



CHAPTER 1

Security Settings

Group Policy is the single most comprehensive and useful feature for managing security in Windows 2000/XP networks. However, it is the security settings specifically, or *Local Security Policy*, that is the crux of some of the most important security configuration options available.

In pre-Windows 2000 days, an administrator had to navigate through a hodgepodge of user interfaces and registry settings in order to properly secure a system. Windows NT 4.0 tools such as Security Configuration Editor and System Policy were the best things available for managing security in a Windows NT network. Then came Windows 2000 and the introduction of Group Policy and the Security Settings user interface (UI). With the evolution to Windows XP/.NET, Group Policy has taken on even more functionality with over 200 new administrative template settings, including new categories such as Software Restriction Policies and Wireless Network Policies.

KNOW YOUR INTERFACES

There are quite a few interfaces available, several of which ultimately do the same thing. A few things should be familiar, such as the MMC (Microsoft Management Console), snap-ins, containers, and objects. These terms and concepts were introduced late in the lifetime of Windows NT 4.0, and were completely integrated into the release of Windows 2000. They continue to play an important role in the administration of Windows XP/.NET systems as well.

The MMC is Microsoft's all-in-one tool for managing the Windows environment. Click Start → Run and type **MMC** to launch it. The MMC basically provides a framework for *snap-ins* to run. The snap-ins provide interfaces to manage most everything on a Windows system or network. For example, in this chapter, we will be using the Local Security Settings snap-in, previously named Local Security Policy. (We will use these terms interchangeably.) To launch the MMC for Local Security Settings, type **secpol.msc** at the command prompt.

Containers and objects can be equated to folders and files, respectively. For example, after launching Secpol.msc, you will see several containers on the left-hand side, as shown in Figure 1-1.

In the MMC and Active Directory alike, containers are everywhere. For example, a domain is a container, and so is an organizational unit. You will see default containers for users in which each user is considered an object. (We cover Active Directory in detail in Chapter 11.)

Since there is often some confusion regarding the tools used to implement security policy, let's take a quick look at what separates them:

- **Local Security Policy (Secpol.msc)** Renamed Local Security Settings in Windows XP/.NET, this is the interface for configuring some of the most important "security settings" related to the operating system. These settings are included as a subset of Group Policy, and contain the relevant configurations for Password Policy, Audit Policy, Security Options, Software Restriction

Policies, and IP Security Policies. Configurations set through Local Security Policy are only applied to the local machine.

- **Group Policy (Gpedit.msc)** This is the MMC snap-in that is used to edit and apply group policy objects (GPOs). It is accessible either by running Gpedit.msc from the command line or by right-clicking a domain or organizational unit in Active Directory and selecting Properties → Group Policy. *Group Policy* is basically a concept term that refers to the use of settings that control computer and user configurations. It is a pretty straightforward concept: you apply the same computer and user configurations to a group of computers or users. On a local machine, Group Policy can also be used to apply local configurations.
- **Group Policy Object** GPOs are a logical collection of specific settings that control operating system and application behavior on a computer or user basis. You can apply one or more GPOs to a relevant Active Directory container such as a domain or organizational unit.
- **Security Configuration and Analysis** This powerful toolbox provides a means of configuring all the security settings related to Local Security Policy as well as many more related to the file system, registry, restricted groups, and services. Access the Security Configuration and Analysis snap-in by opening the MMC and selecting File → Add/Remove Snap-in. This tool, which reappears throughout this book, can be used to perform the following tasks:
 - Analyze system security based on a security template
 - Apply system security based on a security template
 - Export a security template file that can be imported into an Active Directory container GPO

