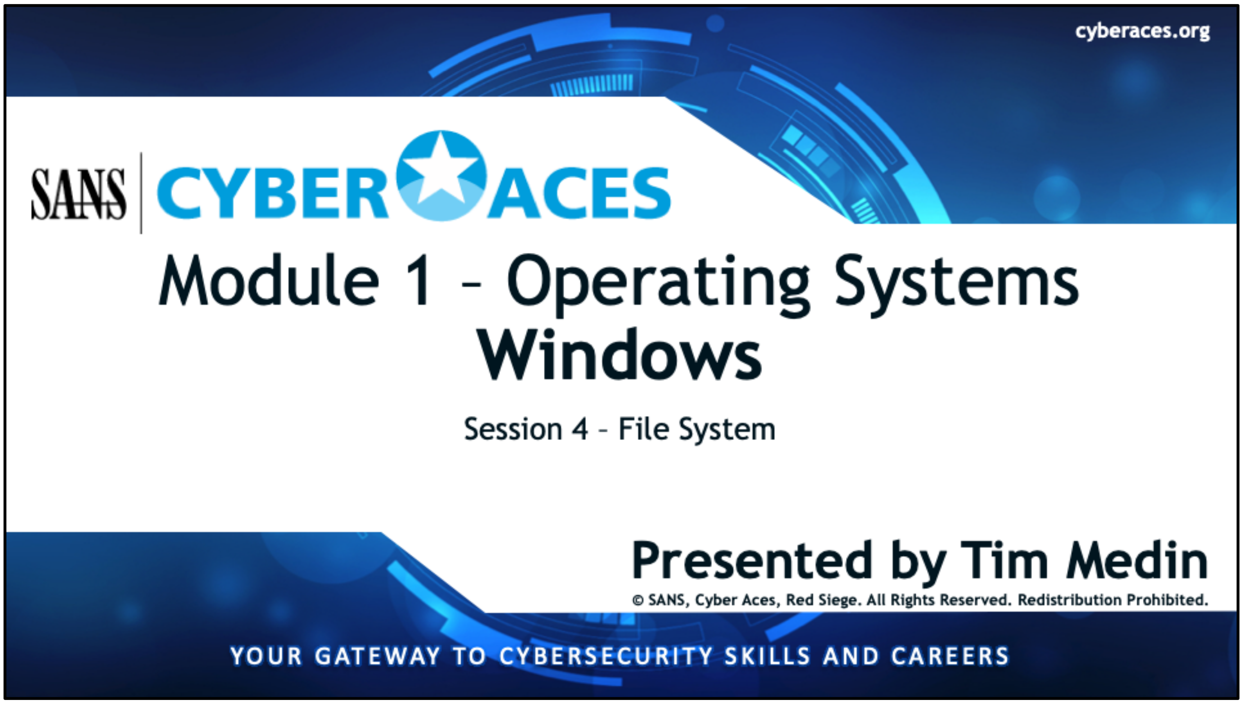


cyberaces.org

SANS | CYBER ACES

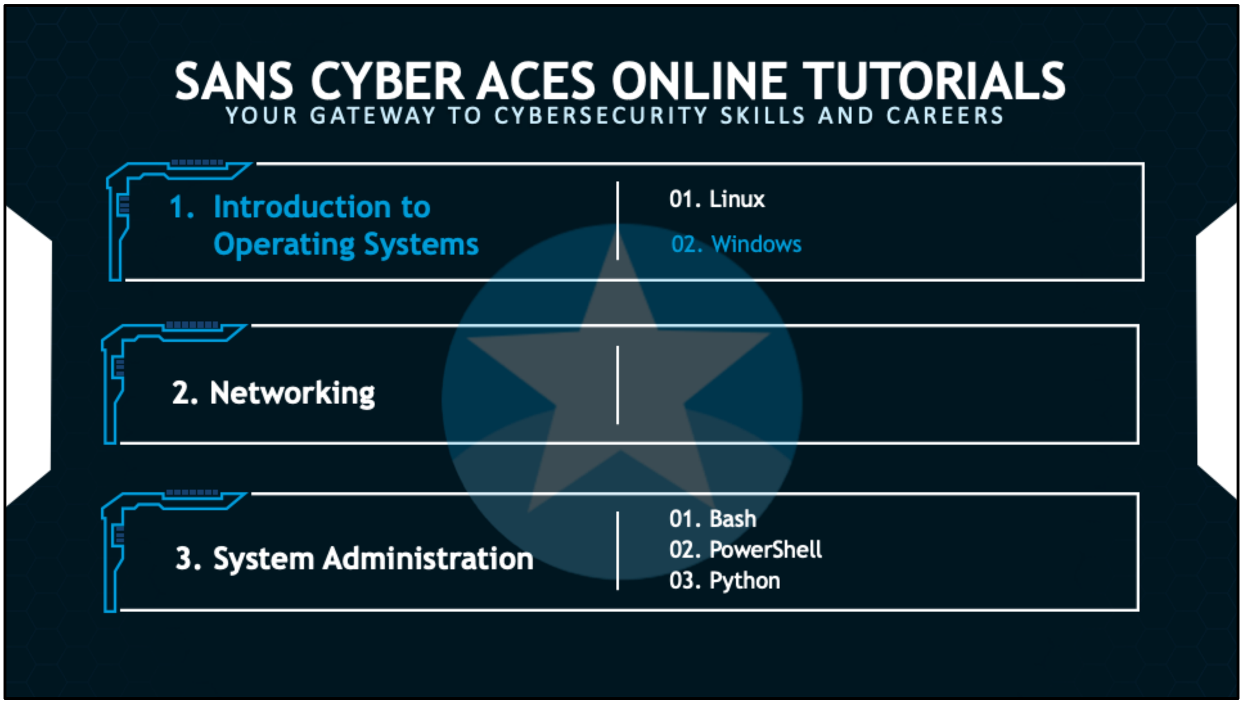
Module 1 – Operating Systems Windows

Session 4 – File System

Presented by Tim Medin

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

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.



SANS CYBER ACES ONLINE TUTORIALS

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems
   - 01. Linux
   - 02. Windows

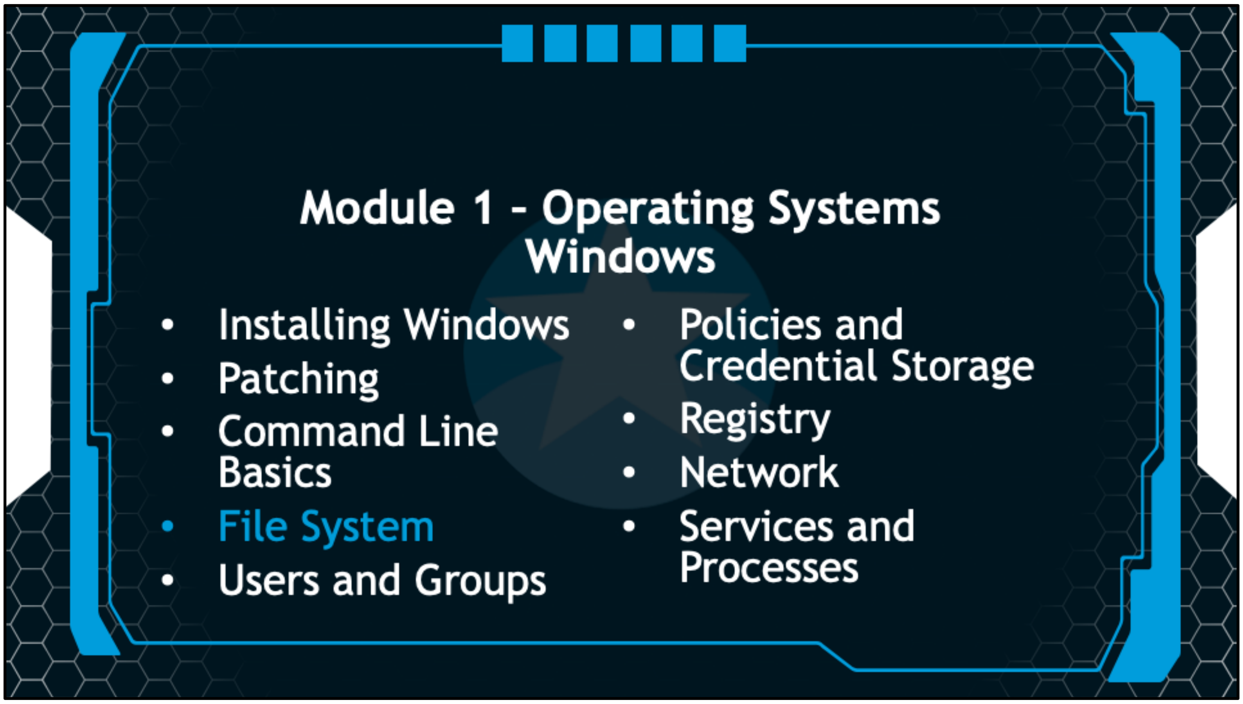
2. Networking

3. System Administration
   - 01. Bash
   - 02. PowerShell
   - 03. Python

This training material was originally developed to help students, teachers, and mentors prepare for the Cyber Aces Online Competition. This module focuses on the basics of what an operating systems is as well as the two predominant OS's, Windows and Linux. This session is part of Module 1, Introduction to Operating Systems. This module is split into two sections, Linux and Windows. In this session, we will continue our examination of Windows.

