GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Securing Windows and PowerShell Automation (Security 505)"
at http://www.giac.org/registration/gcwn

An Examination of Auditing NTFS Permissions

Tom Crow
April 2001

This document was written to fulfill requirements for the practical assignment portion of the GIAC-NT certification. It represents only a portion of items to be addressed in terms of threats and vulnerabilities of a computing environment that utilizes Microsoft's Windows NT. This document should be used as a security baseline and is meant to serve only as a sample guide not a complete list of all best practices that should be followed in attempting to audit NTFS permissions for Windows NT.

The task of securing our desktop and servers grows more difficult every day. Ingenious hackers, new software with unexpected "features," and operating systems with exciting new "innovations" makes the job of systems administrators and security professionals a difficult one. I often think of the Chinese curse, "May you live in interesting times." These are interesting times indeed.

Almost every article and book that covers securing an NT system covers NTFS permissions. In most cases, that coverage is brief. Many mention a favorite tool, but provide no guidance on how to use it or how it might correct the problem. Many audit procedures appear to cater to auditing a brand new system where one is free to make changes. In the real world, administrators are often handed responsibility for systems built and secured by other administrators, sometimes with mission-critical applications.

This paper will attempt to cover auditing of NTFS permissions (Access Control Lists, or ACLs) using built-in, Resource Kit, and third party commercial tools. This paper will attempt to draw together best practices from a variety of sources. Since the average administrator doesn't have much time to devote to auditing permissions, I will mainly be covering command line utilities with an eye towards creating batch files to accomplish this aspect of system security.

Theory

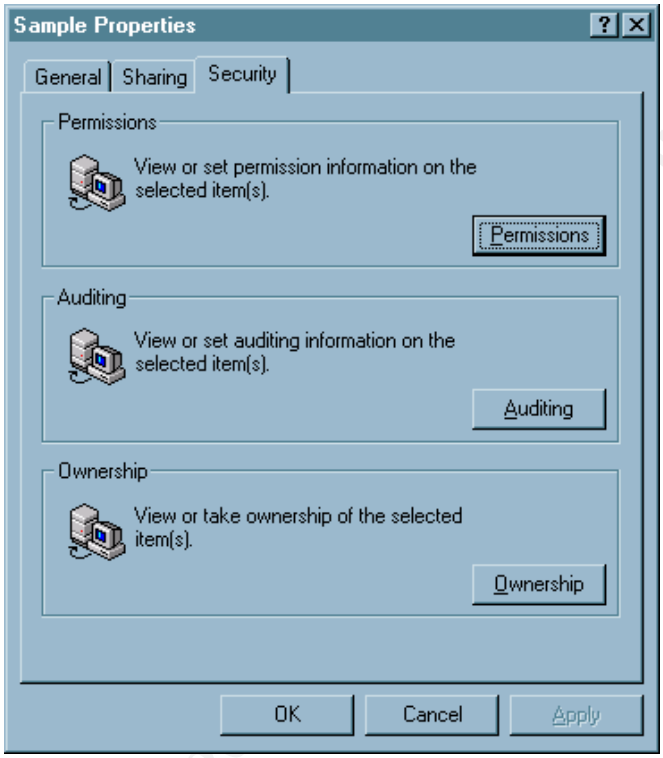
By definition, NTFS stands for New Technology File System. In his paper Choosing Between File Systems, Posey states, "For computers running only NT on the Intel architecture, there is no better choice in file system. NTFS supports larger volume sizes than FAT, better fault tolerance, native file compression, and enhanced security."[1] I would expand on that statement by noting that one aspect of the enhanced security is the ability to assign permissions to files and directories. Under a FAT file system, local users would have no restrictions as to what files they could create, delete, or modify. NTFS provides that additional

[1] Posey, Brien M. Choosing Between File Systems. 15 October, 1999. URL: http://www.microsoft.com/technet/winnt/filesyst.asp

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

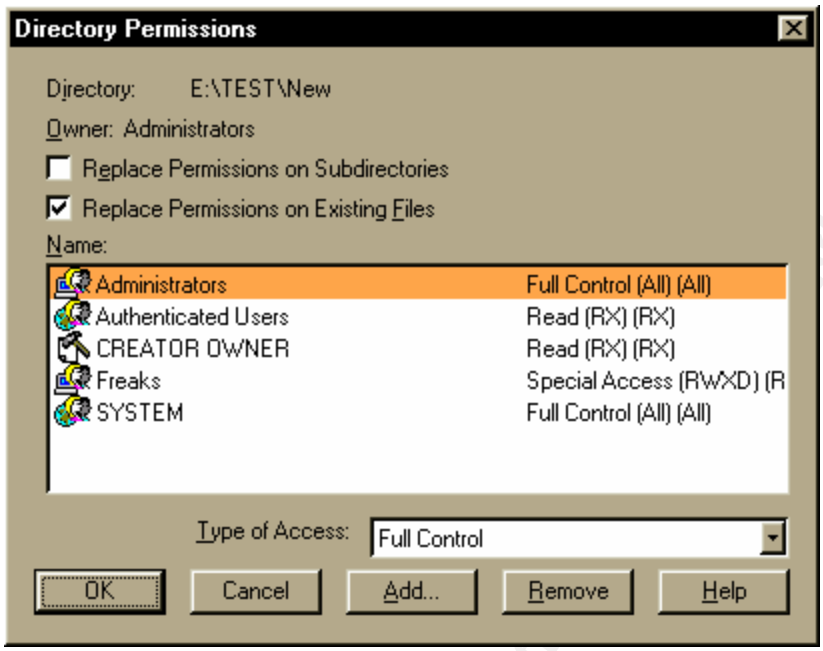
layer of protection, and works with share level protections to safeguard files and folders.