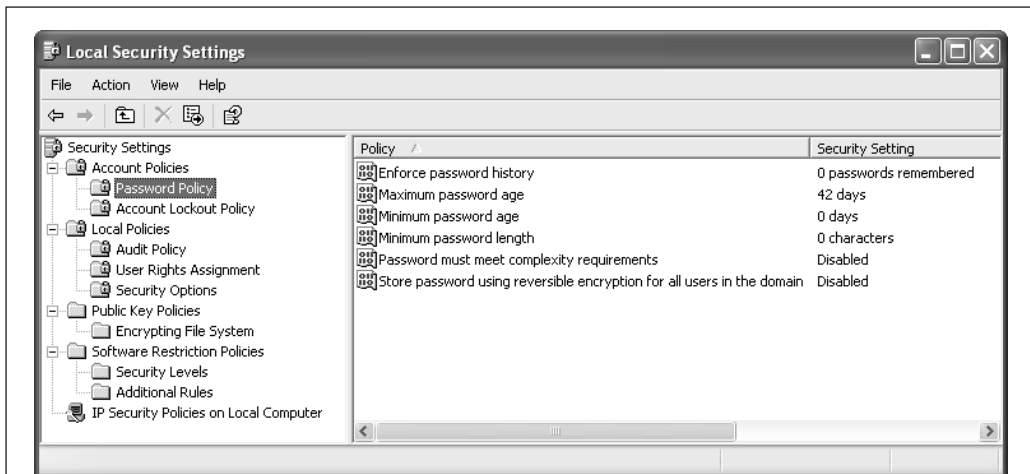


Figure 1-1. Local Security Settings—containers on the left, objects on the right

SECURITY SETTINGS

The core of this chapter is a discussion of security settings. These are configurable locally via Local Security Policy (Secpol.msc) and across a network via Active Directory and the Group Policy Security Settings container. One thing that should come across in the end is the granularity of Windows security. Files are just one of the 14 or so objects capable of being protected by Access Control Lists (ACLs). Others include such obscure objects as named pipes, mutexes, semaphores, events, and devices.

As you go through these security settings, consider how you would use them throughout your network. Remember that with a well-thought-out plan they can most effectively be applied to groups of users and computers through Active Directory. The appendix to this book provides a concise table listing the recommended values for each of these settings.

Account Policies

This container provides settings for password and account lockout security. It is worth repeating that these settings are most effectively enforced at a domain or organizational unit level within Active Directory. These provide one of the most basic means of networkwide security.

Password Policy

Planning and implementing a password policy is crucial. If you do not configure password policies for your domains, then you allow your users to create weak passwords that are never required to be changed. In such a scenario, an attacker will have a much higher degree of success trying basic password-guessing attacks. The following password policy options are available.

Enforce password history Windows will prevent your users from reusing passwords by remembering their old ones. Set the number of passwords you want Windows to remember; otherwise set this to zero (default).

Maximum password age Specify how long in days a password is valid (42 days is the default). At that time it expires and the user is forced to change it. Setting this to zero will prevent a password from ever expiring.

Minimum password age To prevent users from working around the “Enforce password history” setting, set the minimum time in days that a password must be used before it can be changed. For example, if you had a password history of 5 defined, a user could simply change their password 5 times within the same day to get to their old favorite again.

Password must meet complexity requirements Enable this setting to make people use strong passwords. By enabling this setting, Windows will enforce the following minimum password requirements:

- Cannot contain even part of a person's username
- Must be six characters long
- Must contain characters from at least three of the four categories: A–Z, a–z, 0–9, and alphanumeric, such as \$, !, #, %

Store passwords using reversible encryption This setting should never be enabled, as it essentially stores passwords in plain text instead of as a hashed value.

Account Lockout Policy

These settings determine when an account should be locked out and for how long. They should always be configured in a domain to provide maximum security. Without lockout policies defined, you give attackers free rein to guess passwords all day long.

Account lockout duration Specify the time in minutes that an account should be locked out for after a specified number of failed logon attempts have been made (see “Account lockout threshold”). It is often standard practice to set a lockout duration such as 30 minutes, after which time the account is automatically unlocked for use again. It is up to you to decide whether you want this automatic unlock feature enabled, or whether you want your administrators to know about and react to every account lockout (not always a bad idea).

Account lockout threshold This has to be set before any of the other options in this section can be configured. A standard recommended threshold is five failed login attempts, but it's really up to your organizational policy.

Reset account lockout counter This must be set to less than or equal to your account lockout duration (e.g., 30 minutes). This is the time in which failed logon attempts are being tallied up. If you set it to 30 minutes, then failed logins will be tallied until either the threshold is reached and the account is locked out or the lockout counter is reset.

NOTE What happens when somebody starts guessing passwords in the middle of the night? If you have lockout duration set for 30 minutes, and allow five invalid logon attempts before locking out an account, then an attacker can guess about ten passwords in an hour.

Local Policies

This most popular container has very important settings for auditing, user rights, and other security options.

Audit Policy

All auditing is disabled by default. It is important to know just what you want to audit and how to enable it. Some of the overlap in terminology can get a bit confusing (such as “account logon events” versus plain old “logon events”) if you are not certain of exactly what you want to log.

