Computer Security

PENN FOSTER

PC Operating Systems : Computer Security

Lesson 5 Overview

This lesson focuses on the idea of computer security: what it means, core concepts, threats to data and device security, and best practices for minimizing exposure. Topics discussed here include permissions, users, groups, and User Account Control (UAC).

5.1 Explain how to share a Windows device securely; configure permissions, users, and groups; and share devices securely using policies and UAC

Authentication, Permission, and UAC

READING ASSIGNMENT

Read this assignment. Then, read Chapter 13 of your textbook.

Authentication with Users and Groups

No matter your level of experience with computers, you've almost certainly had to use some level of security in your daily life. In practical terms, computer security begins with a **user account**, a unique combination of a username and an associated password that provides the identified user with a certain level of access to a system. User accounts are common across most sectors of society, ranging from banking to Netflix to shopping, in addition to their usage in computers. User account security is based on two mechanisms: authentication and authorization.

Authentication is the identification and granting of access to a system based on user input or properties. Windows handles authentication through password-protected user accounts. **Authorization** is the process of defining what resources an authenticated user can access and what that user can do with the accessed resources. Windows authorization is handled by the NTFS file system, which is responsible for assigning permissions to users and groups. Note that while the lesson discusses how Windows works with these mechanisms, all current OSs utilize user accounts, passwords, and groups.

As stated previously, a user account has a username and password. The **username** is a text string that identifies the account on a system, while the **password** is a unique key assigned to that username. Both pieces of information are encrypted on Windows systems and stored as a **local user account**, which is specific to the machine. A **global user account**, which has different names depending on the provider, synchronizes certain settings so that a user can log in anywhere and have access to the same settings and data. Depending on the OS, a user can be restricted to one type or the other, or have access to both kinds; Windows allows both, for example, while Linux requires local accounts.

When securing accounts through passwords, there are certain guidelines to keep in mind, many of which are (or should be) obvious.

- Never give out passwords over the phone; in general, don’t give them out at all.
- Make strong passwords, meaning at least eight characters long, and including numbers, letters, and symbols.
- Change passwords regularly, preferably every 90–180 days.
- Never use a blank password.
- Make sure not to allow passwords, if written down (not a good idea in itself), to be left out in public view.

In many cases, the organization that holds the user account will have policies in place defining these specific aspects. As a technician, you may not have the opportunity to create such policies, but if you do, make sure they include these guidelines.

User accounts are generally collected into groups, which define what the member accounts are allowed to do on a system or with a resource. An account can belong to multiple groups, as can computers in a network. Having groups allows for simpler and more efficient administration of users and resources, as it provides a centralized mechanism for applying policies and access rules. Windows allows multiple predefined groups with access levels in place. Although these are handled differently in Home editions of Windows, all versions have the following predefined user groups:

- **Administrators** have complete administrative privileges and thus complete control over the machine. The primary user of a machine is often a member of this group, as is the default admin-level Account Administrator. Best practice for this account is generally to create a complex password for the account, store it somewhere safe, and change a different user account to be the default admin-level account.

- **Power Users** have near-administrative privileges, but can't install new devices or access other users' resources unless specifically given access.
- **Users,** also known as **standard users**, can't edit the Registry or access critical system files, but can create groups and manage the groups created by that account (but only those groups).
- **Guests** can access basic functionality and the internet. This level of user is generally disabled.

In most professional environments, the typical Windows setup is to have a single standard user account be the primary user and a local administrator account for updates, installing and removing software, and other higher-level tasks. To perform higher-level tasks on a machine, a technician could log into the machine as the administrator or have a temporary account created with administrative privileges that's deleted once the tech's tasks are finished. But it's generally quicker to right-click the icon or utility to run, select **Run as administrator** from the context menu, and enter the credentials when prompted by the User Account Control (UAC) dialog box. Alternatively, using the User Account Control applet in Control Panel can temporarily elevate privileges as needed. This level of control is useful when creating or modifying users and groups through the Local Users and Groups applet in Control Panel; for greater detail on this tool and its various past iterations (such as Windows 7’s User Accounts), including screenshots, see the “Configuring Users and Groups in Windows” section of Chapter 13.

Authorization Through NTFS

The process of authorization, to define what the account can do with the various resources it can access, happens after a user account is created and a password is associated. Authorization uses NT File System (NTFS) permissions, which are rulesets built into the file and folder structure of the drive, to decide what can be done. NTFS permissions can be very granular and are extremely powerful; for the 1002 exam objectives, however, it’s most important to understand these concepts about NTFS permissions:

- **Ownership** defines who or what created the file or folder. The owner of a resource has total control over the resource, which includes changing permissions to keep any other account (even administrators) from accessing the resource.
- **Take Ownership** allows whoever has this permission to take control of a resource, even if that account can’t otherwise access it. Administrator accounts have this permission for all resources on a system.
- **Change** allows an account to provide or remove permissions for other accounts.
- **Folder permissions** define what a user or account can do to a folder, such as listing its contents or changing the folder name.

