Global Information Assurance Certification Paper

Richard Luhman
GSEC Version 1.4 (Option 1)

Permissions in Windows 2000

Introduction

Security of our belongings is an important aspect of life. We lock our doors at home or at the office to protect our personal/business belongings when we are not around. Most people never leave their keys in the ignition with the automobile running in fears that the automobile may end up in someone else's hands. However, there are times that we may allow someone to check in on those belongings or in some cases give individuals temporary or permanent permissions to utilize certain portions of our belongings.

With the advent of networking and in particular the Internet securing our computer systems has become more complex then just locking our doors. Many individuals on a LAN leave their systems on 24/7. Individuals who have home systems may also leave their systems on 24/7 and these home systems are often connected to the Internet via broadband connections.

This paper will discuss the aspects related to securing Windows 2000 computers at the file and folder level.

What are Permissions?

In Windows 2000 permissions are assigned to objects. Object examples include folders, files, virtual directories (IIS) and printers. These permissions can only be assigned if the NTFS file system is used. Permissions are not to be confused with rights. Microsoft distinguishes permissions and rights by:

"User rights are different from permissions because user rights apply to user accounts, and permissions are attached to objects (such as printers or folders)." (2002 Microsoft Corporation)

Basic File and Folder Permissions:

There is no built in file and folder permissions if the FAT or FAT32 file system is selected during a Windows 2000 installation. This also means that file and folder permissions do not exist in the Windows 9x family.

There are six basic types of Windows 2000 permissions for NTFS folder permissions. NTFS file permissions are similar to folder permissions with the exception of the omission of List Folder contents.

- **Full Control** – Full control of files and folders.
- **Modify** – Make changes in files and folders.
- **Read & Execute** – Read documents and execute programs.
- **List Folder Contents** – View folders and files
- **Read** – Can display files and folders. Can also view folder attributes, ownership and permissions.
- **Write** - Can create new files and folders and make changes to those files and folders.

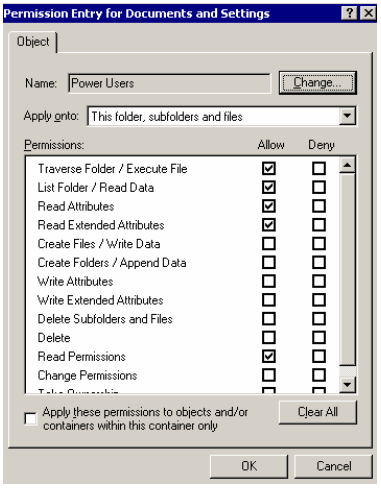
A company's organizational chart could be one source to use when planning permissions on Windows 2000 machines. Personal (access only by one team member and the computer admin) and public (access to all team members) objects could be created for each team member. The simplest approach would be to assign team leaders full control of the object and team members the Read or Read and Write permissions. However, one must remember that if full control is assigned to an object the individual with full control than has the power to re-assign permissions.

For large organizations it could become quite a burden to assign individual user permissions to objects. In these cases security groups should be established and permissions assigned via these groups. If a user belongs to a group and the user and the group have permissions to the same object the permissions are additive or "least restrictive".

Example

1. Settings
    a. User1 has read permission to object1
    b. User2 has read and write permissions to object1
    c. User1 belongs to Group1 and Group1 has full permissions to object1
2. Results
    a. User1 has full permissions to object1
    b. User2 has read and write permission to object1

Advanced Permissions:



The six basic rights can be tweaked with more advanced permissions. Table 1 lists the advanced permissions that are associated with each of the basic permissions. The "Black Hole" list of permissions was described by http://www.uwec.edu/Help/Win00/permissn/types.htm as a set of permissions that could be used in a high secure situation. In the "Black Hole" situation users can add files but not see a list of files, read documents or edit documents.

Table 1: Advanced File and Folder permissions (taken from http://www.atlguide2000.com/eng/win2k/ntfs1.htm and Finder, Kathy http://www.uwec.edu/Help/Win00/permissn/types.htm)

	Full	Modify	Read & Execute	List Folder Contents	Read	Write	Black* Hole
Traverse Folder / Execute File	X	X	X	X			X
List Folder / Read Data	X	X	X	X	X		
Read Attributes	X	X	X	X	X		X
Read Extended Attributes	X	X	X	X	X		
Create Files / Write Data	X	X				X	X
Create Folders / Append Data	X	X				X	X
Write Attributes	X	X				X	X
Write Extended Attributes	X	X				X	X
Delete Subfolders and Files	X						
Delete	X	X					
Read Permissions	X	X	X	X	X		X
Change Permissions	X						
Take Ownership	X						

* not one of the basic permission settings.