CAUTION All auditing is disabled by default in Windows. Make sure you enable auditing in a practical manner.

Audit account logon events This setting is more suited to a domain controller, because you are not auditing where an account logon/logoff occurs but rather where the account lives and is validated. This setting will record every remote logon attempt (success or failure) and logoff, for which this computer confirms the account. For example, if you use a domain account to log on to a workstation named Cottons, this “account logon” will show up in the security event logs of the domain controller that validated the account credentials.

Audit account management This setting logs any successful or failed changes to user or group accounts. These changes include adds, deletes, and modifications such as changing group membership, renaming, or setting a new password.

Audit directory service access This setting logs successful and/or failed Active Directory object access attempts. It is not applicable to a workstation, only to a domain controller.

Audit logon events Once you enable this setting to log success and failure, all logon and logoff events to the local machine will be logged. This setting is suitable to any machine, including workstations and domain controllers. Logon events to the local computer include console logon/logoff, network logon, and RDP/terminal services logons. For example, whether you use a domain or a local account to log on to a workstation named Cottons, the event will be logged in the local security event logs of Cottons.

Audit object access When this setting is enabled, objects such as NTFS files, folders, registry keys, and printers will be ready for auditing. However, you still have to manually configure SACLs, or auditing settings, on the devices and actions you want to log. Additionally, enabling auditing here prepares the system to log actions on global system objects, such as mutexes, semaphores, and events.

Audit policy change Policy changes include modifications to user rights assignments, audit policies, and trusts.

Audit privilege use You can log each occurrence of an account exercising a user right by enabling this audit setting. Not all user rights, however, will actually be audited. Rights such as debugging programs, creating a token object, backing up files and directories,

and restoring files and directories will not be audited unless you enable the “Audit: Audit the use of Backup and Restore privilege” setting in Security Options.

Audit process tracking You can have a log entry created every time a process is successfully or unsuccessfully executed and exited. Process tracking is not normally something you would want to audit successes on, because of the large performance overhead it carries. Many organizations do like to audit failures here so that they can see which processes users are attempting to launch without correct permissions.

Audit system events It’s pretty important to audit such events as system restarts or shutdowns, so this setting should definitely be enabled for both successes and failures.

User Rights Assignment

This section lists a description of each user right as offered in the Local Security Policy GUI. Chapter 6, on user and group management, will go into more detail on some of these specific settings. Through the GUI you can control which users and groups have each right.

Access this computer from the network The groups and users given this right will be allowed to connect remotely to the local computer via SMB sessions (i.e., basic Windows shares). As a general rule, you should define only the specific groups that need remote access via SMB. Remove the Everyone group (added by default) and replace it with Authenticated Users (at a minimum), and remove Guest. This setting has no effect on other services such as FTP and Terminal Server.

Act as part of the operating system There is a good reason that no users have this right by default. A user who possesses this low-level privilege can essentially bypass all other security permissions, rights, and privileges on the system. For example, an anonymous token could be created that includes any or all access permissions, thus evading security and auditing.

Add workstations to domain This right lets the specified users or groups add new workstations to the domain. Many large organizations like to give this right to all Authenticated Users, which is a questionable practice. Obviously, unauthorized or improperly secured machines could be added to the trusted domain. Once on the domain, a computer will be updated with any domain services and service accounts, for which passwords are stored in clear text in the registry.

Adjust memory quotas for a process Adjusting the amount of memory available to a process may be necessary for fine-tuning a system, but it could be used either intentionally or unintentionally to create a denial of service type of attack.

Allow logon through Terminal Services Users can be explicitly allowed the ability to log on through terminal server/RDP sessions.

Back up files and directories Do not give this right away lightly. It defaults to Administrators and Backup Operators, and users allowed the “Back up files and directories” right can essentially bypass all file and folder DACLs (Discretionary Access Control Lists) in order to back up a file. The main thing to keep in mind here is that you don’t want the same user having both backup and restore rights simultaneously.

Bypass traverse checking Users with this right are allowed to traverse directories that some ACL denies them access to. The users cannot list directory contents, but only pass through the directory to get to another one. For example, say you have three chained directories, C:\temp\middle\target. Your user account is allowed access to \temp and \target, but denied access to \middle. With the “Bypass traverse checking” right, you are allowed to pass through the \middle directory in order to access the \target subdirectory.

