Chapter
16

Authentication

Sometimes it is a good idea to be absolutely certain you know who you are dealing with. Authentication is the process by which the identity of a specific entity—a person, a user, or a computer—is verified. Authentication transactions happen in many places, many times a day. Using an ATM card and PIN to withdraw cash from an ATM, providing a driver's license when making a purchase at a home improvement store with a credit card, and presenting a passport when going through customs are common types of authentication. In each of these examples some sort of authority requests proof of identification. This ID verification indicates that the person requesting the transaction is who they say they are. This process is separate from authorization, whereby it is determined that an entity is granted specific rights or permissions. Simply proving identity does not guarantee the desired outcome of the transaction. Once the authority establishes that you are who you say you are, it then attempts to authorize you to complete that transaction: the ATM ensures that you have sufficient funds to cover the requested withdrawal; the cashier contacts the credit card issuer for purchase approval; and the customs agent checks that all necessary paperwork and visas are in place for a traveler to enter or exit a country.

Windows XP Professional, like Windows 2000, provides a sophisticated authentication system, which is examined in this chapter. Specific topics covered here include the mechanics of Windows XP authentication, the log-on process, configuration and management of authentication parameters, and best practices for secure authentication. We cover authorization in detail in Chapter 17, Authorization and Access Control.

Secure Authentication Features in Windows XP

New Features

Because Windows XP is built on a Windows 2000 base, you find that there are a number of familiar secure authentication features. Windows XP, however, goes

beyond Windows NT 4.0 and Windows 2000 in a number of ways. Whether you are connected to a domain, configured as a part of a workgroup, or are using a stand-alone computer, you find that authentication is even more manageable than before. Here are some of the biggest changes and additions to authentication processes, management, and configuration:

- **Everyone Group.** By default, the Everyone Group no longer includes the Anonymous Group. Previously, the Anonymous Group was granted access to any resource to which the Everyone Group was granted access, even though anonymous users are not required to supply usernames and passwords for authentication.
- **Guest Account.** By default, Windows XP workstations not joined to a domain are configured to use **Guest only** network logons. All users, *including anonymous users,* accessing resources on a computer from over a network with this default setting, are forced to use the Guest Account for authentication and are subsequently given all the same access rights and privileges as the Guest Account.
- **Service Accounts.** Two new service accounts have been added to Windows XP to enhance the granularity of service account access: LocalService for services that run locally, and NetworkService for services that run on the network. The LocalSystem account remains available as well, and is the only account that has **Act as part of the operating system** rights by default.
- **Blank Passwords.** By default, Windows XP workstations that are not part of a domain prevent users with blank passwords from logging on over the network. This is especially helpful for preventing unauthorized access to home workstations connected to the Internet. All blank password access is restricted to local logons only.
- **Password Reset Wizard.** Windows XP supplies a recovery mechanism for use in the event a user forgets his or her password. This Wizard creates a disk that can be used to reset a local account password (it cannot be used to reset a domain password). This disk is computer specific so it cannot be used on another workstation, even if the username and password are the same. Others can use this disk without proper authorization to access a local account, so it is a good idea to keep this disk in a safe location.
- **Stored User Names.** Windows XP allows a user to store frequently used username and password combinations for access to other resources, such as secured Web sites or computers in an untrusted domain. This information becomes part of the user's profile and can travel around the network with the user if roaming profiles have been enabled.

- **Fast User Switching.** Fast user switching allows multiple users of the same computer to log on without shutting down applications that may be in use by another user who is currently logged on to the system. Fast User Switching uses Terminal Services technology to provide this ability. This feature is only available on computers that are not connected to a domain.

As you see from this list, there are quite a few changes to the security and authentication strategy in previous Windows versions. However, Windows XP is interoperable with earlier versions of Windows—from Windows for Workgroups and Windows 9*x* on up to Windows NT 4.0 and Windows 2000. The management and configuration of secure authentication is covered later in this chapter, with attention given to interoperability issues where required. We now move into a discussion of credentials that Windows XP does support as the first step in gaining a full understanding of the Windows XP authentication process.

Authentication Services and Components

All users, groups of users, or computers that participate in a domain have accounts and are called security principals. Security principals operate within a security context. The security context defines the rights and permissions a given account has in a specific situation. For example, a user may be limited in capabilities when logging on remotely instead of locally, or be given more capabilities when logging on from Workstation A instead of Workstation B. Before these capabilities can be granted though, the account must be authenticated. How Windows actually authenticates an account is a relatively straightforward process that utilizes a number of components. This section covers the building blocks of authentication in Windows XP, for both stand-alone workstations and domain members.

Credential Types and Validation

Credentials are the pieces of evidence that substantiate a claim of identity. Business cards, drivers' licenses, passports, and so forth are all types of credentials commonly used to verify identity. Some types of credentials are considered a stronger guarantee of someone's identity: a driver's license is a stronger credential than a health club membership card.

Validation is the process by which a credential is confirmed as genuine. The body requesting the credentials verifies that the credentials are acceptable according to specific standards before granting authorization to complete a transaction. Windows verifying that the username and password combo entered by a user is analogous to a cashier verifying that the photo on a driver's license matches the person presenting it.

The strength of the credential is not just based on the credibility of the issuing body though. It is also based on the authenticity of the credential itself (for example, is it possible this credential has been tampered with?). Windows XP supports three types of credentials that offer varying levels of security for the resources that are being protected: passwords, Kerberos tickets, and smart cards. We cover configuration and management of these credential types later in this chapter.

Local Security Authority

The Local Security Authority, or LSA, is responsible for validation of credentials in Windows. The LSA is also responsible for management of local security and audit policies and the generation of tokens. Exactly how authentication occurs depends on where the account was created:

- For a user logging on to a stand-alone workstation, the authentication occurs in the local Security Accounts Manager (SAM).
- For a user logging on to a Windows NT 4.0 domain, authentication occurs in the domain SAM.
- For a user logging on to a Windows 2000 Active Directory domain, authentication occurs in the Active Directory.

The Security Accounts Manager is a protected subsystem that manages the accounts database. The SAM can be located locally or on a Windows NT 4.0 domain controller. The local SAM manages accounts used only on that computer, while the domain SAM manages accounts, both computer and user, for the domain. The Active Directory is only available in Windows 2000 and Windows 2003 domain controllers.

Regardless of where an account is authenticated, however, the LSA still handles all validation tasks at the local level. In other words, no matter where your account resides, the LSA still validates that your account is listed in an account database trusted by the LSA before passing it along to the appropriate authentication provider. Figure 16.1 illustrates the LSA and the various authentication providers.

Logon Process

Having covered the basic components of the authentication process, we now examine how all these come together and function in Windows XP logons. We first examine the log-on types supported by XP.

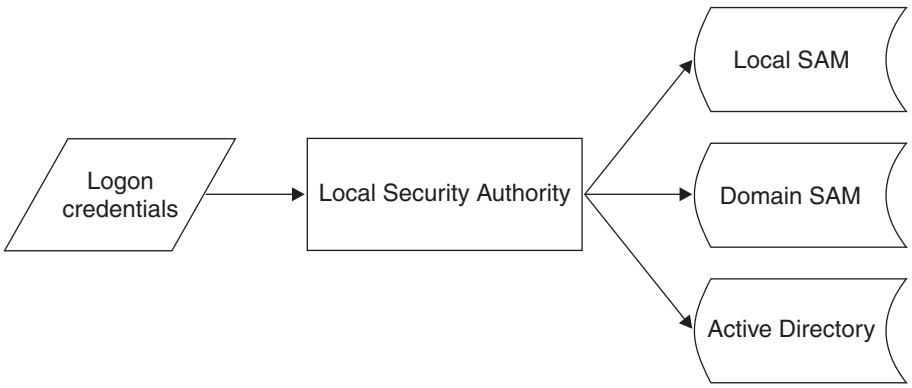

Figure 16.1 The Local Security Authority

Types of Logon

In Windows XP, just as in previous versions of Windows, users can log on over a variety of connections, including network, Internet, dial-up, and local logons. An account holder can attempt to make a connection to a resource and provide the appropriate identifying credentials, such as username and password, to a computer over any of these connection types. The four main types of OS authentication are: interactive, network, service, and batch. Dial-up authentication is covered in Chapter 12, Remote Access.

Interactive Logons

Interactive logons include log-on attempts from a user sitting at the physical workstation where the logon is occurring, users logging in via Terminal Services, and users logging in via Remote Desktop. Interactive logon credentials can be validated by a local accounts database, a domain SAM, or Active Directory.

Interactive logons utilize a number of components to pass credentials entered by the user to the appropriate account database for authentication. The first component is the Winlogon process. Winlogon.exe is a secure user mode process that launches when a user presses Control + Alt + Delete. After a user types that combination of keys, Winlogon calls the Microsoft Graphical Identification and Authentication DLL (MSGINA) to collect username name and password. The MSGINA provides the standard Windows log-on dialog box, but it can be replaced with a custom or third-party GINA.

