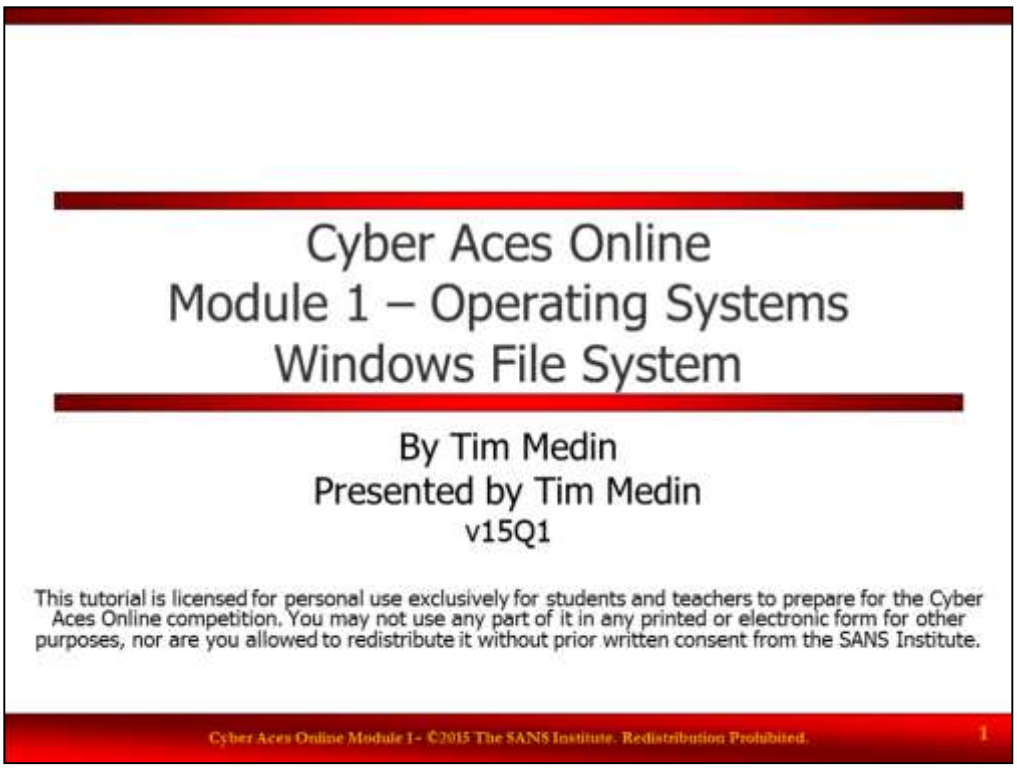
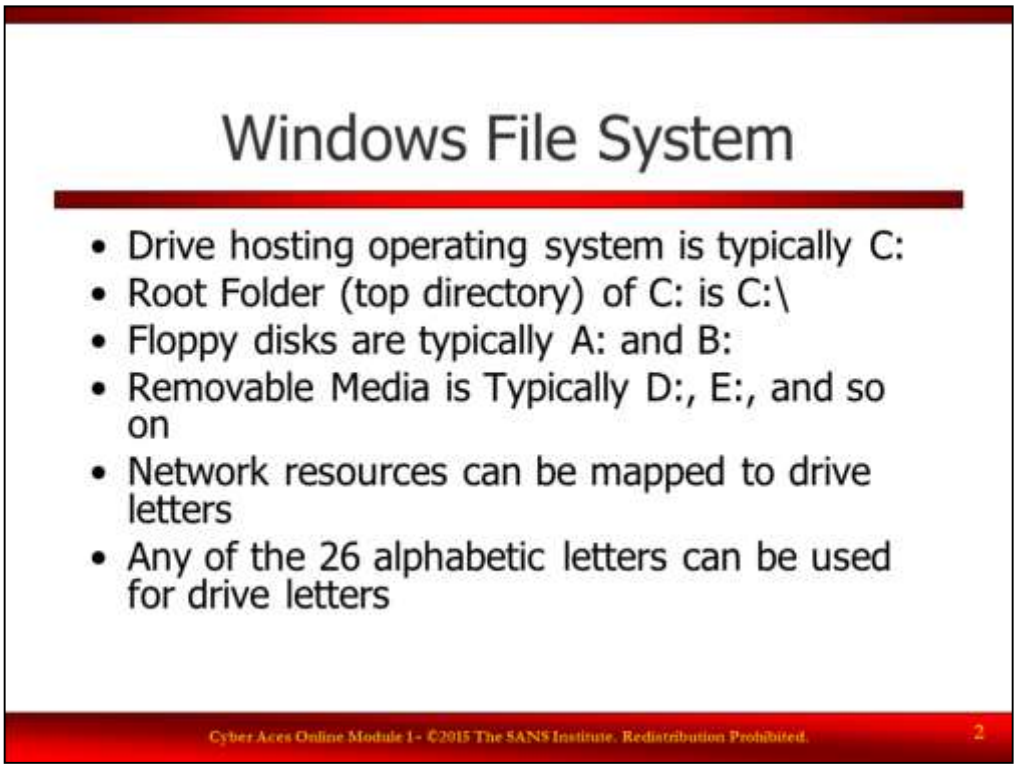
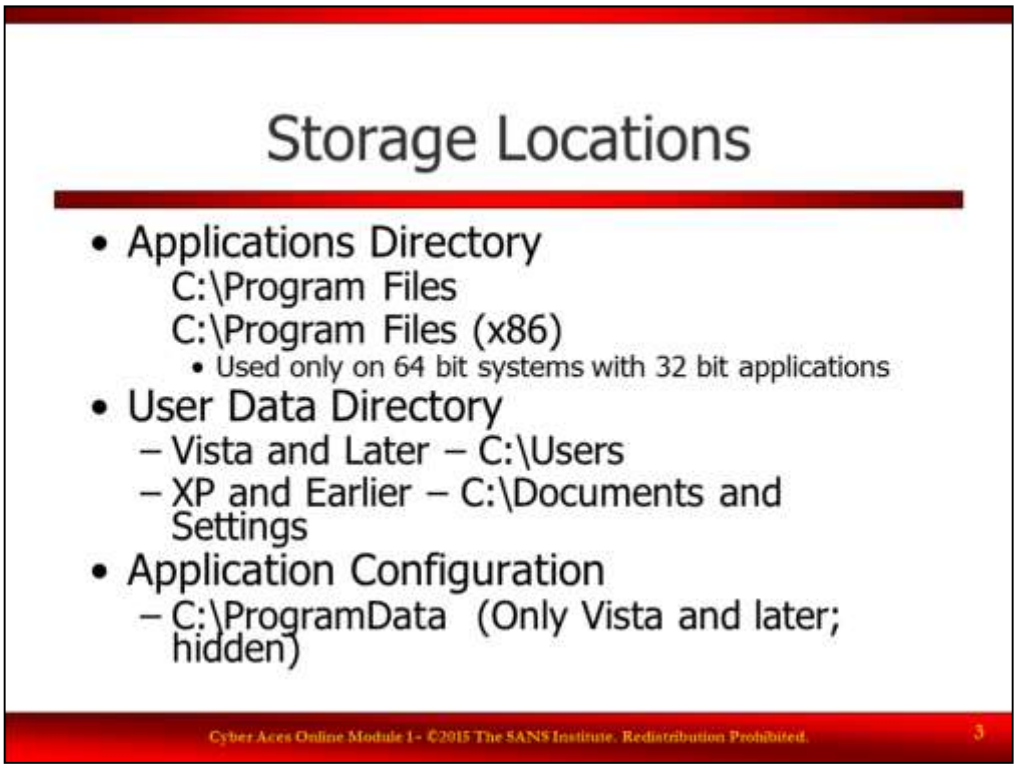


Welcome to Cyber Aces Online, Module 1! A firm understanding of operating systems is essential to being able to secure or attack one. This module dives in to the file system used by the Microsoft Windows Operating System.



