

Getting the Job Done: Comparing Approaches for Desktop Software Lockdown

By: Brien M. Posey, Microsoft MVP

Published: Aug 2008

Abstract:

Preventing the installation and execution of unauthorized software should be a high priority for any IT-conscious organization. Allowing users to install or execute unauthorized software can expose an organization to a variety of stability, security, and legal risks, not to mention the burden of support costs. This paper will compare and contrast a variety of techniques for detecting and preventing unauthorized code.

About the Author:

Brien M. Posey, MCSE, is a five time Microsoft Most Valuable Professional for his work with Windows Server, IIS, Microsoft Exchange Server, and File System Storage.

Brien is currently a freelance technical author with over 4,000 technical articles to his credit. He has also written or contributed material to nearly three dozen books. As a freelance technical writer, Brien has written for Microsoft, TechTarget, Windows IT Professional, CNET, ZDNet, MSD2D, Relevant Technologies, and other technology companies.

Prior to becoming a freelance author, Brien has served as CIO for a nationwide chain of hospitals and was once responsible for the Department of Information Management at Fort Knox. You can contact Brien by e-mail at Brien@brienposey.com

Contents

- Introduction.....3
- What Does it Mean to Be Locked Down?.....3
- The Importance of Locking Down Workstations.....3
- The Challenges of Locking Down Workstations.....5
- Requirements for Desktop Lockdown.....5
- Comparison of Solutions Against Requirements.....6
- User Rights Restrictions.....6
- Group Policy Objects and Software Restriction Policies.....7
- Certificate Rules.....8
- Hash Rules.....9
- Path Rules.....9
- Internet Zone Rules.....11
- Final Thoughts on Software Restriction Policies.....12
- The Windows Terminal Services.....12
- Windows Server 2008 and Windows Vista.....13
- Software Restriction Policies.....13
- The File System Resource Manager.....15
- Preventing Device Installation.....17
- Trusted Software Approval19
- Conclusion / Recommendations.....20

Introduction:

Over the last several years, IT efficiency has become a huge concern for almost every corporation. In spite of the vast resources that many companies dedicate to IT, it sometimes seems that workstation-level management takes a backseat to server management. Although I'm not questioning the need for top-notch server controls, neglecting the controls over your workstations can have disastrous consequences. In this paper, I will discuss the importance of locking down workstations. I will then compare and contrast various techniques and mechanisms used for locking down workstations.

What Does it mean to be Locked Down?

Locking down a desktop can mean a lot of different things. Entire books have been written on the subject of hardening workstations. For the purpose of this white paper, I'm going to refer to a desktop computer as being locked down if it is configured in such a manner that prevents unauthorized applications from being installed or executed.

The Importance of Locking Down Workstations

It is important to lock down workstations for a variety of both business and technical reasons. The strange thing is that a business's needs and the needs of the IT department are often one and the same, but for different reasons. A perfect example of this is the fact that an unauthorized piece of software has the potential to affect system stability.

When the IT department comes up with an approved configuration for desktop machines, the configuration includes a set of applications that are compatible with each other, but that also meet the company's business needs. The approved configuration will also likely include a set of drivers and system services that are proven to be stable.

If a user were to install some random piece of software onto a system that is running a carefully planned and tested configuration, that piece of software may or may not have compatibility issues with the existing configuration. For example, the new application could potentially overwrite critical driver files or registry entries. These types of activities could result in system stability problems, or in a full-fledged system crash.

If the new application did cause problems, the user will no doubt place a call to the helpdesk. This is where business goals and IT goals meet. One of the company's goals should be to minimize calls to the helpdesk both for IT reasons and business reasons. From a business perspective, helpdesk calls cost money. If the user is having a problem and has to call the helpdesk, it means that the user's computer is not performing adequately enough for the user to do his or her job. This means lost productivity.

Likewise, the IT staff usually also wants to minimize helpdesk calls. Typically, IT personnel are overburdened as it is, and helpdesk calls often mean that someone from the IT staff has to stop working on something else and tend to the user's self-inflicted problem.

This leads us to another reason why it is important to lock down workstations. When the person from the helpdesk goes to assist the user who is having problems, they have certain assumptions about the state of the system. They are assuming that the system is running the company's approved configuration, which has been thoroughly tested and proven to be stable. The end user is unlikely to admit to installing an unauthorized application. Since the person from the helpdesk does not know that this application exists, they will likely waste time troubleshooting other areas of the system.

Other reasons for locking down workstations involve mitigating security risks. In my last example, I assumed that the end user installed a benign application. However, many seemingly innocent applications, especially free ones downloaded from the Internet, tend to carry undesirable payloads.

If a workstation is not locked down, then there is absolutely nothing stopping a user from installing an application that contains spyware, a Trojan such as a keystroke logger, a virus, a root kit, or some other form of malware. If a user were to install an infected application, the consequences could be disastrous. For example, the user could accidentally introduce a virus or spyware infection that spreads to every machine in the entire company. If the application happens to contain a Trojan, then sensitive information, such as the user's password or the contents of every e-mail message that the user sends, could be made public.

Here is another reason why it is imperative that you lock down desktop computers for legal reasons. Publicly traded companies and companies within certain industries are required to comply with various federal regulations (Sarbanes-Oxley, HIPAA, etc.). These regulations outline the ways in which computers must be secured. Failure to adhere to such regulations can result in large fines or even in criminal charges.

Even if your company is not subject to any of these types of regulations, it is still important to take security seriously. Some states have laws requiring companies to make public any security breach in which customer information was exposed. At best this type of bad publicity will cost your company a few customers. At worst, the company could become a target for civil litigation.

One last reason why it is so important to control the software that is installed on your workstations is because of software licensing laws. The workstations belong to the company, so therefore the company is responsible for their contents. If a user were to install a piece of unauthorized software, the company is technically in violation of various copyright laws because the company does not own a license for that software. If the company's software were ever to be audited, the company would likely receive a hefty fine, even though the IT staff is unaware of the offending application(s).

The Challenges of Locking Down Desktops

As important as it is to lock down your company's desktops, the task is actually quite challenging. One challenge has to do with the distributed nature of workstations. In most organizations, the IT department is already overburdened, and it would be extremely impractical to expect the IT staff to manually configure a software restriction mechanism individually on each workstation.