- **File permissions define** what a user or account can do to an individual file, such as read the file or run it if it's an executable file.

NTFS permissions are set through the Security tab of the Properties dialog box for a file or folder, as shown in Figure 13.19. The standard NTFS permissions for folders are as follows:

- **Full Control** allows complete control over a resource.
- **Modify** allows reading, writing, and deleting of files and subfolders.
- **Read & Execute** enables viewing of folder and subfolder contents and running of any executable files or associations in that folder.
- **List Folder Contents** allows viewing of folder and subfolder contents.
- **Read** enables viewing of folder contents and opening of any file in the folder.
- **Write** enables writing to files and creating new files and folders.

For files, the NTFS permissions are similar:

- Full Control
- Modify
- Read & Execute
- Read
- Write

NTFS permissions are assigned to both user accounts and groups; it's considered a best practice to assign them to groups, however, to avoid conflicts and other issues. Permissions are **cumulative**, meaning that if an account has more restrictive permissions on a file in a folder than it does on the folder, the account has the less restrictive permissions on the file as well. The creator of a file or folder has ownership permission on that resource, but administrators can take ownership of that resource.

One fundamental concept of permissions that you should understand is **inheritance**, which determines what permissions a newly introduced or created resource in a folder receives. The default setting in Windows is that any new file or folder placed in a folder gets the NTFS permissions of the parent folder; this can be overridden on an individual basis by using the **Deny** checkbox in the Properties dialog box, as shown in Figures 13.22 and 13.23. Note that inheritance isn't the same as **permission propagation**, which can differ depending on how a file or folder is placed into a folder. Whether permission is inherited or propagated depends on whether the resource was **copied** (meaning the original stays in place) or **moved** (meaning the resource is in a new, singular location) and whether it's coming from the same volume (partition). See Table 13.1 for a breakdown of the potential propagation, as well as the text preceding it for an explanation of why. Note also that since FAT doesn't support NTFS permissions, any resource copied or moved to a hard drive or USB drive (whether it uses FAT32 or exFAT) loses its permissions.

Linux and macOS also use permissions and treat them similarly, although the commands used to modify them are different. Modifying permissions requires familiarity with the command line for both Linux and macOS machines. For a breakdown of how these OSs use permissions, as well as the command syntax used to analyze and change them, see the “Permissions in Linux and macOS” section of Chapter 13.

Sharing Resources Securely

NTFS permissions make granular control over files and folders possible, but this can make sharing resources more complicated as well. Different versions of Windows handle sharing in slightly different ways; Windows 7 made sharing simpler through public libraries for certain files, for example, but those libraries are invisible by default in Windows 8 and newer. By default, resources created by a user on a machine are visible only to that user (and administrators), so successfully sharing resources takes a few extra steps.

Windows primarily allows sharing by giving users—or more commonly, groups—access to specific files and folders through the use of **specified permissions**, which are accessible through the Properties dialog box of the resource. Another way to go is to use the Sharing Wizard, which doesn’t allow the same granularity of control but is quicker and easier. The Sharing Wizard allows users to search for specific accounts on a domain if the machine is joined to a domain. See Figures 13.25 to 13.31 and associated text for details and screenshots on how to use this function.

If there are shared folders on a machine already, or if you need to check for any shared folders you may not know about—always a good check for security reasons—you can use the Computer Management applet in

Control Panel. Under System Tools, there's a **Shared Folders** option that will display all the shared folders on a machine, as shown in Figure 13.32. In addition to folders that were shared by you or another user, administrative shares will be visible also. These are default shared resources that allow local administrators to access these system resources locally or remotely. These administrative shares include all hard drives on the machine and the %systemroot% directory, among others. They're hidden from normal usage but can be mapped, and they're indicated by a dollar sign in the name, such as **ADMIN$**.

Part of sharing resources securely is ensuring these resources are safe from unauthorized access, and one of the strongest tools to do that is encryption, as even strict permission levels can be circumvented by administrator-level accounts. Not all versions of Windows support encryption—Windows Home has very little in the way of security tools, for example, and doesn't support encryption—but professional and higher versions allow for different native tools. Professional versions of Windows support the **Encrypting File System (EFS)**, which allows any user to encrypt files and folders through the Properties dialog box from the right-click context menu. Note that although any user can encrypt files with EFS, the encryption key is based on the password for the account, so if the user loses the password or it's reset, any encrypted files are unretrievable. Pulling the hard drive out of the machine and installing it on a different computer won't help, as the system ID will be different. If using EFS for encryption, it's vital to have a valid password reset disk for disaster recovery.