Once the user has entered his or her username and password in the log-on dialog box and pressed enter, the MSGINA passes this information back to Winlogon, which in turn passes the credentials to the Local Security Authority,

running as LSAS.exe. Finally, as discussed previously, the LSA determines whether authentication should occur locally or remotely, as demonstrated in Figure 16.2. The Winlogon process works with the MSGINA to pass user credentials to the LSA.

Network Logons

An account attempting to log on to a computer remotely (with the exception of Terminal Services, Remote Desktop, and dial-up) is said to be performing a network logon. The remote connection is attempted with the credentials you used to log on interactively. The LSA on the remote computer treats that logon as it would a local logon and uses the appropriate accounts database for authentication of the credentials it presented.

Service Logons

Users and computers are not the only entities that require authentication. Many applications require access to resources that are secured by the operating system.

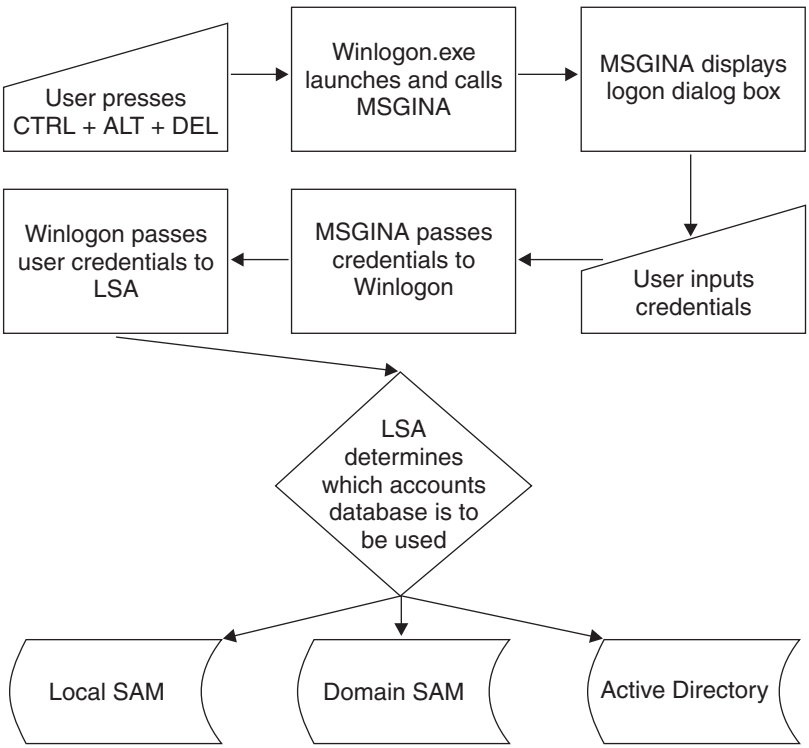

Figure 16.2 Components used in interactive logons

These resources may require the application to have an account so that the appropriate access can be given to the application. Just as a user has to have an account in the domain or on a computer where he or she is trying to print a document, an application or service also has to have the ability to log on to a computer or domain where it must access resources such as files or folders. These applications are given service accounts, which are then granted access to resources after successful authentication. As mentioned earlier in this chapter, there are three built-in service accounts in Windows XP: LocalSystem, Network Service, and LocalService; and it is also possible to create a service account manually. If you want a specific application to have access to certain protected folders or files, you may choose to create a special service account for that purpose.

NOTE: When would you want to use a service account? Well, let's say that you have a Web site running on a Windows XP workstation that requires a specific application to always be running with it. You can add that application to the startup group to run automatically upon login, but that only works if someone is there to log in. If the computer crashes and reboots in the middle of the night and you are not there to log in after the reboot has completed, the application cannot start. However, you can configure that application (using a resource kit utility called *srvany.exe*) to run as a service and log on with a service account. It will automatically start up when the computer starts and does not require any human-user intervention.

Batch Logons

A batch logon is used by applications that run as a batch job, such as a job scheduled with the task scheduler. The job is logged on as a batch user by default in Windows XP, rather than as an interactive user.

Authentication Process

Now that you are familiar with the types of logons supported in Windows XP, as well as the basic components used to gather user credentials for logons, we look at how the actual authentication process takes place.

The Windows XP authentication process, like that of Windows 2000, supports multiple authentication protocols for use in a variety of log-on scenarios. The authentication protocol defines the process by which the supplied account credentials are verified. For the log-on types above, the protocols Windows XP uses for authentication are Windows NT LAN Manager (NTLM) and Kerberos. The default authentication protocol is Kerberos, with NTLM used as

Windows XP's second choice. Windows XP determines which protocol to use based on a simple trial and error mechanism. If Kerberos authentication fails, the backup, NTLM, attempts to authenticate.

Here is how the protocol selection process works. After the username and password passes to the LSA, the LSA passes user credentials to the Security Support Provider Interface, or SSPI. The SSPI is the boundary between the LSA and the Kerberos and NTLM authentication providers. It is a protocol-independent interface, which has the benefit of allowing developers to write applications that can function with both Windows 2000 and Windows NT 4.0 domains. The SSPI hands-off the username and password to the Kerberos server for the appropriate domain. If the Kerberos server recognizes the credentials, authentication continues, and is either deemed successful or unsuccessful. If no Kerberos server is found, however, the LSA is notified to kick off the process once again. The LSA passes the user credentials to the SSPI, which in turn passes it to the NTLM service for authentication. If no authentication provider can provide authentication of the credentials, an error message is returned to the user. Figure 16.3 illustrates this process. This flowchart demonstrates the Windows XP authentication protocol selection process.

Kerberos

The open network is full of risks. The computers connected to it may not be secured against intrusion, and the media over which network data flows can be easily tapped into. This means users can pose as someone else, computers can be fraudulently set up on the network to pose as legitimate servers, and data can easily be monitored or even modified. Both the client requesting logon and the server that provides the authentication service need to be mutually assured that they are speaking to the correct party.

Kerberos protocol is the namesake of the guardian of the gates of the underworld in Greek mythology. Kerberos, also commonly called Cerberus, was charged with preventing the living from entering and the dead from leaving the underworld. In the same way that Cerberus verified the identities of those entering and leaving the underworld, Kerberos also verifies the identity of parties communicating over a network. Kerberos is the default network authentication protocol for Windows XP and provides mutual authentication for both client and server involved in a transaction. This means that it verifies that *both* communicating parties are who they say they are. Kerberos version 5 is the version implemented within both Windows XP and Windows 2000, and is based on RFC 1510.

