Chapter 5 Managing a Server

In this chapter, you will:

- Understand the Web server administrator's view of server management
- Examine networking models
- Learn how users are authenticated
- Manage users and groups
- Manage file system permissions
- Share resources in a network
- Enforce network policies

Because Web server administration is a part of network administration, you should understand the basics of server administration in a LAN environment and network issues in a typical organization. To help you plan the structure and functioning of a network, you use networking models that guide you in determining how users and computers work together.

Managing a server primarily involves controlling access to resources such as files and printers. You can control two areas: users' access to the server, and what users can do once they access the server. You can control these areas for users originating from the Internet as well as users on the LAN. One way to manage user access is through user authentication, which makes sure that only valid users gain access to the server. Although both Linux and Windows share the same objective of controlling access, they implement security features differently. The Windows operating systems are rich in LAN management capabilities, whereas Linux focuses on the advantages of a multiuser server. Although this chapter mainly focuses on the LAN user, most of the concepts discussed here apply to controlling Web access, too.

UNDERSTANDING THE WEB ADMINISTRATOR'S VIEW OF SERVER MANAGEMENT

When you learned about servers in Chapter 2, the focus was on the hardware—the computer and its capabilities. In this chapter, the focus is on the server software that allows you to manage server resources. When sharing files on a Windows network, this kind of software is an integral part of the operating system. When providing a Web page, the Web server software is a separate product that works

with the operating system. The server computer can run more than one server software product. As a consequence, you can use the same computer as a Web server and as your e-mail server and FTP server. Before you as the Web administrator can allow access to the Web server, e-mail server, and other Web applications, access controls need to be in place. In many organizations, the Web server is not isolated from the LAN. Users on the LAN might be responsible for updating Web pages for their departments. You might therefore need to handle a variety of LAN administrator tasks such as sharing folders that reside on your Web site and setting permissions to control access to the folders. The principles involved in controlling access from a LAN and from the Internet are similar. You need to make sure that the users who gain access to your server are valid. The LAN operating system, in fact, is designed to ensure that users are valid. It is your job to match the users with the resources they need. When controlling access from the Internet, you have other options, however. Applications such as e-mail typically are based on user accounts that are part of the operating system. Other applications, such as ones requiring membership to a Web site, often rely on a database of user names and passwords that might also contain other information about the user. If your Web site depends on a database of user information, user names cannot be used to penetrate the operating system, although storing user names and passwords in a database is not as secure as storing them in the operating system. You need to explore the capabilities of the LAN environment to decide which approach is best in your case.

Virtually all LAN workstations are Windows-based. The dominance of Windows in LANs means that Microsoft alone can determine how computers communicate in the LAN environment. As you will learn, the way computers communicate with a Windows server in a LAN environment is very different from the way they communicate with an Internet-based server such as a Web server. In contrast, all computers communicate with a Linux server in a similar way, whether they are part of a LAN or connected only by the Internet.

EXAMINING NETWORKING MODELS

Because a server is part of a network, you should understand how a server fits into the network before you can consider such issues as users, resources, and control. In this section, you examine two approaches to networking models. The first is the Microsoft LAN approach to networking. The second is the client/server approach to networking, which is the basic model used for Web and e-mail servers. Whereas Microsoft does employ the client/server model for some network tasks, Linux primarily uses this model for all functions.

Microsoft LAN Networking Models

You configure a Microsoft Windows LAN using one of two networking models: the workgroup or the domain. The model determines how users are organized. The workgroup networking model, also known as the peer-to-peer model, considers each computer as an independent entity. Any access to resources on a computer depends on local user accounts. The domain model, on the other hand, centralizes users and the control of resources. Note that Microsoft's definition of a domain in a LAN is not related to an Internet domain.

Workgroup Networking Model

The workgroup networking model treats each computer in the network as an equal, or peer. This model does not use a centralized server. Instead, each computer acts as both a server and a client. When you allow other users to access resources on your computer, your computer is acting as a server. When you

access resources on another computer, your computer is acting as a client. Because each user's computer acts as a server, each user must therefore be an administrator.

This decentralized approach has several disadvantages. First, most users are not interested in learning about administration. Second, because each computer must have a complete list of user names and passwords of other users wanting to access resources on the computer, security is compromised. It is also difficult to keep track of changing passwords. When a user changes his or her password, the password must be changed on the other computers that the user accesses. Because of these limitations, the workgroup networking model is best suited to small networks consisting of up to ten computers for which security is not a major concern.

Figure 5-1 shows an example of a workgroup and its limitations. The user Mary Noia (mnoia) can access the printer attached to Bob Cabral's (bcabral) computer only if bcabral does certain things. First, bcabral has to add mnoia's user name and password to the list of users. Second, bcabral must share the printer. Third, bcabral must specifically allow mnoia to use the shared printer.

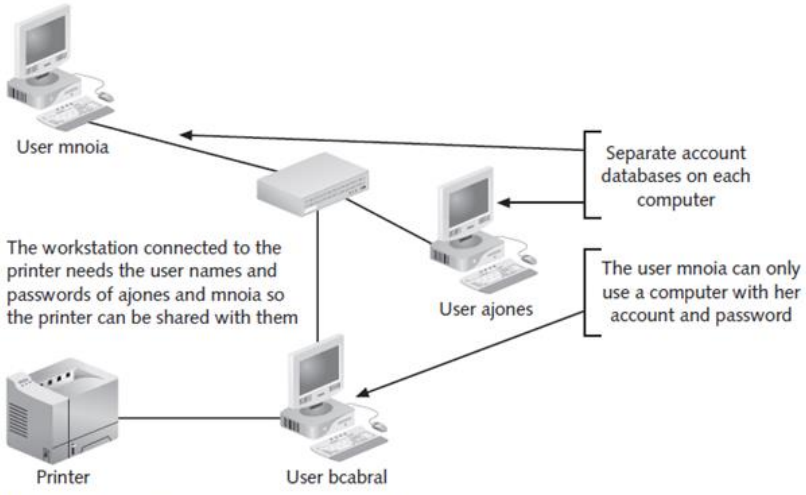


Figure 5-1 Workgroup networking model

Domain Networking Model

The domain networking model uses one or more servers to centralize control. Instead of each computer and user being independent, they are all part of a domain. This centralization allows an administrator to use a single point of control. With a single logon, the user can be given access to any resource in the domain, as shown in Figure 5-2. Because the user name and password are stored only once, changing the password does not have the same negative effects that it would have in a workgroup network. In a workgroup network, if you changed your password on your computer, you would need to provide separate credentials to access resources on other computers. Microsoft suggests that you use the domain networking model for networks with more than ten computers, which makes it the dominant networking model for LANs.

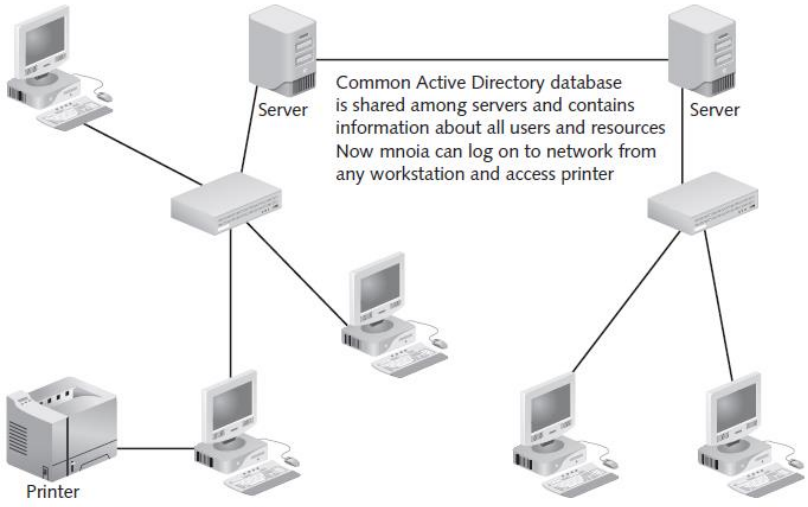


Figure 5-2 Domain network

Client/Server Networking Model

In the client/server networking model, the client represents a program such as a browser that accesses resources. The server is a program such as a Web server that provides resources. The client and the server communicate using a protocol. You already know that TCP/IP is a protocol suite, and that one of the protocols in this suite is HTTP. Browsers and Web servers use HTTP to communicate. Networking in Linux uses the client/server model. Each computer has its own database of users and passwords. If you want to use a resource on another computer, you must supply a user name and password on that computer.