Examples of further tweaking could be to remove the delete permission under modify so users could not delete files. The modify model could also be opened up even further by allowing users to take ownership of files but not change permissions or delete files or subfolders.

There may be cases in which one may desire that some file level permissions within a folder be set differently than the folder. An example could be where an administrator does not want users to have write access to a script he has written. However, the script writes to a text file in the same directory in which the users must have write access to the text file. If permissions are assigned to files within a folder those permissions for that file would over-ride the folder level permissions.

Assigning File and Folder Permissions:

File and folder permissions within the access control list (ACL) can be viewed or modified by right clicking on the file or folder, selecting properties and then selecting the security tab. At this point user permissions can be added, modified or deleted. Permissions to files can also be viewed or modified via the command line (calcs.exe) The syntax for cacls.exe can be viewed by typing cacls /?.

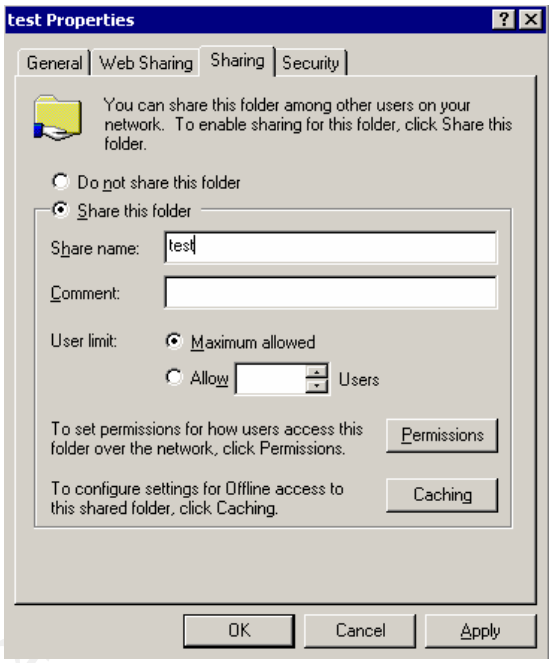
```
Displays or modifies access control lists (ACLs) of files

CACLS filename [/T] [/E] [/C] [/G user:perm] [/R user [...]]
               [/P user:perm [...]] [/D user [...]]
   filename      Displays ACLs.
   /T            Changes ACLs of specified files in
                 the current directory and all subdirectories.
   /E            Edit ACL instead of replacing it.
   /C            Continue on access denied errors.
   /G user:perm  Grant specified user access rights.
                 Perm can be: R  Read
                              W  Write
                              C  Change (write)
                              F  Full control
   /R user       Revoke specified user's access rights (only valid with /E).
   /P user:perm  Replace specified user's access rights.
                 Perm can be: N  None
                              R  Read
                              W  Write
                              C  Change (write)
                              F  Full control
   /D user       Deny specified user access.
Wildcards can be used to specify more that one file in a command.
You can specify more than one user in a command.
```

Inheritance:

Inheritance is a means by which permissions flow down the directory structure. Permissions that are set in the parent directory flow to the folders and files within subfolders of that parent directory. However, there may be cases where one does not want to allow inheritance to occur. In NT4, inheritance was mandatory. You needed to set permissions on the parent directory and then change permissions on each subdirectory if you wanted those permissions to be different then the parent. At a later time if you wanted to change permissions on the parent directory all permissions on subdirectories and files would be changed to the parent permissions. In Windows 2000 you can stop inheritance from flowing to an object by removing the check on "Allow inheritable permissions from parent to propagate to this object." Permissions on that object are then independent of the parent. This feature could be beneficial if one has a directory structure such that the parent directory contains a single administrator who has full control of the entire directory tree and employees have various permissions on the sub-folders. If at a later time a second administrator is granted access to the parent directory all sub-folder permissions would need to be re-created unless the "Allow inheritable permissions from parent to propagate to this object" was unchecked. Of course, another way around this problem is to set permissions via groups rather than individuals and add the new employee to the group.

Folder Sharing:



Permissions discussed up to now have been those that are set on the local system. If an individual needs access to files via a network connection a share would need to be established. Share permissions are set via the share tab when you right click on a folder and select properties.

During a Windows 2000 installation that authenticates against a domain controller there are several shares that are set up automatically. These shares are called the administrative shares and are hidden (designated by a $ after the share name). The administrative shares are accessible only to those having administrator, backup operator, or server operator rights to the machine. If these default shares are removed they are re-created upon system re-boot. However, the shares can be permanently removed with the Policy Editor (poledit) or via the registry. Permissions cannot be changed for the administrative shares. One may wish to remove these administrative shares in a very secure environment. For example, an organization's administrator does not wish to have anyone, including the computer system's administrator, to have access to his/her computer unless the organization's administrator is present.