On top of that, Windows networks, by their very nature, are constantly changing. What this means is that the software restrictions that you implement today probably are going to be insufficient for tomorrow. For example, even if you were somehow able to block all of the software that you have not specifically approved, your policy would quickly become outdated because of the frequent nature of Microsoft patches.

One final issue that complicates workstation lockdown is the fact that the Windows operating system is not really designed to accommodate such an endeavor. There are pieces of the Windows operating system that are designed to prevent unauthorized software from being executed. Even so, these various mechanisms can be difficult to use and tend to be impractical for reasons that I will talk about more later on.

Requirements for Desktop Lockdown

There are lots of ways that you can prevent unauthorized software from being installed or executed on users' workstations. The problem with many of these methods, however, is that they are simply not practical. Some methods place far too heavy of a burden on the IT department in terms of the required workload. Other methods may be too costly, or are too easily circumvented by the end users. In this section, I want to discuss the criteria that would make a workstation lockdown solution practical.

The first criteria of a practical desktop lockdown solution is that it must be flexible. After all, there is nothing static about a corporate network. As business needs change, software requirements will likely also change. A desktop lockdown solution needs to be flexible enough that it can be easily configured to support the mobility of laptops as well as new applications and new versions of existing applications. The solution should also be adaptable to any new security policies, procedures, or workflow requirements. Because time is sometimes of the essence in regard to some IT projects, a solution needs to be simple enough and flexible enough to support ad hoc requests without overburdening the already busy IT staff.

A second criteria is that a desktop lockdown solution must be secure. I know that it's strange to say that a security solution must be secure, but it's the truth. Some desktop lockdown solutions can be circumvented by end users far too easily. For example, one of the solutions that I will talk about later determines whether or not a piece of software is allowed to run by looking at its installation path. If a user were to install a restricted piece of software to a different path, then the restriction would be completely circumvented. Therefore, that is not a good solution. A good solution is a solution that works regardless of whether or not a user changes paths, renames a file, modifies a file, etc.

My third criteria for an effective desktop lockdown solution is that it needs to have a good management interface. The management interface should be able to analyze files across your organization and group those files in a logical manner. Likewise, the solution should be able to produce reports regarding executable files and their locations. Ideally, I think that you should be able to restrict files from within the management interface as you go through the list of executable files in your organization.

My fourth requirement for a desktop lockdown solution to be considered practical is that it needs to be automated. Obviously any solution, no matter how good it is, is going to require some initial configuration. You are going to have to decide which programs should and should not be restricted. There's really no getting around that. Once you arrive at which executables exist on your network and you have decided what you do and don't want to allow, the software should be automated from that point on. You should have the option of configuring the software so that any new executables that happen to show up are automatically restricted and brought to your attention for possible approval.

Comparison of Solutions Against Requirements

Now that I've discussed the various requirements for a practical method of preventing unauthorized software from entering your network, I want to take a look at some of the solutions that are currently available. As I do, I'll discuss both the merits and shortcomings of each solution.

User Rights Restrictions

One of the first approaches that many administrators take in securing workstations is to set the NTFS permissions on the workstation's hard drive so that users have only the minimum necessary set of permissions. Although it is always a good idea to give users only the minimum necessary set of permissions, this technique by itself is completely ineffective in regard to preventing users from installing or executing unauthorized software.

One major issue with relying solely on user rights restrictions is that a user has to have rights to their profile directory. A profile directory stores a user's documents and all of their user-specific application settings. Since a profile is a required part of Windows, and a user has to have rights to their profile directory, a user could place an executable file into their profile directory and run it from there.

There are workarounds for some of the profile-related security issues though. For example, you could redirect a profile so that it is stored on a server rather than on each individual workstation. Once the profile folders have been redirected, there are other utilities that I will talk about later that can search profile folders for unauthorized executables.

Another option is to implement mandatory profiles. Mandatory profiles are designed so that any changes that a user makes to their profile directory are automatically overwritten with a clean and pristine copy of an approved profile when a user logs off.

Even using profile redirection or mandatory profiles will not completely prevent users from running unauthorized software though. The reason is because regardless of where a profile is located, a user must still have write permissions to it in order for applications to function correctly. In the case of a mandatory profile, a user can write to a local copy of the mandatory profile, and that copy is later overwritten by a clean copy of the mandatory profile when the user logs out. While a user is logged in

though, they have write access to their profile directory.

To see why this is a problem, think about the way that Internet Explorer works. When a user visits a Web page, the contents of that page (HTML code, images, etc.) are downloaded to a cache directory. If a user happened to visit a malicious Web page, any malware that might exist on the page is also written to the cache directory, where it would then be executed. If the user had a mandatory profile, the contents of the profile directory would eventually be overwritten, but by that time the damage may already be done.

This is just one example of why limiting a user's permissions over the workstation's hard drive does not guarantee that the user will not be able to run unauthorized code. Even if you could completely lock down a workstation in this way, you would likely impact the user's ability to do their jobs. For example, without sufficient permissions, a user would not even be able to install a print driver. Likewise, some applications will not even run without an elevated set of permissions.

One more reason why adjusting NTFS permissions alone isn't a good desktop lockdown solution is because there is no central way of managing the permissions. Microsoft does not offer a built-in console that allows you to set NTFS permissions across all of your workstations. Even if they did, performing blanket lockdowns at the NTFS level could make it difficult to install new applications or software patches.

Group Policy Objects and Software Restriction Policies

For computers that are attached to Windows Server 2003 or Windows Server 2008 domains, the primary mechanisms for locking down workstations are group policies. Group policies are a collection of security settings (called group policy objects or GPOs) that can be applied to either all of or a subset of the computers in a domain.

Group policies can be quite large and can cover a wide variety of security-related settings. The group policy objects that are probably of the most interest from a desktop lockdown standpoint are the software restriction policies.

Software restriction policies are made up of a collection of rules. There are two primary components to a rule: a scope and a security level. A rule's scope defines what the rule applies to. For example, if you look at the sample rule, shown in Figure A, you can see that the scope in this case is a registry path.

Figure A

A rule is made up of a scope and a security level.