The three modules of Cyber Aces Online are Operating Systems, Networking, and System Administration.

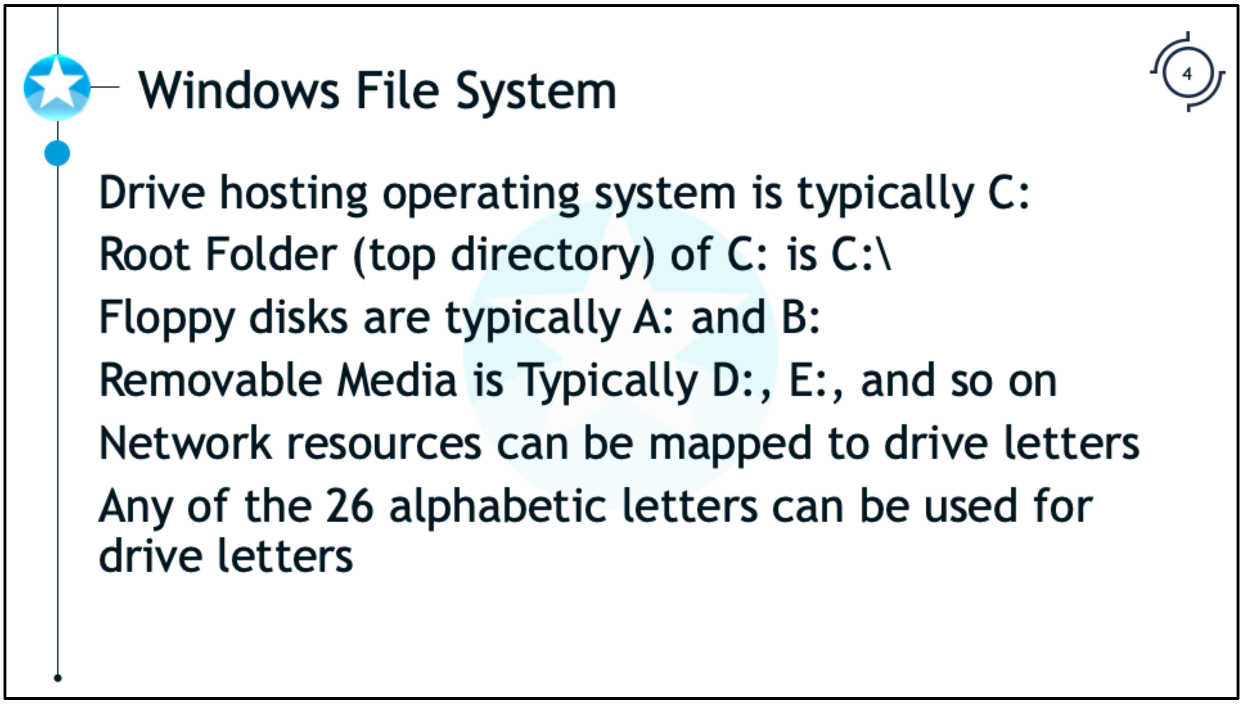
For more information about the Cyber Aces program, please visit the Cyber Aces website at https://CyberAces.org/.



Module 1 - Operating Systems Windows

- Installing Windows
- Patching
- Command Line Basics
- File System
- Users and Groups
- Policies and Credential Storage
- Registry
- Network
- Services and Processes

In this session we will discuss the Windows file system.



Windows File System

Drive hosting operating system is typically C:

Root Folder (top directory) of C: is C:\

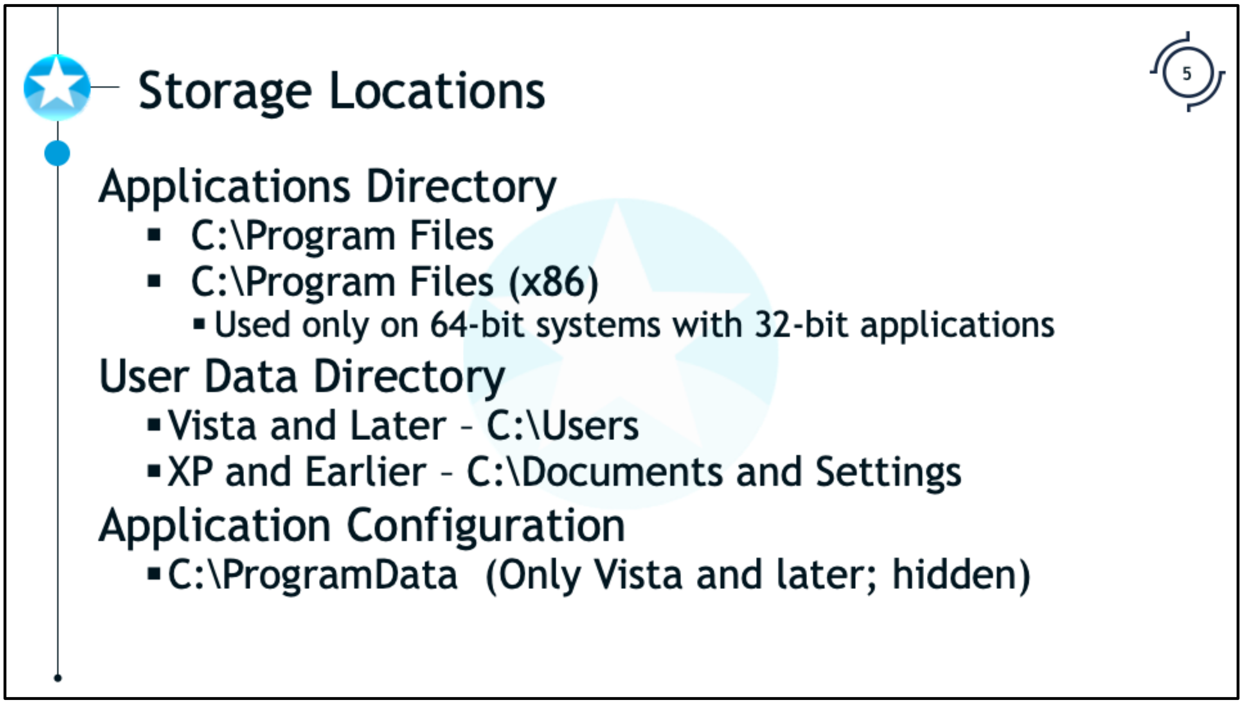
Floppy disks are typically A: and B:

Removable Media is Typically D:, E:, and so on

Network resources can be mapped to drive letters

Any of the 26 alphabetic letters can be used for drive letters

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

Applications Directory

- C:\Program Files
- C:\Program Files (x86)
  - Used only on 64-bit systems with 32-bit applications