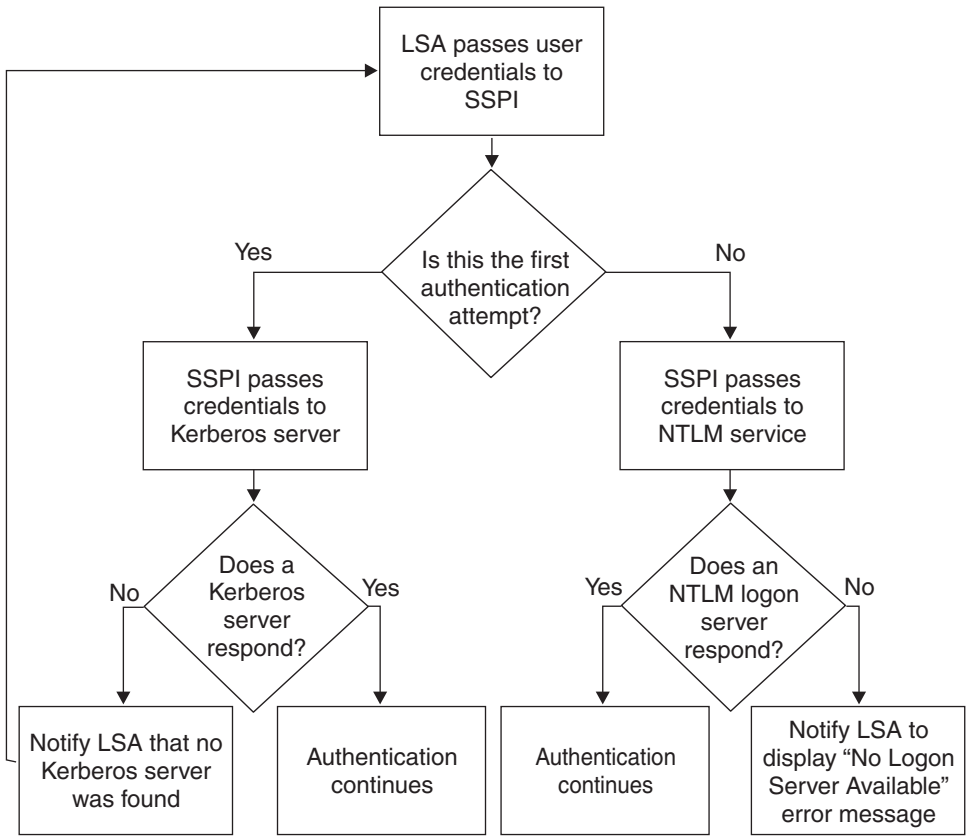

Figure 16.3 Selection of authentication protocol

Shared Secret Overview

Kerberos uses shared secrets to validate identity. A shared secret is something that is known by all parties involved in a transaction, but by no one else. Let's say two spies, Boris and Alexandra, are meeting in a park to exchange information. When they arranged the meeting, they also came up with a secret code—"73"—so they can be mutually assured that Boris is speaking to Alexandra and Alexandra is speaking to Boris. Both parties arrive at the right place and time, and Boris walks up to the woman he believes to be Alexandra and says, "By any chance do you know the current temperature in San Jose?" to which Alexandra replies, "It's 73 degrees Fahrenheit in San Jose." Boris then says "I was last in San Jose in '73. It was a lovely place." At this point both parties have shared the mutual secret with each other and the information exchange can now occur, as illustrated in Figure 16.4. Shared secrets are used to mutually validate identity of both parties in a transaction.

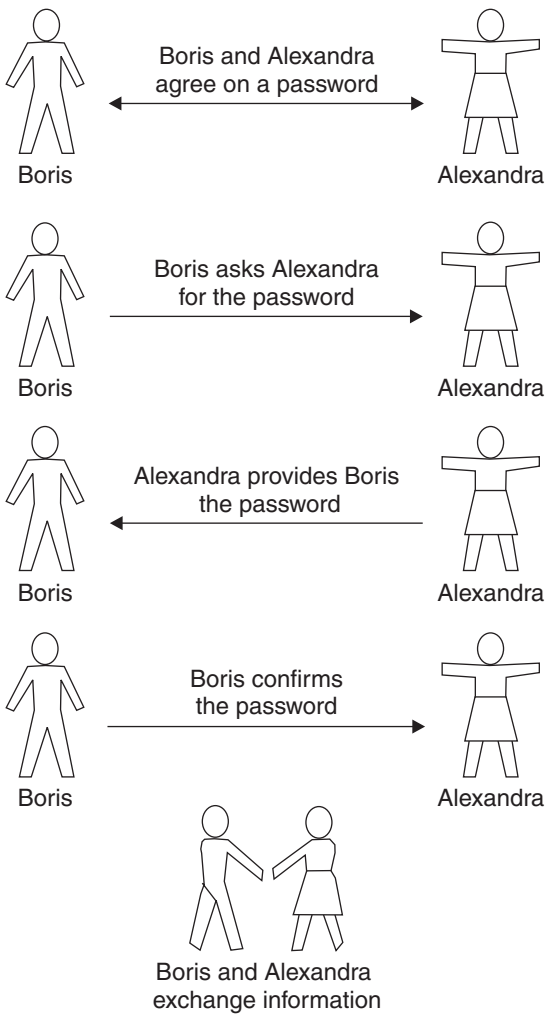

Figure 16.4 Shared secret authentication

In order for this to have qualified as mutual authentication, the secret has to be repeated by each party to ensure that both sides know the secret. By hiding the secret in the conversation, it makes it difficult for eavesdroppers to understand exactly what portion of the conversation was the secret. Kerberos handles both these tasks as well as the added feature of acting as a trusted third party, distributing the secrets to parties that wish to conduct a transaction with each other, much as a secret intelligence team would provide this service to their own agents. Let's take a look at how this works with Kerberos.

Imagine that now Alexandra and Boris want to exchange information via computer, rather than in person. In the physical world, they have all sorts of spy

gadgets to protect their conversations in which they planned meetings and shared secret codes. They need to do the same thing with their computer-based communications. The first thing they have to do is determine how to share the secret code, or password, that will prove to both Boris and Alexandra that they are communicating with each other.

Because networks can be very insecure, they cannot just e-mail the password that they want to use. They need a way to ensure confidentiality. Someone could be sniffing packets as they come over the network for just such an e-mail. So how do you send a password to someone so that that person, and only that person, knows what the password is? Easy—you encrypt it and share the key to decrypt it. Kerberos does this by providing a single key that will encrypt and decrypt the password. This is known as symmetric secret key cryptography, and the "package" that contains the shared secret password is called an authenticator.

Another challenge facing Boris and Alexandra is that although they have protected the password, they need a way to be sure that someone scanning the network does not take those packets containing authenticators and reuse them to fraudulently pose as one party or the other. By using an authenticator that is different each time it is sent, it is possible to prevent these types of replay attacks. Kerberos provides this function by using a unique shared secret. It is encrypted with the secret key and decrypted by the secret key at the other end.

Now that you are familiar with the basic concepts of shared secret authentication with Kerberos, let's walk through an example of Boris and Alexandra using this protocol.

Boris and Alexandra need to send sensitive documents to each other across the network. Because this is top-secret information, Boris needs to be sure that it is actually Alexandra that he is contacting, and Alexandra likewise needs assurance that the person contacting her is actually Boris. Boris and Alexandra have decided that shared secret authentication is the way to handle this mutual identification validation, and now Boris has some information he needs to share with Alexandra. Here is the process that they use to authenticate each other:

1. As shown in Figure 16.5, Boris sends Alexandra a message that is encrypted with their secret key. The encrypted message contains the

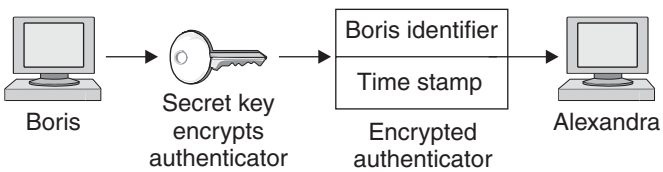

Figure 16.5 Boris makes initial contact with Alexandra.

authenticator, which in turn contains two pieces of information. One piece identifies the sender as Boris and the other is the time on Boris' computer. The time stamp acts as a unique identifier, to prevent fraudulent reuse of the authenticator.

2. Alexandra receives the message from Boris, as shown in Figure 16.6. She decrypts the authenticator with the shared key and takes a look at the time stamp from Boris' computer. The time shown must be within the acceptable range of difference from the time on Alexandra's computer. For the sake of this discussion, let's say it must be within plus or minus two minutes. If the time is within that range, Alexandra can be reasonably sure it is Boris, but if it is not, then she can refuse to communicate with the person claiming to be Boris. It is still possible for that packet to have been replayed from a previous attempt by Boris to communicate with Alexandra, but the time stamp also acts as a unique identifier. If Alexandra had previously received an authenticator from Boris with an identical time stamp, the second one could be rejected. The same holds true of any time stamp that is from a time earlier than the last time stamp received.
3. Because Alexandra is now reasonably sure that the authenticator that she received is from Boris, she responds to the message as shown in Figure 16.7. Alexandra removes just the time stamp from the message and encrypts it with the secret key. By doing so, she not only proves that she knows the secret, she also proves that she is able to decrypt and modify the message with the secret key that only Boris and Alexandra share. This assures Boris that it is actually Alexandra who is responding.

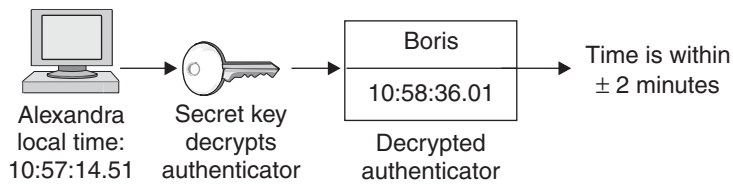

Figure 16.6 Alexandra receives Boris' message.

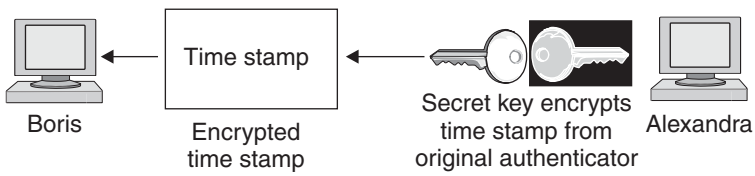

Figure 16.7 Alexandra replies to Boris' message.

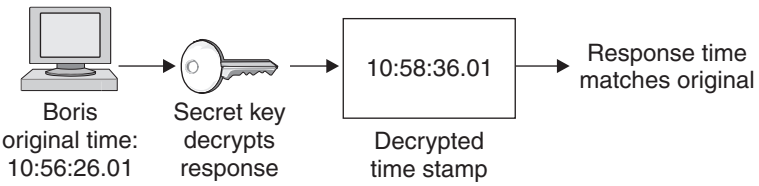

Figure 16.8 Boris receives Alexandra's response.