Change the system time Because accurate system time is critical to auditing, application, and authentication functions such as Kerberos, this right is limited to Administrators and Power Users by default. Depending on how much power they really need, best practice indicates that Power Users should be removed so that only Administrators hold this right. Keep in mind that changing the system time can corrupt audit logs and disable Kerberos—and don’t give this right away lightly.

Create a pagefile Administrators are the only ones who by default can create the pagefile on a Windows system. The pagefile is necessary to normal computer operation, and it is security sensitive because it acts as virtual memory, storing process variables and functions. Unless you have a unique situation, Administrators should be the only group with this right.

Create a token object No accounts should ever possess this right, unless you want to completely jeopardize all other security defenses. This right provides the ability to call system APIs that create an access token which defines an account’s permissions at logon. If access tokens can be arbitrarily created then there is no security. Access tokens are described further in Chapter 6.

Create permanent shared objects Certain kernel mode components already possess this right, which they use for extending an object namespace. This right should never be given to a user or group account.

Debug programs Administrators only have this right by default, and they should be removed from it. This right allows a user to attach a debugger to a process, which may be necessary for developers and Visual Studio installations. Having this right also allows tools such as LsaDump2.exe to be used; it dumps clear text service account passwords from the registry.

Deny access to this computer from the network Use this right to force accounts to log on at the console, instead of via SMB across the network. They will receive a “permission denied” error message when attempting SMB connections remotely. A Deny rule takes precedence over an Allow rule, so, for example, if an account is given both the Allow and Deny access right, the remote logon will be denied.

Deny logon as a batch job Batch job logons may be used instead of interactive logons. A batch logon can be executed, for example, by the Schedule service. A Deny rule takes precedence over an Allow rule, so, for example, if an account is given both the Allow and Deny logon right, the logon will be denied.

Deny logon as a service Service accounts are used to register a process as a service. Use this setting to specifically deny an account this ability. A Deny rule takes precedence over an Allow rule, so, for example, if an account is given both the Allow and Deny logon right, the logon will be denied.

Deny logon locally Accounts specified here will not be allowed to log on at the console. They can still log in via Terminal Services, Telnet, SMB, and other services. As usual, a Deny rule takes precedence over an Allow rule.

Deny logon through Terminal Services Accounts specified here will be explicitly denied logon via Terminal Services. They can still log in locally via the console, Telnet, SMB, and other services. Again, a Deny rule takes precedence over an Allow rule.

Enable computer and user accounts to be trusted for delegation Typically, on a client workstation such as Windows XP, no one will have this right, as is the case by default. Delegation is a Windows function used most often in multitiered applications where servers need to pass user contexts from one to another. Users with this right have the ability to set the “Trusted for delegation” setting on a user or computer object in Active Directory. This option is accessible on a user object in Active Directory, for example, under the Account tab and Account options. On a Windows 2000/.NET domain controller, Administrators possess this right by default.

Force shutdown from a remote system Users with this right can call APIs to shut down the computer from a remote system. There is not much reason to let anyone but the Administrators group have this right.

Generate security audits A process will be allowed to use accounts specified here to add events to the Security Event log.

Increase scheduling priority Users with this right can increase the priority of a process. For example, a process’s priority can be increased through the Task Manager by right-clicking the process and selecting Set Priority. The highest definable level is named Realtime, and it essentially commits the operating system to give all available CPU

resources to the process. Setting high priorities can quickly consume all computer resources and result in a denial of service attack.

Load and unload device drivers Device drivers run in privileged kernel mode and should therefore be scrutinized and controlled. Best practice dictates that only Administrators have this right, as is the case by default.

Lock pages in memory Locking pages in physical memory (RAM) means preventing them from being written to the system pagefile. If someone were to abuse this right, they could create a denial of service condition by consuming all available RAM while preventing the use of the pagefile. For good reason, nobody has this right by default.

Log on as a batch job Batch job logons may be used instead of interactive logons. A batch logon can be executed, for example, by the Schedule service.

Log on as a service Service accounts are used to register a process as a service. Use this setting to specifically allow an account this ability. Accounts will be automatically given this right once you specify that a service can “Log on as” a certain account through the Service GUI of the MMC (Services.msc). Be aware that a service accounts password will be stored in clear text in the registry, so refrain from using domain administration level service accounts.