You can designate one computer to act as a server and store hundreds or even thousands of user accounts on it. On the client computer, you can use a program such as SSH (Secure Shell) to log on to the server. Using this configuration, everyone can share the applications and resources on that centralized computer. A big difference between the client/server approach and Microsoft's domain model is that Linux typically lets you access only a single computer at a time. In Microsoft, your identity can be carried throughout the network to allow you to access any resources.

Even when you use a Web browser to connect to a Web page on another computer, you first have to log on to that computer. Typically, the Web server provides a guest account that it uses automatically when someone wants to access a Web page. A guest account is a very restricted user account that can access only resources related to the Web site. A Web server requires an account to be associated with anyone who accesses the computer.

Access to a server is controlled by a program that runs in the background, called a **service** in Windows and a **daemon** (pronounced "demon" or "daymon") in Linux. Multiple services can run simultaneously, such as a Web server and an e-mail server. Client programs are designed to access the service. For example, a Web browser connects to a Web server, an e-mail client connects to an e-mail server, and a SSH client connects to an SSH server, even if both reside on the same computer.

To allow all these server services to run on a single computer, the server uses **port numbers** to distinguish them. You can imagine ships sailing to specific ports in a certain city, where each port has a specific purpose and can handle only certain types of cargo. Likewise, a packet of data from a Web browser goes to a specific port on the Web server, while a packet of data from an e-mail client goes to a different port. For example, the default port for a Web server is 80, the port used to send e-mail is 25, and the port used for DNS is 53. Each server application listens at a port, waiting for packets of data destined for it. The server sends data back to a client in a similar manner, with client programs using ports to accept data from various servers. A detailed understanding of these concepts is critical in designing a secure environment for your Web server, e-mail server, and other related applications.

AUTHENTICATING USERS

Authentication is the process of determining a user's true identity. That is, when you log on to a network or supply user information to a Web site, authentication is how the system verifies whether you are who you say you are. Authentication involves two processes. First, you need a mechanism such as user names and passwords to identify users. Second, you need to know how secure the process is to get the identification information to the server. A complex password may not ensure security if it is sent to the server in such a way that it is easy to intercept.

Identifying Authentication Methods

Networks can employ three methods to authenticate users. These methods can be used alone or in combination with other security-related methods. **Multi-factor authentication** is more secure because multiple authentication methods are used. You can prove your identity by using the following methods:

- What you know
- What you have
- Who you are

The following sections examine these authentication methods in more detail.

What You Know

This method is the most common form of authentication, and it typically uses passwords. That is, when you log on to a computer network, you are prompted to type a password that you have chosen or that a network administrator has given to you. The password is *what you know*. The computer bases its authentication on this password by checking it against a list. If the password you typed matches a password on the list, you are allowed access to the system. If it does not, you are locked out.

If you give your password to someone else, the computer grants this other person access because the authentication scheme is based on knowing the password. In this case, the security measure has not failed—the other person gained access by a legitimate authentication method.

What You Have

This method requires that you use a physical item, such as a key, for authentication. An example would be an entry card that you insert into a card reader to gain access to a room or building. Anyone who runs the card through the reader is granted access to the building. In this case, the authentication is based on *what you have*.

Of course, if someone takes the card from you, he or she can enter the building even though the card was originally given to you. Therefore, to create a more sophisticated authentication system for entering the building, an administrator may require not only a card, but also a password. Taking both a card and a password from someone is more difficult. ATM cards, which use personal identification numbers (PINs), are based on this combination of what you have and what you know.

Smart card logon contains information that provides the most secure logon procedure—namely, encrypted codes that uniquely identify the user. Putting this information on a card is much more secure than putting the information on a computer, because a computer is more readily accessible.

Who You Are

Biometrics is the science of connecting authentication schemes to unique physical attributes. Examples of this method include using fingerprints, visual and photographic identification, and voice recognition. Each method attempts to validate an individual's claim concerning his or her identity by verifying a specific physical characteristic. These *who you are* methods of authentication are becoming increasingly common as the hardware verification tools become less expensive and the recognition tools are built into operating systems.

Each of the three authentication methods is used in systems today, either individually or in combination with the others. How they are implemented varies from system to system.

Implementing an Authentication System

Kerberos is an authentication system developed at the Massachusetts Institute of Technology (MIT). It is designed to enable two parties to exchange private information across an otherwise open network. Kerberos works by having an authentication server assign a unique key, called a ticket, to each user who logs on to the network. The ticket is then embedded in messages to identify the sender of the message and is used to grant access to other resources. Many implementations of Kerberos are available, including a free implementation available from MIT at *http://web.mit.edu/kerberos/www*. This site provides more detailed information about Kerberos.

Kerberos has been implemented in Windows as the authentication protocol in a domain environment. It is designed specifically for authenticating users who have accounts in the operating system. Windows has another mechanism, called **certificates**, for authenticating users over the Internet. Certificates guarantee the identity of an organization or user.

MANAGING USERS AND GROUPS

Users need accounts to access resources on a server. Even when the resource is a Web page, the Web server has a default user account that it uses on your behalf. This default account has restricted access, but at least allows you to view the Web page. In a LAN environment, access to resources such as printers and files are controlled based on user accounts. If a number of user accounts have common resource needs, the administrator can organize them into groups. For example, suppose everyone in the accounting department needs access to the accounting software and the printers in the accounting department.

All of these users could be organized into a single group. The users are members of the group, and the administrator gives the group access to the resources. As resource needs change, the administrator simply modifies the group's access instead of setting new access limits for each member of the group. This section discusses adding users and groups; the next section focuses on giving the users and groups access to resources (that is, managing file system permissions).

Identifying Special Accounts

Applications that operate as a service need to use accounts to perform work tasks. For example, assume that you have a DBMS such as MySQL on Linux or SQL Server on Windows. Even though your personal user account does not have the access necessary to modify the physical DBMS files that exist on the computer, when you use the DBMS, the DBMS has the ability to modify its own files. The DBMS is associated with an account.

Windows has a special **system account**. The system account represents the operating system and has many of the same privileges as the administrator. This powerful account is a favorite target of hackers. If the hacker's program becomes associated with a system account or the hacker can manipulate a program associated with a system account, the hacker has almost complete access to the computer. When you install SQL Server, for example, you have the choice of using the system account or a user account that is created especially for operating under it. Normally, you would use the system account. However, you should use a special user account if you have multiple computers with SQL Server and they merge data. Services such as the Web server use special, highly restricted accounts. When you install the IIS Web server in Windows, for example, a guest user account is created to permit **anonymous logon** to the Web server. Basically, this user has only permission to read Web pages.

Be careful about deleting user accounts that you do not recognize. They could be special accounts that are used by applications such as the Web server or DBMS. By default, these special accounts are not displayed.

Linux implements daemons in a different way. Although Linux does not have a system account, it does have the **root account**, which is similar. The root account has full access to everything in Linux. Even though programs could run as if they were root, the standard in Linux is that each daemon is associated with an individual account that restricts the daemon to specific directories and files.