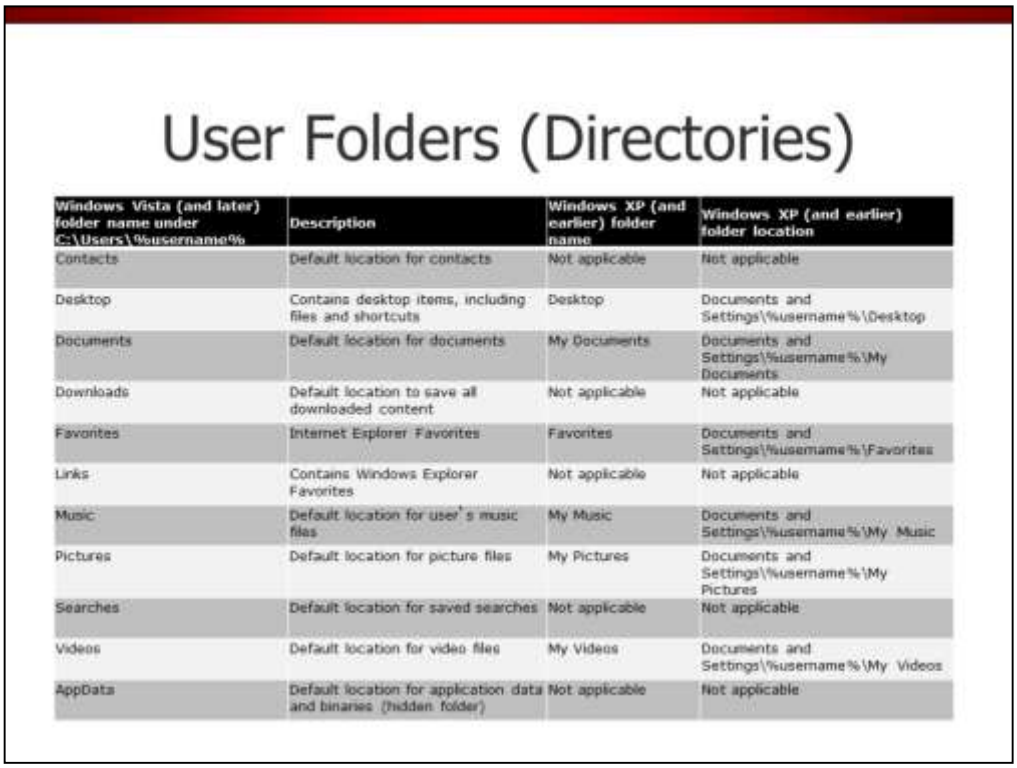
Windows File System

In Windows, the root folder (highest directory) is "C:\" by default on the majority of Windows machines. Different logical drives, whether they are physically individual disks or partitions within a single disk, are represented in Windows as letters followed by a colon. For example, in a standard Windows setup, the hard drive hosting the operating system is "C:", the floppy drive is "A:" (or "B:" if there are two drives), and additional drives (floppy/hard disk/CD-ROM/etc.) are "D:", "E:", "F:", and so on. Additionally, remote network resources are also sometimes mapped to drive letters. Any of the 26 letters of the alphabet can be mapped to hardware and network file resources.



Storage Locations

Microsoft has released a whitepaper on the directory structure of Windows Vista. Although Windows 7 has introduced some new concepts such as "Libraries", the underlying directory naming standard has remained unchanged on Windows 7, 8, and 8.1. An attacker will be very familiar with the directory structure and know where interesting files such as a user's browser cache, NTUSER.DAT (registry database), and the SAM database containing passwords are kept. For example, applications are stored in the "C:\Program Files" directory while user data is stored in "C:\Users". Application Configuration files are often stored under the "C:\ProgramData" directory.



User Folders (Directories)

Windows Vista (and later) folder name under C:\Users\%username%	Description	Windows XP (and earlier) folder name	Windows XP (and earlier) folder location
Contacts	Default location for contacts	Not applicable	Not applicable
Desktop	Contains desktop items, including files and shortcuts	Desktop	Documents and Settings\%username%\Desktop
Documents	Default location for documents	My Documents	Documents and Settings\%username%\My Documents
Downloads	Default location to save all downloaded content	Not applicable	Not applicable
Favorites	Internet Explorer Favorites	Favorites	Documents and Settings\%username%\Favorites
Links	Contains Windows Explorer Favorites	Not applicable	Not applicable
Music	Default location for user's music files	My Music	Documents and Settings\%username%\My Music
Pictures	Default location for picture files	My Pictures	Documents and Settings\%username%\My Pictures
Searches	Default location for saved searches	Not applicable	Not applicable
Videos	Default location for video files	My Videos	Documents and Settings\%username%\My Videos
AppData	Default location for application data and binaries (hidden folder)	Not applicable	Not applicable

User Folders (Directories)

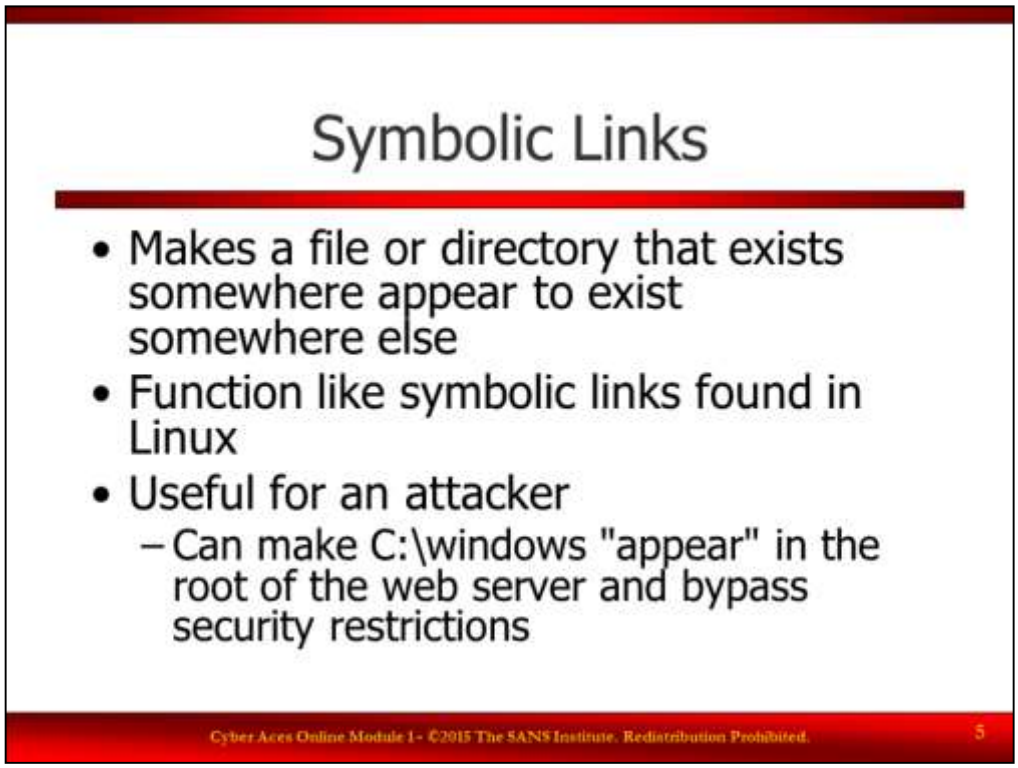
In Vista and later, the user folder structure has changed from that of the older versions. The above table shows the folders, and if applicable, the location of the same folder in Windows XP and earlier. The new folder names and locations means fewer nested folders, easier navigation and the new names better describe the contents of the folders.