4. Boris receives Alexandra's response (shown in Figure 16.8) and decrypts it. Once he examines the time stamp and successfully compares it with the time stamp in his original authenticator, Boris can be confident that it was from Alexandra, since only he and Alexandra share that key.

Key Distribution and Tickets

The above scenario explains how Boris and Alexandra use secret keys to authenticate each other. But there is a large piece of information missing: Exactly how and where did Boris and Alexandra get their secret keys? They would like to exchange keys with each other and only each other, prevent others from being able to use the keys, and at the same time guarantee that the sender and receiver of the keys are really Boris and Alexandra. Involving a trusted party to provide the secret key is the way this problem is solved in Kerberos.

Let's take the example of Boris and Alexandra a step further. Both of these people need to communicate with other people and exchange information over the network, as well as connect to shared resources such as databases, printers, and e-mail. Each of these resources requires a secret key to mutually authenticate both the client and server in the transaction. If each user is required to have a different secret key for each resource he or she wants to access, the total number of secret keys required for all users and resources on the network could be huge—a potential management nightmare!

Kerberos solves the two problems of key distribution and management with a Key Distribution Center, or KDC. The KDC maintains a central database of keys and the accounts that they belong to. The group of accounts that the KDC is responsible for is called a realm. You can think of a realm as being analogous to a domain. The KDC is a service running on a Windows 2000 or Windows 2003 domain controller. In fact, *all* Windows 2000 and Windows 2003 domain controllers are Key Distribution Centers. In this section, we walk through the process of key distribution.

When a client wants to talk to a server to access resources located on that server, such as files or printers, the client sends a request for authentication to the KDC. Kerberos caches account passwords as a piece of encrypted data,

called a *long-term key*, on both the client and the KDC. This long-term key is used to secure communications between the KDC and clients that use Kerberos for authentication. The authentication request is composed of two parts:

1. An identifier for the client that is requesting authentication and the service or resource the client wants to access.
2. An authenticator for the KDC that contains a time stamp from the client and the client's long-tem key.

Figure 16.9 illustrates such a request from a client.

The KDC responds with a session key to be used by the client and server that wish to communicate. The session key is encrypted with the long-term key of the parties that will be communicating. The client's session key will be encrypted with the client's long-term key while the server's session key will be encrypted with the server's long-term key. The server's session key is grouped with the client's authorization level for the requested server or service. The two are encrypted together with the server's long-term key, and the resulting piece of information is called a *session ticket.* You can think of a ticket as a permit or license for accessing a server or service within a Windows 2000 domain. Tickets are required for accessing all resources in a Windows 2000 domain, including the log-on process. A session ticket is shown in Figure 16.10.

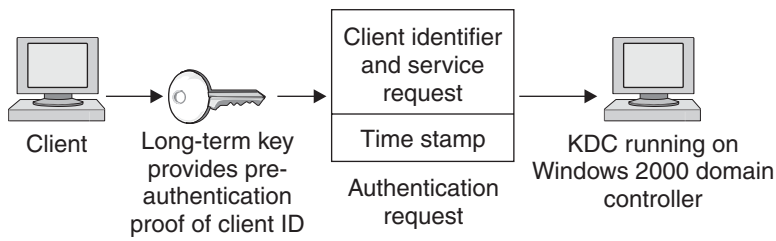

Figure 16.9 A client requests authentication assistance from the KDC.

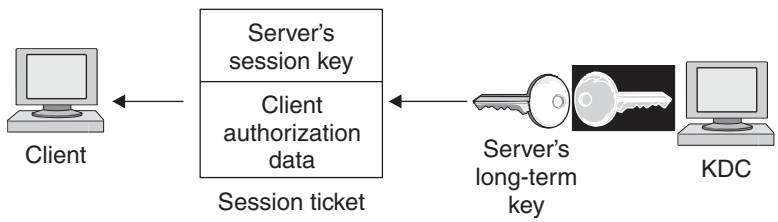

Figure 16.10 A session ticket

The first time the client contacts the KDC for authentication assistance, which occurs at logon, the KDC generates a session key for that client. In this case, the KDC is the service for which client authentication assistance is being requested. The KDC responds with a special session ticket called a TGT, shorthand for Ticket-Granting Ticket, which is to be used for further communication with the KDC itself. It contains two pieces of information:

1. A copy of the logon session key generated by the KDC
2. Client authorization data

Just as in the case of ordinary session tickets, the session key for the client is encrypted with the client's long-term key, while the TGT is encrypted in the KDC's long-term key. The TGT is illustrated in Figure 16.11.

Sending session keys to both the client and server would place a significant load on the system resources of the server where the KDC is running. Instead, the burden of session management is placed squarely upon the client. Both of these session keys are sent directly to the client that wishes to initiate communication with a server. The client is responsible for managing the ticket and all subsequent attempts to access resources with that ticket. By eliminating the need for the KDC to act as the manager of session messaging between client and server, the potential load on the memory of the server or servers running the KDC service is reduced. Making the client responsible for the session ticket management provides the additional benefit of allowing the client to directly contact the server without going through the KDC for a new session key each time it needs to access a network resource for which it has already been given session-specific information.

The session ticket is good for as long as the client remains logged on or until the ticket expires, whichever comes first. Typically, session tickets expire after eight hours. If a ticket is still valid at the time a user logs off, the ticket is destroyed to prevent unauthorized reuse of that ticket. The client can attempt to access the network resource for which it has been issued a ticket for as long as the

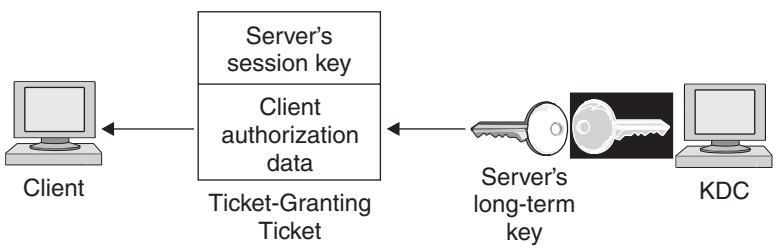

Figure 16.11 A Ticket-Granting Ticket

session-specific data it has been given is valid. The client simply presents his or her ticket for a specific resource or server each time he or she attempts to access that resource.

We now look at how the Windows XP logon process works with Kerberos.

The Kerberos Logon Process

If a workstation is not part of a Windows 2000/2003 domain, there is no Kerberos authentication, so there is not a requirement for stand-alone workstations or Windows NT 4.0 domain members to have tickets to access resources. (Kerberos is an industry standard method of authentication. Because Microsoft's Kerberos implementation is interoperable with other implementations of Kerberos, it is possible that some sort of Kerberos authentication scheme could be used in the above scenario. That is beyond the scope of this book, however.) Windows 2000 domain members must have tickets for accessing any resources or services within the domain, including logon or services running on the local computer where a user has logged on.

So how does the user get a ticket from the KDC before logging on? Recall that all Windows 2000 domain controllers are also KDCs. When a user logging on from a Windows XP workstation attempts to contact the Windows 2000 domain controller for logon, he or she needs two pieces of data:

- A TGT that allows access to the ticket-granting service
- A ticket that allows access to the workstation used for logon

Figure 16.12 illustrates this activity.

Now we walk through how this process takes place.

1. The user (Boris, for example) presses Control + ALT + Delete, also known as the Secure Attention Sequence. The MSGINA appears, Boris enters his username and password, and clicks OK. MSGINA hands the log-on credentials to Winlogon, which in turn passes them along to the LSA.
2. The LSA takes Boris' password and converts it to a preauthentication encrypted key that is stored in the workstation's credential cache and can be used by whatever authentication provider is indicated for the logon

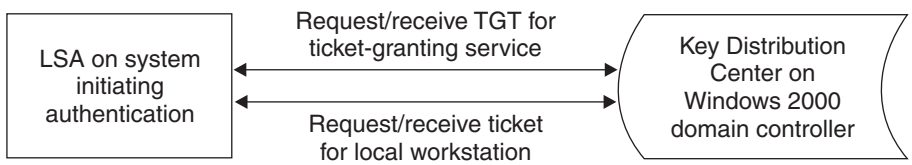

Figure 16.12 The components required for Kerberos-based logon.

Figure 16.13 Kerberos and LSA interaction

type, which in this case is for a Windows 2000 domain. The LSA hands this information to the Kerberos SSP, which handles communication with the KDC. The LSA interfaces with the Kerberos SSP as shown in Figure 16.13.