Understanding Users and Groups in Windows

You need to create user accounts for individuals, and in some cases for applications, that need access to your resources. Windows has two types of user accounts: **local accounts** and **domain accounts**. When you create a local account on a computer, it exists only on that computer and can be used to control access to resources on only that computer. When you create a domain account, it is recognized throughout the whole domain. Recall that a **domain** is a logical grouping of computers that administrators use to organize common resource needs. A domain user can access resources on any computer in the domain.

Understanding Local User Accounts

Web servers typically have local user accounts. When you view a Web page or use FTP to upload or download files, you are doing so while logged on as a local user. You would add a local user if you wanted to give individuals the ability to use FTP to upload files to their Web directory. Although some user accounts are necessary, you should add only the ones that are absolutely essential to a server that is connected to the Internet. After all, the more users you have, the more opportunities hackers have to gain access to your system.

The user name is the account name that users need to log on to the computer. When you set up your user accounts, establish a consistent naming convention. For example, you could use the first letter of the user's first name plus the last name. A user named Cristina Salinas would then have the user name csalinas. Although the steps to add a user appear later in this section, Figure 5-4 shows the principal dialog box involved.

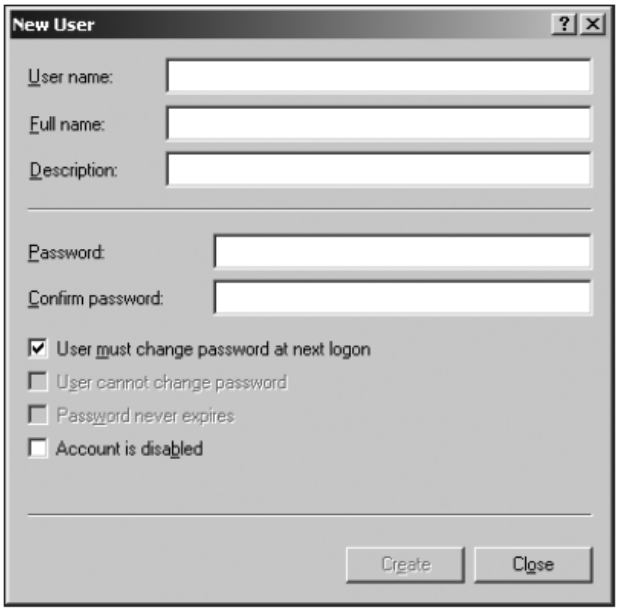


Figure 5-4 New User dialog box in Windows 2003

Understanding User Passwords

In the New User dialog box, the Full name and Description text boxes are optional, but are useful to help you remember the details of the account. The password can be up to 128 characters, and you can use all the symbols on a keyboard. Make sure that passwords are difficult to discover because hackers (and even coworkers) can easily obtain programs that can find passwords. The more complex the password, the more difficult it is to discover. Here are some rules to follow when creating passwords:

- Include at least eight characters.
- Use a mix of uppercase and lowercase letters and numerals.
- Use non-alphanumeric symbols.
- Do not use a recognizable word as part of the password.

To create a complex password that is easy to remember, start by thinking of an easily remembered sentence such as, "I really want to go to San Diego." Replace any occurrence of "I" with a 1. Replace a "to" with 2. Alternate uppercase and lowercase letters. Add one or more symbols. Now you have: 1Rw2G2sD:). This password would be extremely difficult to crack.

Some security researchers recommend using larger pass phrases because they are resistant to hacking and easy to remember.

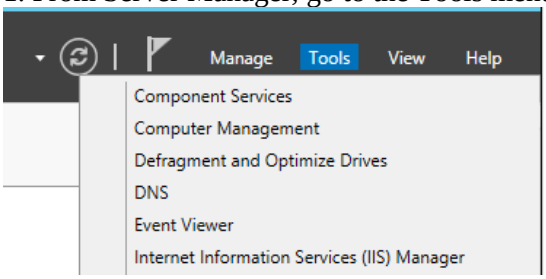
The four check boxes in the New User dialog box let you control account properties. By default, Windows selects the "User must change the password at next logon" check box. This option forces the user to change the password the next time he or she logs on to one that the administrator does not know.

If you uncheck “User must change password at next logon,” two check boxes become available. One is “User cannot change password.” Select this check box for a common account that is used by more than one person. An administrator is then responsible for changing the password and notifying the users. Select the “Password never expires” check box if you are creating a user that is associated with a service. In this case, you want the service to run without having to change a password periodically. Select the “Account is disabled” check box if you want to suspend the use of the account but do not want to delete it. For example, if a user were taking a six-month family leave, you could disable his or her account. In the following steps, you add a user called ajones with a password of P@ssw0rd.

Activity 5-1 - Adding a Windows User Account

To add a user account:

1. From Server Manager, go to the Tools menu and open **Computer Management**.



2. In the left pane of the Computer Management window, click the **plus sign** (+) next to Local Users and Groups.

3. Click the **Users** folder. The current users for this computer appear in the right pane of the Computer Management window.

4. To add a user, click **Action** on the menu bar, and then click **New User**. The New User dialog box opens. (Refer back to Figure 5-4.)

5. In the User name text box, type **ajones**.

6. For the Full name, type **Arda Jones**. For the description, type **Director of Accounting**.

7. In the Password text box, type **P@ssw0rd**

8. Type the same password in the Confirm Password text box. Because you entered a password that is not secure, make sure the “User must change password at next logon” check box is selected. **Take a screenshot of this step.**

9. Click the **Create** button. Windows creates a new user account for the user named ajones.

10. Close all open windows.

Understanding Domain Accounts and Active Directory Domain Services

Although Web servers often use only local accounts, there are typically other servers in the Web environment that have more complex requirements. For example, you can set up an e-mail server with Microsoft Exchange. Exchange requires the capabilities of the **Active Directory Domain Services (ADDS)**. ADDS allows users to use only a single logon for the whole network. All resources throughout the network are then available to them. ADDS organizes the domains in your network so you can administer them as a whole. It requires DNS, which you set up in Chapter 4.

ADDS is designed for large networks. Any server that has ADDS installed is called a domain controller. Domain controllers share information about the network. If one domain controller cannot be contacted, the other domain controllers can take over its duties.

Computers and other resources in ADDS follow the same naming format as the naming scheme that the Internet uses. That is why ADDS requires DNS. A major planning decision is whether the internal naming scheme, or **namespace**, should match the external namespace. If both are the same, your logon name (internal namespace) would be the same as your e-mail name (external namespace). For example, if your domain is *technowidgets.com*, your user with the account ajones could use *jones@technowidgets.com* for both a logon name and an e-mail name. If you keep the namespaces separate, configuration would be more flexible because for the external namespace, you need to focus on only those computers that will be accessed from the Internet. You could then use a DNS like the one you created in Chapter 4 and it could reside either within your organization or at your ISP.

Once ADDS is installed, you add users with a special administrative tool called Active Directory Administrative Center (ADAC). The process is similar to the one you used to add a new user in the previous steps, except that there are two possibilities for logon names. The logon name looks like an e-mail address, such as *ajones@technowidgets.com* but you can still enter the user name alone, such as ajones.

Configuring Groups in Windows

You use groups to organize common needs among users. Typically, these needs are related to accessing resources such as printers and files. For example, only certain people may require access to a high-speed color printer. You would put the user accounts of those people needing to use this printer into a group, and then you would give the group access to the color printer. You could use the same technique to restrict the actions of a specific group of users while using a Web site.

If you do not install AD, Windows has only one local group, which you use for your local users. With AD, you can assign users to two types of groups. One type is called a security group; you use the groups in this category to assign permissions to users and thereby control access. You use the other type, called a distribution group, for combining users for other purposes, such as e-mailing groups of users.