Pages 4 through 9 of the document below describe the new folder structure and contains a full mapping of the relevant XP folder name to the new format.

http://www.microsoft.com/download/en/details.aspx?DisplayLang=en&id=22322

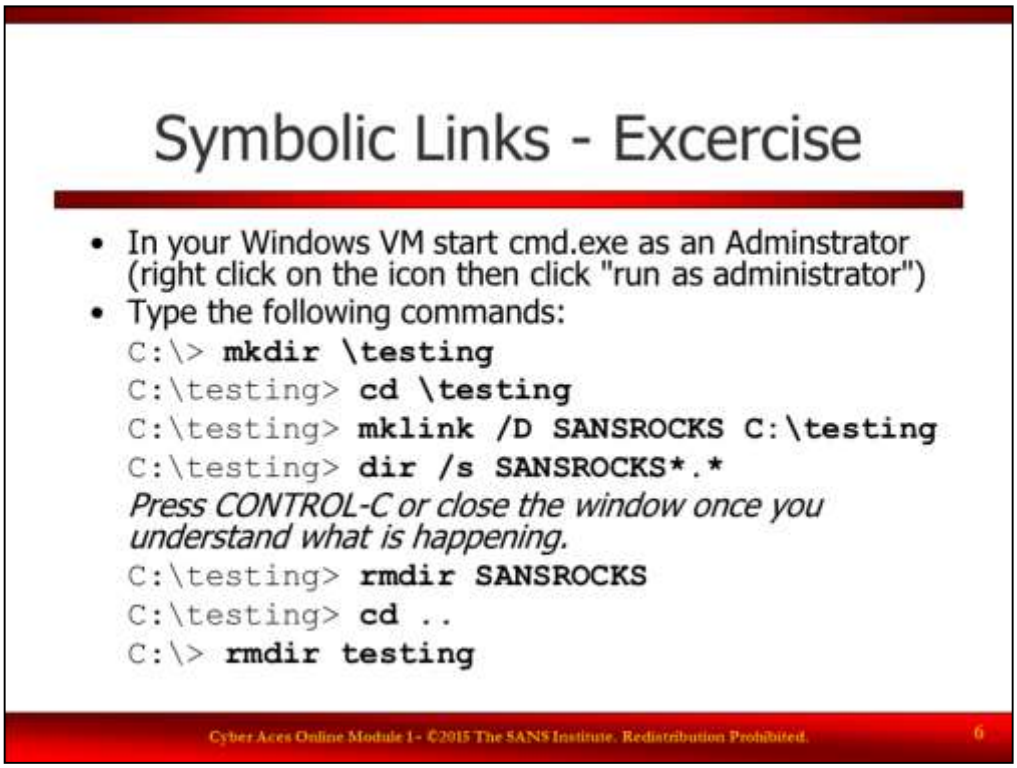
The "AppData" folder used in Vista and later is different from the "Application Data" folder used in XP. A number of folders in XP were moved under the "AppData" folder.

Vista & Later	**XP & Earlier**
\AppData\Roaming	Application Data
\AppData\Local	Local Settings
\AppData\Local	Local Settings\Application Data



Symbolic Links

Windows Vista introduced Symbolic Links to the Windows world. A symbolic link makes a file or directory that exists somewhere appear to exist somewhere else. File based symbolic links have similarities to Shortcuts. Symbolic Links function like symbolic links in the Linux world and replace Junctions introduced with Windows 2000. Symbolic Links are often interesting to attackers because they can be used to access directories outside of the current subdirectory tree. For example, an application may have software-enforced restrictions that do not allow the users to access any file above the c:\inetpub\wwwroot directory. However, a poorly placed symbolic link by the administrator, or one created by an attacker, might be used to bypass that restriction. A poorly placed symbolic link can lead to several problems.



Symbolic Links Exercise

In your Windows VM open cmd.exe as an Administrator. To do this, press the windows key to access the start screen. Type "cmd.exe", then right click on the icon and click on "run as adminsitrator" at the bottom of the screen. You will be asked if you want to run the executable with full permissions, click "Yes". You should now have a cmd.exe window and the title bar should start with "Administrator". Now, type the following command.

```
C:\> mkdir \testing
C:\testing> cd \testing
C:\testing> mklink /D SANSROCKS C:\testing
C:\testing> dir /s SANSROCKS*.*
```

Press CONTROL-C or close the window once you understand what is happening.

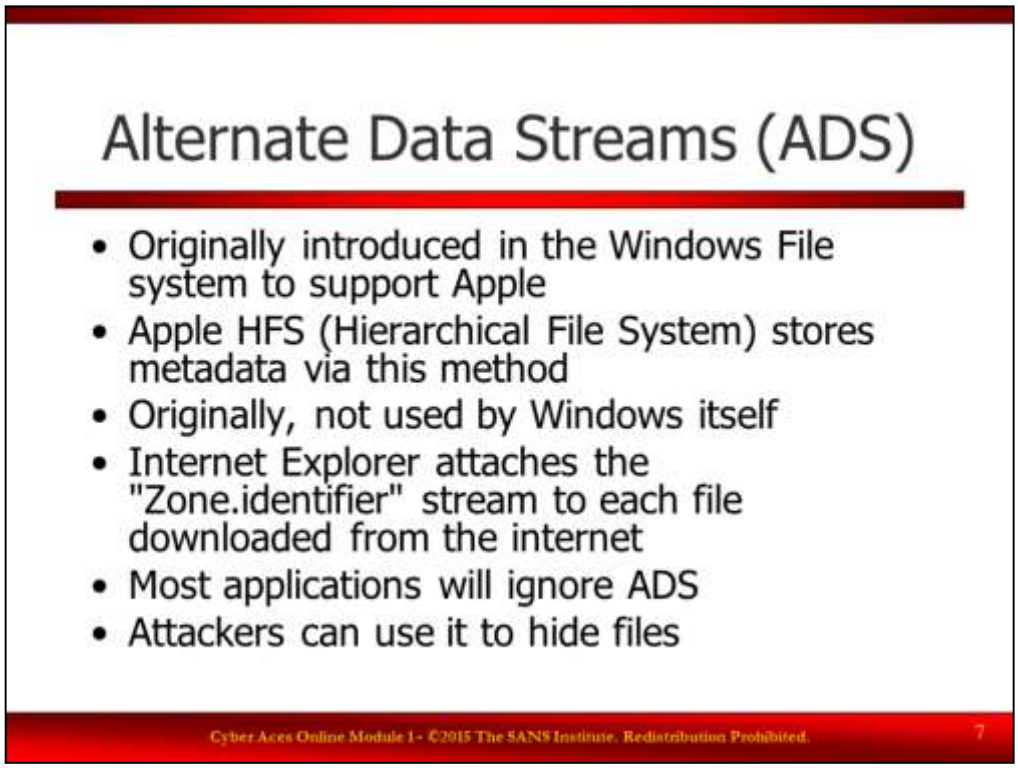
```
C:\testing> rmdir SANSROCKS
C:\testing> cd ..
C:\> rmdir testing
```

Notice that we created the SANSROCKS link in the C:\testing directory and not in the ROOT of the file system (i.e. in C:\). If we do a recursive directory listing it will enter the SANSROCKS directory which warps us to \testing. Within that directory we see SANSROCKS (again) and will try to enter it (again). This results in an infinite loop.