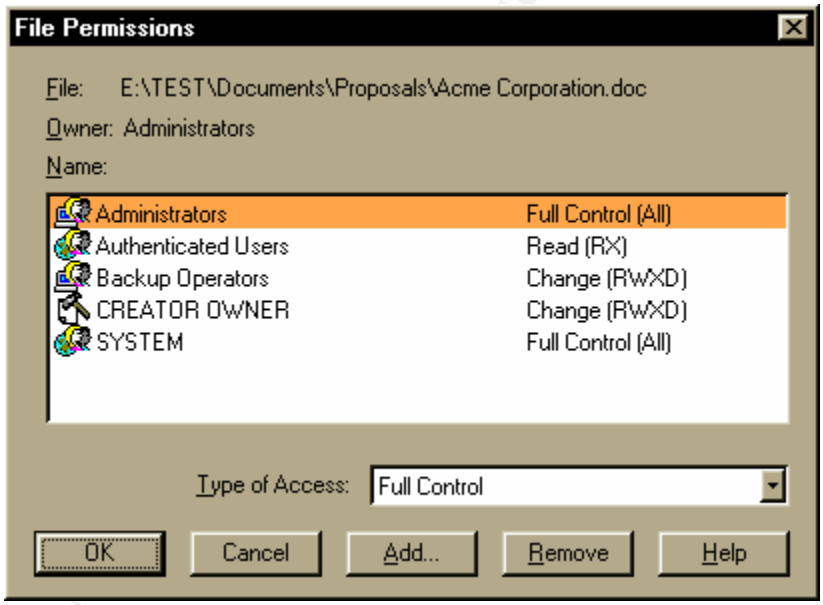
To access the NTFS permissions on a drive, file or folder; right click on its icon in the Windows NT Explorer. This will result in a window like the following:



Click on the Permissions button next. The resulting window will look like one of the following depending on whether it is a file or a directory:

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.



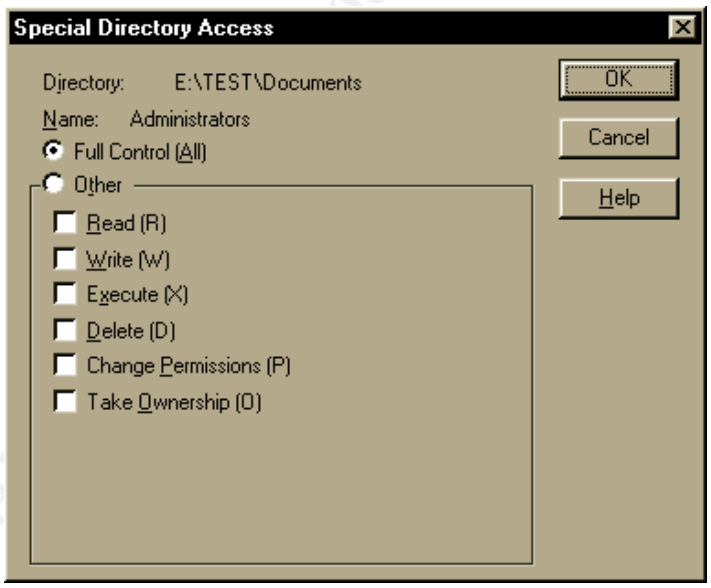


Under NTFS, a file or directory has what is known as an "Creator Owner." In the above examples the owner is the Administrators group. Usually, the owner of a file or directory is the user who created it. One exception to this rule is when the creator is a member of the Administrators group, in which case the ownership is set to the Administrators group.[2] To take ownership of a file or directory one should select the Security tab in the Properties window of the file or directory. The Ownership button is at the bottom of the resulting window. Selecting that button will result in a window like the following:

[2] Frisch, Æleen. Essential Windows NT System Administration. O'Reilly & Associates. 1998. P. 166.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Access to a file or directory is controlled by an Access Control List (ACL) which consists of Access Control Entries (ACE). An ACE can be created for a Local User, a Local Group, a Global User or a Global Group.[3] In the Directory Permissions, please note that there are two sets of braces after the Standard Permission. The first set of braces shows the permissions set on the directory itself; the second is the permissions set on new files and directories under that directory. If NTFS Standard Permissions are being used, both entries will be the same. If the permissions are different, then Special Permissions have been set. To set Special Permissions on either files or directories, select the ACE you wish to modify in the Permissions window. Next click on the down arrow on the "Type of Access" field. Towards the bottom there will be entries called "Special Directory Access" and "Special File Access" for directories and "Special Access" for files. Select one of these and a window like the following will result:



At this point, the user can select any combination of permissions for the directory.

Here is a convenient table listing what the code letters mean from Essential Windows NT System Administration:

[3] Frisch, Æleen. Essential Windows NT System Administration. O'Reilly & Associates. 1998. P. 166-167.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Permission (code letter)	Meaning for a file	Meaning for a directory
Read (R)	View or access a file's contents	List files within the directory.
Write (W)	Change or delete the file's contents	Add items to the directory.
Execute (X)	Run an executable image.	Change the current directory to that directory.
Delete (D)	Delete the file.	Delete the directory itself.
Change Permissions (P)	Change the permissions on the file.	Change the permissions on the directory itself.
Take Ownership (O)	Become the owner of the file.	Become the owner of the directory.

[4]

Each ACE can be granted a range of permissions as seen in the table below for folders from Essential Windows NT System Administration:

Permission Set	Equivalent Basic Permissions on Directory	Equivalent Basic Permissions for New Files
No Access	None	None
Special File Access	N/A	Any Desired Subset.
Special Directory Access	Any desired subset.	N/A
List	RX	Not Specified
Read	RX	RX
Add	WX	Not Specified
Add & Read	RWX	RX
Change	RWXD	RWXD
Full Control	RWXDPO (ALL)	RWXDPO (ALL)

[5]

Here are the NTFS Standard permissions for files from Essential Windows NT System Administration:

Permission Set	Equivalent Basic Permissions
No Access	None
Special Access	Any desired subset.
Read	RX
Change	RWXD
Full Control	RWXDPO (ALL)

[6]

[4] Frisch, Æleen. Essential Windows NT System Administration. O'Reilly & Associates. 1998. P. 167-169.
[5] Frisch, Æleen. Essential Windows NT System Administration. O'Reilly & Associates. 1998. P. 167-169.
[6] Frisch, Æleen. Essential Windows NT System Administration. O'Reilly & Associates. 1998. P. 167-169.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Given that users and groups can be assigned ACEs in a file or directory's ACL, there will be times when a user will belong to more than one group with access to that file or directory. NTFS handles this by applying the permissions that are least restrictive. However, if a user or group is given the "No Access" standard permission, that permission is upheld no matter what other permissions a user might have. Be very careful when assigning the "No Access" permission.

The Problem

At this point, one might be wondering: So what is the problem? The problem is that given this powerful system for controlling access to files and directories, Microsoft elected to make the default protections for those files and directories very weak. The Everyone group has full control of almost everything. This comes from personal observation on freshly installed systems. According to Fossen, "Assigning permissions to the Everyone group can be a risk because the Everyone group includes literally everyone."[7] He continues, "The Everyone group includes users from untrusted domains, users who have no Windows NT domain, anonymous Internet users, and null session users."[8] It also means that sensitive documents could be accessible to local users who should not have access. Users like temporary workers, summer interns, etc. might be able to access accounting or payroll data.