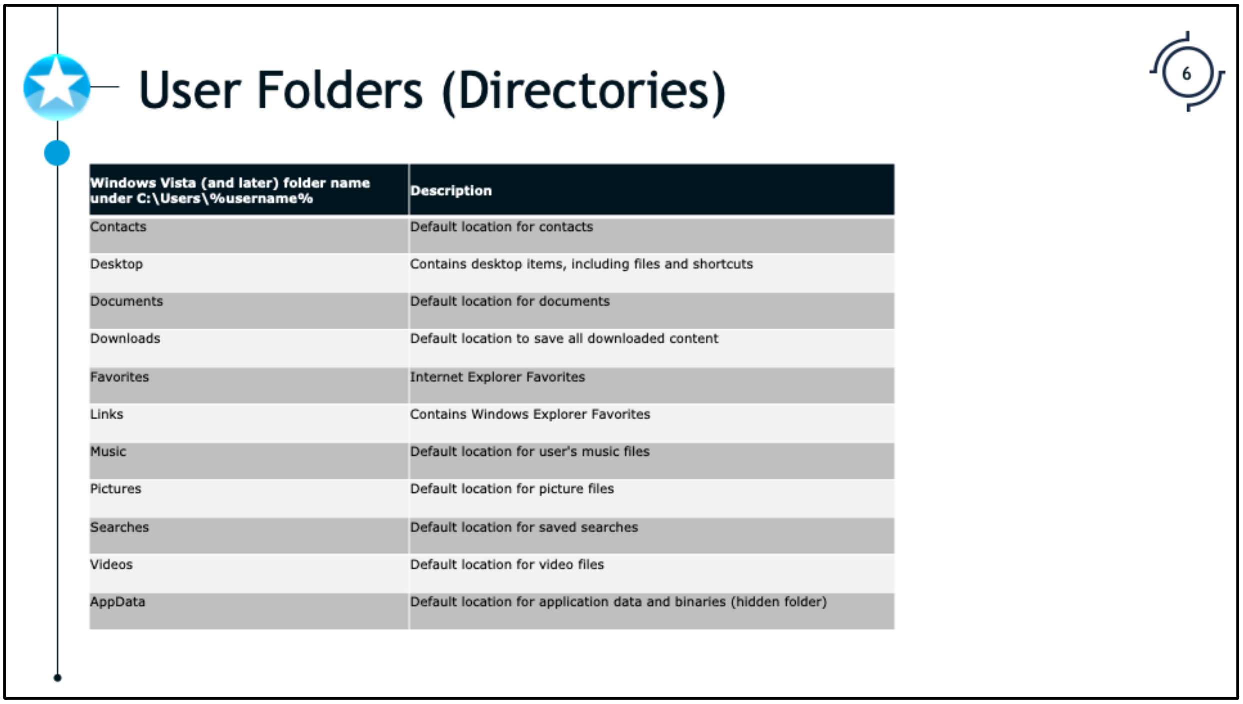
User Data Directory

- Vista and Later – C:\Users
- XP and Earlier – C:\Documents and Settings

Application Configuration

- C:\ProgramData (Only Vista and later; hidden)

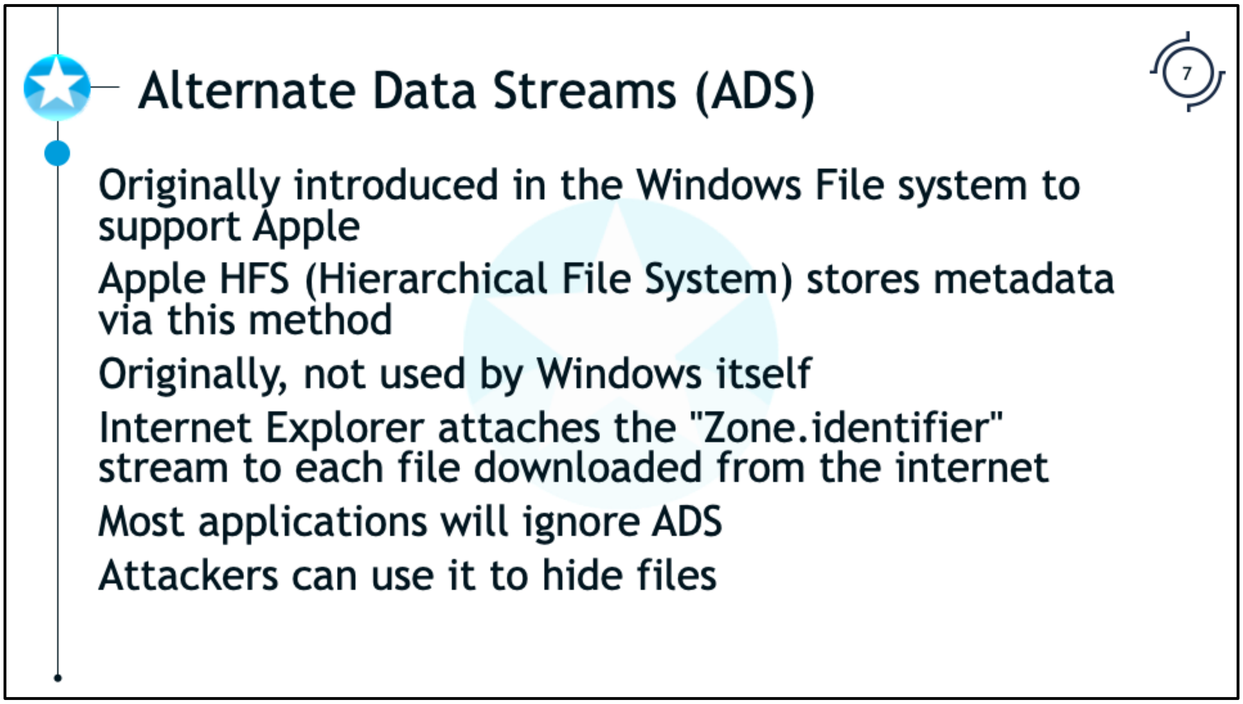
Microsoft has released a whitepaper on the directory structure of Windows Vista. Although Windows 7 has introduced some new concepts such as "Libraries", the underlying directory naming standard has remained unchanged on Windows 7 through Windows 10. An attacker will be very familiar with the directory 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
Contacts	Default location for contacts
Desktop	Contains desktop items, including files and shortcuts
Documents	Default location for documents
Downloads	Default location to save all downloaded content
Favorites	Internet Explorer Favorites
Links	Contains Windows Explorer Favorites
Music	Default location for user's music files
Pictures	Default location for picture files
Searches	Default location for saved searches
Videos	Default location for video files
AppData	Default location for application data and binaries (hidden folder)

Microsoft uses a new folder structure for storing user data, starting with Windows Vista. User profile folders as stored in C:\Users\%username% by default. The table above provides the default directory locations for storing specific kinds of data in Windows 10.

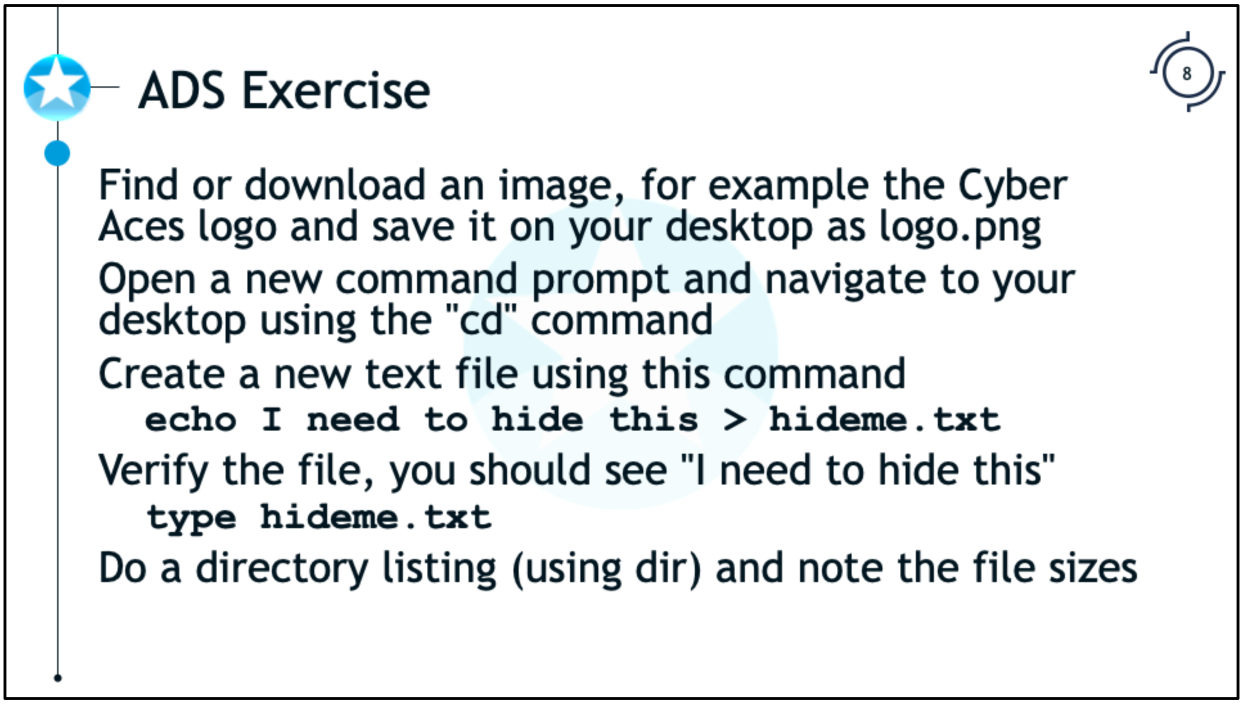


Alternate Data Streams (ADS)

- Originally introduced in the Windows File system to support Apple
- Apple HFS (Hierarchical File System) stores metadata via this method
- Originally, not used by Windows itself
- Internet Explorer attaches the "Zone.identifier" stream to each file downloaded from the internet
- Most applications will ignore ADS
- Attackers can use it to hide files

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.