The security level portion of the rule allows you to control whether or not applications within the defined scope are allowed to execute. For example, if you wanted to ban Real Player, you would set up a rule that defines Real Player's characteristics, and then you would set the rule's status to Disallowed. A security level can be set to either Disallowed or Unrestricted.

There are four different techniques that you can use to define a scope for a rule. Each of these techniques has its own strengths and weaknesses. Scopes can be defined by using certificate, hash,

path, and Internet zone rules.

Before I begin discussing the various rule types, you might be wondering why there are four different types of application definitions. The primary reason why there are four different types of rules is because no one rule will work for all applications. Another reason why there are four different types of rules is because each type has its strengths and weaknesses. Some rules are easier for users to circumvent than others. Therefore when creating a software restriction policy, you must consider not only which rules will be compatible with the applications that you're trying to restrict, but also which rules will be the least likely to be circumvented by your end users.



Certificate Rules

Certificate rules are probably the most secure of the various rule types. The idea behind certificate rules is that many companies digitally sign the code they publish. A certificate rule allows you to permit or restrict code based on the signature. For example, you could create a certificate rule that allows any code signed by Microsoft to run.

Although certificate rules tend to be effective, there are two problems with them. First, the majority of software publishers do not sign their code. This means that in most cases, you probably won't be able to use certificate rules. The other problem is that certificate rules are catch-all by nature. For example, if you were to create a rule that allowed any code signed by Microsoft to run, then you couldn't create

another certificate rule that prevents a specific Microsoft product from running. To do that, you have to create a hash rule, and prioritize that rule so that it takes precedence over the blanket certificate rule.

Hash Rules

Hash rules allow you to permit or restrict an application based on its hash. The nice thing about a hash rule is that unlike some of the rules that I will talk about later, hash rules are effective regardless of where the file is located on the system. Even if a user were to rename a restricted file, the hash rule would still be in effect.

The biggest weakness associated with hash rules is that they are created with the assumption that the restricted file is not going to change. If a change is made to the restricted file, then the file's hash value also changes, which means that an existing hash rule would no longer apply to the file.

A few years ago, this was no big deal. Most of the time the only way that file's hash would change would be if a new version of the application were released, or if a user used a hex editor to modify one of the bytes in the file. Today though, it is common practice for software publishers to frequently release patches for their products. Patches almost always overwrite the previous code, therefore rendering any existing hash rules against that code useless.

Path Rules

Path rules can be extremely powerful, but their downfall is that they are difficult to implement effectively. The basic idea behind a path rule is that you can prevent applications located in a certain path from executing. For example, suppose that you found out that one of your users had Microsoft Flight Simulator installed on the workstation. Microsoft Flight Simulator installs to the C:\Program Files\Microsoft Games folder by default. You could therefore create a path rule that prevents any application in C:\Program Files\Microsoft Games\ or in any of its subfolders from executing.

There are a couple of problems with this type of path rule though. First, the Program Files folder could potentially be located in a different location on some users' systems. Although it's rare for the Program Files folder to be located in a location other than C:\Program Files\, it is possible for the folder to exist elsewhere.

To get around this issue, you will need to use environment variables in place of an absolute path. Windows supports many different environment variables, but in this case the %programfiles% environment variable points to the Program Files folder regardless of its location. Therefore it's best to specify the path like this: %programfiles%\Microsoft Games\

Another problem with this type of path rule is that if the user wanted to circumvent it, all they would have to do is install the application to a different location. For example, if you created a path rule to block %programfiles%\Microsoft Games\, then all a user would have to do to get around the rule would be to install Microsoft Flight Simulator into a location other than the \Microsoft Games\ folder.

You can get around this problem. Almost all applications write data to the Windows registry. Typically, when an application is installed, data related to the application is written to the HKEY_LOCAL_MACHINE\SOFTWARE portion of the registry. If you can find a registry key that contains

the installation path to the application, you can specify that registry key in a path rule.

This is a way of telling Windows that the application could be installed anywhere, so you want to pull the application's path from the registry rather than referencing it directly.

To see how this works, let's go back to our example in which someone in the organization has Microsoft Flight Simulator installed. Microsoft Flight Simulator, like most applications, stores its set-up path in the registry. The location is `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft Games\Flight Simulator X\1.0`. The name of the registry key containing the set up path is `SetupPath`, as shown in Figure B.

Figure B

Most applications list their installation path in the Windows registry.