For example, I was issued a PC running Windows NT at work to use for personal productivity applications, which my employer has standardized on. I noticed that the Windows NT Explorer does not show who owns the file in any of the standard views of a directory or drive. Permissions are also nowhere to be seen. I opened a DOS command window and performed a listing there. Again, no indication of who owns the files or what the permissions might be was given. The average user would have no idea what protections are on their files. When I looked at the permissions via the Properties popup window, I was dismayed to find that the group Everyone had Full Control of the folders I store my documents in.

What makes this scenario even scarier is that in my organization, user logins are maintained by the domain controller. Any other user with a valid username and password in that domain and physical access to my PC can log in and access data. Thankfully, I am not responsible for sensitive information like contracts and financial data.

What files and directories need to be locked down tight? Who absolutely, positively needs to get access to key directories and files? What tools can help a systems administrator or security professional audit permissions on a Windows NT system? This paper will attempt to answer some of these questions.

[7] Fossen, Jason. Securing Windows NT, Step-by-Step, Part 4. Document version 1.0. 24 July, 2000.
[8] Fossen, Jason. Securing Windows NT, Step-by-Step, Part 4. Document version 1.0. 24 July, 2000.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Steps to More Secure NTFS Permissions

I have derived this process for securing permissions:

1. Establish a baseline for which users and groups need access to which files and folders and at what level of privilege.
2. Establish what the current NTFS Permissions are on those same files and folders.
3. Correct differences between baseline and current conditions.
4. Periodically re-evaluate the baseline.
5. Periodically re-evaluate the host.

Like most processes in information security, this process is a circular one.

Establishing a Baseline For NTFS Permissions

The best practice for NTFS Permissions is to follow the Principle of Least Privilege.[9] Although each organization should derive its own permissions policy, there are certain directories and files that should always be locked down. The following list comes from Security Focus' document Securing NT:

Directory	Local Group	Permissions
C:\ (or System Drive)	Administrators: CREATOR OWNER: Authenticated Users: SYSTEM:	Full Control Read Read Full Control
TEMP	Administrators: SYSTEM: CREATOR OWNER: Authenticated Users:	Full Control Full Control Full Control Full Control
Program Files	Administrators: Account Operators: Backup Operators: Server Operators: CREATOR OWNER: Authenticated Users: SYSTEM:	Full Control Special (All)(None) Special (All)(None) Special (All)(None) Read Read Full Control
Program Files\NTReskit (or alternative location)	Administrators: SYSTEM:	Full Control Full Control
WINNT	Administrators: CREATOR OWNER: Authenticated Users: SYSTEM:	Full Control Full Control Read Full Control

[10]

[9] Fossen, Jason. Securing Windows NT, Step-by-Step, Part 4.
[10] Security Focus. Securing NT.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The Ntreskit directory is sometimes installed as C:\ntreskit instead of C:\Program Files\Ntreskit. Within the WINNT tree, Securing NT recommends the following exceptions to the parent directory Access Control List:

Directory	Local Group	Permissions
WINNT\REPAIR	Administrators: SYSTEM:	Full Control Full Control
WINNT\SYSTEM32\CONFIG	Administrators: CREATOR OWNER: Authenticated Users: SYSTEM:	Full Control Full Control List Full Control
WINNT\SYSTEM32\SPOOL	Administrators: CREATOR OWNER: Authenticated Users: System Operators: SYSTEM:	Full Control Full Control Read Change Full Control
WINNT\COOKIES WINNT\FORMS WINNT\HISTORY WINNT\OCCACHE WINNT\PROFILES WINNT\SENDTO WINNT\Temporary Internet Files	Administrators: CREATOR OWNER: Authenticated Users: SYSTEM:	Full Control Full Control Add Full Control

[11]

Securing NT also suggests applying the following ACLs to specific files:

FILES	LOCAL GROUP	Permissions
Boot.ini Ntdetect.com Ntldr.	Administrators: SYSTEM:	Full Control Full Control
Autoexec.bat Config.sys	Administrators: SYSTEM: Authenticated Users:	Full Control Full Control Read
\winnt\poledit.exe \winnt\regedit.exe \winnt\system32\cacls.exe \winnt\system32\convert.exe \winnt\system32\dhcpadmin.exe \winnt\system32\eventvwr.exe \winnt\system32\inetins.exe \winnt\system32\musrmgr.exe \winnt\system32\ntbackup.exe \winnt\system32\rasadmin.exe	Administrators: SYSTEM:	Full Control Full Control

[11] Security Focus. Securing NT.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

\winnt\system32\rdisk.exe \winnt\system32\regedt32.exe \winnt\system32\rplmgr.exe \winnt\system32\srvmgr.exe \winnt\system32\syskey.exe \winnt\system32\sysedit.exe \winnt\system32\tftp.exe \winnt\system32\usrmgr.exe \winnt\system32\windisk.exe \winnt\system32\winmsd.exe \winnt\system32\winsadmin.exe		

[12]

I consider this list a very good start. The Windows NT Security FAQ goes even further to recommend locking down all ".exe" and ".dll" files.[13] Of course, each NT Server or Workstation would have additional files and folders that would require special attention. A good example would be a database with payroll information or users' home directories. The Windows NT Security FAQ recommends that, …"[i]f users are allowed to store files locally, make sure that they have full rights to their own directories."[14] User directories are commonly found under %system root%\profiles. Before deploying proposed ACL changes on critical servers, I highly recommend trying them out on test systems. I managed to mess up a few test installations of Windows NT during the course of my investigations. Another best practice is to only give permission to groups in ACLs, not users. Where possible, it is advisable to only use local groups, as opposed to global groups.[15] This makes sense since a local administrator has more control over the members of the local group. A common recommendation is to use the Authenticated Users group instead of the Everyone group.

For the remainder of the paper, the above list of files and folders will be used to compare the strengths and weaknesses of auditing NTFS Access Control Lists with various tools. Where possible, batch files will be constructed to automate the process of reporting on current ACLs, and methods will be shown for changing the current ACLs to an accepted standard.

Auditing NTFS Access Control Lists with Built-In Commands

Windows NT provides the CACLS.exe program, which provides command line functionality to display and change ACLs on files and directories. The window dump included below shows the on-line help for the CACLS command.

[12] Security Focus. Securing NT.
[13] Klaus, Christopher. Windows NT Security FAQ. 31 January, 1997. URL: http://www.cs.uu.nl/wais/html/na-dir/computer-security/ntsecurity.htm.
[14] Klaus, Christopher. Windows NT Security FAQ. 31 January, 1997. URL: http://www.cs.uu.nl/wais/html/na-dir/computer-security/ntsecurity.htm.
[15] Posey, Brien M. Making Effective Use of Permissions. 12 January, 2000. URL: http://www.microsoft.com/technet/winntas/Tips/techrep/permiss.asp.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Command Prompt
C:\>cacls.exe
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
                              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
                              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.