ADS Exercise

Find or download an image, for example the Cyber Aces logo and save it on your desktop as logo.png

Open a new command prompt and navigate to your desktop using the "cd" command

Create a new text file using this command

```
echo I need to hide this > hideme.txt
```

Verify the file, you should see "I need to hide this"

```
type hideme.txt
```

Do a directory listing (using dir) and note the file sizes

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:

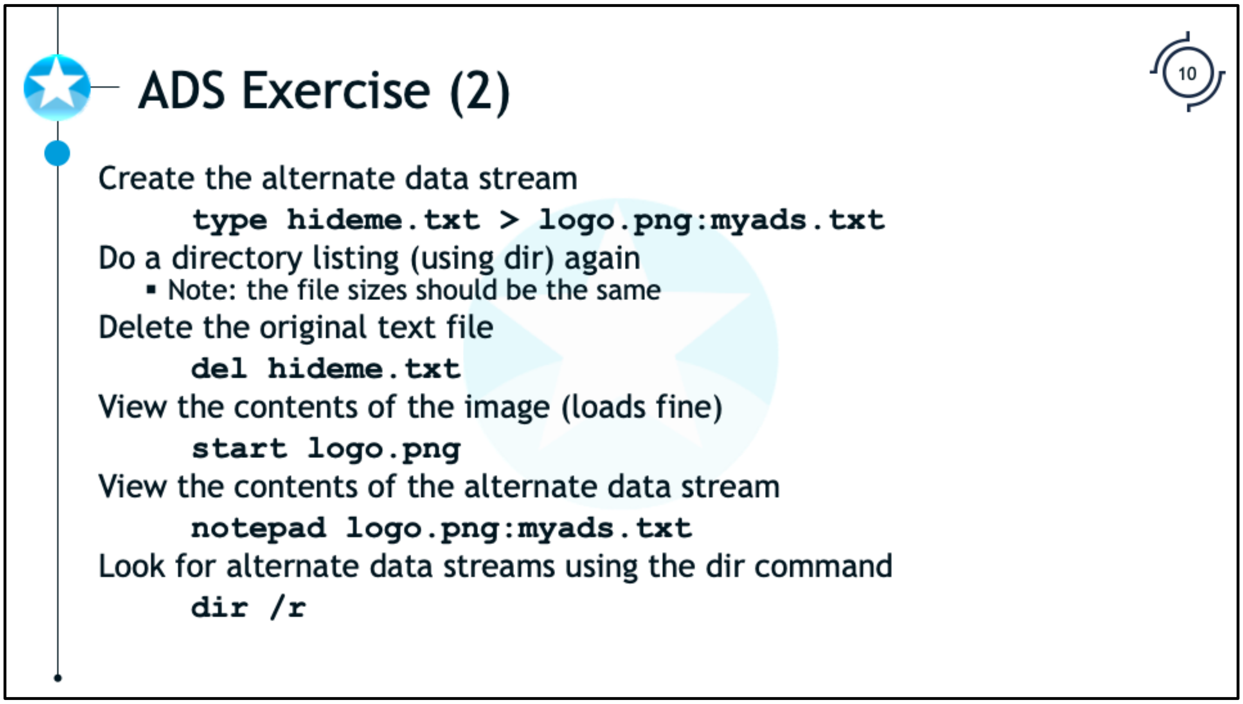
Open your browser, and navigate to cyberaces.org. Right click on the logo in the top left corner and then click on "Save Image As…". Select your desktop and use the default file name.

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`**



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
```

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):

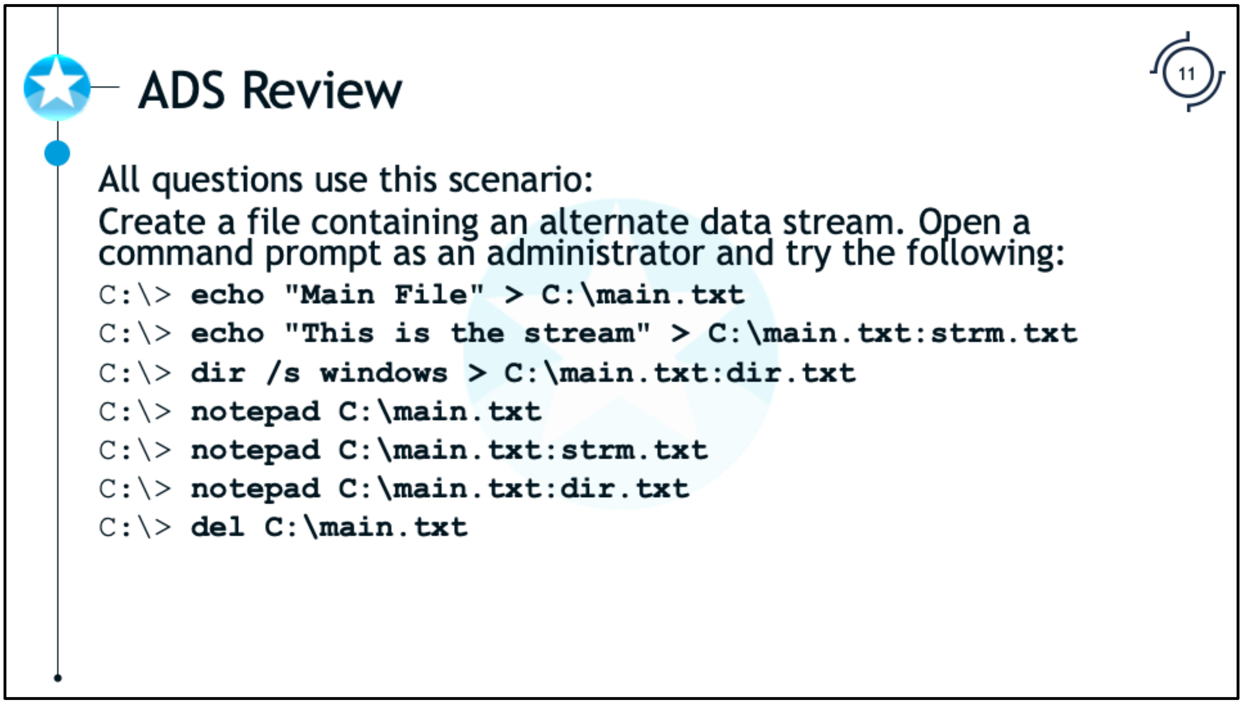
```
start logo.png
```

View the contents of the alternate data stream:

```
notepad logo.png:myads.txt
```

Look for alternate data streams using the dir command:

```
dir /r
```



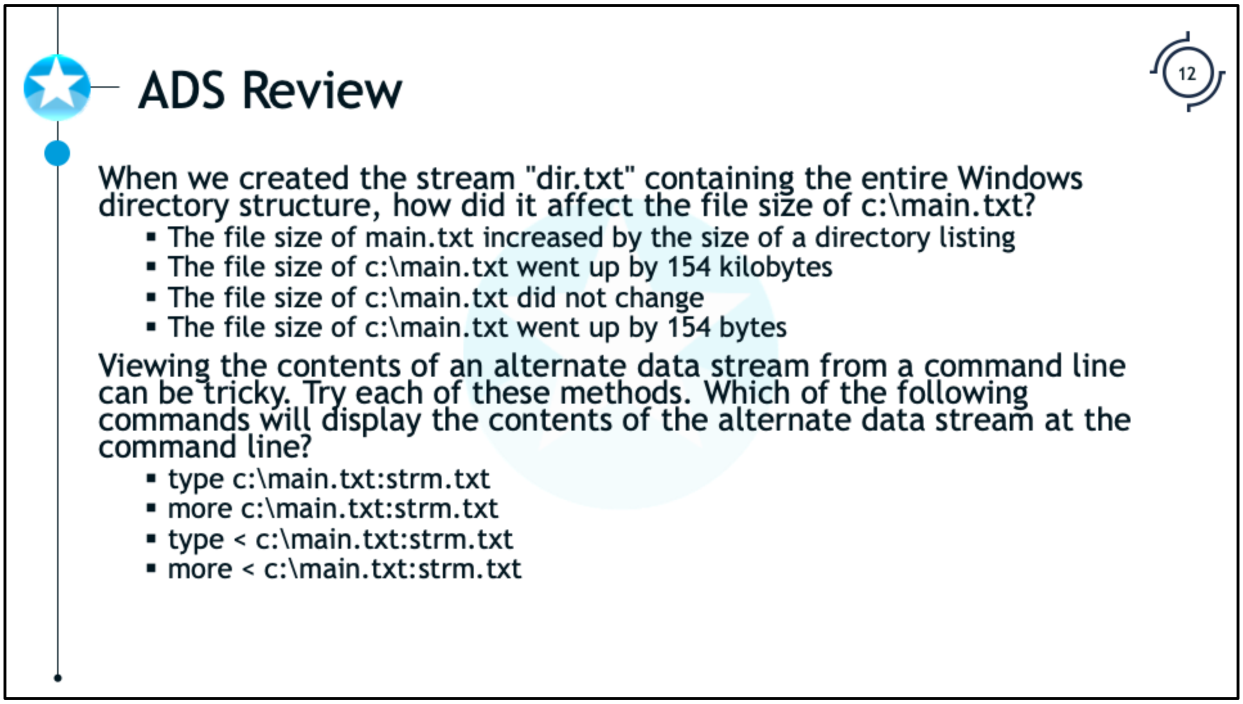
ADS Review

All questions use this scenario:
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
```

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?

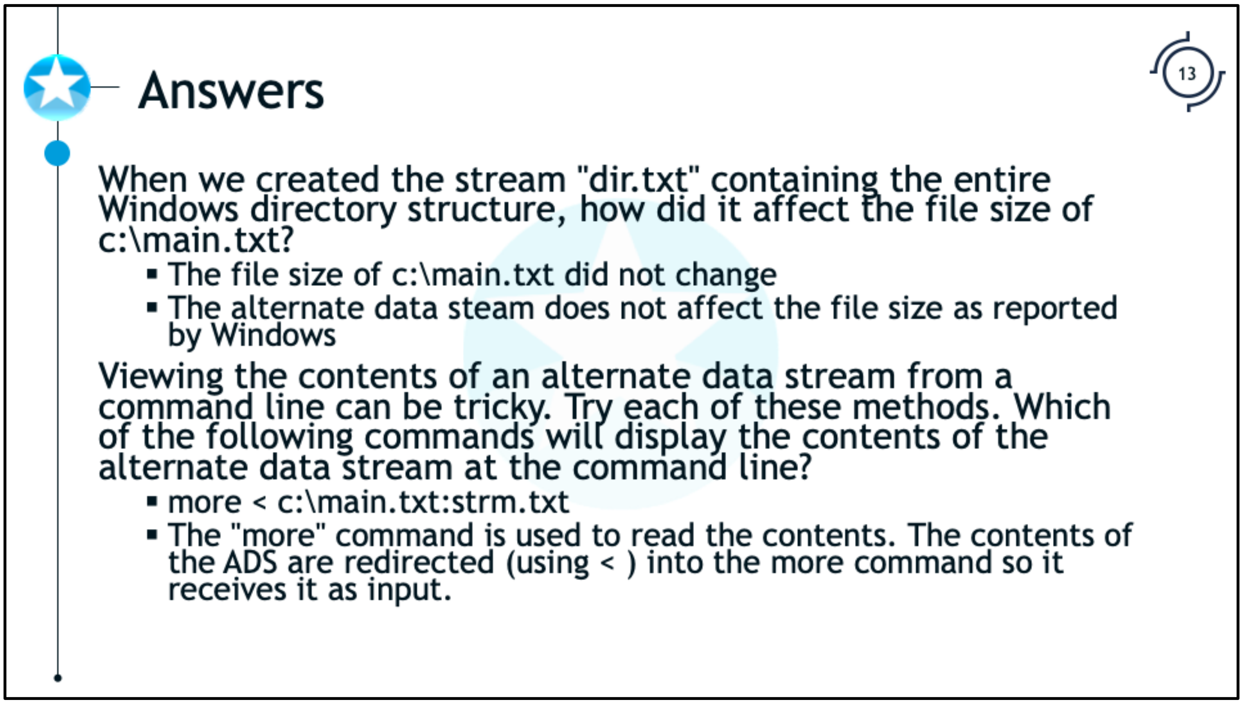
- type c:\main.txt:strm.txt
- more c:\main.txt:strm.txt
- type < c:\main.txt:strm.txt
- more < c:\main.txt:strm.txt

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



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?

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

- 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.

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

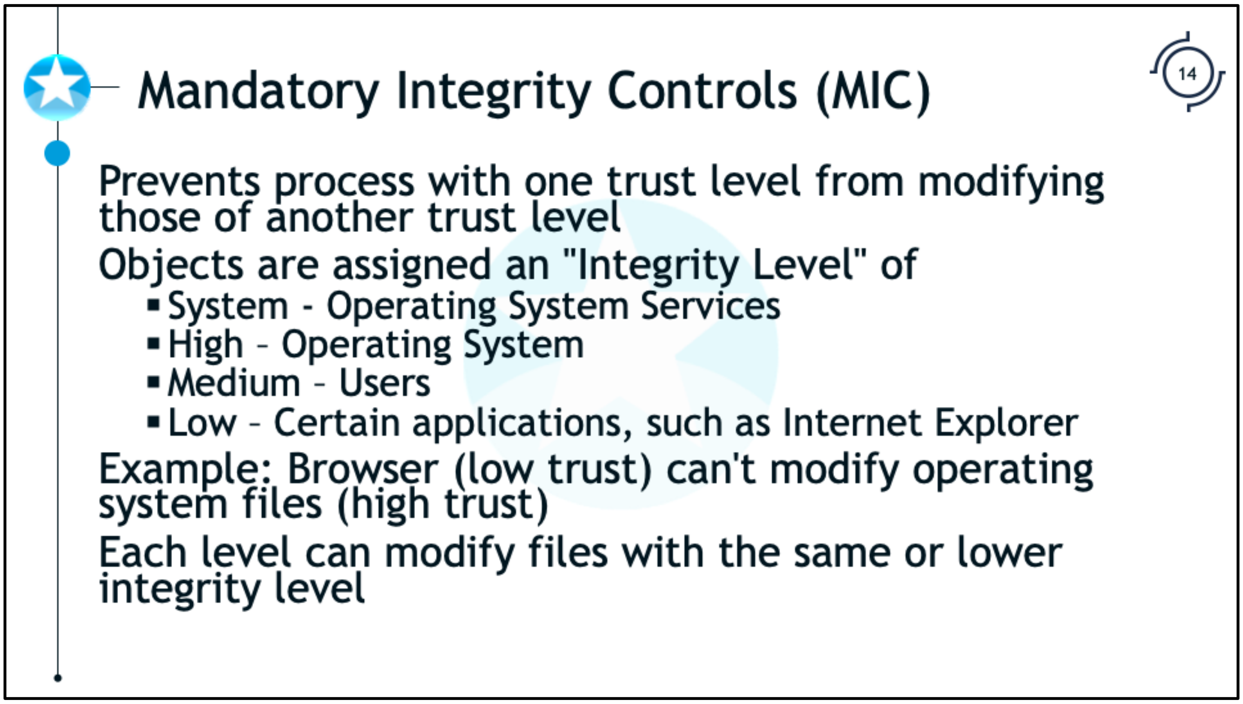
Answer: The file size of c:\main.txt did not change.