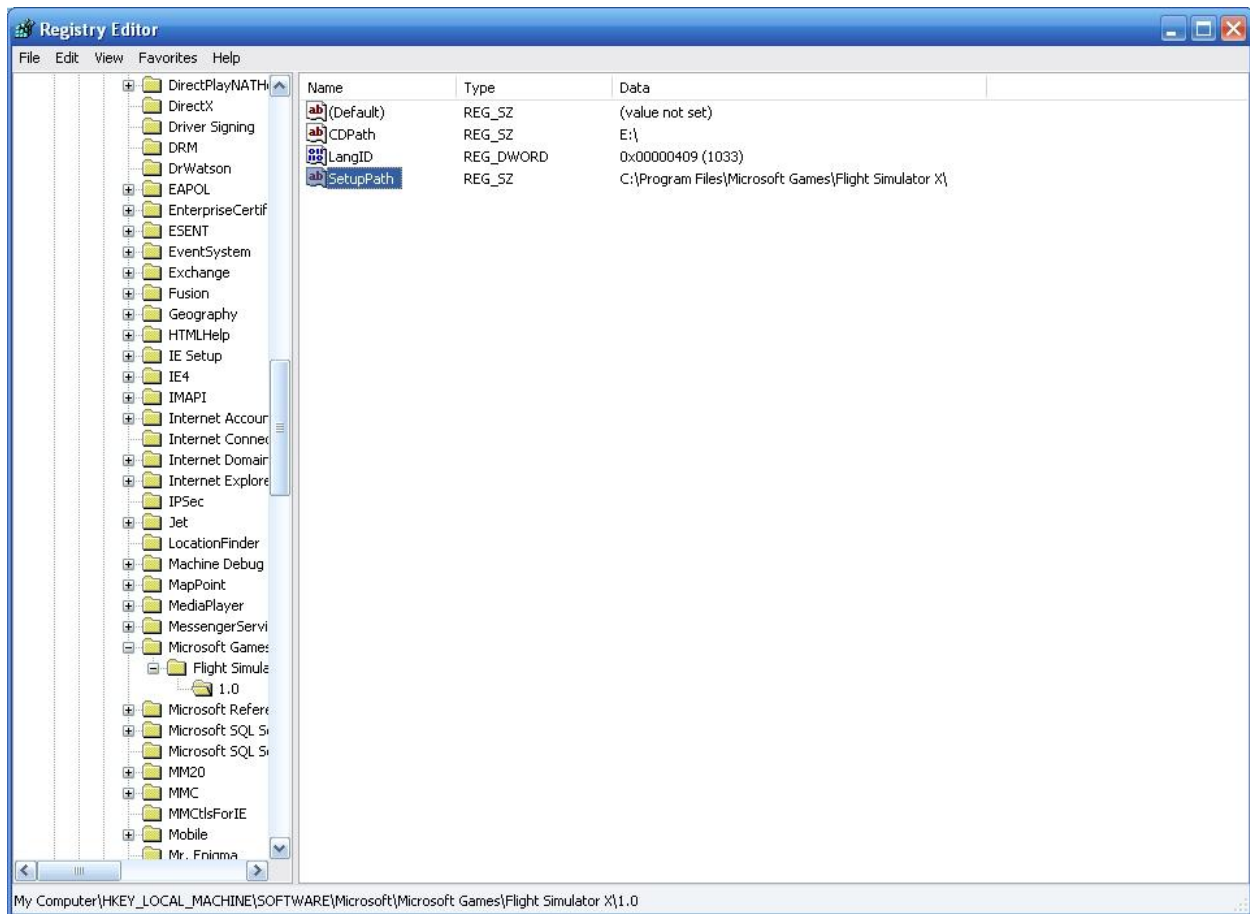
If you wanted to create a path rule that blocks this application based on the installation path listed in the registry, the path portion of the rule would look like this:

```
%HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft Games\Flight Simulator X\1.0\SetupPath%
```

Notice that the registry path contains a percentage sign on both ends.

There are a few other things that you can do with path rules. You can create path rules that reference UNC-based paths. You also have the option of blocking individual files or file extensions.

I mentioned earlier the importance of rule precedence. Some types of path rules automatically take priority over other types of path rules. Specific path rules take precedence over general rules. The order of precedence is as follows:



Path Rule Example

A specific file C:\folder1\folder2\filename.ext

A file extension in a specific location C:\folder1\folder2*.ext

Globally blocking a file extension *.ext

Blocking a sub folder C:\folder1\Folder2\

Blocking a root level folder C:\folder1\

If a registry rule is used, its precedence is based on the path specified within the registry.

Internet Zone Rules

An Internet zone rule is the least powerful of all the software restriction policy rules. An Internet zone rule allows you to allow or restrict an application based on what Internet zone the site that the file was downloaded from falls into.

There are two reasons why this type of rule isn't very powerful. First, it only works against files that are downloaded from the Internet. If a file exists on the local hard disk, then this type of rule is ineffective regardless of where the file came from.

The other reason why this type of rule isn't usually very useful is because it only works against Windows

installer files. If the user downloads executable code from the Internet, an Internet zone rule will only go into effect if the file being downloaded is an MSI package.

Final Thoughts on Software Restriction Policies

As you can see, each type of software restriction policy rule has its strengths and weaknesses. If you're going to use software restriction policies in your organization, then it usually works best to use a combination of rule types. The main problem with using multiple rule types is that it is possible for rules to be contradictory. Therefore, it is important to establish rule precedence so that Windows knows how to deal with any contradictions that may occur.

The complexities of developing a set of rules that are effective is not the only problem with using software restriction policies though. Probably the biggest problem, aside from complexity, is that software restriction policies must be managed through the Group Policy Editor. Because the Group Policy Editor is a catch-all tool for working with all different types of group policy objects, it lacks the management features that would make it more useful. For example, the existing interface does not have a way of automatically discovering new applications on the fly.

There is also no easy way to provision for new applications. Yes, it is easy to create an exception that would allow an application to run. The problem is that you must take a look at rule precedence to make sure that the exception will be effective. You must also make sure that the exception policy is designed in such a way that it does not interfere with any other rules that you have already established.

At the end of the day, the biggest problem with software restriction policies has to be their complexity. Each type of rule has its strengths and weaknesses, and most rules can be circumvented in some way. The only way to really accomplish your goal of locking down workstations is to use multiple rule types, and to apply those rules in the correct order. This requires careful planning and lots of testing. As the organization's needs change, and new applications are adopted or old applications are phased out, you may find yourself having to completely re-evaluate the existing rules.

The Windows Terminal Services

One way that many organizations are attempting to gain tighter control over the end-user experience is to implement the Windows Terminal Services. In case you're not familiar with a Terminal Services environment, it's a type of Windows Server deployment in which all applications run on the server itself rather than on the user's workstations. The users rely on either a PC or a terminal to establish a session with the terminal server. Once the user establishes the session, all applications are run on the server itself. Windows uses the RDP protocol to transmit screen images from the server to the user, and to transmit keyboard and mouse inputs from the user to the server.

When it comes to locking down desktops, a terminal server deployment initially seems ideal. After all, all of the applications are installed on the server, and only the system administrator has the rights to install additional applications or to remove existing applications.

The one major problem that I often see with Terminal Services deployments in regard to desktop lockdown is that some administrators completely neglect the desktops. The logic behind this is that everything is running on the server, so why worry about the desktops?

While it is true that all applications are running on the server, this is only the case after the user establishes a session. Up to that point, the user's workstation functions just like any other PC. The computer is running a Windows operating system as well as the client software that is used to attach to the terminal server, and if an administrator chose to neglect security on the PC, it is also running anything else that the user might have installed. The PC is left completely vulnerable.

If your terminal server deployment uses terminals rather than PCs as user workstations, then the workstation's contents are not really an issue. If you are using PCs though, the contents of the workstation's hard drives can be particularly difficult to control.