Permissions can be set for user created shares. Walthall (2001) indicated that users who have administrator, power user, or server operator rights can create shares. New shares can be created via either a GUI interface or via a command.

The Net.exe Windows NTR/2000 program can be used for setting up shares. Sheppard (2002) listed the commands on how to use the net.exe program. The user command for setting shares is

"NET SHARE sharename=drive:path /REMARK:"remark text". The share can be deleted by NET SHARE sharename /DELETE" without the quotes.

There are several ways to create a share via a GUI interface in Windows 2000. One way is via the properties of the folder.

1. Right click on the folder that you want to share
2. Select sharing

3. Check the item "Share this folder"
4. Give the share a name
5. Type in a description if you would like
6. Set permissions
7. click OK

Cerelli (2001) documented a second way to create a share via the program SHRPUBW.EXE program. This is the same program that will run when using the computer management snap-in in the Microsoft Management Console if you right click shares and select new file share.

1. Click start
2. Click run
3. Type shrpubw.exe and click OK
4. Browse to or type in the folder you want to share
5. Give the folder and share name
6. Type in a description if you would like
7. Click next
8. Set permissions and click finish

Permissions can be set for shared folders. The default value for permissions is to allow everyone who has access to your computer full permission to the share. In most cases, allowing everyone who has access to your system full permissions to a share would not be recommended. Chien (2001) has indicated that these open network shares are one of the ways that the Nimda worm has infected systems. The other vehicles of Nimda infection include via Microsoft Outlook email package and via an Microsoft's Web Server, Internet Information Server. It is therefore recommended that systems be monitored so that only those needing access to file shares have access.

In a Windows 2000 server environment, if the OU administrator has granted rights to create shares by establishing administrator, power user, or server administrator users, that administrator must train the staff to create shares with only the permissions that are needed. In addition, the OU administrator should monitor systems in their OU for open shares via the active directory and users snapin to the Microsoft Management Console. Once the snapin is loaded the administrator would.

1. Right click on the computer name
2. Select Manage
3. Expand Shared Folders
4. For each share
    a. Right click the share
    b. Select properties
    c. Select share permissions

d. Verify that only those users/groups needing access have access to that share. If an open share is found the administrator would have the rights to remove that share or tweak the permissions so that the share is no longer open.

Permissions at the share level, like the file and folder level, are additive. That is, if an individual belongs to a group, and the individual and group both have share permissions, the least restrictive permissions are in effect.

Having permissions assigned both at the file and folder level and at the share level now adds a dilemma in determining the effective permissions of the folder. Since sharing is a network phenomenon, Microsoft has chosen to utilize the most restrictive permission scenario (as opposed to the least restrictive when looking at groups and users assigned to the same file or folder permission) between the network share permissions and the file and folder share permissions.

Example

1. Settings
    a. User1 folder level permissions are add files, list files, read their own documents, and edit their own documents (advanced folder permissions set to Traverse folder / Execute file, List folder / Read data, Read attributes, Read extended attributes, Read permissions).
    b. User1 share level permissions on that folder are full permissions.
2. Effective permissions of User1 for that folder would be advanced folder permissions set to Traverse folder / Execute file, List folder / Read data, Read attributes, Read extended attributes, and Read permissions, the most restrictive of the two permissions.

Internet Information System (IIS 5.0)

Permissions in IIS may have an effect on the effective file and folder permissions that a user has. IIS permissions are divided into Web permission levels and executable permission levels. The IIS permissions are only for permissions to execute HTTP verbs. IIS includes the following permissions:

The Web permissions levels are

- **Read**-users can view files and file properties
- **Write**-users can change files and file properties
- **Script Source Access**-available only if read or write is checked
- **Directory browsing**

The executable permissions include

- **None**-no executables are allowed

- **Scripts only**-only scripts are allowed to run (e.g. asp and php)
- **Scripts and Executables**

IIS permissions can be set via a couple of routes. If IIS is installed on a Windows 2000 system, you can right click on any folder or partition select properties and see a Web Sharing tab. If "share this folder" is selected a box will appear that has the permission options that are listed above. Once that folder is WEB shared the folder becomes a virtual directory in the IIS default WEB.