What would happen if we created our link to the root of the drive and to remove the directory you typed "del /s SANSROCKS"? (The answer is on the next page)

- It would create a paradox in the space-time continuum, ending life as we know it
- It would only delete the SANSROCKS directory and stop
- It would traverse the symbolic link and attempt to delete everything off the C: drive

• It would only remove the symbolic link



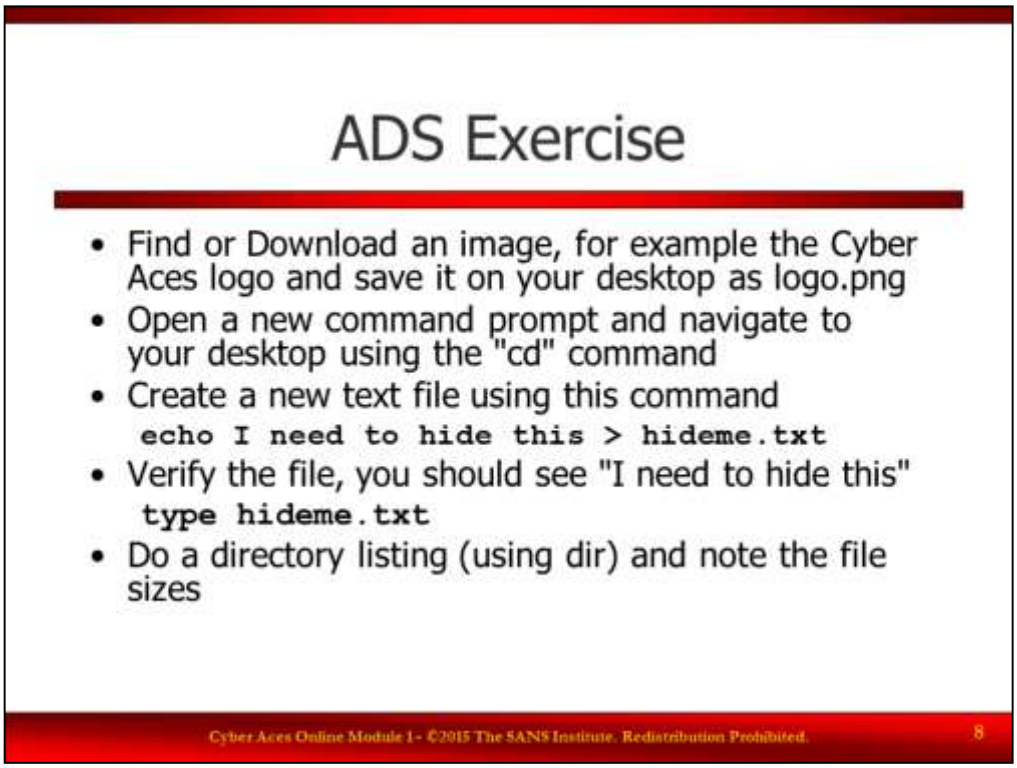
Alternate Data Streams (ADS)

The Windows NTFS file system also support Alternate Data Streams or "ADS". Alternate Data Streams were originally introduced to the Windows File system in order to provide support for Apple computers. Apple HFS stores information about a file such as the name of the program that created the file in a file "resource fork". This information is not normally used by the Windows Operating system and is ignored by most Windows applications. However, Internet Explorer adds a "Zone.Identifier" stream to each file downloaded from the internet. Since information is stored and accessible, but ignored by normal operations, Alternate Data Streams can be used by attackers to hide malicious files from view.

ADS gives you the ability to inject/add file data into existing files without affecting their functionality, size, or display in utilities like Windows Explorer or even "dir" under command line.

Answer from question on previous slide:

It would traverse the symbolic link and attempt to delete everything off the C: drive



ADS Exercise

In this exercise you will be using the commands and functions below.

type – as previously discussed, this will output a file. It also supports ADS

echo – used to output text

> (greater than) – The "redirect" is used to redirect output to a file or stream that would normally be displayed

: (colon) – The delimiter used to specify the stream. Make sure you use a colon and not a semi-colon

start – Run a program in a new session. Just typing "start" will open a new command prompt

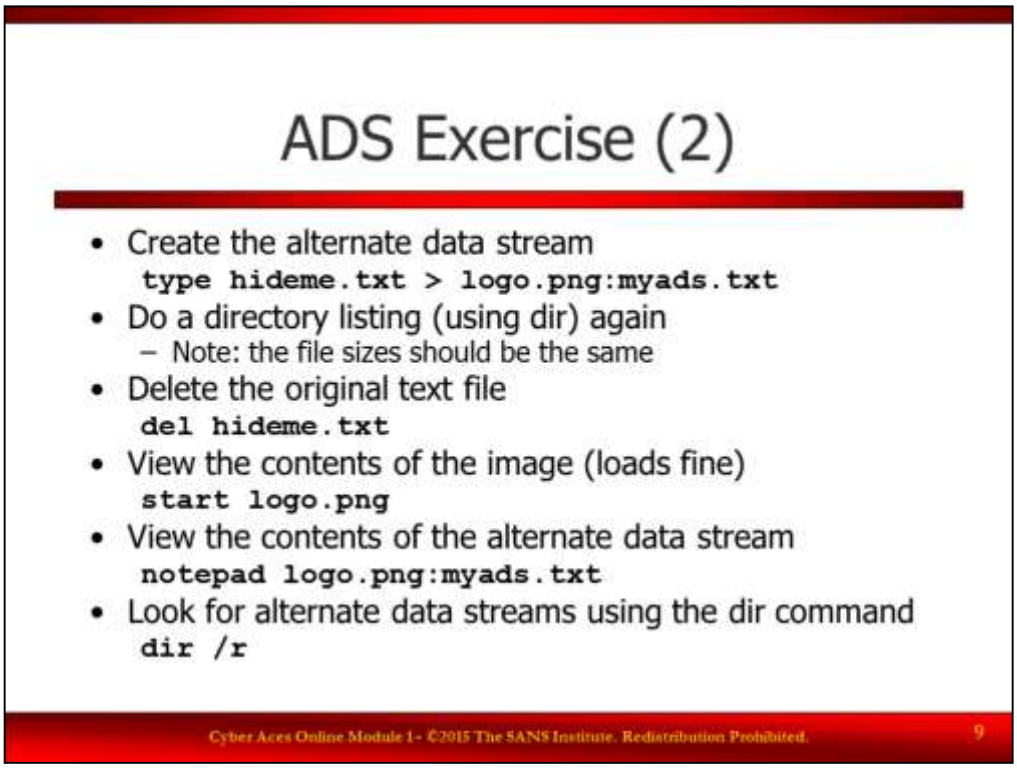
Follow these steps:

- Open your browser, and navigate to www.cyberaces.org. Right click on the logo in the top right corner and then click on "Save Image As…". Select your desktop and use the default file name.
- Open a command prompt by clicking on the Windows button, then click on "Run…". Type "cmd.exe" (no quotes) and hit enter
- Change to your desktop by typing: **`cd %USERPROFILE%\Desktop`**
- Create a new text file with the contents of "I need to hide this": **`echo I need to hide this > hideme.txt`**
- Verify the file, you should see "I need to hide this": **`type hideme.txt`**
- Type "dir" to see the directory listing. It should look similar to this:

```
C:\Users\yourusername\Desktop> dir
 Volume in drive C has no label.
 Volume Serial Number is CAFÉ-BABE

 Directory of C:\Users\yourusername\Desktop

04/05/2015  02:28 PM    <DIR>          .
04/05/2015  02:28 PM    <DIR>          ..
04/05/2015  02:15 PM                22 hideme.txt
03/21/2015  01:42 PM            63,341 logo.png
               2 File(s)         63,363 bytes
```