3. The Kerberos SSP sends the preauthentication key to the authentication service on the Kerberos server. This request contains the following items:
   - An identifier for the client that it wishes to be authenticated and the service or resource the client wants to access. In this case, the client is Boris and the specific service is the authentication service.
   - An authenticator for the KDC. It contains a time stamp from Boris and his preauthentication data (Boris' password as encrypted by the LSA).
4. On receipt of this authentication request, the KDC verifys that the time stamp contained within the package is within the range of time that is permissible for that realm. If it is acceptable, the KDC sends the Kerberos SSP a ticket-granting ticket for Boris to use. The actual data structure contains the following items:
   - The session key for Boris to use.
   - A TGT for the KDC that has been encrypted with the KDC's secret key, and includes a session key for Boris and the KDC to use when communicating with each other, as well as information detailing Boris' authorization level for the KDC. Now the Kerberos SSP is in possession of one of the two items Boris needs to complete his logon.
5. The Kerberos SSP receives this message and turns its attention to the matter of obtaining access to the ticket-granting service. The ticket-granting service provides the ticket that is needed to log on to the local workstation. The Kerberos SSP sends a message to the KDC that includes these pieces of information:
   - The name of the workstation Boris is attempting to log in from and the domain the workstation belongs to.
   - Boris' TGT and authenticator that are encrypted with the session key obtained in Step 4.

6. The KDC examines the request for access to the ticket-granting service and ensures its authenticity as discussed in the previous section. If the request is deemed legitimate, the KDC returns the following items to the Kerberos SSP for use by Boris:
   - A session key, encrypted with Boris' secret key, for Boris and the workstation to use when communicating with each other.
   - A session ticket for accessing the local workstation, which is encrypted with the workstation's secret key.
7. Now that the Kerberos SSP has both items required for logon, it can pass this information to the LSA, which handles the task of querying the local SAM to determine what authorization levels Boris has for this workstation. On completion of that query, an access token is created. (Access tokens are discussed in Chapter 17, Authorization and Access Control.) The LSA passes this token and confirmation of Boris' identity to Winlogon, which completes the logon process and displays the Windows XP desktop.

Windows NT LAN Manager

Windows NT LAN Manager (NTLM) is the authentication protocol used for credential validation when one of the computers is running Windows NT 4.0 or when computers involved in the authentication process are configured to act as stand-alone or workgroup members, specifically:

- Windows XP authenticating to Windows XP computers in the absence of a domain
- Windows XP authenticating to a Windows NT 4.0 domain
- Windows XP authenticating to Windows for Workgroups, Windows 95, Windows 98, Windows 98 Second Edition, and Windows Me
- Windows XP authenticating to Windows NT 4.0 computers running in a Windows 2000 domain

NOTE: This is not an exhaustive list of all possible instances where NTLM could be used, but is instead a list of where Windows XP would use NTLM.

NTLM Types

NTLM relies on a challenge/response method for validation of username and password. Username and password are sent across the network as a hash, rather than in clear text. It does not provide mutual authentication; that is, verification

that both the user and the authentication provider are who they say they are. Instead, it only provides authentication of the account being used to log on to a server.

There are three types of NTLM supported in Windows XP. Multiple versions provide backward compatibility to older versions of Windows and provide varying levels of security. The three NTLM types are as follows

- **LAN Manager:** The oldest version of NTLM challenge/response provided by Windows XP is also the least secure. It is used when a Windows XP computer is attempting to authenticate to a computer running older versions of Windows: Windows for Workgroups, Windows 95, or Windows 98. If you are not planning to access any resources shared from workstations running these operating systems, you may wish to disable this protocol, because it is not as strong as the other versions of NTLM provided. The algorithm used to encrypt these passwords effectively limits password strength to seven characters and is not case sensitive. We cover disabling LAN Manager authentication in the next section, "Configuration and Recommended Practices."
- **NTLM version 1:** NTLM version 1 provides a more secure method of authentication than LAN Manager. It uses 56-bit encryption, allowing it to have an effective password strength of 14 characters and also allows both upper- and lowercase characters to be used. It is used on Windows NT 4.0 Service Pack 3.0 and earlier domains.
- **NTLM version 2:** With the advent of Windows NT 4.0 Service Pack 4, a newer and more secure challenge/response mechanism was made available. NTLM version 2 offers message integrity, 128-bit encryption, and session-level security. Session security is provided by the use of separate keys for message integrity and confidentiality. The RFC-compliant HMAC-MD5 algorithm used in NTLMv2 provides message integrity checking, and 128-bit encryption is used for message confidentiality.

Authentication Process

As you have learned, user authentication in Windows XP uses the LSA to pass credentials to the authentication provider. NTLM is the fallback authentication provider for Windows XP, used when Kerberos is not available. NTLM uses the MSV1_0 Authentication Package, which references the SAM database as its user accounts database.

There are two parts of MSV1_0; one that runs on the computer where the logon was initiated, and the other that runs on the computer where the account is located, as shown in Figure 16.14. If the log-on computer also houses the user

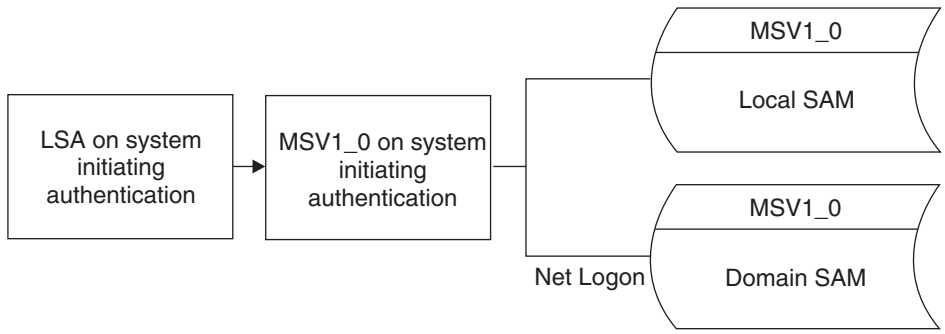

Figure 16.14 NTLM uses MSV1_0 to authenticate accounts.

account, then both portions run on the same computer. If the user account is located on a remote machine, MSV1_0 hands the request to the Netlogon service, which in turn sends the request to the remote machine.

Configuration and Recommended Practices

Part of a good security plan includes determining what rights and features will be available to a user that wishes to log on to Windows XP. Even if your Windows XP computer is not a member of a domain, it is important to ensure that it is protected from unauthorized access by other users on your network. The network to which you are connected may be just two devices using a wireless network, but if you are connected to the Internet via an "always on" connection such as DSL, you need to consider that your network now extends beyond just the boundaries of your home or office. Your username and password are the two pieces of information that you supply to get access to resources. It is important to protect these credentials, and establishing (and subsequently enforcing) a general policy on how these items will be treated in your network will go a long way toward keeping your network as safe as possible. Besides just protecting the log-on credentials, you need to devise a policy for limiting log-on access to resources on your network, to prevent unauthorized users from using "accidental backdoors" such as blank passwords or a guest account that has been given an exceptionally high level of privileges. You want to create the most restrictive local authentication policies you can for your network to prevent intruders from accessing information that you have not explicitly designated as public (such as a Web site).

This section discusses configuration and management of Windows XP credentials, authentication policies, and recommended practices. While previous sections referred to the domain log-on process, the information in this sec-

tion is primarily for stand-alone Windows XP computers, because domain account management is beyond the scope of this book. Where applicable, however, specific caveats or recommendations are made in cases where domain-related issues are of particular importance.

Individual Account Settings

Some account settings affect only a single account. These settings govern certain aspects of that account's password and are configured from within the Local Users and Groups console. To access these settings:

1. Launch **Control Panel**, then select **Administrative Tools**, and then select **Computer Management**. (There are a number of other ways to reach this console besides this, but for the sake of discussion we limit it to this way for now.)
2. In the left-hand pane of the console select **Local Users and Groups** and then click on the **Users** subfolder. The users configured for this system appear in the right-hand pane of the console.
3. Right-click the user you wish to modify, and select **Properties** from the pop-up menu. The property sheet for that user appears, as shown in Figure 16.15. Click the boxes to enable or disable the settings as you desire.

The configuration options for user account passwords are as follows:

1. **User must change password at next logon:** If you reset a user's password for them and assign a generic password, it may be helpful to require the user to change the password the next time they log on. The user selects a password that is easier for them to remember and reduces the possibility of other unauthorized users attempting to guess another user's password by using the generic password that was previously assigned to the unauthorized user. If this option is selected, however, the next two options are disabled.
2. **User cannot change password:** This option is useful when used in conjunction with accounts that are used by multiple people, such as Guest Accounts. This prevents one user from changing a password, thereby locking out all users of that account. It also reduces the administrative workload from having to reset this password and disseminating the new password to all users of that account.
3. **Password never expires:** This is another option useful for multiple-user accounts, such as Guest or Administrative Accounts. If you set a maximum password age in the global account policies (to be covered in the

Figure 16.15 Configure individual account settings from within Local Users and Groups.

section below), it affects all accounts except for the accounts where this has been designated.