The other method for selecting IIS permissions is within the Internet Services Manager. With this method expand any WEB within the Internet Services Manager, right click on the folder and select properties to assign permissions. If you want to add a virtual directory to the WEB and assign permissions you would right click on the WEB, select new and then select virtual directory. Follow the Wizard to set up the virtual directory. The permissions are set during the Wizard setup procedures.

As with file shares, the interaction between NTFS and IIS shares are that the most restrictive permissions are utilized.

Example A

1. Settings
    a. User1 WEB permissions-Read and Write
    b. User1 NTFS permissions-Read permissions
2. Results
    a. WEB User1 cannot write to the file because NTFS does not allow the write permission
    b. File manager User1 cannot write to the file because NTFS does not allow the write permission

Example B

1. Settings
    a. User1 WEB permissions-Read
    b. User1 NTFS permissions-Read and Write
2. Results
    a. WEB User1 cannot write to the WEB site because NTFS can understand that the WEB user only has read permission and therefore the most restrictive case takes over.
    b. File manager User1 can paste to the directory via the file system because IIS can only control access via HTTP requests.

Example C

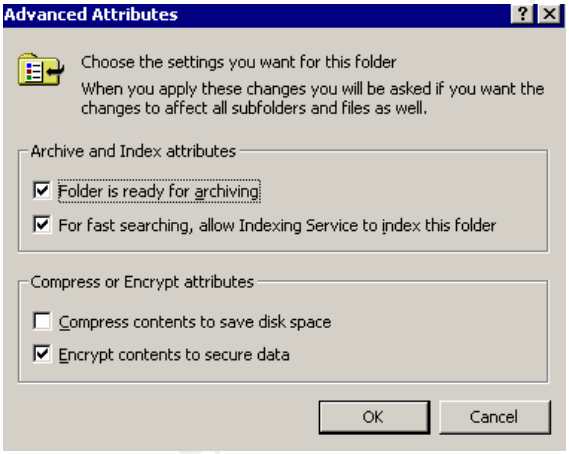
1. Settings
    a. User1 WEB permissions-Read, Write and Directory Browsing

b. User1 NTFS permissions-Traverse Folder/Execute File; Read Attributes; Create Files/Write Data; Create Folders/Append Data; Write Attributes; Write Extended Attributes; Read Permissions
c. User1 Share permissions-Full control

2. Results
   a. Web User1 can write a file to the WEB directory
   b. Web User1 cannot edit anything on the WEB directory
   c. Web User1 cannot browse the WEB site because NTFS rights are more restrictive then IIS permissions
   d. File manager User1 can write a file to the directory
   e. File manager User1 cannot edit anything in the directory
   f. File manager User1 cannot browse the directory

Encryption:

The use of encryption will have an effect upon the read permission of any file that has encryption set. Windows 2000 offers users the ability to encrypt files and folders with the encrypting file system (EFS). As with file and folder permissions the drive must be formatted as an NTFS drive. Windows 2000 utilizes the DESX algorithm to encrypt folders and files. The file is encrypted with a 128-bit random file encryption key. The EFS then utilizes RSA and the user's public key to encrypt the file encryption key which is saved with the encrypted file.

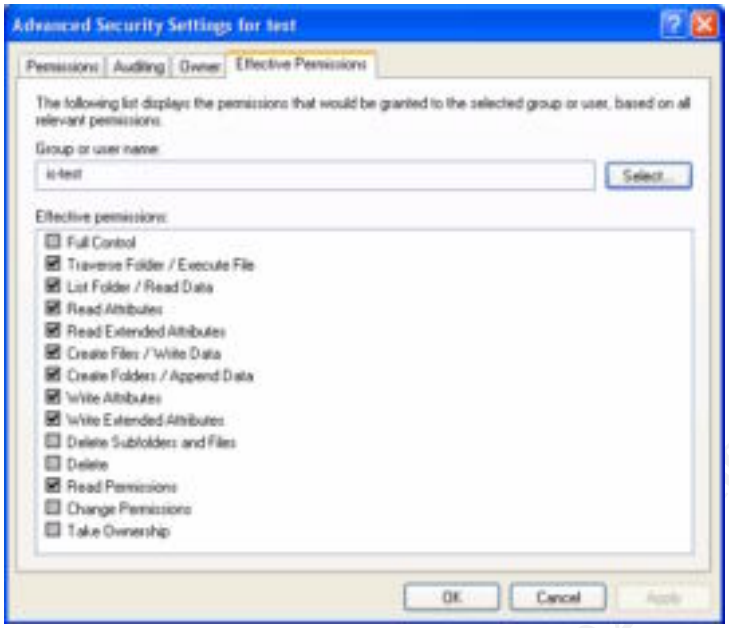


Accessing the encryption system can be accomplished either of two ways. The first is via the properties of the file or folder. The second method is via the cipher command.