C:\>_
```

To view the ACL of a file or folder one would use following format:

```
C:\>cacls C:\
C:\ BUILTIN\Administrators:(OI)(CI)F
    Everyone:(OI)(CI)C
    CREATOR OWNER:(OI)(CI)F
    NT AUTHORITY\SYSTEM:(OI)(CI)F
```

In the above example the Administrators group, the Creator, and the SYSTEM user have Full Control; whereas, the Everyone group just has Change permission. CACLS will also accept wildcard entries like "*.*" or "*.exe."

To grant access for a user or group, use the /g option like the following:

```
E:\>cacls *.doc /g geeks:r
```

The /g option grants additional privileges. To replace the current set of permissions in an ACE, use the /p option. While the /d option will deny access to a user or group, it is generally safer to remove the ACE for a user or group by usubg the /r option like the following:

```
E:\>cacls important.doc /r everyone
```

However, I would like to note that in both of the above examples the ACL for the file in question is being replaced by the ACL specified by that command. In practice, this is not a very good idea. CACLS has a /e option which will edit the ACL and add, remove, or modify ACEs within the ACL. Here is an example:

```
E:\>cacls important.doc /e /g freaks:c administrators:f /p everyone:r
```

Note that if the Freaks group already has the permissions in the Change set, nothing would happen; whereas, the ACE for the Everyone group would be

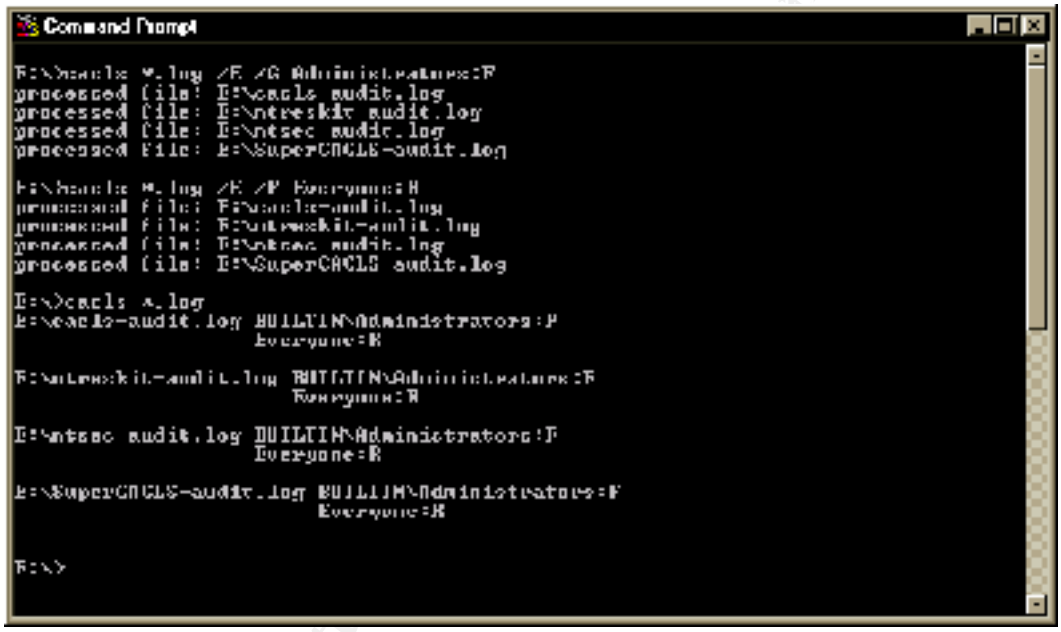
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

replaced by the Read set. Another useful option is the /t option, which will propogate the ACL changes down into the subdirectories and files of a specified directory. For example:

```
E:\>cacls Documents /e /t /p freaks:c  /r everyone
```

The above command would edit the ACL for the Documents directory, change the ACE for the Freaks group to Change, remove the ACE for the Everyone group, and propogate that change to all files and subdirectories of the Documents directory.

Below is a sample of the output of CACLS.EXE:



The first example, I am adding the Administrators group to the ACL with Full Control. I used the /E switch to edit the ACL, keeping pre-existing entries. In the second example I am changing the entry for the Everyone group to Read, again using the /e switch. The third example is obtaining a listing of the affected files. Notice that the Administrators group has Full Control and Everyone has Read. There are several other options available for this command.

Using CACLS.EXE in combination with the list of files and folders in the preceding section in a batch file, I was able to quickly produce a report on the ACLs on critical files. When used to display ACLs CACLS.EXE does not have the capability to recurse down through a directory tree. The "/T" option only applies when using CACLS to set permissions. Therefore, I was forced to call CACLS for each file or directory on the Securing NT list. The tool also does not have the capability to save a snapshot of the current permissions for later comparisons. I used output redirection to send output to a file for later viewing.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

I also found that CACLS does not support setting an ACE with special permissions, where folder and file permissions are different. This can be done through Windows Explorer under Properties→Security→Permissions. Another anomaly I observed is that the following command does not actually change the permissions of subdirectories and files:

```
C:\>cacls e:\ /t /e /g Administrators:R
```

However this one will:

```
C:\>cacls e:\* /t /e /g Administrators:R
```

This behavior only seems to happen at the drive level of the file system.

There appears to be no built-in command to take or assign ownership of files. As shown earlier, the Security tab under the Properties window provides a button for the current user to take ownership of a file or folder.

I have provided a sample of a batch file to assess current ACLs on files and folders from the Securing NT list as Appendix A.

Auditing NTFS Access Control Lists With the NT Resource Kit

The Windows NT Resource Kit is a free download from Microsoft's web site, or it can be purchased on a CD with a large reference manual in most book and computer stores. The following table is a list of Resource Kit tools useful in auditing permissions on a Windows NT computer:

PERMS.EXE	Displays a user's permissions to specified files and directories.
SHOWACLS.EXE	Displays the access control list associated with a file or directory.

The following two screen shots show the on-line help for the two commands.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Command Prompt
E:\>perms /?

Displays a user's permissions to specified files and directories.

PERMS  [domain\]computer\]username path [/i] [/s]

 [domain\]computer\]username Name of user whose permissions are to checked.
 path                        A file or directory, wildcards (*,?) accepted.
 /i                          Assumes the specified user is logged on interactively
                             to computer where the file/directory resides.
                             With this switch, PERMS assumes the user is a member
                             of the INTERACTIVE group. Without this switch, PERMS
                             assumes the user is a member of the NETWORK group.
 /s                          Check permissions on files in subdirectories.

The output access mask contains the following letters:

R Read
W Write
X Execute
D Delete
P Change Permissions
O Take Ownership
F General All
- No Access
* The specified user is the owner of the file or directory.
# A group the user is a member of owns of the file or directory.
? The user's access permissions can not be determined.

E:\>
```

```
Command Prompt

E:\>showacls /?

Usage:

showacls /s /u:domain\user filespec

/s  include sub-directories
/u  specify domain\user

ACE header values:

0x1 - Object Inherit ACE
0x2 - Container Inherit ACE
0x4 - No Propagate Inherit ACE
0x8 - Inherit Only ACE

Access mask values:

A  Generic All        l  List Directory
R  Generic Read       d  Read Data
W  Generic Write      S  Synchronize
X  Generic Execute    r  File Read
w  File Write         a  File Append
fx File Execute       D  Delete
rb Read EA            rW Write EA

E:\>_
```