Multiple ADDS domains can be grouped into a **forest**. Three types of security groups exist. **Domain local groups** can have members from multiple domains, but you can only use such groups to assign permissions to resources in the same domain as the user. **Global groups** can only have members from the same domain, but you can use them to assign permissions to resources in multiple domains. **Universal groups** can have members from many domains, and you can use them to assign permissions to resources in many domains. What a group can have as members varies by group. When you create a user account, the account becomes a member of a group called **Users** if the network has no domain, and a member of the **Domain Users** group if it does have a domain. Both of these groups are built-in groups. That is, when you install Windows, they are created automatically to help with managing users.

Windows has a number of other built-in groups. The most commonly used are Domain Admins and Administrators, for administrators; Account Operators, who can administer user accounts; and Server Operators, who can shut down the server; Backup Operators who can back up data and restore data.

Understanding Linux User and Group Accounts

Setting up user accounts in Linux is simpler than the equivalent process in Windows. Like Windows local users, Linux users have permissions only on the computer where the user account is created. Linux offers two ways to create users. You can use the command-line utility useradd, which is found in all Linux distributions. Alternatively, you can use a GUI tool User Accounts.

When you add a user in Linux, you specify the properties of the user account. Table 5-1 lists these properties.

Table 5-1 Properties of user accounts in Linux

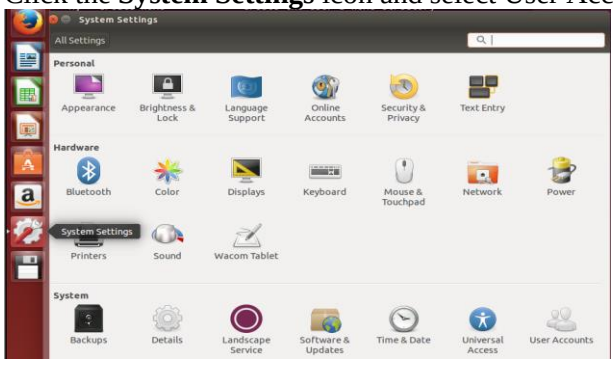
Item	Description
User name	Logon name of the user
Full name	The full name of the user or any comment
Password	The password must be at least six characters
Home directory	The default is /home/*username*
Group	The default is to create a group with the same name as the user
Login shell	The default is /bin/bash, which determines the characteristic of the shell environment

In the following steps, you will set up a new user account in Linux for a user named Mary Noia.You will use mnoia as her user name and Azore$ as her password.

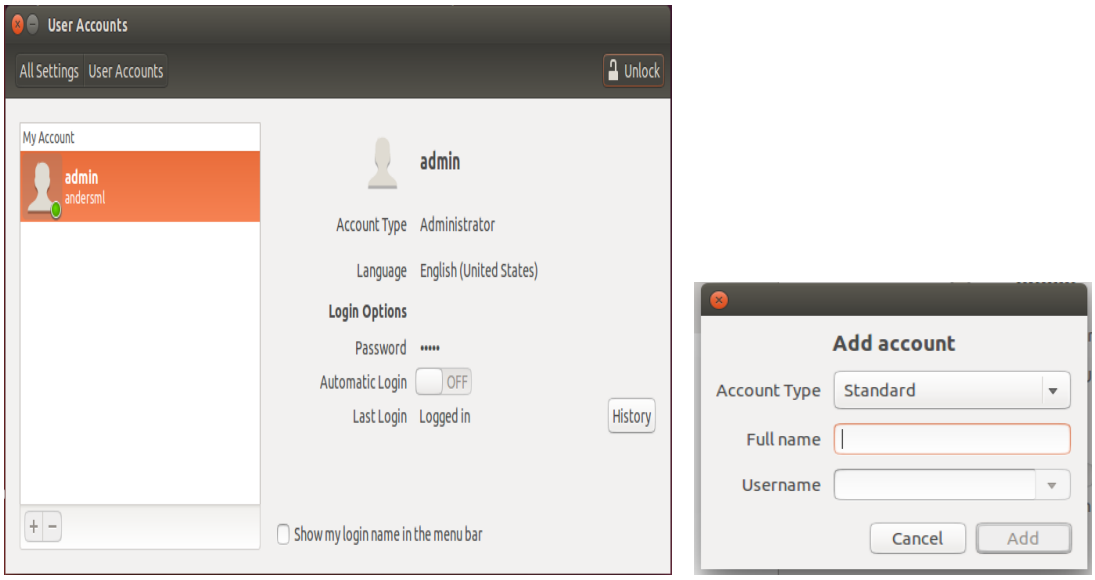
Activity 5-2 - Adding a Linux User Account.

To add a user with Ubuntu's GUI:

1. Click the **System Settings** icon and select User Accounts.

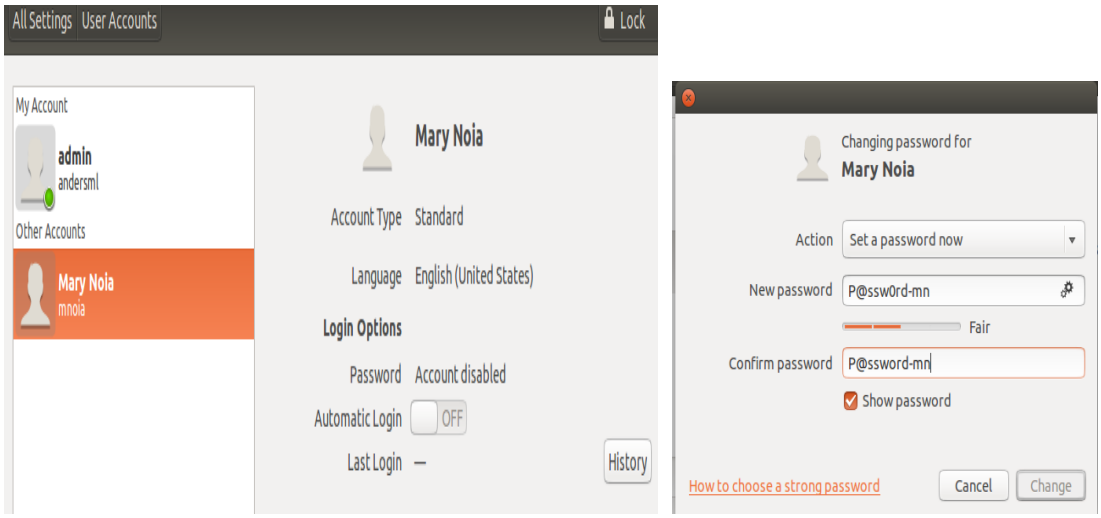


2. Choose the Unlock icon and then choose + to start the add account dialog



3. Enter the Full name: Mary Noia and the Username: mnoia.

4. Next enable the account by clicking 'Account disabled' at the Password prompt and enter a reasonably strong password (eg P@ssw0rd-mn) and click change



5. Now mnoia can login to the server

To add a user from the command line using useradd and passwd in Ubuntu:

6. From the terminal app, type **sudo adduser ajones** and then press **Enter**. This creates the user and the user's home directory, which is /home/ajones.

7. Enter the password **ajones** and then press **Enter.**

8. You'll be prompted for optional information which you may enter if you like.

9. To display the users you just added, return to **System Settings**, and then click **Users Accounts**.

10. Make a **screenshot of User Accounts.**

When you create a user, Linux enters information in three files. The first file is /etc/passwd shown in Figure 5-13 which is a text file that contains user names and information related to user names (**sudo cat /etc/passwd)**. You can edit this file directly to change the full name of the user or other attributes. However, this file does not contain passwords. Instead, Linux stores the encrypted passwords in the second file, /etc/shadow. The information is kept separated so that only the person who logs on as root (or gains root privileges using **sudo**) and the authentication application can read /etc/shadow (**sudo cat /etc/shadow)**. Some distributions of Linux leave the encrypted passwords in /etc/passwd, which can be more easily cracked by hackers. By default, a group is created with the same name as the user account and then stored in **/etc/group**.