ADS Exercise (2)

- Create the alternate data stream:

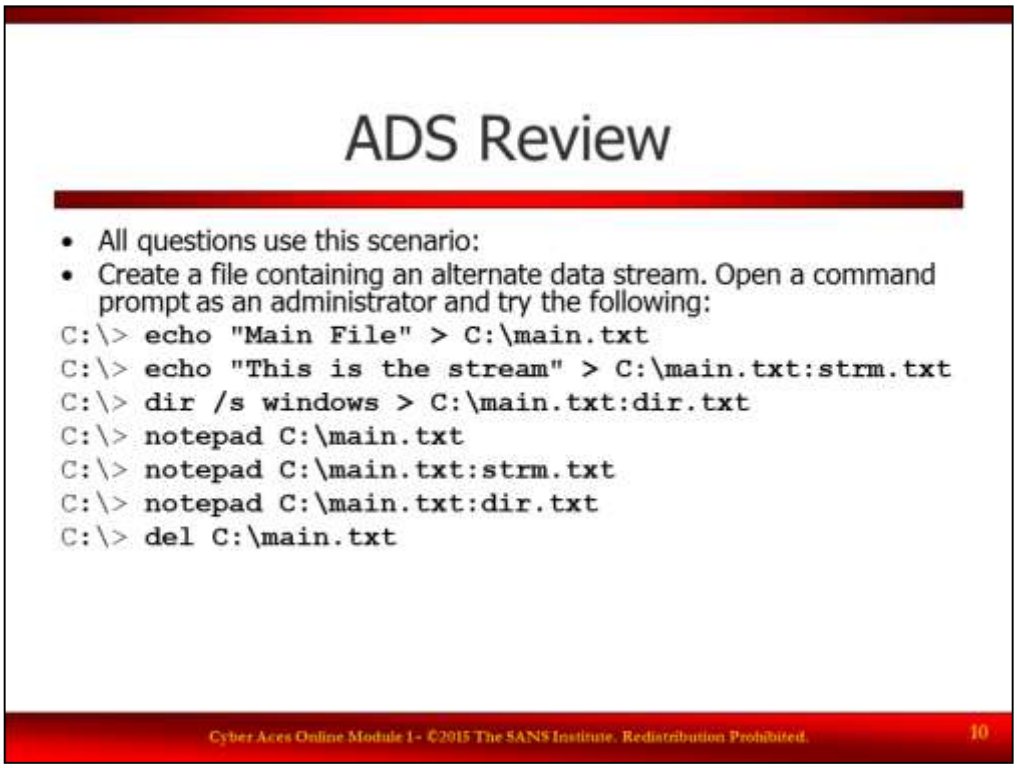
  `type hideme.txt > logo.png:myads.txt`
- Do a directory listing (using dir) again. Note: the file sizes should be the same
- Delete the original text file:

  `del hideme.txt`
- View the contents of the image (loads fine):

  `start logo.png`
- View the contents of the alternate data stream:

  `notepad logo.png:myads.txt`
- Look for alternate data streams using the dir command:

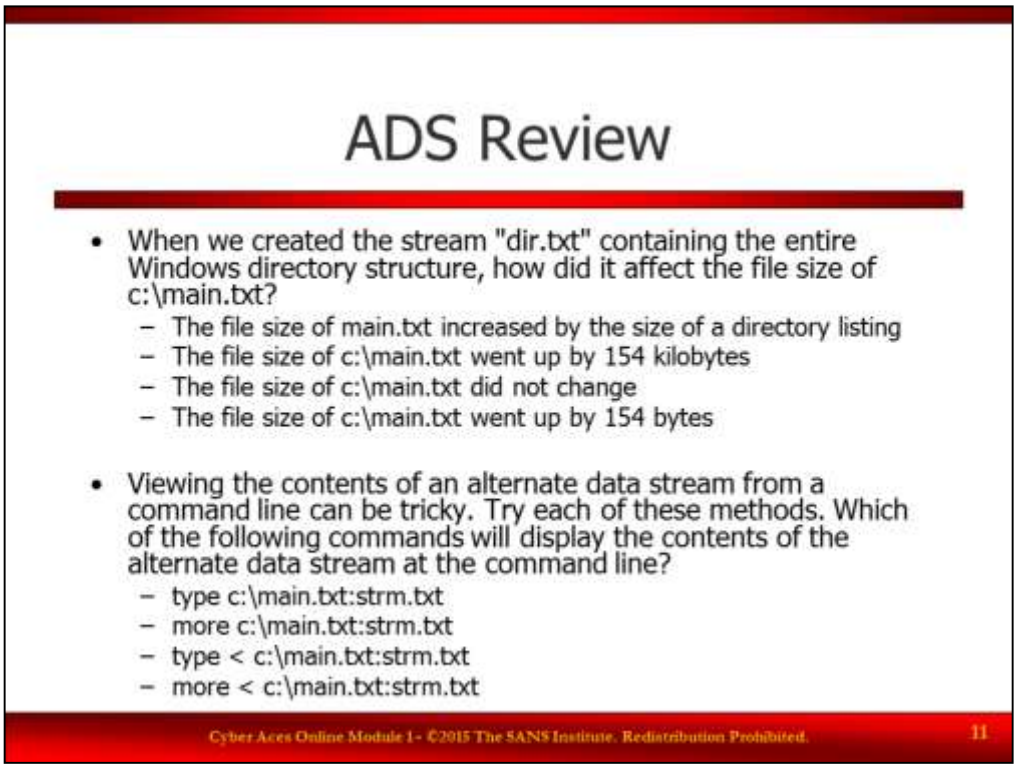
  `dir /r`



ADS Review

All questions use the scenario below. Create a file containing an alternate data stream. Open a command prompt as an administrator and try the following:

```
C:\> echo "Main File" > C:\main.txt
C:\> echo "This is the stream" > C:\main.txt:strm.txt
C:\> dir /s windows > C:\main.txt:dir.txt
C:\> notepad C:\main.txt
C:\> notepad C:\main.txt:strm.txt
C:\> notepad C:\main.txt:dir.txt
C:\> del C:\main.txt
```



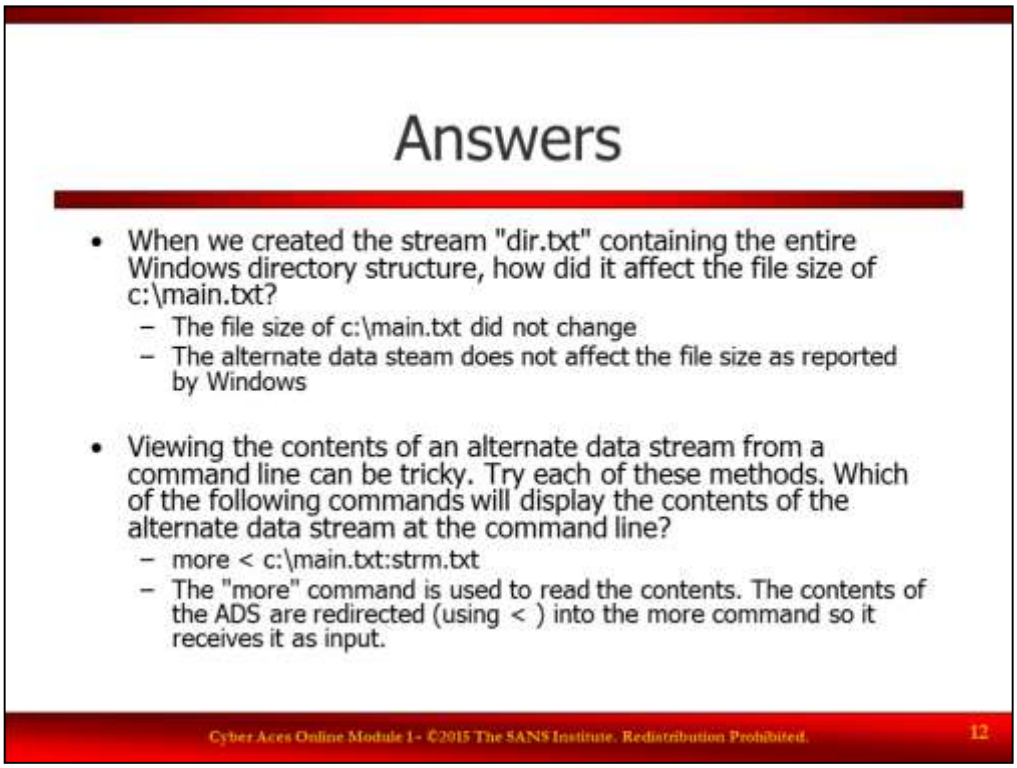
ADS Review

When we created the stream "dir.txt" containing the entire Windows directory structure, how did it affect the file size of c:\main.txt?