To view the ACE of a particular user or group on a file or folder one would use PERMS like the following:

```
E:\> perms freaks file
```

With the /s option, PERMS can recurse down into subdirectories of a specified directory. PERMS can also accept wild cards for the file to be evaluated.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

To view the ACL of a file or folder, one would use SHOWACLS like the following:

```
E:\>showacls e:\documents
```

Like PERMS, SHOWACLS has a /s option to recurse through subdirectories, provided that the command is evaluating a directory. SHOWACLS can also limit it's output to displaying the ACE of a single user by using the /u option and specifying a user. SHOWACLS does not evaluate wild cards.

Here are some examples of the two commands in action:

Please note the difference in the output. The output of PERMS is fairly straight-forward. The output for SHOWACLS is not as intuitive, but it can be interpreted with the on-line help (SHOWACLS /?). I could see PERMS being very useful in searching for what permissions a given user or group might have within a directory structure. Here is a very simple batch file that would search for objects that the Everyone group has access to:

```
Rem
Rem  Batch file to check for files the Everyone group has
Rem  Access to.
Perms /s Everyone e:\ >> c:\temp\perms.out
Rem  End of batch file
```

If the drive in question is large, PERMS will need a large amount of memory to process the request. Given that a freshly-installed Windows NT system will have Everyone in full control of everything, I would not advise using the above script until after some locking down has been accomplished.

The Resource kit does not provide a utility to change an ACL or to take or assign ownership of files or directories. Luckily, CACLS can fill this role nicely.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Auditing NTFS Access Control List With SuperCACLS

SuperCACLS is a commercial suite of tools for managing NTFS ACLs, Registry ACLs, and Audit SACLS written by Trusted Systems Services, Ltd. This suite takes the functionality of the CACLS program breaks it into three pieces, extends the functionality of the resulting pieces, and adds a much needed command line utility to take ownership of files and folders. Trusted Systems provides a demo version for evaluation purposes. The following table lists the SuperCACLS utilities and describes their functions:

PRACL.EXE	Print ACLs in a form familiar to Windows NT/2000 users. The /bat option allows the user to take "snapshots" that can be later run to restore a baseline configuration.
REACL.EXE	Sets the ACLs on objects, overwriting the previous ACL. This command can set or take ownership of an object.
MODACL.EXE	Changes parts of an ACL, leaving the rest unchanged.
TAKEOWN.EXE	Take ownership of an object.

SuperCALCS commands use many of the same options.

/f	Do not process directories
/d	Do not process files
/h *host*	Remote host
/t	Target objects
/s	Recurse through subdirectories
/?	On-line help
/L	Look before you leap

PRACL might be used like the following:

```
E:\>pracl /t C:\
*** Super CACLS Demonstration Version *** (use "PRACL /about" for info)

C:\ ----------------------------------------------------------------
    Administrators:  full (owner)
    CREATOR OWNER:   full
    Everyone:        change
    SYSTEM:          full
```

Targets are specified by the /t option. Wild cards can also be used in the target definition. Another useful option is the /s option which will recurse through subdirectories of the specified directory. Using PRACL with the /bat option produces a batch file with a series of REACL commands for each object in the snapshot. This batch file could be put on non-volatile media like a CD and used to baseline permissions on systems. This could come in handy to secure new systems or to bring old installations back to an established standard. The "/bat" option will overwrite existing files—no option exists to append to an existing file.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

If the following command were executed:

```
E:\>pracl /t C:\temp /bat C:\temp.bat
```

C:\temp.bat would contain:

```
Reacl %1 %2 %3 %4 /v40 /t "C:\TEMP" /a "Administrators:full"
"Everybody:change" "CREATOR OWNER:full" "SYSTEM:full"
```

Using PRACL with the "/s" and "/bat" options would create a batch file full of entries like the above.

REACL commands take a similar form. As seen below, the "/a" option is used to define the ACEs for the ACL.

```
C:\>reacl /t e:\Documents /a Quasi:change Administrators:full
Everyone:-full
```

In the above example, the user Quasi has been given Change, the Administrators group has been given Full Control of the Documents directory. The "-full" permission given to the Everyone group denies members of this group any access to this Directory. REACL can also assign permissions like this:

```
C:\>reacl /t e:\Documents /a Tom:RWXD
```

The "RWXD" permissions effectively give the Tom user Change rights on the Documents directory. REACL can set special permissions like the following with the "/" separating the different permissions:

```
C:\>reacl /t e:\Documents /a Bill:RWX-PO/RX/Full
```

The above command would give the user Bill Read, Write, and Execute permissions on Documents, but deny Bill from changing permissions or taking ownership of the file. For files under the Documents directory Bill will have Read and Execute permissions. Full control would be granted to Bill for any subdirectories under Documents. It is important to re-emphasize that REACL replaces the current ACL for a folder or directory with the ACL provided as an argument. To add a new ACE or to change the permissions on an existing ACE, one must use the MODACL command.

MODACL commands follow the same syntax as REACL commands with the "/a" option replaced by the following:

Option	Description
/an	Add ACE if not already present
/ap	Add permissions to an ACE
/ar	Add or replace ACE
/de	Delete an ACE
/re	Replace an ACE if present

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

/rn	Rename an ACE's trustee
/rp	Remove permissions from an ACE

These options provide greater flexibility in how ACEs can be set. In practice, REACL is better suited to enforcing a standard set of ACLs; while MODACL is more useful in tweaking ACLs for specific situations. Both MODACL and REACL can use the "/L" option, which prints out the changes that will take place without actually making the changes. I would recommend making frequent use of that option.

TAKEOWN provides a command line method of taking ownership of files and directories. While the capability to take ownership of files and folders from the command line can be useful, the ability to assign ownership to another user would be more useful.

I have provided a sample batch file to assess current ACLs on files and folder from the Securing NT list as Appendix B.