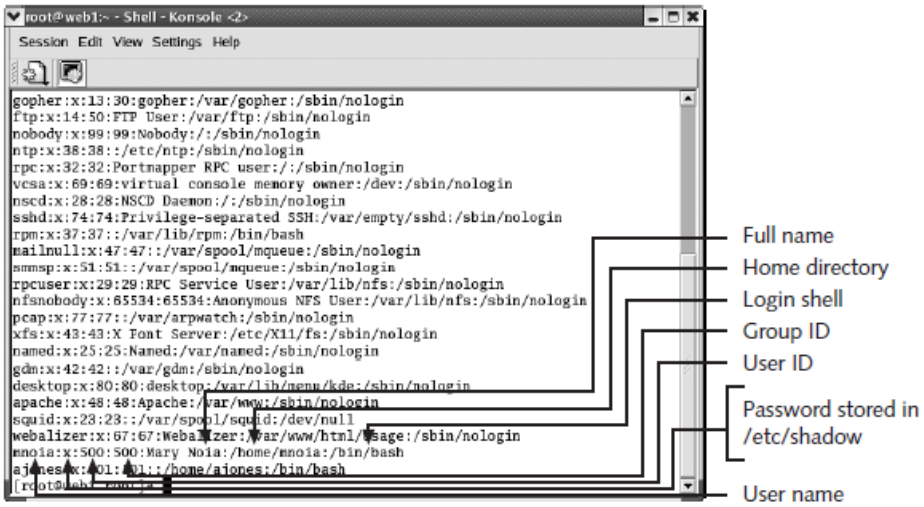


Figure 5-13 The /etc/passwd file

Figure 5-13 shows a partial listing of users in /etc/passwd. Items in the /etc/passwd file are separated by a colon (:). The "x" in the second position represents the password, which is actually stored in /etc/shadow. The next-to-last line is for Mary Noia, who was the first user added. The number 500 appears in two places. The first 500 represents the **user ID** (**uid**). The second 500 represents the **group ID** (**gid**). By convention, uids less than 100 are reserved for special system users and programs. User numbering starts at 500.The root user has a uid of zero and a gid of zero—the defining factor for the root account is the zeros for uid and gid. You could change the account name from root to whatever you want, however.
Wikipedia reference: http://en.wikipedia.org/wiki/Passwd

MANAGING FILE SYSTEM PERMISSIONS

Permissions allow you to control access to the resources on a computer. A resource may be a Web page, a document, a program, or a printer. You give permissions to users and groups. File system permissions exist in Windows using NTFS and ReFS but not with FAT.

Managing File System Permissions in Windows

File system permissions in Windows offer more detailed control than their counterparts in Linux. Windows has 15 individual file system (NTFS) permissions organized into seven standard permissions; only one is geared toward folders. This section focuses on the standard permissions, which are listed in Table 5-2. All the permissions are the same for folders and files except the List Folder Contents permission, which is for folders only. When a permission is set at a **folder level**, the permission applies by default to the **files in the folder** and is inherited by all subfolders.

Table 5-2 Windows permissions

Permission	Description
Full Control	Full Control includes all other permissions, such as Modify and Read, and allows you to take ownership of the file or folder and change the attributes of a file.
Modify	To modify a file, you need to be able to read it and write it. Because the modification could be to delete the contents, this permission lets you delete a file. When you have this permission, you have Read, Write, Read & Execute, and List Folder Contents permissions.
Read	With this permission, you can read files but cannot execute them. For example, you can view a text file or your local program can read a configuration file. You must have at least Read permissions in all folders above the folder containing the file. For example, if you have permission to read a file called test.cfg in C:\config\app, but you do not have Read permission in config and app, you cannot read test.cfg.
Write	When set on a file, this permission allows you to write to files. When set on a folder, you can write to the folder, meaning that you can create and delete files in the folder.
Read & Execute	In addition to the Read permission, this permission allows you to run programs. It also includes the List Folder Contents permission.
List Folder Contents	This permission allows you to view the contents of a folder. It can only be set at the folder level. It allows you to see the files and folders inside the folder.
Special Permissions (Windows 2003 only)	This is not a specific permission. Under the list of permissions for users, when this permission is checked, it means that this user has one or more of the 14 individual permissions set. These individual permissions are combined to form the other permissions in this table, which are appropriate in the vast majority of circumstances.

If a user is a member of multiple groups that have certain permissions on a folder or file, the permissions are added together. For example, if the Users group has Read permission and the Managers group has Write permission, and you are a member of both groups, then you have both Read and Write permissions. When a particular permission is denied, that denial takes priority over when it is allowed. For example, if you were a member of Users and Managers, and also a member of a group that was denied the Read permission, you would have only the Write permission.

Configuring File System Permissions in Windows

Assume that you just created a folder called config, a subfolder called app, and a file in app called test.txt. In Windows the default settings for users are Read & Execute. But what if you wanted to allow users to write to the file? You have to configure the properties of the file and click the check box next to the Write permission in the Allow column. Figure 5-14 shows the default settings for the test.txt file.

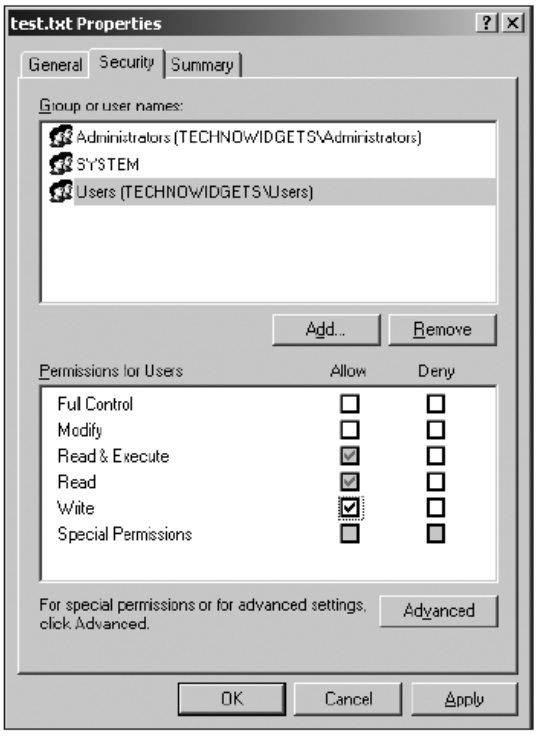


Figure 5-14 Setting Write permissions on a file

In the dialog box shown in Figure 5-14, the Read & Execute (and the related Read) permission is gray, meaning that the file inherited this permission. The Special Permissions check box is grayed because the special permissions are actually set on another screen, accessed with the Advanced button. In the following steps, you will set the Write permission on the test.txt file.