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)

Prevents process with one trust level from modifying those of another trust level

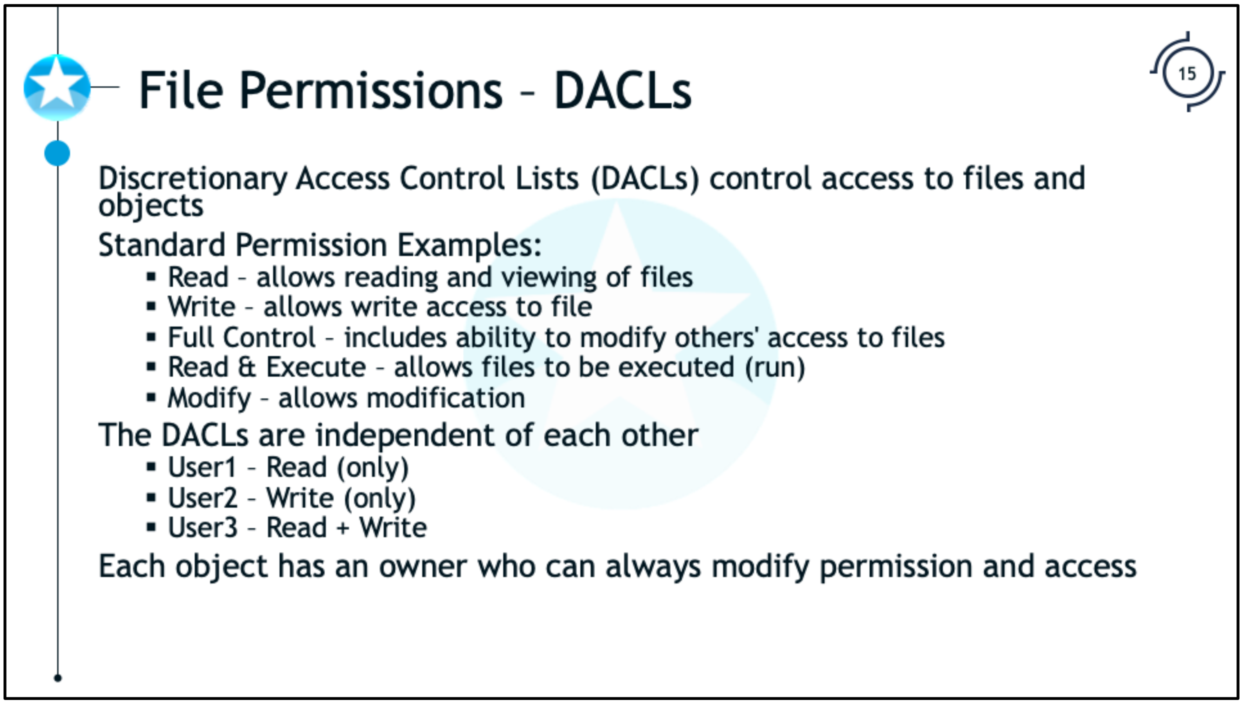
Objects are assigned an "Integrity Level" of

- System - Operating System Services
- High - Operating System
- Medium - Users
- Low - Certain applications, such as Internet Explorer

Example: Browser (low trust) can't modify operating system files (high trust)

Each level can modify files with the same or lower integrity level

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. Operating system services operate with the System "Integrity Level." Unprivileged users are assigned an "Integrity Level" of Medium or Low. Administrative users are assigned an "Integrity Level" of High. 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

Discretionary Access Control Lists (DACLs) control access to files and objects

Standard Permission Examples:

- Read - allows reading and viewing of files
- Write - allows write access to file
- Full Control - includes ability to modify others' access to files
- Read & Execute - allows files to be executed (run)
- Modify - allows modification

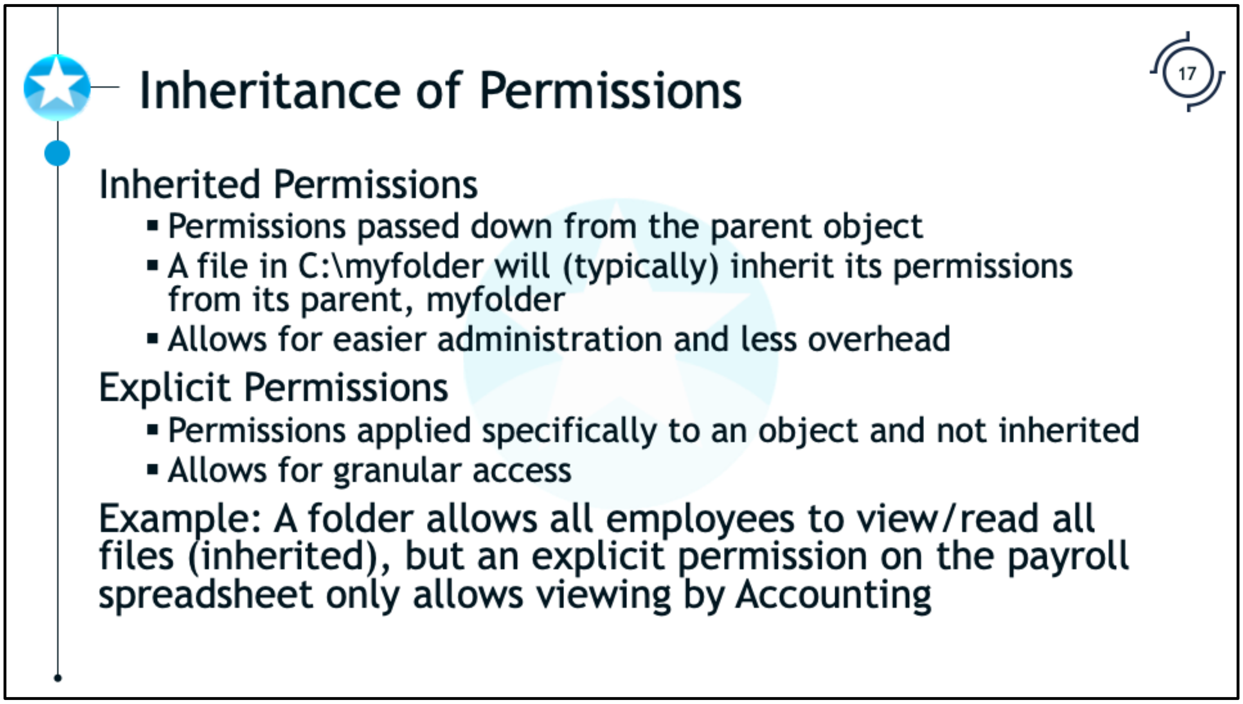
The DACLs are independent of each other

- User1 - Read (only)
- User2 - Write (only)
- User3 - Read + Write

Each object has an owner who can always modify permission and access

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



Inheritance of Permissions

Inherited Permissions

- Permissions passed down from the parent object
- A file in C:\myfolder will (typically) inherit its permissions from its parent, myfolder
- Allows for easier administration and less overhead

Explicit Permissions

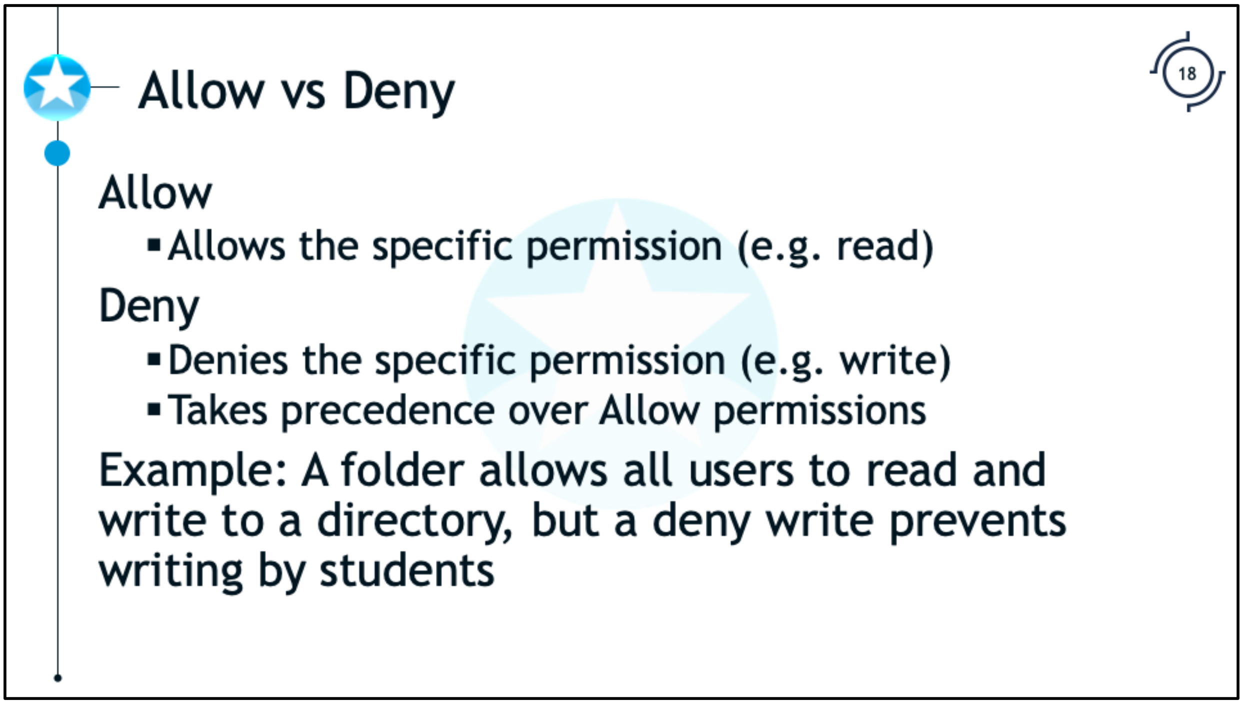
- Permissions applied specifically to an object and not inherited
- Allows for granular access