This has to do largely with the way that the Terminal Services work. In a terminal server environment, users are not required to authenticate until after a terminal server session has been established. What this means is that the workstations do not even have to be domain members.

Earlier I talked about how software restriction policies could be used to regulate the contents of workstation hard drives. Keep in mind that software restriction policies are nothing more than group policy objects. Group policy objects are applied by domain controllers to users and computers connected to the domain. If a workstation is not a member of a domain, then none of the group policy objects that you establish will apply to that workstation.

My advice is that if you're going to run a Terminal Services environment, then use terminals rather than PCs as user workstations. If terminals simply are not an option, then you need to consider making the workstations domain members or investing in third-party desktop lockdown software.

Windows Server 2008 and Windows Vista

Windows Server 2008 and Windows Vista both use the same underlying code base, and are therefore very similar to each other. Microsoft claims Windows Server 2008 and Vista are the most secure operating systems that Microsoft has ever released. These operating systems contain improved versions of many of the security features found in Windows Server 2003, and contain some new security features as well.

Software Restriction Policies

As with Windows Server 2003, the primary mechanisms in Windows Server 2008 for preventing unauthorized software are the software restriction policies. Although software restriction policies have not changed a great deal since Windows Server 2003, some changes have been made.

Software restriction policies still use the same basic types of rules in Windows Server 2008 as they did in Windows Server 2003. One change that has been made to the rules though, is that the Internet Zone Rules have been changed to Network Zone Rules, as shown in Figure C. The significance of this change is that there is now a Local Computer entry in the list of zones. This means that you could potentially create a policy that would prevent any code from executing unless it is running on the local computer. Of course this is just an example, in real life you may also want to allow code to run if it is located on the local intranet or on a trusted site.

Figure C

Zone rules will support the local computer in Windows Server 2008.

Another significant change to software restriction policies in Windows Server 2008 has been made to the security levels. As you may recall, in Windows Server 2003 there were only two security levels available: Disallowed and Unrestricted. A third security level has been added to Windows Server 2008. This new security level is called Basic User, as shown in Figure D.