4. **Account is disabled:** By selecting this setting, the user cannot log on. You might wish to disable an account rather than deleting it; perhaps for a user that has left, but will be replaced with someone who will need to have access to the same resources as the previous user. Disabling it prevents unauthorized access to the network by users who know the current username and password combination. Once the replacement user has

been determined, you should rename the account, change the password, and enable the account. This way the user account has the same SID and retains all previous rights and permissions.

5. **Account is locked out:** If a user has attempted to log on with incorrect credentials that exceed the maximum number of allowable attempts, this box is checked. An administrator can unlock the account by removing the check in this box. This setting cannot be selected by the administrator, only deselected. If you wish to quickly terminate user access, use the **Account is disabled** setting.

NOTE: If a user account is locked out and auditing has been enabled, take a look at the Event Viewer to verify that the bad log-on attempts took place within acceptable parameters, such as a specific time or location. It may be that the correct user made only one or two of those bad attempts and the rest came from an unknown or unauthorized source.

Stored User Names and Passwords

Windows XP includes a new tool that helps users maintain a library of passwords for accounts they have for a variety of resources, such as Web sites (sites that require Passport or SSL logon only), network resources, and so forth. It provides the illusion of having a single set of credentials for these resources, even for resources that you do not manage directly. For example, you have an administrative account on several computers in your network, but for security reasons you do not wish to log on with an administrative account. You also have a .NET Passport for use with a variety of Web sites, as well as the credentials you need for your online brokerage. By configuring these credentials in the Stored User Names and Passwords tool, they are saved as part of your profile and automatically presented to the resource to which you request to be logged on. To configure Stored User Names and Passwords:

1. Launch **Control Panel** and open **User Accounts**.
2. Select your account name and in the Related Tasks box, click **Manage my network passwords**. However, if this computer belongs to a domain, you select the **Advanced** tab, and then click **Manage Passwords**. The Stored User Names and Passwords dialog box appears, as shown in Figure 16.16.
3. Click the button that corresponds to the action you wish to perform. If you select **Add** or **Properties**, the Logon Information Properties dialog box appears, as shown in Figure 16.17. Simply add to or change the information stored here.

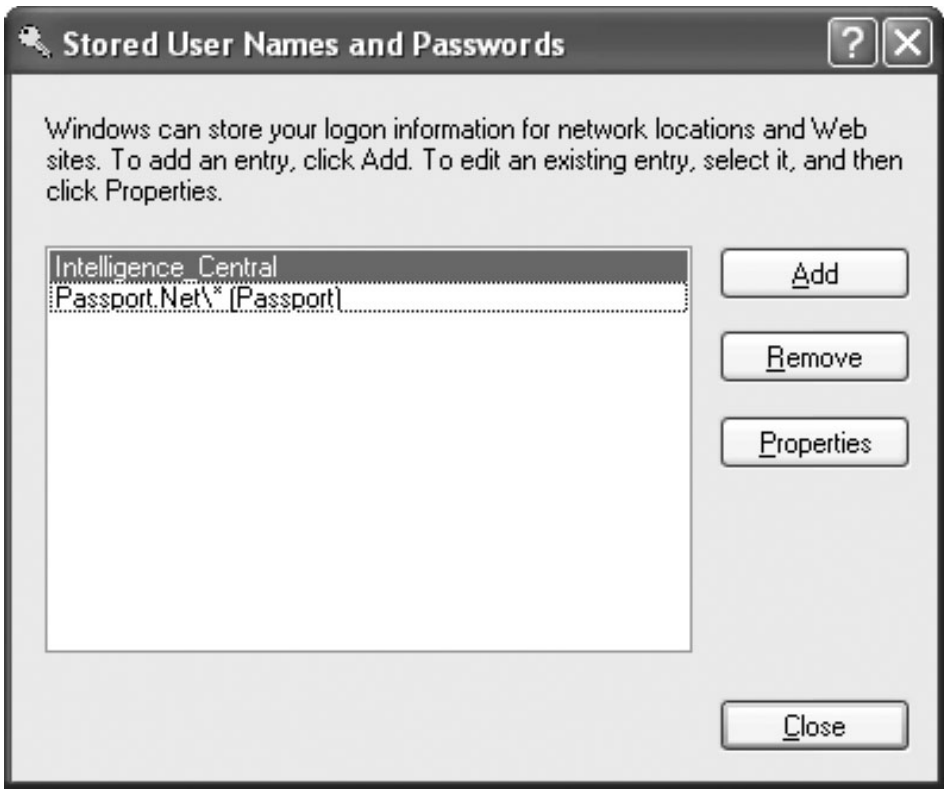

Figure 16.16 Manage Stored User Names and Passwords from within User Accounts in Control Panel.

You can also save credentials from a command line, if you prefer to use the command line rather than a GUI tool. The **net use** command provides the **/savecred** switch, used when the user is prompted for a username and/or password. The syntax for this command is either of the following examples:

1. When prompted only for a password:

```
net use * \\computer_name\share_name /savecred
```

2. When prompted for both username and password:

```
net use * \\computer_name\share_name /u:domain_name\user_name
/savecred
```

Note that /savecred is used only in conjunction with **devicename** and is ignored when the remote resource does not prompt the user for credential presentation automatically. /savecred, when used with password *, only caches those

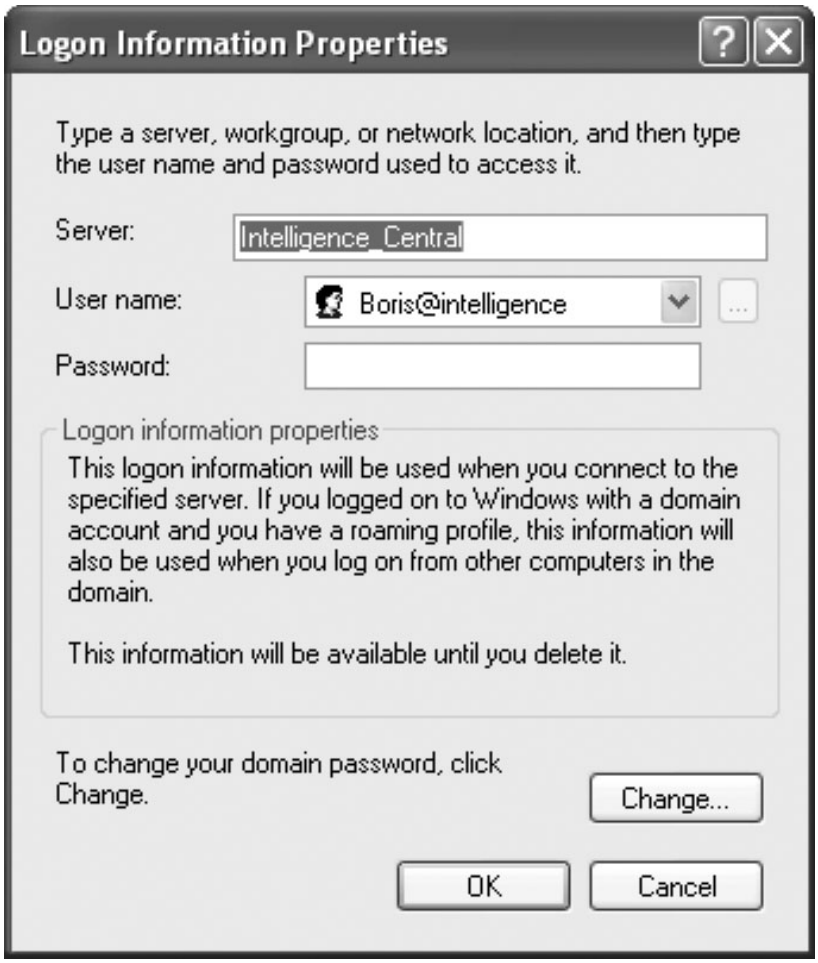

Figure 16.17 Provide Server Name, User Name, and Password in Logon Information Properties.

credentials for the duration of the logon session, rather than saving them in Stored User Names and Passwords. Further, /savecred is only available in Windows XP Professional. It is ignored in Windows XP Home Edition.

Global Account Settings

Settings that affect all accounts are configured from within the Local Computer Policy console. They are grouped by function, which is how they are covered here. To open this console, launch **Control Panel**, then select **Administrative Tools**, then select **Local Security Policy**. Alternatively, you can open the Local Computer Policy snap-in within the Microsoft Management Console. The groups discussed in this section are found in **Account Policies** and **Local**

Policies. Local account policies are applied in stand-alone Windows XP workstations, while domain members logging on to a Windows XP computer are subject to domain account policies.