Example: A folder allows all employees to view/read all files (inherited), but an explicit permission on the payroll spreadsheet only allows viewing by Accounting

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

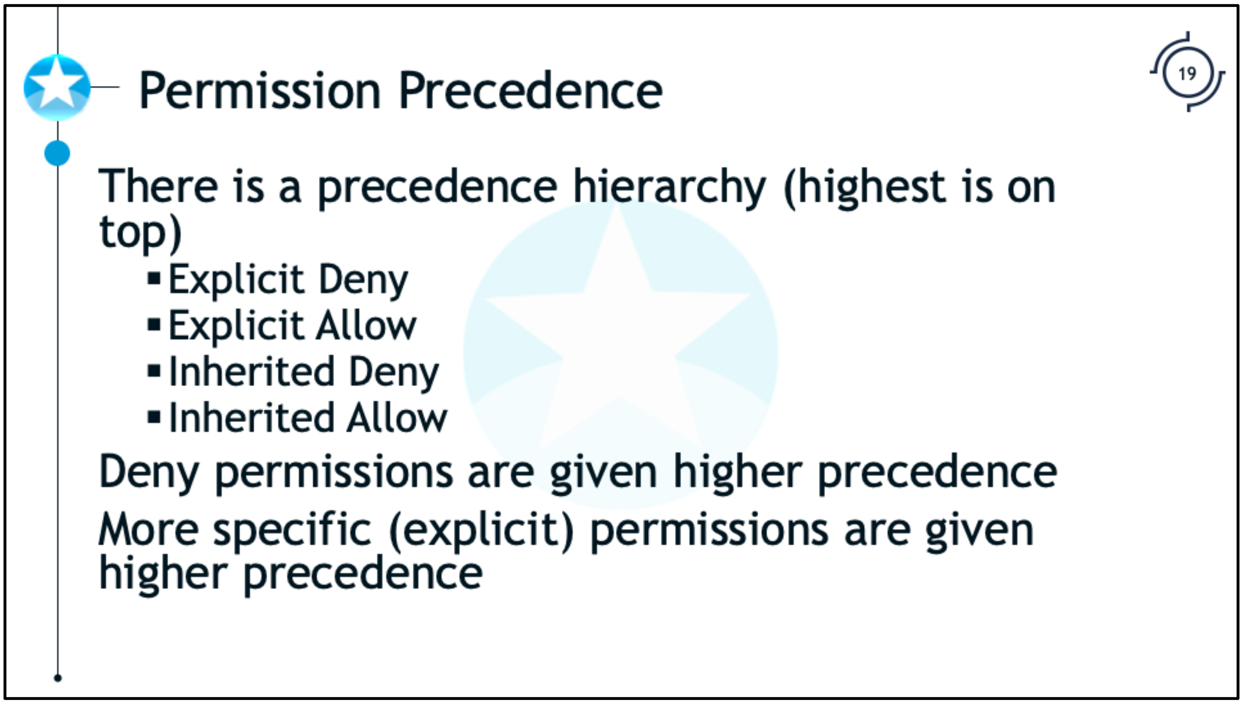
Allow
- Allows the specific permission (e.g. read)

Deny
- Denies the specific permission (e.g. write)
- Takes precedence over Allow permissions

Example: A folder allows all users to read and write to a directory, but a deny write prevents writing by students

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. The administrator can set a deny write permission for students. This is simpler than adding each group that isn't a student and providing write access to all of them.



Permission Precedence

There is a precedence hierarchy (highest is on top)

- Explicit Deny
- Explicit Allow
- Inherited Deny
- Inherited Allow

Deny permissions are given higher precedence

More specific (explicit) permissions are given higher precedence

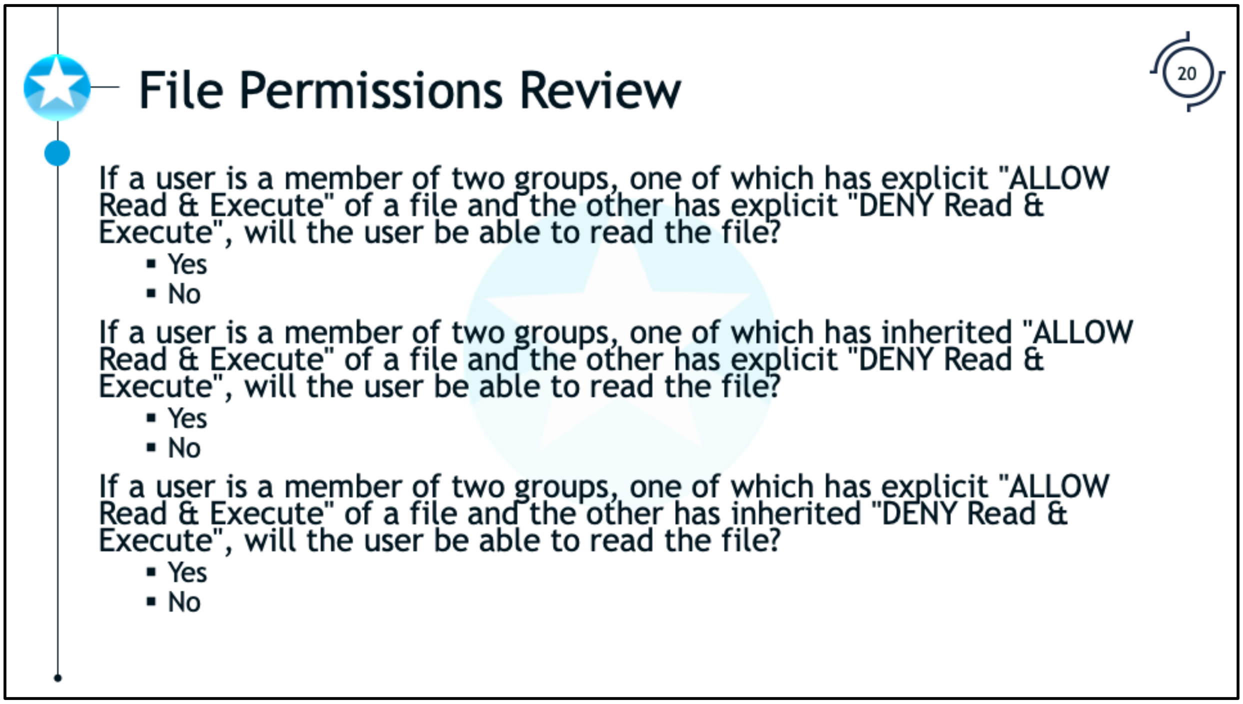
Deny permissions are given a higher precedence than allow permissions. Explicit permissions are given a higher precedence than inherited permissions.

Consider the following example:

```
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 Accounting Group
 |--EngineeringFolder – Write access for users in Engineering Group
 +-MarketingFolder – Write access for users in Marketing Group
```

In the above example, 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 could be added, 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 permission 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 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?