Activity 5-3 - Setting File Permissions in Windows

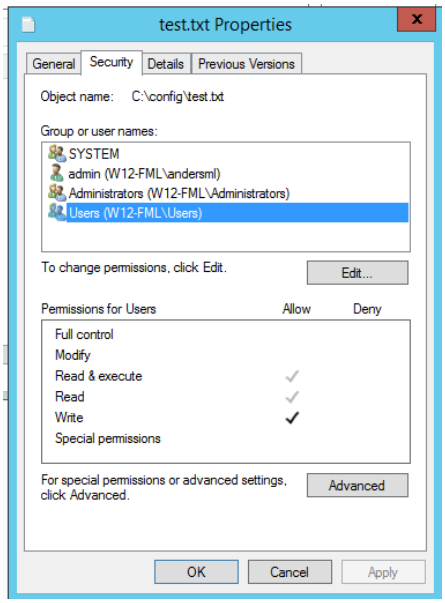
To create a sample file called test.txt in Windows:

1. Right-click **Start**, and then click **File Explore**.
2. In the left pane, click the drive that corresponds to the root of the Windows installation. Typically, it is labeled **Local Disk (C:)**. The contents are displayed in the right pane.
3. **Right click**, select **New**, and click **Folder**. A text box and a folder icon opens.
4. Type **config**, which is the name of the folder, and then press **Enter**.
5. Double-click **config** to display the contents in the right pane.
6. **Right click** in the right pane, select **New > Text Document**. A text box and a text icon opens.
7. Type **test.txt**, and then press **Enter**.

To set a file permission in Windows:

8. In Windows Explorer right-click **test.txt**, and then click **Properties** on the shortcut menu. The test.txt Properties dialog box opens.
9. Click the **Security** tab. You use this property sheet to assign permissions to the test.txt file.
10. In the Group or user names window, click **Users** to view permissions for user accounts in the Users group.
11. To change permissions for user accounts in the Users group, click the **Edit** button and click the **Write** check box in the allow column to allow the users who have accounts in the Users group to have Write permission on the test.txt file. **Take a screenshot of these permissions**

12. Click **OK** to close the dialog box.



Managing File System Permissions in Linux

To better understand file system permissions in Linux, you have to understand the file system itself. Linux does not have a file system equivalent to the Windows FAT file system, which does not have any security. (All Linux file systems have security.) In Linux, a directory is nothing more than a file that contains other files. This structure helps in determining what the permissions allow you to do. Linux directories correspond to folders in Windows. Linux has three basic permissions that you can apply to directories and files: read, write, and execute. Table 5-3 describes these permissions.

Table 5-3 File and directory permissions in Linux

Permission type	When used with files	When used with directories
Read	Read a file or copy a file	List the contents of a directory
Write	Write to the file, including deleting the file	Create files
Execute	Execute programs and shell scripts, which are text files containing Linux commands	Modify the file permissions

The read, write, and execute permissions can be applied to three categories of users. First, they can be set for the owner of the file. When a file is created, ownership is given to the user who created it. Second, the permissions can be set for a group that is assigned to the directory or file. Third, you can set permissions for accounts that are not members of the group (often called “others” or “the world”). This approach is different from Windows.

In Windows, the file or folder remains separate from the permissions assigned to it. Thus, in Windows, you could have dozens of groups and users with differing permissions assigned to a single file. In contrast, in Linux, the permissions are part of the file. Groups exist only in the context of the file. As a consequence, there can be only one group assigned to a file and only one user assigned to the file. In Windows, zero or more users and groups can be assigned permissions in a file. In Linux, the three categories are the only assigned permissions, and they are assigned to every file.

Because three sets of permissions are assigned to every directory and file, the designers of Linux had to come up with an efficient method of designating permissions. Their approach was to have three bits represent each set of permissions. The first bit corresponds to the read permission and has a value of 4. The second bit corresponds to the write permission and has a value of 2.The third bit corresponds to the execute permission and has a value of 1. The permission can either be described as a single digit ranging from 0 to 7 or a combination of r, w, and x. Table 5-4 lists the various combinations of permissions, which are always represented as rwx, meaning read, write, and execute permissions, respectively. The dash (-) indicates that no permissions are set for that item. For example, r-x means that the write permission is not given, only the read and execute. For a directory, the execute permission allows you to use the directory name when accessing files in it. The numeric equivalent can be used to change permissions.

Table 5-4 Linux Permissions

Permissions r=4, w=2, x=1	Decimal equivalent	Binary equivalent
---	0	000
--x	1	001
-w-	2	010
-wx	3	100
r--	4	011
r-x	5	101
rw-	6	110
rwx	7	111

The primary utility for changing permissions is **chmod**. The format of the chmod utility is chmod nnn name, where nnn represents the three digits for each of the three permissions and name represents the name of the directory or file. Each digit in the chmod parameter corresponds to one of the numeric values in Table 5-4. The first digit represents the permission for the owner, the second digit is for the group, and the third digit is for everyone else ("other" or "the world"). Table 5-5 shows several examples of using the chmod utility.

Table 5-5 Using chmod to set permissions

Command	Permissions		
	Owner	Group	Other
`chmod 755 myfile`	rwx	r-x	r-x
`chmod 540 myfile`	r-x	r--	---
`chmod 744 myfile`	rwx	r--	r--

Linux provides another way to use chmod. Instead of setting all the permissions at once, you can change existing permissions. To do so, use the following syntax: chmod x +|- p filename. The x represents which set of permissions is being changed. The values can be u for user, g for group, o for others, s for user and group, and a for all sets of permissions. The a is the default. The + or the - designates whether the permission will be added or deleted. The p represents the permissions. Instead of digits, you use a combination of r, w, and x. Table 5-6 lists some examples. For more information, refer to the Linux command-line help manual, **man.** The syntax is **man <utility>** or in this case, **man chmod.**

Table 5-6 Use of the chmod command

Command	Description
chmod g+rx myfile	For members of the group for myfile, add the read and execute permissions.
chmod o-wx+r myfile	For anyone outside of the group for myfile, delete the write and execute permissions and add the read permission.
chmod +rwx myfile	Change all the permissions for myfile to rwx. Because there was no designator, the a, for all sets, is assumed.

To display the permissions for a file or directory, you use the -l modifier of the ls command that is used to list the contents of a directory. The -l contains the letter "l," not the digit one. It stands for long listing. For example, the command ls -l may produce the following output:

```
drwxr-xr-x 2 ajones   ajones   1024  Oct 17 11:38  apps
-rw-r--r-- 1 ajones   ajones   349   Dec  3 10:44  myfile
-rwxrwxrwx 1 root     root     3245  Oct 18 12:12  mfile
```

The first line states that it is a directory because the character in the first position is a d. The permissions for the owner are rwx, the permissions for the group are r-x, and the permissions for others are r-x. The owner is ajones. The group is also ajones. The size of the file is 1,024 bytes. The file was last modified on October 17 at 11:38 A.M. The name of the directory is apps. Figure 5-18 shows how to create file permissions to allow others to edit a file. Then you change the permissions to allow others to create files in a directory. In Figure 5-18, the text after the "#" prompt is what you type.