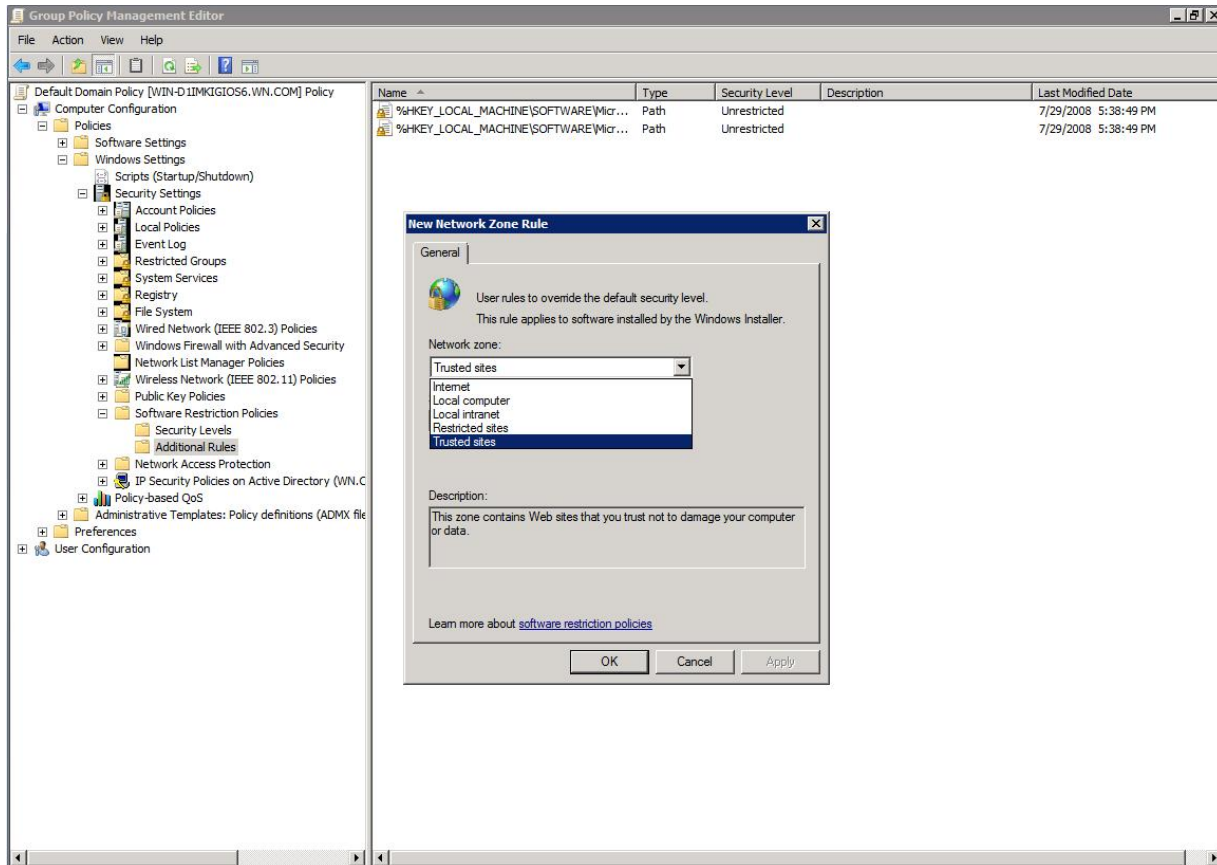


Figure D

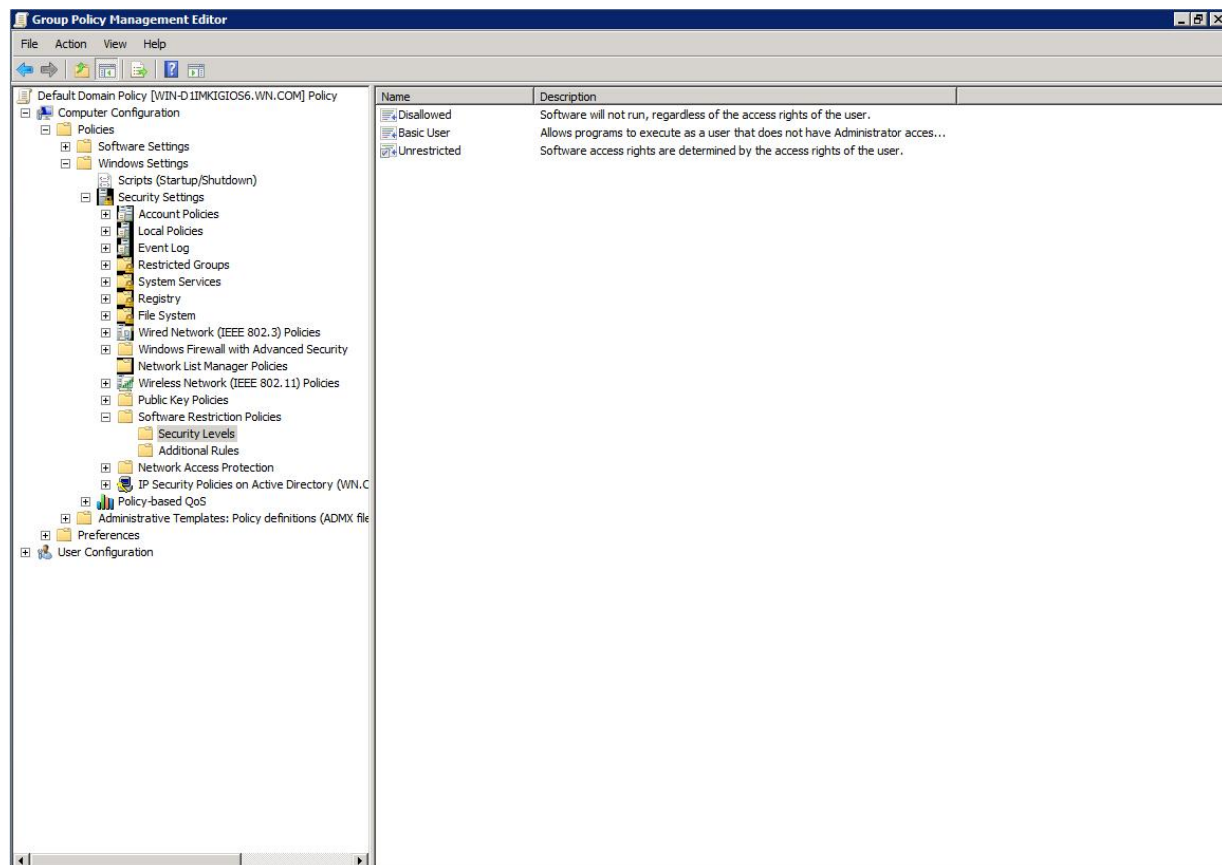
The Basic User security level will allow an application to run, but with a reduced set of permissions.

The Basic User security level will not prevent users from running software for which the security level is applied, but it will restrict their rights when running the application. If the Basic User security level is applied to an application, then all users for whom the rule applies will only be able to run the application as a standard user, regardless of what other permissions they may have.

The File System Resource Manager

The File System Resource Manager was actually introduced in Windows Server 2003 R2, but was not highly publicized. This tool remains virtually unchanged in Windows Server 2008. The File System Resource Manager is the closest thing that the Windows operating system has to a file analysis and reporting tool.

The File System Resource Manager acts as both a policy enforcement tool and as a file system reporting tool. At first, this probably sounds like the perfect tool for locking down a file system, but as you'll see in a moment, this tool is not appropriate for locking down desktops.

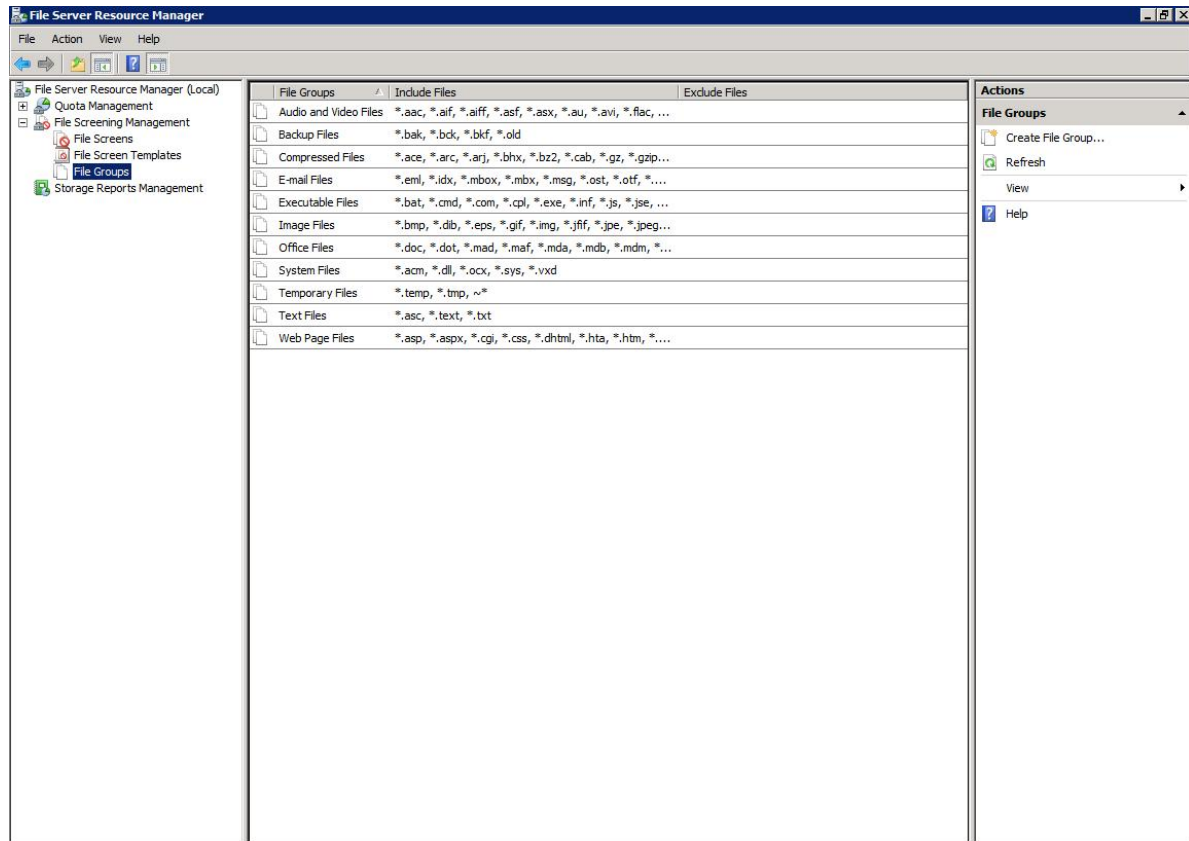


From a policy enforcement standpoint, the File Server Resource Manager works by using file screening technology to prevent certain types of files from being stored in designated locations. For example, you could use file screening to prevent users from storing executable files within home directories.