- The file size of main.txt increased by the size of a directory listing
- The file size of c:\main.txt went up by 154 kilobytes
- The file size of c:\main.txt did not change
- The file size of c:\main.txt went up by 154 bytes

Viewing the contents of an alternate data stream from a command line can be tricky. Try each of these methods. Which of the following commands will display the contents of the alternate data stream at the command line?

- type c:\main.txt:strm.txt
- more c:\main.txt:strm.txt
- type < c:\main.txt:strm.txt
- more < c:\main.txt:strm.txt



ADS Review Answers

When we created the stream "dir.txt" containing the entire Windows directory structure, how did it affect the file size of c:\main.txt?

Answer: The file size of c:\main.txt did not change.

Why: The alternate data steam does not affect the file size as reported by Windows.

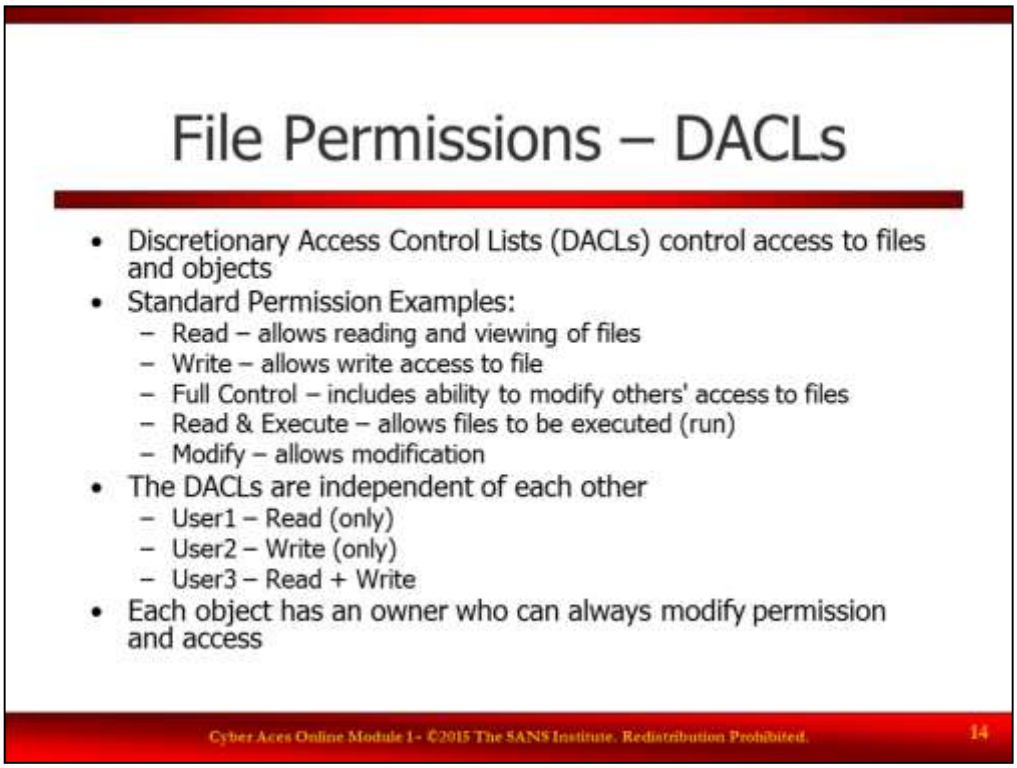
Viewing the contents of an alternate data stream from a command line can be tricky. Try each of these methods. Which of the following commands will display the contents of the alternate data stream at the command line?

Answer: more < c:\main.txt:strm.txt

The "more" command is used to read the contents. The contents of the ADS are redirected (using <) into the more command so it receives it as input.

Mandatory Integrity Controls (MIC)

Windows uses integrity control to prevent users and processes that have one level of trust from modifying files at another level of trust. For example, Windows will prevent Internet Explorer from modifying operating system files in the c:\windows\system32 directory. Users are assigned an "Integrity Level" of High, Medium or Low. Operating system objects such as files are also assigned an "Integrity Level" of High, Medium or Low. Users can only modify files with an integrity level that is equal to or lower than their own. So a user who has a "Medium" integrity level can only modify Medium or Low Integrity files. By default, users have a Medium Integrity level, but the Operating System will drop the user to Low Integrity when the user does things like browsing the web or reading email. The operating system and some applications such as Internet Explorer also create a "LOW" directory to make files available to the user when their integrity level is demoted.



File Permissions – DACLs

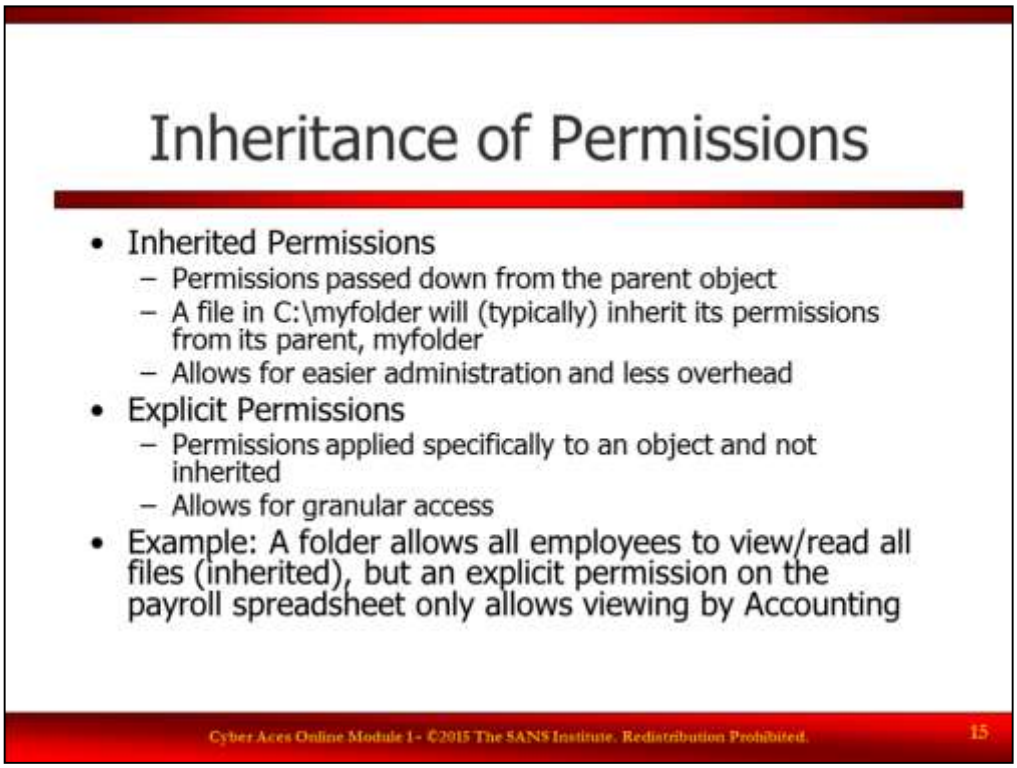
Windows uses "Discretionary Access Control Lists" (DACLS) to control access to file and system objects. Each directory or file has a list of permissions associated with it. Those permissions detail who can access the files and what they can do with the file. Some users will have read only access while others have the ability to read, write or execute the files. Other users might be assigned "full control" of the files, including the ability to change other users' access to the file. File and directory objects also have an "owner". The "owner" can always modify permissions on the object and control who can access it. In addition to Standard Permission there are Advanced Permissions which allow very granular settings on the security of objects.