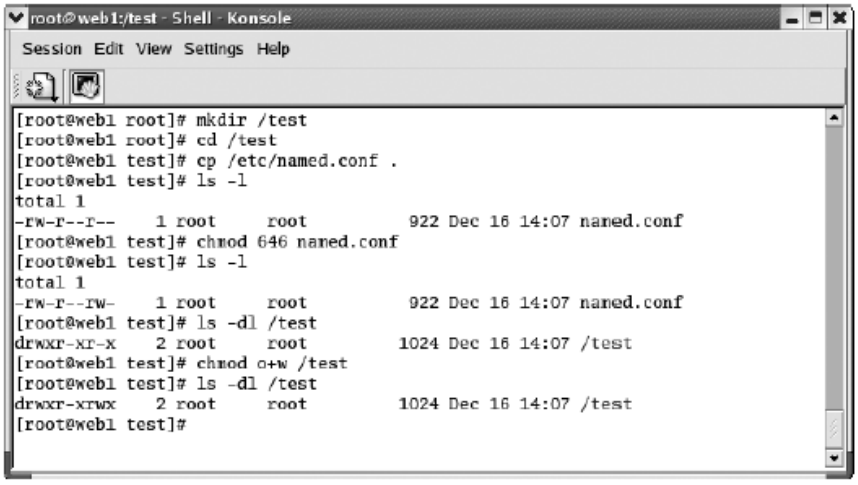


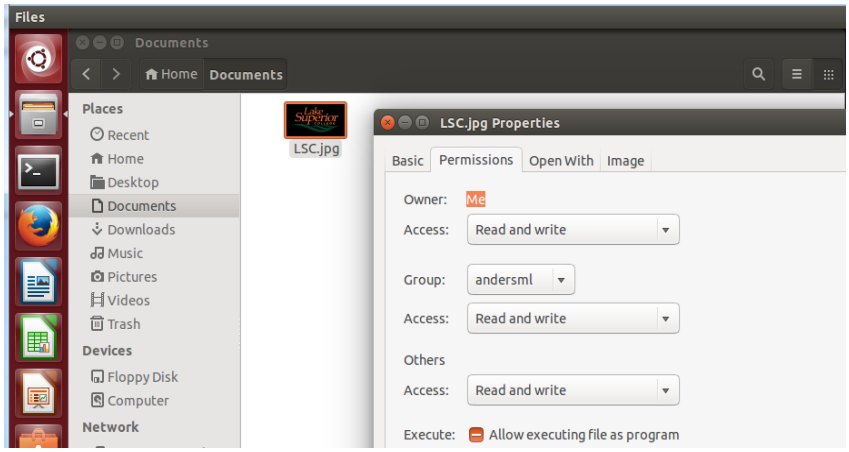
Figure 5-18 Exploring file permissions in Linux

Activity 5-4 - Setting File Permissions in Linux

1. In a terminal window, create the /test directory by typing
   mkdir test
2. Change to the /test directory by typing
   cd test
3. **cp /etc/bind/named.conf.local .**
   to copy the named.conf.local file from /etc/bind to your current directory (/home/…/test).
4. **ls -l**
   A file listing appears, similar to the one shown earlier in Figure 5-18.The permissions allow the owner (root) with read and write permissions.

5. To allow others write permission, type
   chmod 646 named.conf.local
6. **ls –l**
   to see the changes to the named.conf.local file. Now you could log on as any user and edit the file. **Make a screenshot of this step.** Although you can change the file, you need write permissions in the test directory to make a backup of the file. By default, the permissions for the test directory are drwsrwxr-x. Verify this with an **ls.**
7. To allow others write permissions in the test directory, type
   chmod o+w /test

The **files** utility (which is the second icon from the top in the left navigation pane) gives you a GUI method for managing permissions by selecting the file's Properties and going to the Permissions tab.



SHARING RESOURCES IN A WINDOWS NETWORK

Sometimes a user needs files on other computers. If you have the needed files, you can share the folder with the rest of the Windows network. When you create a shared folder, you need to set permissions on it. Although you will learn how to share folders in this section, note that the steps for sharing a printer are similar. You need to determine who can access the shared folder and what they can do. For example, you may just want the user to read files from your folder but not modify existing files or store new files there. You have already learned about file system permissions. Shares have their own permissions, which apply only when a user accesses the shared folder over the network. Share permissions are not as complex as NTFS permissions for files. The three share permissions are listed in Table 5-7.

Table 5-7 Share permissions

Permission	Description
Full Control	Allow files to be added, deleted, changed, and read
Change	Allow existing files to be written to
Read	Can only read files

When you compare the permissions on a shared folder to the file system (NTFS) permissions that were described earlier, note that the most restrictive permissions always take priority for a user accessing the folder over the network. For example, if the shared folder has given a user Full Control, but the

underlying NTFS permissions are Read & Execute, the effective permissions are Read & Execute. If the shared folder permission is Read and the NTFS permission is Full Control, the effective permission is Read. It can be confusing to keep track of the differences between shared folder permissions and NTFS permissions. Microsoft suggests that you set shared folder permissions at Full Control and then implement the restrictive permissions using NTFS permissions.

Activity 5-5 - File Sharing in Windows

Create a shared folder named config:

1. In a previous section, you created a folder called config that you will modify. In Windows Explorer, right-click the folder **config**, choose properties and click the **Sharing** tab.
2. To share the config folder with other users, click the **Advanced Sharing** button and click the checkbox **Share this folder**.
3. The name of the shared folder appears in the Share name text box. By default, the name of the share is the same as the folder name. See Figure 5-19. Click the **Permissions** button to open the permissions for the config folder, which indicates that, by default, everyone gets Read permission. **Take a screenshot and** click **OK** twice return to the sharing tab.
4. Note that the folder is now shared and click **OK**.

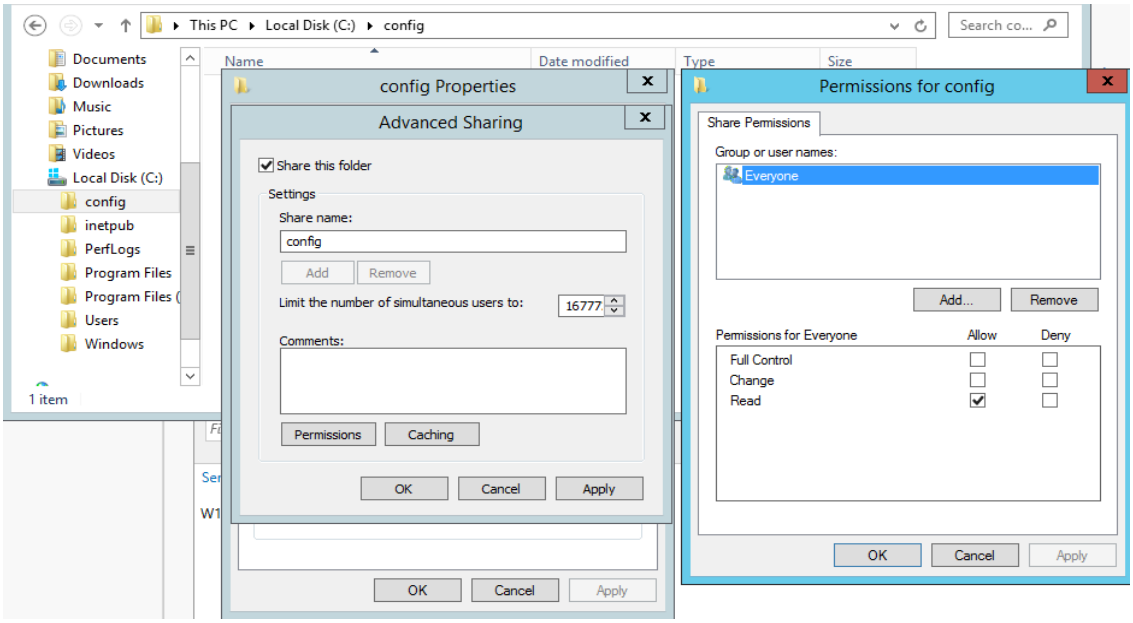


Figure 5-19

If you want to access a folder that is stored on another computer, that folder must first be set up as a shared folder. Then the shared folder on the other computer can become a virtual drive on your computer. A **virtual drive** is a drive that does not physically exist on your computer. For example, when you install Windows, it creates a partition to which it assigns the identifier C:.Your CD-ROM drive may be drive D. Both C: and D: are physical drives. In contrast, a virtual drive assigns a drive letter to a shared folder on another computer. Microsoft uses the term **map a drive**, meaning that the folder corresponds (maps) to a drive letter. Actually, you can even map a drive using a shared folder on the same computer for practice.

ENFORCING NETWORK POLICIES

You may want to exert even more control over users who have an account on your network. You can set network policies in both Windows and Linux, although Windows has significantly more policies. Both Windows and Linux, however, have policies concerning passwords. For example, you can set policies such as the number of days before the user's password must be changed.