Auditing NTFS Access Control Lists With NTSEC

NTSEC is a suite of security tools from Pedestal Software. The following is a list of tools useful in auditing NTFS ACLs:

SAVEACL.EXE	Saves file, directory, and ownership permissions to a file.
RESTACL.EXE	Restores file permissions and ownership from a saveacl file.
LISTACL.EXE	Lists file permissions and auditing settings in human readable format.
IGRANT.EXE	Grant a user or group access permissions to a folder or file
IREVOKE.EXE	Revoke a user or group
SETOWNER.EXE	Set the ownership of a file or folder to a user or group.

The combination of SAVEACL and RESTACL offer something similar to the "/bat" option in the RECACL utility in SuperCACLS. Output format can be the default SAVEACL format, tab-delimited text, or unicode text. The following table describes some of SAVEACL's options:

-r	Recurse into subdirectories
-dirsonly	Store ACLs for directories only
-filesonly	Store ACLs for files only, skipping directories
-x	Exclude these files
-text	Save in textual, comma-separated format

The next table does the same for RESTACL:

-listonly	Just list the affected files. Do not apply

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

	permissions
-listdetail	Same as –listonly, but dump the ACLs too
-owneronly	Restore file ownership only
-noowner	Restore all ACLs but not ownership
-merge	Merge permissions instead of replacing them
-x	Exclude these files

Use of either command with the "-h" option will yield a full listing of all options.

```
C:\>saveacl -r e:\test e:\test.acl

C:\>restacl e:\test.acl
```

In the above example, SAVEACL is used to take a snapshot of the e:\test directory and everything below it. On the next line, RESTACL will restore the original settings to e:\test and everything below it. This process could be used as a safety net when experimenting with new permission schemes, or as a method to save and deploy a baseline of permissions.

LISTACL acts very much like SAVEACL with the exception that LISTACL is designed to report back in human-readable text like the following:

```
E:\> listacl C:\

C:\
      Owner: Administrators (lg)
      Administrators (lg)                (All)(All)*
      CREATOR OWNER                      (All)(All)*
      Everyone                           (RWXD)(RWXD)*
      SYSTEM                             (All)(All)*

1 file found, 0 errors.
```

The output can be customized with the following options:

-r	Recurse into subdirectories
-owneronly	List only file ownership
-bare	List only the file name
-dirsonly	List only the directories
-where *expression*	List only files matching 'expression'
-x	Exclude these files from the list (wild cards permitted)

SETOWNER is a tool that will allow files or folders to be assigned to another user or group. The syntax is very flexible, allowing for recursion, exclusion, and selection based on pattern-matching. Here is an example:

```
C:\>setowner -r -where filename()='*.bat' Administrators e:\
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

The above example will set all of the files ending with ".bat" on the e: drive to be owned by the Administrators group. As with the previous NTSEC utilities, the "-r" option indicates recursion. Given that the owner of a file can always change the permissions on that file, it is a very good idea to make sure that executables and libraries (*.dll) are owned by the proper user or group.

IGRANT and IREVOKE are related tools to, respectively, grant and revoke privileges to files and directories. The following table lists some of the important options available for these utilities:

-r	Recurse into subdirectories
-replace	IGRANT: Replace user's permissions with ones specified.
-clear	Clear ALL permissions first (this could be dangerous)
-dirsonly	Only affect directories (default)
-filesonly	Only affect files
-f	Affect files and directories
-where *expression*	
-ru	Remove 'account unknown' access control entries
-rd	Remove deleted and invalid entries
-x	Exclude files or directories

Here are some examples of how to use the utilities:

```
C:\igrant -r -dirsonly Administrators:full c:\temp
```

In this example, the Administrators group is being granted full access to the c:\temp directory and every directory under it. If the Administrators group already has those permissions, IGRANT will not change anything. The next example shows IREVOKE in action:

```
C:\irevoke -r -f "Authenticated Users":P e:\Important
```

In this example, the "-f" option is being used. The end result of this command is to remove the ability of the Authenticated Users group to change the permissions on the e:\Important directory and the files and directories under it. In reality, the IGRANT utility can perform some of the tasks of the IREVOKE command by using the "-replace" option.

Auditing NTFS Access Control Lists With DumpSEC

DumpSec by SomarSoft offers the best of both worlds for the viewing of ACLs, a graphical user interface and a command line interface suitable for batch files. Unlike some of the other reporting tools covered in this paper, DumpSEC dumps its output to a file instead of to the screen. The utility's primary deficiency is that it is only a reporting mechanism. A second tool set is required to make any necessary changes to the ACLs.

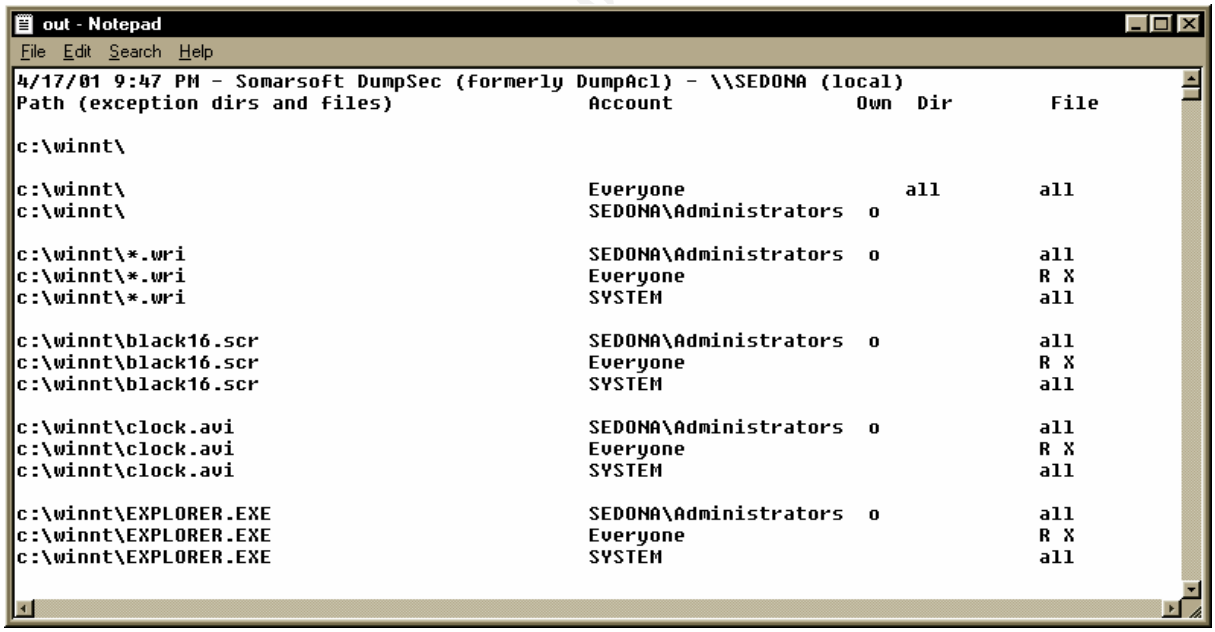
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Here are some of the options available for ACL Reports:

/rpt=dir=path	Directory permissions report (manditory)
/outfile=path	File in which to store report (manditory)
/saveas=format	Format to save report: Native – binary format which can be reloaded into DumpSEC (default) CSV – comma separated columns TSV – tab separated columns Fixed – fixed width columns, padded with blanks
/noowner	Do not dump owner. Default is to dump owner.
/noperms	Do not dump permissions. Default is to dump permissions.
/showexceptions	Show directories and files whose permissions differ from those of the parent directories. This is the default.

Below is a portion of the output for the command line:

DumpSec /rpt=dir=c:\winnt /saveas=fixed /outfile=out.txt