- Full Control
- Traverse Folder/Execute File
- List Folder/Read Data
- Read Attributes
- Read Extended Attributes
- Create Files/Write Data
- Create Folders/Append Data
- Write Attributes
- Write Extended Attributes
- Delete
- Read Permissions
- Change Permissions
- Take Ownership

The "Standard Read Permission" allows these Advanced Permissions:

- List Folder/Read Data
- Read Attributes
- Read Extended Attributes
- Read Permissions

Reference: http://www.windowsecurity.com/articles/Understanding-Windows-NTFS-Permissions.html

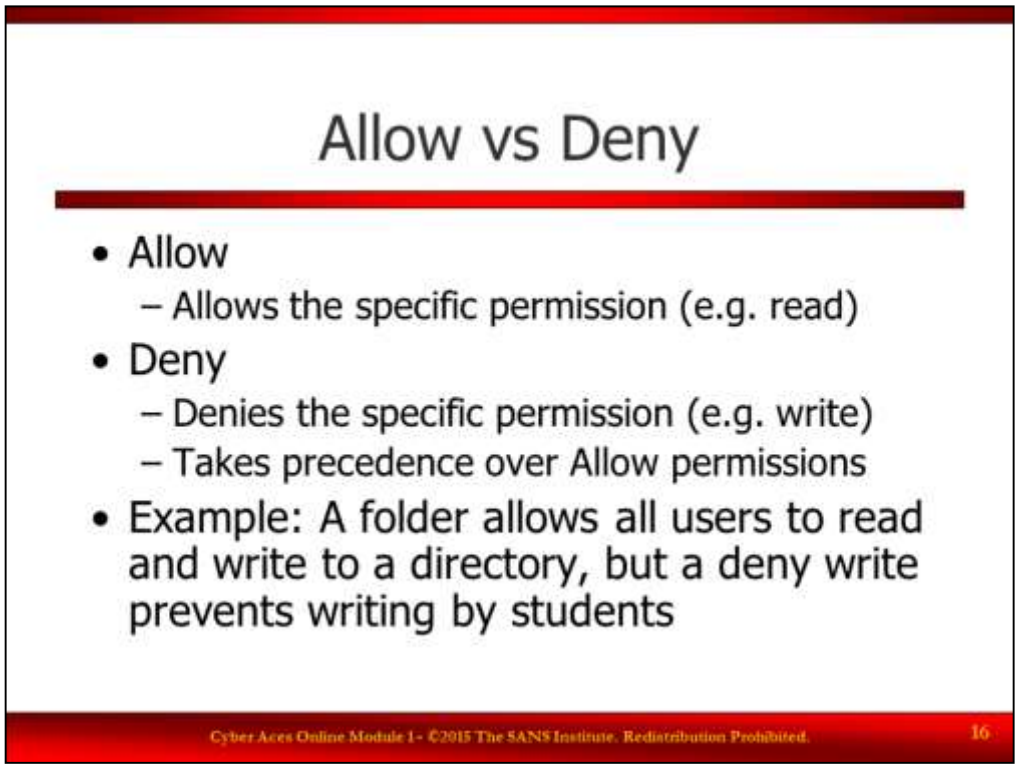


Inheritance of Permissions

Inherited permissions are permissions applied to a parent directory and "inherited" from that parent. For example, if folder B is under (a child of) folder A, then folder B will inherit permissions applied to A.

Explicit permissions are applied to a specific object (file or directory) and are not inherited.

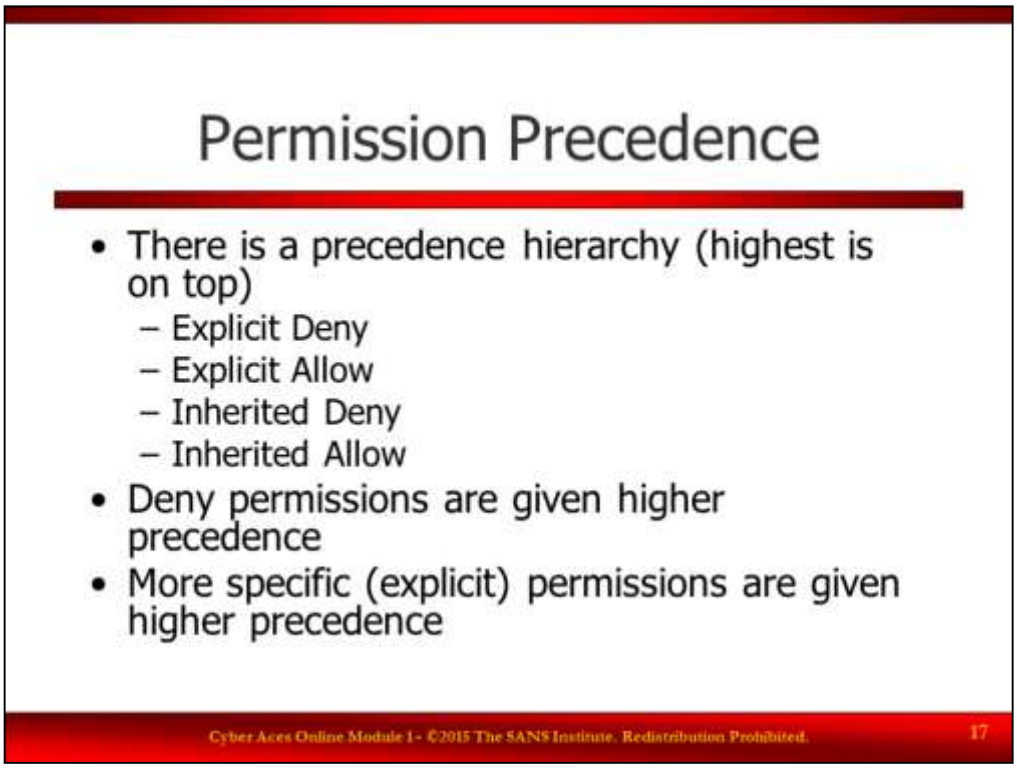
The combination of inherited and explicit permissions allows administrators to define broad permissions and then adjust them at a granular level. For example, we could have a folder with permissions that allow all employees to view/read all files (inherited by all objects), but an explicit permission on the payroll spreadsheet that only allows viewing by people in Accounting.



Allow vs Deny

Permissions can be highly complicated and the deny permission can make it easier to administer and specify permissions.

A specific folder may allow all employees, faculty, adjunct, consultants, etc (all authenticated users) to read and write to a specific directory, but they may not want to allow students to write to the directory so they have a deny write permission applied. This is simpler than adding each group that isn't a student and providing write access to all of them.



Permissions Precedence

Deny permissions are given a higher precedence.

