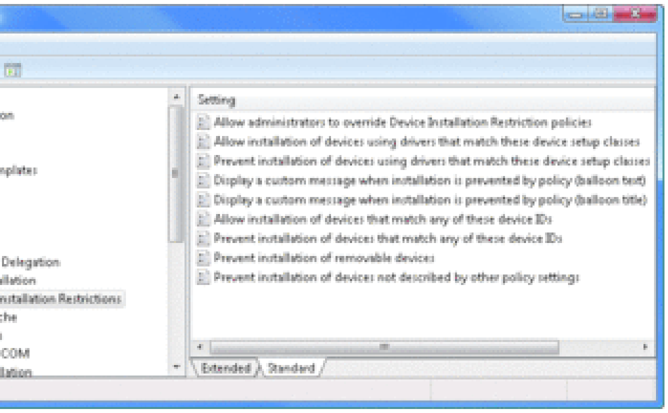


many users would fall prey to this?

king

ws Vista, Group Policy now has a set of policies to govern device installation (see **Figure 6**). As shown, the administrator can blo

e. If the driver states that they are not, however, the policy does not take effect. Thus, this policy can be circumvented using cus



Setting
- Allow administrators to override Device Installation Restriction policies
- Allow installation of devices using drivers that match these device setup classes
- Prevent installation of devices using drivers that match these device setup classes
- Display a custom message when installation is prevented by policy (balloon text)
- Display a custom message when installation is prevented by policy (balloon title)
- Allow installation of devices that match any of these device IDs
- Prevent installation of devices that match any of these device IDs
- Prevent installation of removable devices
- Prevent installation of devices not described by other policy settings

dows Vista device installation policies (Click the image for a larger view)

nular control, the administrator can use the policies to act on specific device setup classes. However, those require you to know t

considerably more difficult to manage.

ge Really Matters

DMA scenario for a moment, the success of the attacks I have discussed, as well as the success of the countermeasures, will dep

, the amount of damage the exploit can do is limited. It can still steal that user's data and anything that user has access to. How

ne user being exploited is an administrator, the consequences can be a whole lot worse. In the worst-case scenario, the user is a

attacker could extract the user's password hash from memory. The system caches the hash within the LSA process space at logo

user. An attacker with administrative privileges can extract this hash and access any network resources with it. In other words, i

.

to the topic of dependencies. So far, I have outlined a specific attack on workstations. I have not, however, explored what the a

f dependencies, showing you what they mean for the environment as a whole and, more importantly, what you can do to harden

hansson is a Security Engineer working on software security issues and is a contributing editor to *TechNet Magazine*. He holds a