Account Policies

Account policies include both password and account lockout–related settings. Password policy dictates password strength and lifespan, while account lockout specifies what, if anything, happens in the event that a user attempts to logon too many times with an incorrect credential set. To access these options, navigate to **Computer Configuration**, then **Windows Settings**, then **Security Settings**, then **Account Policies**, in the left-hand pane.

Let's look at Password Policy first. Clicking the **Password Policy** folder under **Account Policy** displays the configuration options, as shown in Figure 16.18.

All of the options here are configured in accordance with a well thought out password policy that you must devise for your network. In addition to keeping passwords secret and safe, they should also be difficult to guess. You can accomplish this by requiring them to be strong, unique, and changed frequently.

- **Maximum password age:** Requiring users to change their passwords frequently helps prevent fraudulent use of passwords. The shorter the duration you set, the more often a user has to create a new one. This can be difficult for users when initially implemented, especially when this imple-

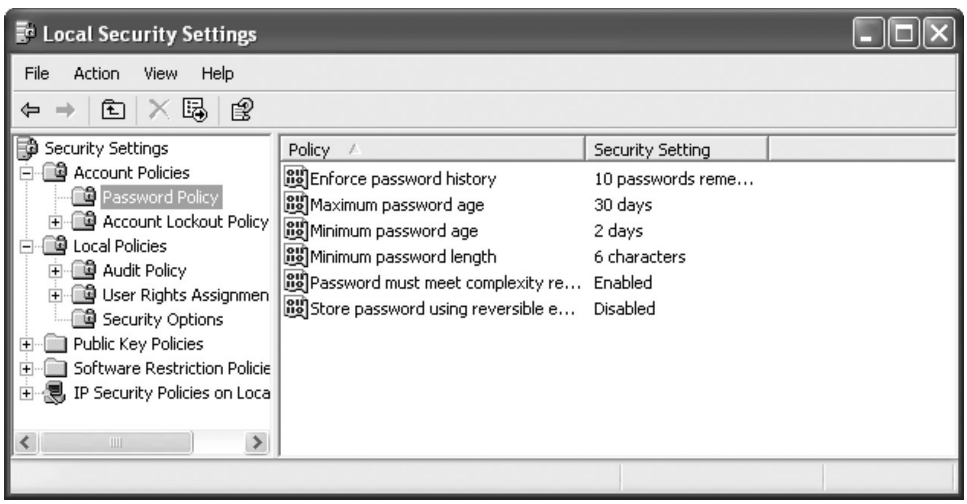

Figure 16.18 Password Policies are managed from within the Local Security Policy console.

mentation is in conjunction with other settings. It also means an additional level of work for you, the administrator, who may have to assist with the reset of their passwords (although the Password Reset Wizard can provide some relief). You need to balance the aging requirement with the actual level of security requirements in your network.

- **Enforce password history:** This is a great one to set to keep users from defeating the requirement to change their password every so many days. Some crafty users will try to change their password to the exact same password, which really does not do much in the way of keeping things secure. By setting a history, users have to create a unique password each time they are forced to change their password.
- **Minimum password age:** This is another way to defeat the users who want to reuse the same password. This prevents him or her from changing the password *n* times, where *n* represents the number of unique passwords required in a single day, and reverting back to their original password.
- **Minimum password length:** By specifying a specific length of a password, you are doing two things—making passwords difficult to guess and preventing the use of blank passwords. Six characters is a recommended minimum, while 14 characters is the maximum permitted. By default, Windows XP workstations that are not part of a domain prevent users with blank passwords from logging on over the network. This is especially helpful for preventing unauthorized access to home workstations connected to the Internet. All blank password access is restricted to local logons only.
- **Passwords must meet complexity requirements:** In addition to requiring passwords of a minimum length, you can also require that they consist of both alphanumeric and special characters and use both upper- and lowercase letters. In addition, by enabling this policy, Windows XP prohibits users from including their username or their "friendly name" as configured in their account properties sheet as part of their password.

NOTE: To make it easier for users to remember long and seemingly random combinations of letters, symbols, and numbers, encourage them to create "passphrases." A good strong passphrase can be a nonsensical expression such as "Bob is your uncle" or "The White Sox won the pennant!" but with a twist—substitute numbers and symbols for letters and spaces. "Bob is your uncle" becomes B0b_1s_y0ur_unc1e, and "The White Sox won the pennant!" becomes The_Wh1te_S@x_W0n_the_pennant! Voila! Instant, difficult to guess (and type), but easy to remember password.

- **Store passwords using reversible encryption for all users in the domain:** This option should be selected only when CHAP authentication is required for Remote Access Service or Internet Authentication Service. Enabling this is equivalent to storing passwords in plain text, which is a definite security no-no! Enable only when absolutely required.

Now that you have made it difficult for fraudulent users to guess passwords, you need to limit their opportunity of trying to crack those passwords with dictionaries or manual attempts. The account lockout options are designed to do just that. These options are found in the same Account Policy area as Password Policy, shown in Figure 16.19.

- **Account lockout threshold:** The first option you need to configure is the number of bad log-on attempts permitted before the account is administratively disabled. By default, this is set at zero or disabled. No other account lockout options can be set until the threshold is set to some number above zero. The maximum number of attempts allowed is 999. The more secure the network, the lower this number should be. Three to five attempts should be adequate for most networks.
- **Account lockout duration:** This setting sets the amount of time that an account will be locked out. In other words, how many minutes will elapse from the time the account is locked out to the time that the account is unlocked and log-on attempts can begin again. A setting of zero means

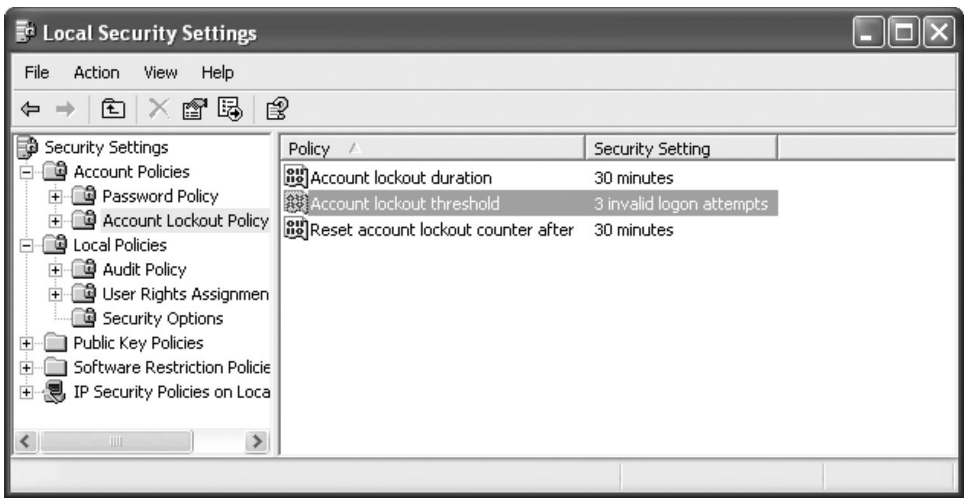

Figure 16.19 Account Lockout Policy dictates when and how user accounts are disabled after too many bad attempts to log on.

that the administrator must manually unlock the account. Setting the duration to 25 minutes strikes a nice balance between making a dictionary attack difficult and letting the administrator have enough time to get out of the office for lunch without having to return to unlock someone's account.

- **Reset account lockout counter after:** This setting provides yet another way to cut down on some of the administrative work of unlocking accounts. Windows XP tracks the number of bad log-on attempts in preparation to locking out an account. Specifying a span of time between a number of bad log-on attempts that are not sufficient to lock out the account and the point at which the bad log-on attempt counter is set not only slows down an intruder guessing passwords, but also provides an additional bit of room for error for a user who may have repeatedly fat-fingered a recently changed password.

When setting an account lockout policy, it is important to set up auditing and check the logs any time a user account is locked out and you are made aware of it. This way you can become aware of any suspicious log-on activity as soon as possible.

Changing the Way Users Log On

There are a number of ways the log-on experience for users can be modified, from requiring smart cards to automation of the log-on process to policies that limit time and location of logons. This section reviews these log-on process modifications.

Smart Cards

Smart cards are an excellent way to physically and logically secure access to your computer's resources. A smart card is very similar to an ATM card: In order to complete a transaction, both the card and a PIN are required. The card is read in a reader attached to the computer, and the user is prompted to enter his or her PIN. When smart cards are used, this credential combination is used in place of the account password stored within the operating system.

Smart cards use public and private key cryptography and can only be used for domain account authentication; local account authentication is not supported on Windows XP. Kerberos version 5 protocol is used with smart cards to authenticate to Windows 2000 domains, while Kerberos with x.509 v3 certificates are used with non–Windows 2000 domains. Unlike passwords, smart card PINs are not sent over the network. Like Windows accounts, smart cards can be set to lock after too many bad log-on attempts. An administrator must unlock any locked cards.