Enforcing Network Policies in Linux

In Linux, network policies are part of the entries in the /etc/shadow file. As you learned earlier, the /etc/shadow file contains the encrypted password for each user. It also contains network policies. The items are separated by a colon (:) and appear in a specific order.

Here is an example of an older hashing algorithm used with /etc/shadow:

```
mnoia:$3498jhhd8:11816:20:40:10:15:12379:-1
```

Table 5-8 explains each of the fields in the record.

Table 5-8 Fields in the /etc/shadow record

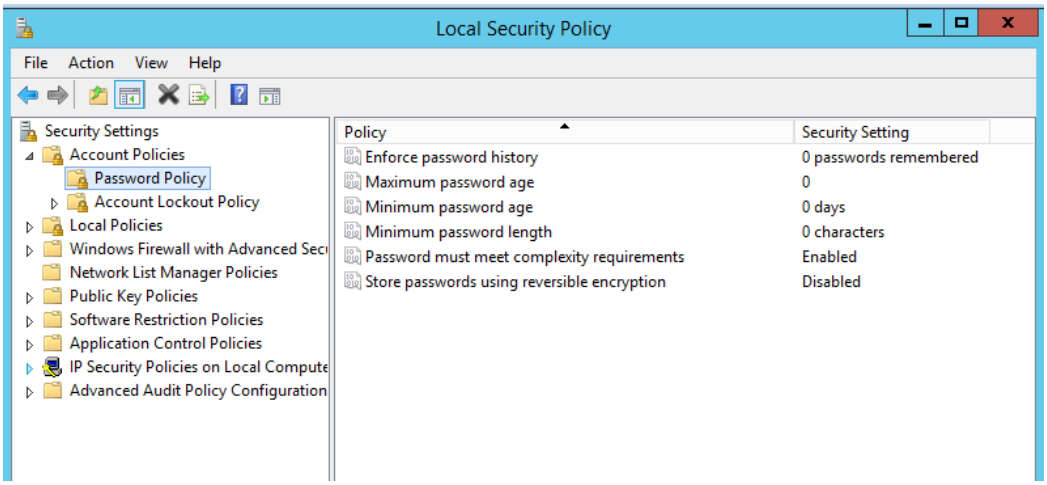
Field	Description
mnoia	User account name
$3498jhhd8	Encrypted password
11816	Starting at January 1, 1970, the number of days since the password was changed
20	The number of days before a change is allowed
40	The number of days before a change is required
10	The number of days of warning before a change is required
15	The number of days before the account becomes inactive after the password has expired
12379	The number of days since January 1, 1970, that the account is set to expire
-1	Reserved field

This Wikipedia link describes more robust hashing algorithms currently used with /etc/shadow:

https://en.wikipedia.org/wiki/Passwd

Enforcing Network Policies in Windows

Windows supports **group policies** which include network security policies affecting the local computer or an entire domain. An individual server has security policies at the local level.



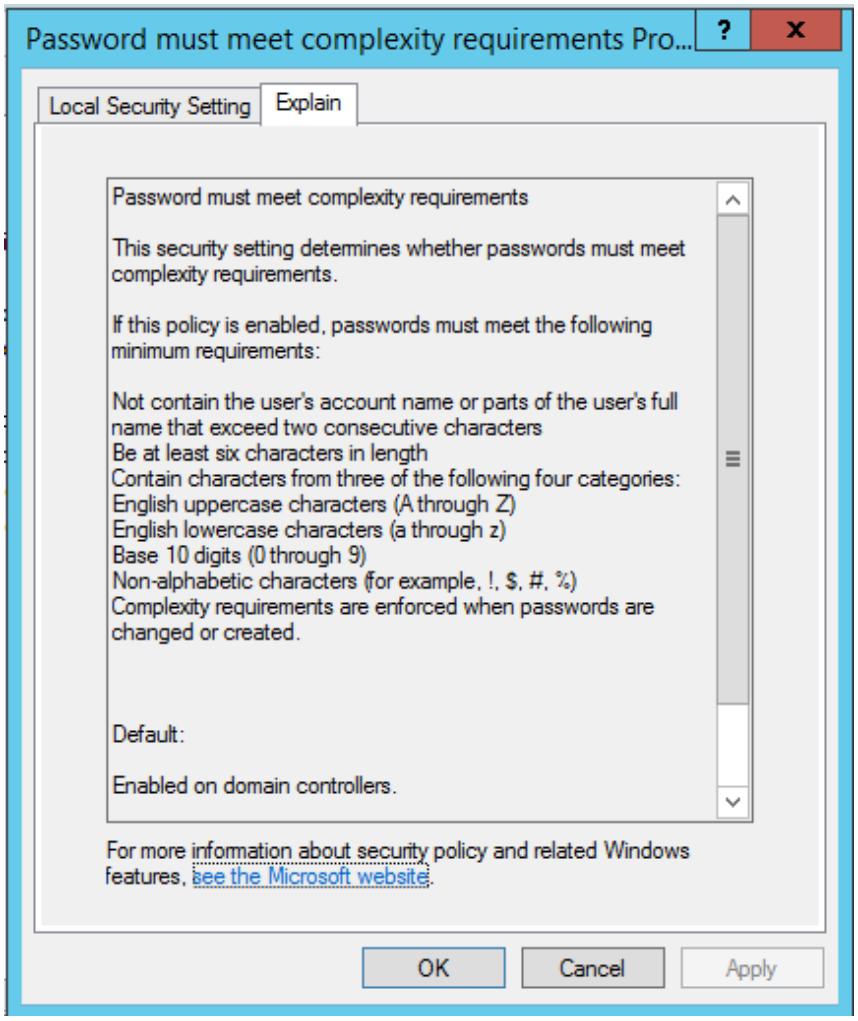
Activity 5-6 - Local Security Policies in Windows

To display a password complexity policy on the local server:

1. From Server Manager > Local Server choose the Tools dropdown menu and select Local Security Policy

2. Expand Account Policies > Password Policy and click 'Password must meet Complexity Requirements'

3. **Take a screenshot of the** Explain tab.



CHAPTER SUMMARY

❒ The Web Administrator's view of server management focuses on controlling access to resources but from a different perspective than a LAN Server Administrator. Typically, the Web server has a guest user account that allows people to contact the Web server without having a user account and password. However, the Web server can also be connected to the LAN, so LAN techniques for controlling access are important as well.

❒ Networking models are split between the methods typically used in a LAN and the client/server networking model. The Microsoft LAN models are divided between the workgroup networking model, in which each computer stores user accounts for that computer, and the domain networking model, in which user accounts are stored in a central location. The workgroup networking model is designed for groups of 10 or fewer computers. The domain model is designed for larger groups. The client/server networking model uses programs running on the server and client programs that access them. Some examples include a Web server and Telnet. The client/server networking model is used to log on to a Linux server.

❒ Authenticating users can be based one of three things: (1) what you know, such as a user name and password; (2) what you have, such as a smart card; and (3) who you are, or biometrics. Kerberos is a method used in the Microsoft environment to authenticate.

❒ User accounts and groups are created to organize and secure users. You should put users into global groups, assign global groups to local groups, and then assign permissions to local groups. Microsoft servers can operate in two modes. First, they can be standalone servers that just have local user accounts;

this is the typical mode for Web servers. Second, the servers can be in a domain with domain user accounts. In Windows this means adding the Active Directory Domain Service. ADDS is a comprehensive repository of information related to users, computers, and other resources on a network.

- Go to D2L **Discussions** to respond to Ch 5 Permissions and **Quizzes** to complete the Ch 5 Review Questions. You have three attempts at the review questions and your score is the average of all three. Submit your completed worksheet to the Chapter 5 **Assignments** folder.