```
4/17/01 9:47 PM - Somarsoft DumpSec (formerly DumpAcl) - \\SEDONA (local)
Path (exception dirs and files)                Account                Own  Dir        File

c:\winnt\

c:\winnt\                                      Everyone                   all        all
c:\winnt\                                      SEDONA\Administrators  o

c:\winnt\*.wri                                 SEDONA\Administrators  o              all
c:\winnt\*.wri                                 Everyone                              R X
c:\winnt\*.wri                                 SYSTEM                                all

c:\winnt\black16.scr                           SEDONA\Administrators  o              all
c:\winnt\black16.scr                           Everyone                              R X
c:\winnt\black16.scr                           SYSTEM                                all

c:\winnt\clock.avi                             SEDONA\Administrators  o              all
c:\winnt\clock.avi                             Everyone                              R X
c:\winnt\clock.avi                             SYSTEM                                all

c:\winnt\EXPLORER.EXE                          SEDONA\Administrators  o              all
c:\winnt\EXPLORER.EXE                          Everyone                              R X
c:\winnt\EXPLORER.EXE                          SYSTEM                                all
```

The /rpt option is mandatory for the command line version and specifies what type of report is to be performed. This example is a report on a directory; however, DumpSEC can also report on the registry, users, shares, policy, rights, groups, printers, and services. DumpSEC normally writes to a file in its own particular format, but the /saveas option overrode the default to write in a text file with fixed columns. Comma or tab-separated text output could be imported into a spreadsheet application or perhaps a database for further manipulation.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

This example uses DumpSEC's default setting to only report on files and folders whose permissions are different than those of their parent. DumpSEC can also show all files and directories, show all directories (no files).

DumpSEC cannot append its output to a file. Therefore, a batch file of DumpSEC calls would have to write to a different file for each call.

Conclusions

In performing my research for this subject, I read through several GCNT practicals that were submitted by previous students. I was particularly looking for suggestions of tools to audit NTFS permissions. Of the five I use as references, two don't even cover NTFS permissions, one uses the Windows NT Explorer to change permissions, one suggested DumpSEC, and the fifth suggested a tool called FIXACLS, which I could not find. I believe that setting proper NTFS permissions is more important than that.

I was genuinely surprised by the capabilities of CACLS. For a tool not often mentioned for handling NTFS permissions, it was quite effective. Given that CACLS should be available on any NT system, I recommend that administrators of Windows NT systems become very familiar with it.

SuperCALCS and NTSEC are very similar in capabilities; however, NTSEC does provide the SETOWNER utility, which is unique in its abilities. Both sets of tools provide a means to establish a baseline template that can be easily used over and over again.

NTFS offers the security-minded Administrator great flexibility in determining what users can access on an NT computer. However, Posey offers some words of caution, "…with so many ways of controlling access, it's easy to make a mess of things and end up with overlapping permissions, which usually result in unexpected side effects."[16] It is very possible to mess up access to files and folders using the tools reviewed in this paper. There is no substitute for testing.

There is another reason for Administrators to be familiar with NTFS permissions and tools to assess and modify them. Tools that available to Administrators are also available to hackers and malicious insiders.[17]

[16] Posey, Brien. Making Effective Use of Permissions. 12 January, 2000. URL: http://www.microsoft.com/technet/Tips/techrep/permiss.asp.
[17] Fossen, Jason. Securing Windows NT, Step-by-Step, Part 4. page

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

References

Security Focus. Securing NT. 6 December, 1999. URL: http://www.securityfocus.com/focus/microsoft/nt/ntsecure.html.

Fossen, Jason. Securing Windows NT, Step-by-Step, Part 4. Document version 1.0. The SANS Institute. 24 July, 2000.

Frisch, Æleen. Essential Windows NT System Administration. Sebastopol, CA: O'Reilly & Associates. 1998. 166-184.

Klaus, Christopher. Windows NT Security FAQ. 31 January, 1997. URL: http://www.cs.uu.nl/wais/html/na-dir/computer-security/ntsecurity.htm.

Evers, Dan. NTFS Security, Part 1: Implementing NTFS Standard Permissions On Your Web Site. 15 November, 1999. URL: http://www.microsoft.com/technet/iis/ntfssec.asp.

Posey, Brien M. Choosing Between File Systems. 15 October, 1999. URL: http://www.microsoft.com/technet/winnt/filesyst.asp.

Posey, Brien M. Making Effective Use of Permissions. 12 January, 2000. URL: http://www.microsoft.com/technet/Tips/techrep/permiss.asp.

Payne, Jeff. GCNT Practical Assignment. The SANS Institute.

Horn, Michael. GCNT Practical Assignment. The SANS Institute.

Hutchinson, George. GCNT Practical Assignment. The SANS Institute.

Otis, Brig. GCNT Practical Assignment. The SANS Institute.

Bollig, Jerry. GCNT Practical Assignment. The SANS Institute.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Appendix A
Batch file to check permissions Using CACLS