CompanyFileShare – Full-Access for Administrators group, Read for all Users

|--HumanResourcesFolder – Write for users in HR Group, Deny Read/Write for users in the "Non-HR" group

| |--PayrollSpreadsheet.xlsx – Read Access by Executives

| |--EmployeeInfo.xlsx

| +-ResumesFolder

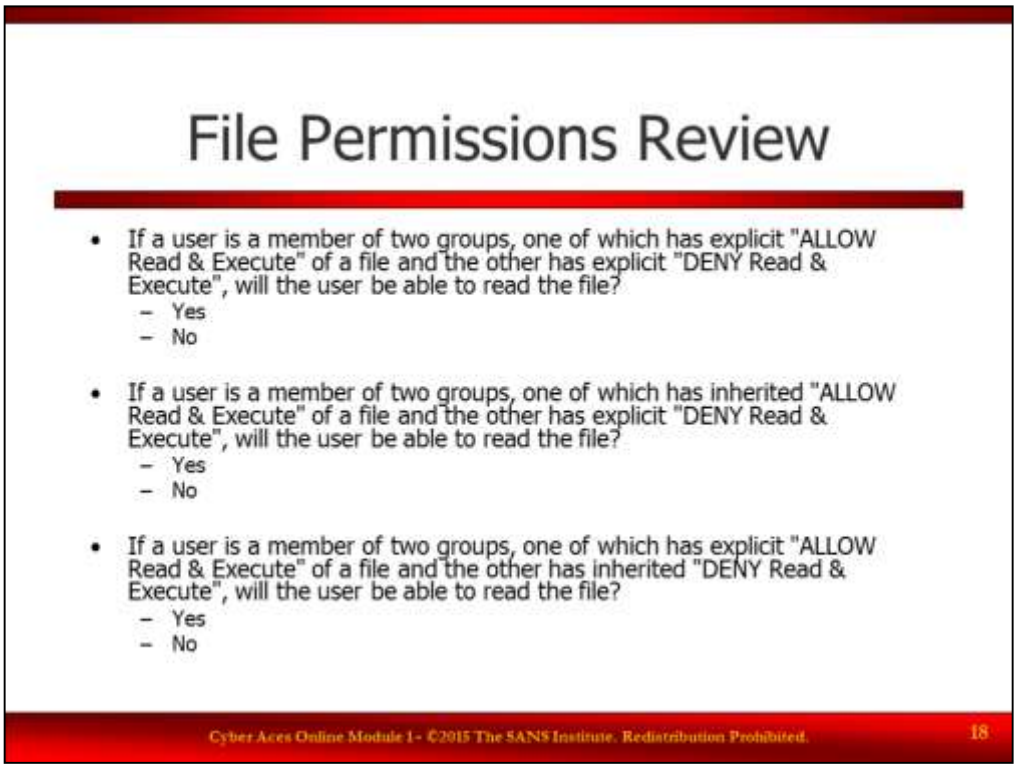
|--AccountingFolder – Write access for users in Account Group

|--EngineeringFolder – Write access for users in Engineering Group

+-MarketingFolder – Write access for users in Marketing Group

For example, a directory structure could be used to allow users to share files. Administrators would be given full access at a high level and it would be inherited to each object further down the directory tree. Other permissions would also add, so the AccountingFolder directory can be read by all users and written to by users in the Accounting group.

The objects (directories and files) in the HumanResources folder have a deny read/write for any users in the "non-HR" group. However, the explicit allow read access on the PayrollSpreadsheet.xlsx allows Executives to read the files since the Explicit Allow permissions has a higher precedence than the Deny access that is inherited from the parent folder (HumanResourcesFolder). This file would have to be accessed directly as the Executives are not allowed to read the directory that contains the file.



File Permissions Review

If a user is a member of two groups, one of which has explicit "ALLOW Read & Execute" of a file and the other has explicit "DENY Read & Execute", will the user be able to read the file?

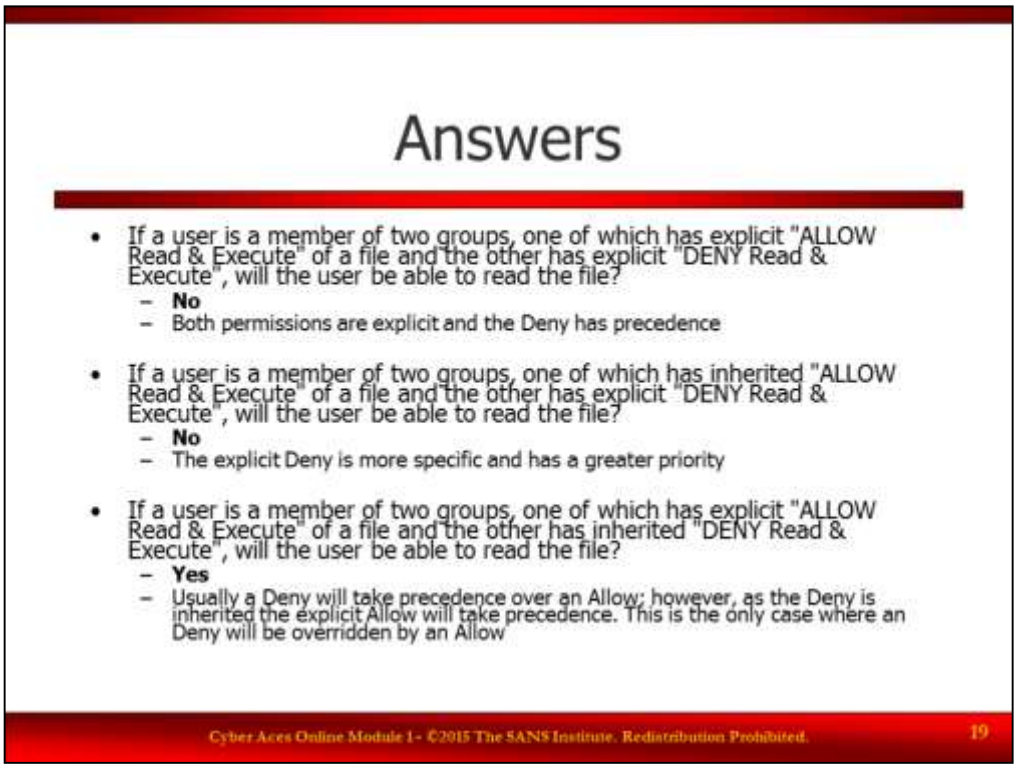
- Yes
- No

If a user is a member of two groups, one of which has inherited "ALLOW Read & Execute" of a file and the other has explicit "DENY Read & Execute", will the user be able to read the file?

- Yes
- No

If a user is a member of two groups, one of which has explicit "ALLOW Read & Execute" of a file and the other has inherited "DENY Read & Execute", will the user be able to read the file?

- Yes
- No



File Permissions Answers

If a user is a member of two groups, one of which has explicit "ALLOW Read & Execute" of a file and the other has explicit "DENY Read & Execute", will the user be able to read the file?

Answer: No, as both permissions are explicit and the Deny has precedence

If a user is a member of two groups, one of which has inherited "ALLOW Read & Execute" of a file and the other has explicit "DENY Read & Execute", will the user be able to read the file?

Answer: No, the explicit Deny is more specific and has a greater priority

If a user is a member of two groups, one of which has explicit "ALLOW Read & Execute" of a file and the other has inherited "DENY Read & Execute", will the user be able to read the file?

Answer: Yes. Usually a Deny will take precedence over an Allow; however, as the Deny is inherited the explicit Allow will take precedence. This is the only case where an Deny will be overridden by an Allow

Congratulations, you have completed the tutorial on the Windows file system