Microsoft has even gone through the trouble of creating file groups that define various file types. If you look at Figure E for example, you'll see that the File Server Resource Manager contains a file group named executable files. This file group tells Windows which file extensions are considered to be executable. That way, if you wanted to restrict executable files from being stored in a specific location, you would simply select the File Screens container and create a new file screen. When creating the new file screen, you would pick the location that you want to protect and the file types that you want to block. In this case that would mean choosing the Executable Files group.

Figure E

The File Server Resource Manager allows you to restrict files based on file type.



The File Server Resource Manager also allows you to generate a wide variety of reports. Among the reports that you can generate are a file screening audit and a report of files by file group. The report of files by file group is especially useful because it allows you to view all of the executable files within a specific location.

As I said earlier, the File Server Resource Manager initially seems ideal for securing a file system. There are two primary problems with this tool though that make it impractical for desktop lockdown. The first problem is that the tool is easily overwhelmed. As you can see in Figure F, the tool will only report the first 100 files in each file group.

Figure F

The File Server Resource Manager's Files by File Group report only displays the first 100 files in each file group.

The other major problem that is even more of an issue is the fact that this tool is designed to lock down

the file systems of servers, not workstations. This tool is great for locking down file servers, but is powerless to lock down desktops.

Least Recently Accessed Files Report
Generated at: 7/29/2008 5:49:03 PM

Report Description: Lists files that have not been accessed recently. Use this report to quickly identify stale files that can be either deleted or archived. This can help you to reclaim disk space that isn't being actively used.

Machine: WIN-D1MKIGIOS6

Report Folders: C:\

Parameters: Minimum accessed days: 90 Days

⚠ More than the maximum number of files matched the report criteria. Only the top 1000 matches are shown.

[Least Recently Accessed Files Report Table of Contents](#)

[Report Totals](#)
[Size by Owner](#)
[Size by File Group](#)
[Report statistics](#)

Report Totals			
Files shown in the report		All files matching report criteria	
Files	Total size on Disk	Files	Total size on Disk
1000	332 MB	53028	10,547 MB

[To top of the current report](#)

Size By Owner

Owner	Size (MB)	Percentage (%)
NT AUTHORITY\SYSTEM	5,391	51.11
NT SERVICE\TrustedInstaller	3,218	30.51
BUILTIN\Administrators	1,859	17.63
Others	78.9	0.75

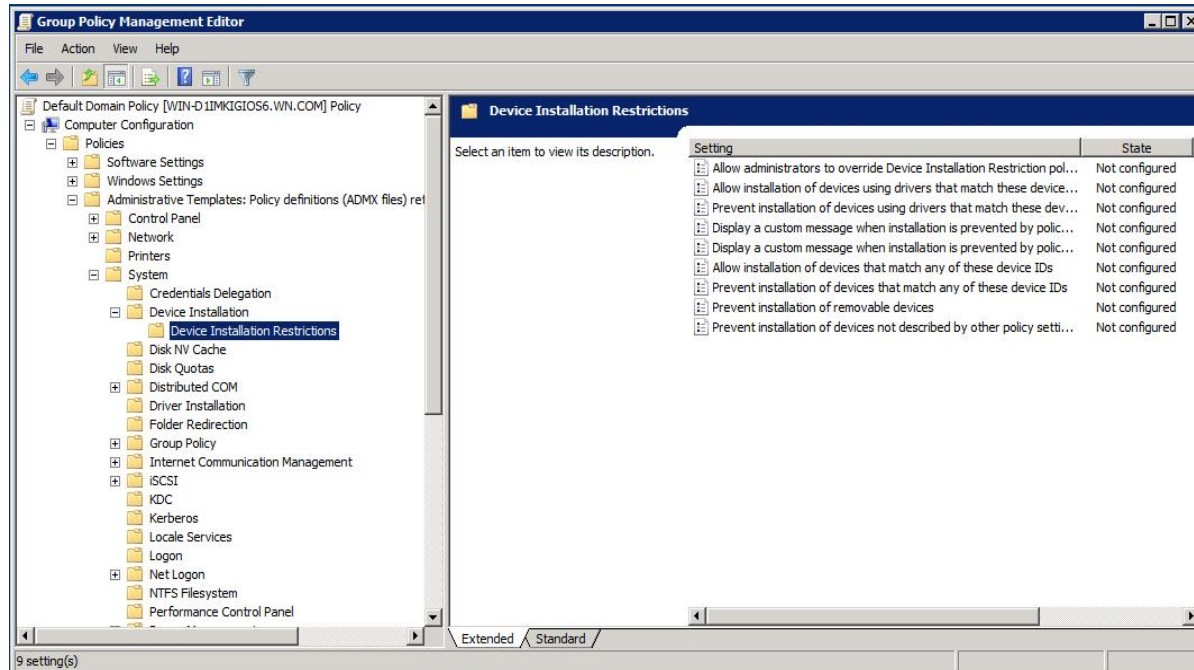
In spite of its weaknesses, the File Server Resource Manager is an effective tool in situations in which a group policy setting is being used to redirect a user's profile folders onto a file server. In situations like this, it would be easy to use file screening to make sure that users are not placing executable code into their profile folders. You could also use the File Server Resource Manager to generate reports of what file types actually exist within the various folders.

Preventing Device Installation

Another new feature in Windows Server 2008 that can help in the fight against unauthorized software is a new set of group policy settings that allow an administrator to prevent users from installing removable storage devices, as shown in Figure G. The reason why this feature is important is because of the ubiquitous nature of USB flash drives.