```
REM Batch file to check permissions based on list suggested by
REM Security Focus
cacls C:\
cacls C:\TEMP
cacls "C:\Program Files"
REM  The following path may be different on your machine
cacls C:\NTRESKIT
cacls C:\WINNT
cacls C:\WINNT\REPAIR
cacls C:\WINNT\SYSTEM32\CONFIG
cacls C:\WINNT\SYSTEM32\SPOOL
cacls C:\WINNT\COOKIES
cacls C:\WINNT\FORMS
cacls C:\WINNT\HISTORY
cacls C:\WINNT\OCCACHE
cacls C:\WINNT\PROFILES
cacls C:\WINNT\SENDTO
cacls "C:\WINNT\Temporary Internet Files"

cacls C:\boot.ini
cacls C:\NTRESKIT\ntdetect.com
REM This may not exist on your machine. Test for it anyway.
cacls C:\ntldr

cacls C:\autoexec.bat
cacls C:\config.sys

cacls C:\WINNT\POLEDIT.EXE
cacls C:\WINNT\REGEDIT.EXE
cacls C:\WINNT\SYSTEM32\CACLS.EXE
cacls C:\WINNT\SYSTEM32\CONVERT.EXE
cacls C:\WINNT\SYSTEM32\DHCPADMIN.EXE
cacls C:\WINNT\SYSTEM32\EVENTVWR.EXE
cacls C:\WINNT\SYSTEM32\INETINS.EXE
cacls C:\WINNT\SYSTEM32\MUSRMGR.EXE
cacls C:\WINNT\SYSTEM32\NTBACKUP.EXE
cacls C:\WINNT\SYSTEM32\RASADMIN.EXE
cacls C:\WINNT\SYSTEM32\RDISK.EXE
cacls C:\WINNT\SYSTEM32\REGEDT32.EXE
cacls C:\WINNT\SYSTEM32\RPLMGR.EXE
cacls C:\WINNT\SYSTEM32\SRVMGR.EXE
cacls C:\WINNT\SYSTEM32\SYSKEY.EXE
cacls C:\WINNT\SYSTEM32\SYSEDIT.EXE
cacls C:\WINNT\SYSTEM32\TFTP.EXE
cacls C:\WINNT\SYSTEM32\USRMGR.EXE
cacls C:\WINNT\SYSTEM32\WINDISK.EXE
cacls C:\WINNT\SYSTEM32\WINMSD.EXE
cacls C:\WINNT\SYSTEM32\WINSADMIN.EXE
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Appendix B
Batch file to check permissions Using SuperCALCS

```
REM  This batch file will report back on NTFS permissions
REM  in key directories and on key files
pracl /t C:\
pracl /t C:\TEMP
pracl /t "C:\Program Files"
pracl /t C:\NTRESKIT
pracl /t C:\WINNT
pracl /t C:\WINNT\REPAIR
pracl /t C:\WINNT\SYSTEM32\CONFIG
pracl /t C:\WINNT\SYSTEM32\SPOOL
pracl /t C:\WINNT\COOKIES
pracl /t C:\WINNT\FORMS
pracl /t C:\WINNT\HISTORY
pracl /t C:\WINNT\OCCACHE
pracl /t C:\WINNT\PROFILES
pracl /t C:\WINNT\SENDTO
pracl /t "C:\WINNT\Temporary Internet Files"

pracl /t C:\boot.ini
pracl /t C:\NTRESKIT\ntdetect.com
pracl /t C:\ntldr

pracl /t C:\autoexec.bat
pracl /t C:\config.sys

pracl /t C:\WINNT\POLEDIT.EXE
pracl /t C:\WINNT\REGEDIT.EXE
pracl /t C:\WINNT\SYSTEM32\CACLS.EXE
pracl /t C:\WINNT\SYSTEM32\CONVERT.EXE
pracl /t C:\WINNT\SYSTEM32\DHCPADMIN.EXE
pracl /t C:\WINNT\SYSTEM32\EVENTVWR.EXE
pracl /t C:\WINNT\SYSTEM32\INETINS.EXE
pracl /t C:\WINNT\SYSTEM32\MUSRMGR.EXE
pracl /t C:\WINNT\SYSTEM32\NTBACKUP.EXE
pracl /t C:\WINNT\SYSTEM32\RASADMIN.EXE
pracl /t C:\WINNT\SYSTEM32\RDISK.EXE
pracl /t C:\WINNT\SYSTEM32\REGEDT32.EXE
pracl /t C:\WINNT\SYSTEM32\RPLMGR.EXE
pracl /t C:\WINNT\SYSTEM32\SRVMGR.EXE
pracl /t C:\WINNT\SYSTEM32\SYSKEY.EXE
pracl /t C:\WINNT\SYSTEM32\SYSEDIT.EXE
pracl /t C:\WINNT\SYSTEM32\TFTP.EXE
pracl /t C:\WINNT\SYSTEM32\USRMGR.EXE
pracl /t C:\WINNT\SYSTEM32\WINDISK.EXE
pracl /t C:\WINNT\SYSTEM32\WINMSD.EXE
pracl /t C:\WINNT\SYSTEM32\WINSADMIN.EXE
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Appendix C
Batch file to check permissions Using NTSEC

```
REM  Batch file to audit NTFS permissions using NTSEC tools.
REM  Using full path to executable.
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\TEMP
"C:\Program Files\Pedestal Software\NTSEC\listacl" "C:\Program Files"
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\NTRESKIT
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\REPAIR
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\CONFIG
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\SPOOL
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\COOKIES
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\FORMS
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\HISTORY
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\OCCACHE
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\PROFILES
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\SENDTO
"C:\Program Files\Pedestal Software\NTSEC\listacl" "C:\WINNT\Temporary
Internet Files"

"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\boot.ini
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\NTRESKIT\ntdetect.com
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\ntldr

"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\autoexec.bat
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\config.sys

"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\POLEDIT.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl" C:\WINNT\REGEDIT.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\CACLS.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\CONVERT.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\DHCPADMIN.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\EVENTVWR.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\INETINS.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\MUSRMGR.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\NTBACKUP.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\RASADMIN.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\RDISK.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\REGEDT32.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\RPLMGR.EXE
```


© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\SRVMGR.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\SYSKEY.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\SYSEDIT.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\TFTP.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\USRMGR.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\WINDISK.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\WINMSD.EXE
"C:\Program Files\Pedestal Software\NTSEC\listacl"
C:\WINNT\SYSTEM32\WINSADMIN.EXE
```

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Last Updated: April 28th, 2017

Upcoming Training

Click Here to
{Get CERTIFIED!}

SANS Security West 2017	**San Diego, CA**	**May 09, 2017 - May 18, 2017**	**Live Event**
SANS Northern Virginia - Reston 2017	**Reston, VA**	**May 21, 2017 - May 26, 2017**	**Live Event**
SANS Cyber Defence Canberra 2017	**Canberra, Australia**	**Jun 26, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
San Francisco Winter 2017 - SEC505: Securing Windows and PowerShell Automation	**San Francisco, CA**	**Nov 27, 2017 - Dec 02, 2017**	**vLive**
SANS San Francisco Winter 2017	**San Francisco, CA**	**Nov 27, 2017 - Dec 02, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**