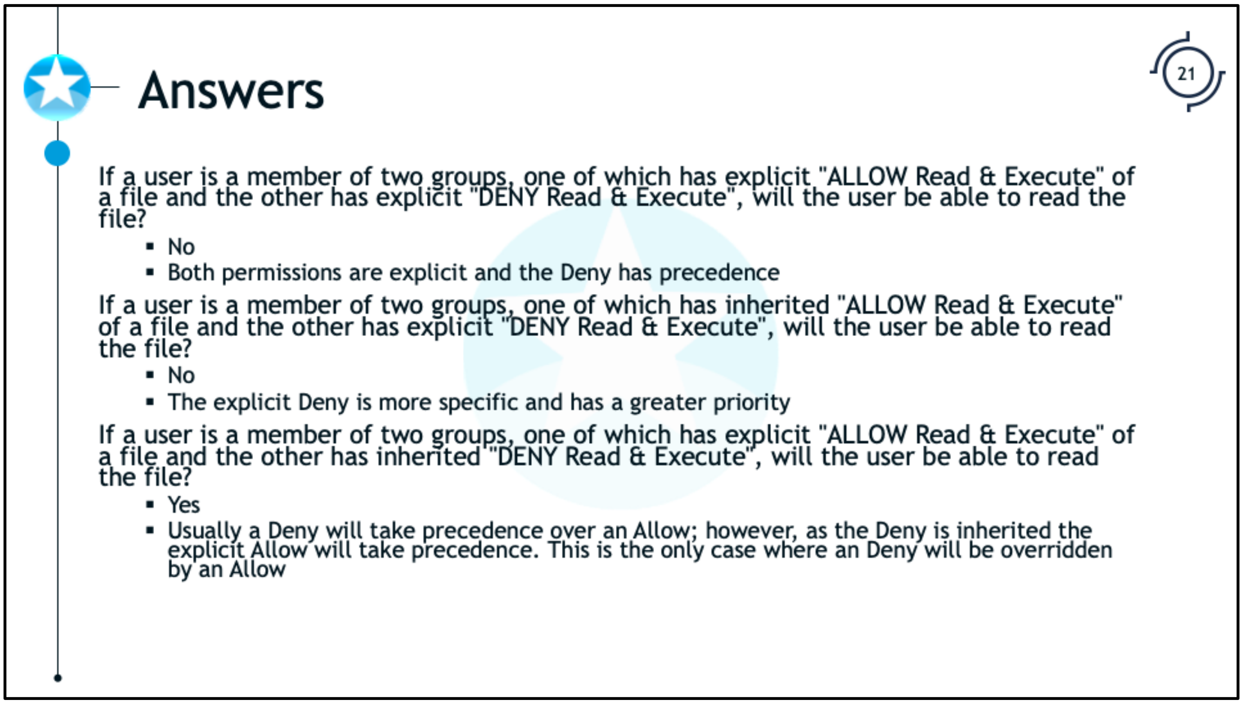
- Yes
- No

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



Answers

21

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?

- No
- 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?

- 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?

- 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

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?

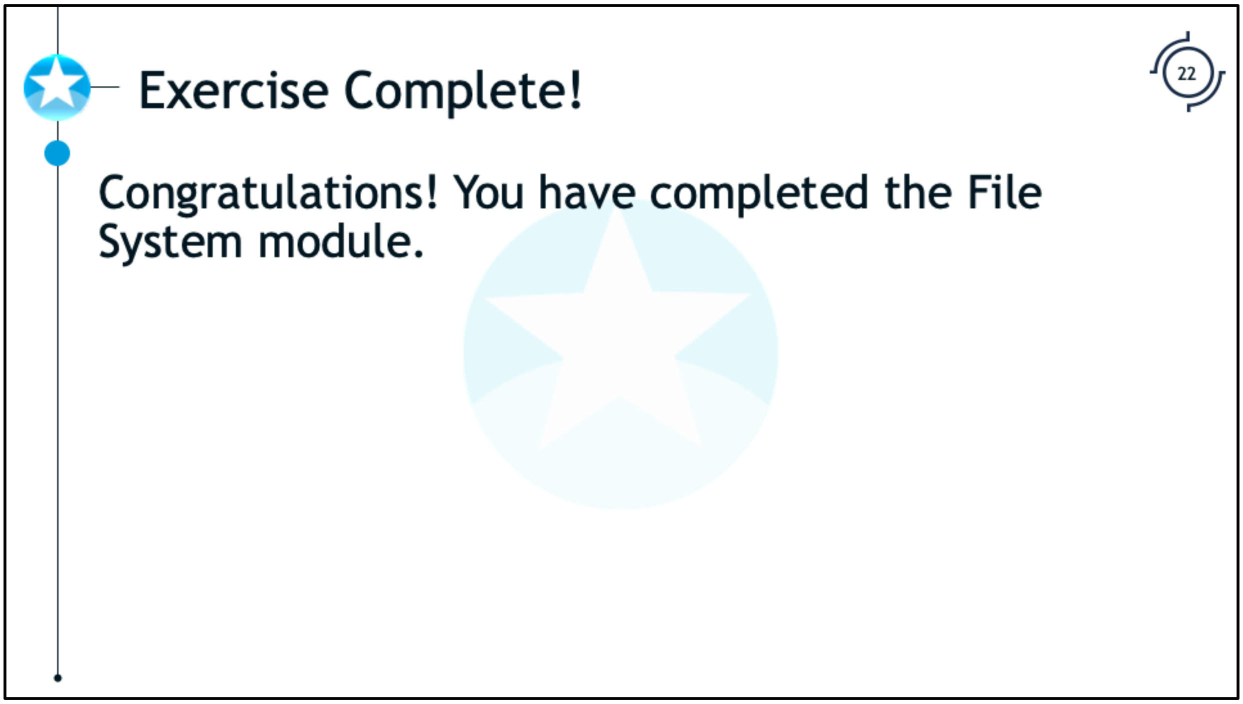
Answer: No, as both permissions are explicit and the Deny has precedence

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?

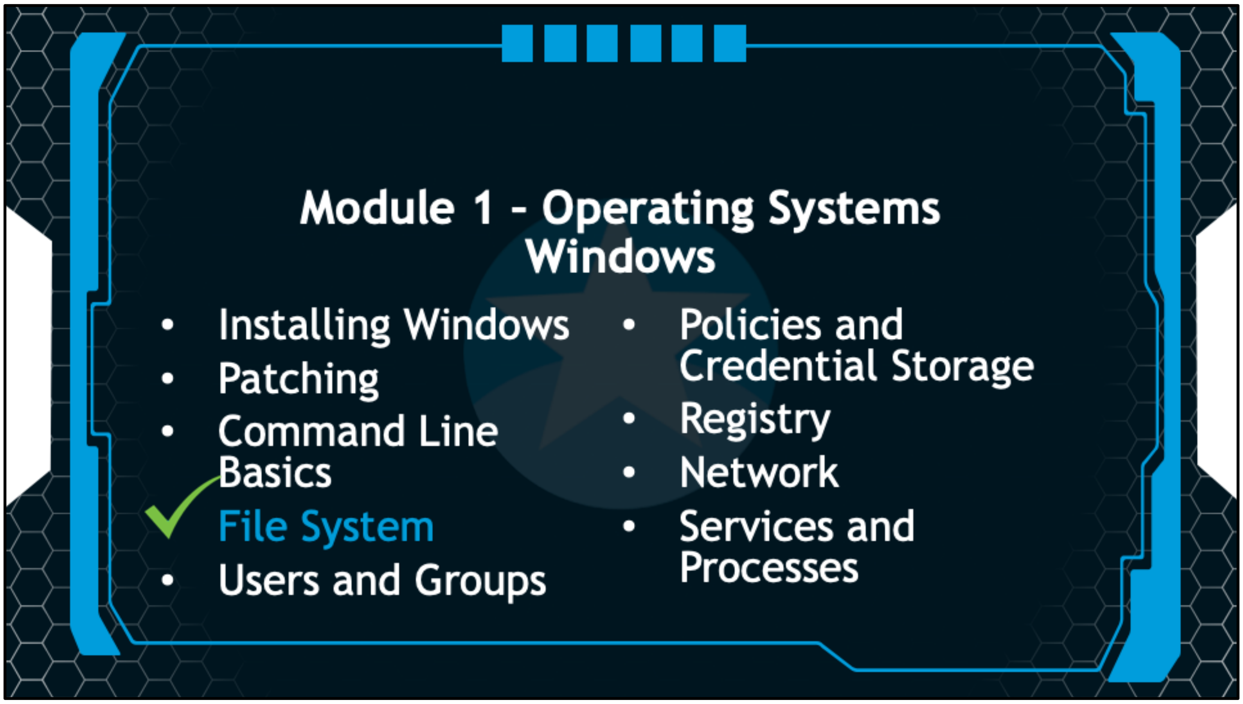
Answer: Yes. Usually a Deny will take precedence over an Allow; however, as the Deny is inherited the explicit Allow will take precedence. This is the only case where an Deny will be overridden by an Allow



Exercise Complete!

Congratulations! You have completed the File System module.

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



Module 1 - Operating Systems Windows

- Installing Windows
- Patching
- Command Line Basics
- ✓ File System
- Users and Groups
- Policies and Credential Storage
- Registry
- Network
- Services and Processes

In the next session we will discuss Windows users and groups.