Once the file or folder is encrypted the contents within that file or folder become inaccessible to everyone except the owner of the private and public keys or to those who have been granted as recovery agents. Therefore an individual who has read access to a folder that does not have access to the public and private keys cannot read the contents of files that have been encrypted. However, they can still delete the encrypted file if they have been given the delete permission to the file or folder.

Figueroa (2001), has indicated that some virus protection programs will have difficulty scanning files that are encrypted because the virus program may not have access to the public and private keys. Therefore, it would be recommended to scan any file for viruses before it is encrypted and to exclude encrypted folders from a full virus scan.

Beyond Windows 2000:



Determining effective permissions can be a tedious process. This would be especially true for persons new to Windows 2000 or when there is turnover within the computer management of an organization. Good records of what has been accomplished in a Windows 2000 environment are essential in the organization of these permissions. However, sometimes records are not kept or records are inaccurate. Apparently, Microsoft has realized that effective permissions are difficult to determine and have given users the ability to see these effective permissions in a GUI interface with the Windows XP Professional installed into a domain environment. This GUI interface of effective permissions does not, however, negate the importance of determining how these permissions were established. Were the permissions established via folder and file permissions, via shared permissions, via IIS or via some combination of permissions? If permissions indicate one type of access and the user indicates that he/she has other types of access one may wonder what is going on. Examples could include the use of encryption or password protection on a file (for example a password protected Microsoft Outlook personal folder.

Conclusion:

I was once asked if it would be advisable to store credit card information and passwords in a Microsoft Excel file on a Windows 95 system that was connected to the Internet via a local area network without a firewall or any intrusion detection software. My response was a shocked, definitely not. I recommended that if the user must store that information on a computer hooked to the Internet, he should at least upgrade his system to Windows 2000 with all current patches and utilize the NTFS file system to add some file and folder level security. Additionally, he may want to utilize the encryption file system to encrypt the file that includes his valuable information. Finally, I recommended that this individual may want to invest in some additional protection by purchasing a program that is specifically designed to store passwords. Some of these programs offer additional encryption protection. The bottom line is that those who utilize computers in a networked environment should be aware that security of their information is something

that needs to be looked at. Never leave your door wide open to intrusion of unwanted guests and it would be preferable to have multiple protection layers, including, but not limited to permission settings in a NTFS volume, encryption, virus protection, intrusion detection, and some form of perimeter defense.

Sources:

Authorization on the files level – NTFS. 22-1-2002.
http://www.atlguide2000.com/eng/win2k/ntfs1.htm

Cerelli, Bob. Windows 2000 Tips.
http://www.onecomputerguy.com/windows2000_tips.htm#shares

Chien, Eric. Dec. 13, 2001. Symantec Security Response - W32.Nimda.A@mm.
http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html

Figueroa, Michael. 2001. Windows 2000 Security Overview.
http://www.ams.com/Amscat/Downloads/Win2000Security.pdf

Finder, Kathy. Windows 2000: Permission Types: An Overview. Feb. 6, 2002.
http://www.uwec.edu/Help/Win00/permissn/types.htm

Ivory, Shaun. Whisper 32. 1988. http://www.ivory.org/whisper.html.

Microsoft Corp. Cannot Change Default Administrative Shares. Aug. 8, 2001.
http://support.microsoft.com/default.aspx?scid=kb;en-us;Q100517.

Microsoft Corp. HOW TO: Remove Administrative Shares in Windows 2000. March, 15, 2002. http://support.microsoft.com/default.aspx?scid=kb;en-us;Q318751.

Microsoft Technet. 2002.
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windows2000pro/reskit/part3/proch13.asp.

Morey, James. Microsoft Technet: The Permissions Maze. Oct. 15, 1999.
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/iis/deploy/config/permmaze.asp

Posey, Brien M. Working With The Windows 2000 File System Tools, Part 2.
http://www.brienposey.com/working_with_win2k_file_system_tools,_part_2.htm

Selke, Erik. Windows 2000: Setting Up Share Access. 1999-2000.
http://www.msu.edu/service/sns/windows/win2k_sharing.html

Sheppard, Simon. 2002. http://www.ss64.demon.co.uk/nt/net_share.html.

Walthall, John T. File System & Share Permissions. Oct. 29, 2001. http://www.research.umbc.edu/~jwalth1/Fallsix.ppt (April 15, 2002).

Windows 2000-NT Tip #310: Blocking Inheritance. 2002. http://windows.about.com/library/tips/bltip310.htm (April 15, 2002)