Figure G

Windows Server 2008 contains group policy objects that allow you to restrict the use of removable storage devices.



As it is now, users could potentially place executable code onto a USB flash drive on a computer at home. They could then bring that USB flash drive into the office, plug it into their workstation, and copy the code to their hard disk. The new group policies available through Windows Server 2008 can be configured to prevent anyone other than administrators from using external storage devices.

This feature can go a long way in protecting your organization against unauthorized software, but it does have its shortcomings. For example, a user could still bring executable code into the organization by downloading it from the Internet. Being that Windows Server 2008 does not appear to have a mechanism for detecting new executable code, preventing the use of removable storage devices will only go so far in protecting the organization.

Trusted Software Approval

The term graylisting is usually associated with anti-spam software. When it comes to anti-spam software, there are whitelists of approved senders and blacklists of senders from whom you never want to accept mail. The idea behind graylisting is that any sender who has not previously sent a message to the recipient, and who is not listed on a whitelist or a blacklist, is graylisted. This means that the sender is not trusted until the message is proven to be trustworthy, but the sender is not immediately blacklisted either. Instead, the mail server returns a code to the sender indicating that the server is too busy to accept the message. Legitimate mail servers will automatically resend the message later, whereas most automated mailing list applications won't. If the sender does send the message again later on, it is assumed to have come from a legitimate source and is accepted.

I don't want to get into a discussion of anti spam-related graylisting, because spam control is beyond the scope of this paper. I mention it because similar graylisting techniques can be applied to applications.

Think about the applications on your workstations for a minute. There are applications that need to be able to run, such as Microsoft Office, or maybe Adobe Acrobat Reader. These are applications that were installed by the IT staff, and can therefore be thought of as whitelisted applications.

On the other hand, there are applications that you know for sure that you never want to run on your workstations. Some examples of these types of applications might include games or peer-to-peer file sharing applications. These can be thought of as blacklisted applications.

Finally, there are the executable files that show up on workstations every day that the IT staff did not install. These can be considered to be graylisted applications, because until you check to see what they are, you really don't know if those applications should be allowed to run or not.

With this type of definition, it's easy to think of graylisting as simply quarantining any new executable files that have not been specifically approved. That really isn't the idea though. Earlier when I talked about the requirements for an effective desktop lockdown solution, I mentioned that one of the requirements should be that the solution should not place a huge burden on the IT department. If graylisting were just a matter of quarantining executable code until an administrator could approve it, then the process would place a huge burden on the IT department.

Imagine for example that you have a carefully orchestrated set of software restriction policies in place that block any executable file that has not been specifically approved. Now imagine that a critical patch is released for one of your applications. The patch can't be applied because your software restriction policies are blocking it. In this type of situation, an administrator would have to go into the software restriction policies and write an exception rule that allows the patch to run. Of course they would have to be careful to write the rule in a way that wouldn't compromise any of the other rules.

Adding exception rules when necessary can be done, but it is a lot of work, and you run the risk of accidentally circumventing existing rules. That's why process graylisting isn't just about quarantining executable files until they are approved.

If you think back to my description of e-mail graylisting, you will recall that when a message was received from an unknown sender, the server returned a "server too busy" error to test if the sender's mail server would resend the message. The network administrator does not have to manually flag each sender as either being a legitimate sender or a spammer. Instead, the anti-spam software uses a test to determine if the sender is a spammer or not.

Just as anti-spam graylisting involves a test, so too does process graylisting. The administrator can define a series of tests for new executables. New executables can then be either approved or banned automatically based on the outcome of the tests. This automatic approval process frees the IT staff from the burden of having to manually approve or ban each executable file that comes along.

Because process graylisting capabilities are not built into the Windows operating system, it means that you will not be able to implement process graylisting through group policies. In a way this is a good thing, because it means that you will not be affected by limitations in the Group Policy Editor.

Instead, software publishers wishing to implement process graylisting capabilities in their products would likely have to rely on placing an agent on each workstation since policies cannot be pushed out via group policy. The agents would likely be used to relay information about executable code to and from a server running a software approval application.

One of the nice things about using this approach is that it means that software publishers creating such an application could develop a very comprehensive management console that takes advantage of the underlying architecture. At a minimum, such a management console could probably be used to perform software audits and inventories and to generate reports based on those functions. I'm sure that such a console would also allow administrators to perform manual point-and-click approvals if necessary.

A nice side effect of having an agent monitoring the contents of workstation hard drives is that the chances of malware infections are greatly reduced. Malware code must execute in order to do any damage. If an agent is monitoring workstation hard drives and preventing new code from executing until it has been proven trustworthy, then theoretically no malware would be allowed to run.

Since all of the agents communicate with a centralized management console, it could even be possible for software publishers to include a panic button in the management console. This panic button could be used to instruct the agents not to allow any code (approved or otherwise) to run, as a response to a zero-day malware attack.

Conclusion / Recommendations

Of the various desktop lockdown solutions that are available, process graylisting is the only one that seems practical.

It is possible to lock down a desktop using a combination of software restriction policies and NTFS rights manipulation, but doing so leaves a great deal of room for error. These techniques also lack the flexibility that would allow them to automatically adapt as an organization's needs change.

Although rights management and software restriction policies are capable of locking down workstations, Windows lacks a built-in software inventory utility. Without such a utility, there is no centralized way of checking to see which executable files exist on the desktop machines without performing some sort of manual search. This is even true of Microsoft's Windows Server 2008 operating system, which is not slated for release until sometime in 2007.

About the Whitepaper Sponsor

About Bit9, Inc.

Bit9 provides the easiest and most effective way for Windows Desktop Administrators to enforce desktop application usage policies via automatic whitelisting. By centrally controlling which applications can and cannot run, Bit9 drastically reduces the volume of desktop support calls, thereby maintaining the highest levels of availability, compliance, and security. Unlike other application control solutions that require significant setup and intervention, Bit9 installs in minutes and automates the approval or banning of new applications. Founded in 2002 by the founders of Okena (acquired by Cisco) and headquartered in Cambridge, Massachusetts, Bit9 is a privately held company. For more information, visit www.bit9.com.