Automating Logon

Going in the opposite direction from the security provided by smart cards is the ability to automate user logons. When automated logon is configured, users do not have to use the Control + Alt + Delete command to initiate logon, nor do they enter a username or password. Instead, user credentials are stored in the registry, which is then parsed for this information when required for loading a specific user's operating system environment.

It is highly recommended that any computer configured with automatic logon be physically secure and protected from unauthorized network access. While automatic logon offers a great deal of convenience to users, because they can boot up and get right to work on their Windows XP computer, it also leaves a gaping hole where a certain level of security would have been provided by entering user credentials. All users of that system have access to all network resources that are granted to the account that was logged on automatically. As you can imagine, this is a dangerous situation. Even weak passwords can provide a higher level of local logon deterrence to unauthorized users. The log-on screen works in much the same way a sign alerting intruders to the presence of a burglar alarm works. A trespasser is not sure if there really is an alarm system, and may not be willing to risk finding out one is truly installed. An unauthorized user may not be willing to take the 10 or 15 extra seconds to figure out what the weak password is before he or she is caught with the proverbial hand in the cookie jar. Another danger point with automated logons is that user passwords are stored in plain text in the registry. Any remote user that is a member of the Authenticated Users group by default has access to the key where this is stored. This makes it an easy target for users looking for unauthorized access to the files of another user.

To set up automated logon you must edit the registry. As always, use extreme caution when making modifications to the registry and be sure you have a good backup before you start. Any registry change can have unexpected and unpleasant results. When you are ready to proceed, you perform the following steps:

1. Open a registry editor, such as **regedit.exe**.
2. Locate HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.
3. Double-click **DefaultUserName**.
4. In the Value data box, type the username, and then click **OK**.
5. Double-click **DefaultPassword**.
6. In the Value data field, type the password and click **OK**.

7. Double-click **AutoAdminLogon**.
8. Enter **1** in the Value data box and click **OK**.

Changes take effect on the next reboot.

Disabling the Welcome Screen

The Windows XP Welcome screen provides the usernames of all users with local accounts on a computer. Users viewing this screen can attempt to log on with the username of another user by guessing the password. If this is an unacceptable risk in your network and you are not in a domain, you can use the traditional Control + Alt + Delete screen instead. To do so, perform these steps:

1. Open **Control Panel**, click **User Accounts**.
2. Click **Change the way users log on or off**.
3. Remove the check in the **Use the Welcome screen** check box.

Authentication Policies

A number of specific settings can be applied through policies defined either locally, across a number of computers, or domainwide. Local policies on a single computer are configured with the Local Security Policy tool. Settings that are to be applied to a number of computers are configured with the Group Policy tool, and templates can be used to make configuration of a number of options easier. Domain account policies are defined in the Domain Group Policy. Account policies were discussed earlier in this chapter, so this section provides an overview of the authentication policy options that are available for the Windows XP network administrator in a nondomain environment. To modify any of these policies:

1. Launch **Control Panel**, then select **Administrative Tools**, then select **Local Security Policy**.
2. In the left-hand pane of the console select **Local Policies**, and then select the appropriate folder for the option you wish to modify.
3. In the right-hand pane, right-click the policy you wish to modify and select **Properties** from the pop-up menu. The property sheet for that policy appears. Make changes as appropriate for the policy you are configuring.

User Rights Assignment

User rights pertain to the specific activities that users may perform, such as loading or unloading device drivers, limiting access to specific computers, or changing the system time. These rights can be assigned to a user or a group; Microsoft's stance is and always has been that it is better to set permissions and rights by group, rather than by individual users. If you only have a handful of users in your network, management by group may be more difficult than administering each user independently. However, if you have more than 10 to15 users, group management is usually easier in the long run. Regardless of whether you manage by users or groups, assigning rights uses the same process: Double-click the policy option under the **User Rights assignment** folder, click the **Add User or Group** button, and supply the appropriate user or group name(s).

The following user rights deal with authentication-related issues.

- **Access this computer from the network:** Permits a user to connect to a specific computer via network.
- **Deny access to this computer from the network:** Prohibits a remote user or group from accessing a specific computer from the network. Useful for securing servers with large numbers of directories that should not be accessed by anyone other than a specific group or user.
- **Allow logon through Terminal Services:** Users are permitted to connect to the computer via terminal services.
- **Deny logon through Terminal Services:** Disables terminal service access to specific users or groups, while still allowing those with "Allow logon through Terminal Services" rights to continue doing so.
- **Log on as a batch job:** Allows a user, such as a service account, to log on by means of a batch job.
- **Deny logon as a batch job:** Prevents specific users and groups from logging on as a batch.
- **Log on as a service:** Enables accounts to log on as a service, in order to access resources on behalf of an application.
- **Deny logon as a service:** Prevents specific users and groups from logging on as a service.
- **Log on locally:** Permits users to use this computer locally.
- **Deny logon locally:** Prohibits specific users or groups from using the computer locally. Useful in instances where a computer is in a public area but you want to limit the use to a specific user or two.

Security Options

Security Options govern specific configuration options that are used to increase or decrease security on a given computer. These options are assigned to the computer, not users or groups, unlike User Rights Assignment. All users of this computer are subject to these policies. The following Security Options deal with authentication-related issues and are located in the Security Options folder under Local Security Policy. To modify a policy, double-click it and make the necessary changes.

- **Do not display last user name:** Hides the last username in the Control + Alt + Delete log-on screen. This makes it difficult for fraudulent users to access your network by guessing passwords based on known usernames.
- **Do not require CTRL+ALT+DEL:** Eliminates the need to press Control + Alt + Delete to log on. This puts the user credentials at risk, as the MSGINA is no longer able to use a secure channel to transmit credentials, and a fraudulent user can capture the password, even without sending data over a network.
- **Message text for users attempting to log on:** You can specify a legal warning or disclaimer that appears before users are permitted to log on.
- **Message title for users attempting to log on:** This controls the title of the window where pre-log-on messages appear. Title suggestions include Warning, Notice, and Attention.
- **Number of previous logons to cache** (in case a domain controller is not available)**:** Sets the number of log-on attempts to be stored by a domain account. This is helpful for remote computers where an Internet connection must be made before a domain controller can be contacted. If the ISP has unreliable service, a domain user can be locked out if the connection cannot be made. The downside of this is that a fraudulent user can disconnect a computer from the network and use recently changed credentials to log in to the network without being blocked by a domain controller. Zero disables caching.
- **Prompt user to change password before expiration:** Gives users a warning that they will be required to change account passwords in the near future.
- **Require domain controller authentication to unlock:** Prevents a user from using cached credentials to log on to a computer that has been locked.

- **Smart card removal behavior:** If smart cards are used, you can determine what will happen if a card is removed while the computer is in use. The choices are Lock workstation, Force Log off, and No action.
- **Allow anonymous SID/Name translation:** Enables anonymous users to translate SIDs into usernames and vice versa.
- **Do not allow anonymous enumeration of SAM accounts:** Prohibits anonymous users from enumerating the SAM.
- **Do not allow Stored User Names and Passwords to save passports or credentials for domain authentication:** Prevents passport or domain authentication credentials from being saved after user has logged off, if Saved Passwords has been configured.
- **Sharing and security model for local accounts:** Guest only requires all network logons to use the Guest account. Classic allows network users to use their own credentials.
- **Let Everyone permissions apply to Anonymous users:** Grants Everyone group permissions to anonymous users.
- **Do not store LAN Manager hash value on next password change:** LAN Manager hash is deleted after a password change.
- **Force log off when log-on time expires:** Requires a user to be forcefully logged off the computer if a time constraint has been placed on his or her account. The account equivalent of Cinderella's carriage becoming a pumpkin at midnight.
- **LAN Manager Authentication Level:** Sets LAN Manager authentication requirements for networks with down-level Windows hosts.
- **Minimum session security for NTLM SSP based (including secure RPC) clients:** Permits configuration of the following options:
  - Require message integrity
  - Require message confidentiality
  - Require NTLMv2 session security
  - Require 128-bit encryption
- **Allow system to be shut down without having to log on:** Permits or prohibits a computer from being shut down without the user being logged on.

Conclusion

In this chapter we covered authentication, the process of verifying the identity of the user attempting to access a computer or other network resource. Once a

user's identity has been proven to a reasonable level, authorization can occur. Windows XP Professional can provide total authentication services or it can interoperate with Windows NT, Windows 2000, and Windows 2003.The specific procedures of Windows XP authentication were covered including Kerberos and NTLM and how the log-on process works with each of these authentication types. Also covered in this chapter were new authentication features within Windows XP and best practices for configuration and management of Windows XP authentication. From here we move to the next step for a user to access a network resource or computer. Authorization, or determining that a verified user has been granted permission to access a specific resource or given the right to perform a specific task, is covered in the next chapter.