Log on locally Accounts must have this right if they are to have console access. This setting has no effect on an account’s logon permissions for Terminal Services, Telnet, SMB, and other services, including IIS (see the following Caution). Best practice dictates that the defaults here are too permissive, and should be tightened up to only allow the exact groups who need it. This typically means removing all groups except the Administrators and Users groups.

CAUTION When running IIS, the IUSR account must be given the “Log on locally” right. That may not be of as much concern, however, as that any user who needs to access a Basic Authentication-protected virtual directory also needs the “Log on locally” right! That’s correct—if you have a site set up to require Basic Authentication, then any remote user needing access will require this right.

Manage auditing and security log Users with this right have two very important capabilities. First, they can enable auditing on individual objects such as files, folders, and registry keys. Second, they can view and clear the Security Event log. For this reason, it is recommended that only Administrators carry this right.

Modify firmware environment values This right allows an account to modify the systemwide environment variables, as opposed to the individual user environment variables. Because system environment variables can be used to point at malicious programs, it is recommended that only Administrators possess this right, as is the default.

Perform volume maintenance tasks These tasks include using such built-in APIs as disk cleanup and disk defragmenting, both actions that should only be carried out by Administrators.

Profile single process This setting defines who can monitor the performance of non-system-related process counters.

Profile system performance This setting defines who can monitor the performance of system-related process counters.

Remove computer from docking station Users with this right can undock a workstation without logging on. Without this right, a user must log on and use the Start → Eject PC menu option to undock the computer. Obviously, without physical controls, a computer could be forcefully undocked regardless of this right.

Replace a process level token An account with this right can use a parent process to replace the access token for a subprocess. This is a highly privileged right, usually reserved for kernel mode components.

Restore files and directories As with the complementary “Back up files and directories” right, users allowed this right can essentially bypass all file and folder permissions (DACLS) in order to restore a file. It defaults to Administrators and Backup Operators, which means that your secret design plans can be restored, or overwritten, by any member of these groups, despite your specified DACL. Again, the thing to keep in mind is that the backup and restore rights should be separated between different user accounts.

Shut down the system Users with this right can shut down the system, typically something that needs to be done for a workstation, especially a laptop that travels.

Synchronize directory service data This right is relevant only to domain controllers and gives an account the ability to synchronize directory service data (i.e., the Active Directory database).

Take ownership of files or other objects A user that can take ownership of objects (including files, folders, and registry keys) can bypass all security permissions on that object. *Read that one more time.* This functionality is designed to give administrators the ability to access files and folders regardless of their DACL. For example, if an employee hastily leaves your organization, you may still want access to their protected files.

Security Options

One of the first things seasoned administrators will notice is that the settings in this container have been newly categorized. In Windows 2000 they were simply presented in alphabetical order. In Windows XP/.NET they are presented in this format:

Category: title or short description

The *Category* piece is fairly straightforward, meaning that “Accounts” will be settings applied to user accounts, while “Network Security” will be settings meant to affect network security. The short description is not always as straightforward, which is why we are presenting a longer description in this chapter. The appendix will include a table of the recommended settings.

Accounts: Administrator account status This setting is not applicable (by default) unless you have created another user account that is a member of the local Administrators group. There is nothing fancy here; by setting this to Disabled, you are disabling the built-in Administrator account just as you could through another UI such as the Local Users and Groups MMC snap-in.

Accounts: Guest account status This is another place from which you can enable or disable the local Guest account, which is disabled by default. Be aware that if you have the “Network access: Sharing and security model for local accounts” setting set to “Guest only,” then the local Guest account must be enabled or else network logons (such as SMB-based logons) will fail.

Accounts: Limit local account use of blank passwords to console logon only This setting is enabled by default. Remote interactive logons, such as those by Terminal Services and Telnet, will not be allowed if the local user account being used has a blank password. Additionally, remote connections to SMB services (such as network shares) will not be allowed for users with blank passwords.

Accounts: Rename administrator account This is another example of how security settings that can be made in other user interfaces are being brought together under one roof. There are several reasons for this, including making the administrator aware of what can be done and providing the ability to propagate such settings through Group Policy. Rename the Administrator account to something less obvious if you want to, but be aware that its RID (Relative Identifier) will always be 500 (see Chapter 6). A more paranoid technique is to actually rename and disable the built-in Administrator account.

Accounts: Rename guest account Renaming the Guest account to something less obvious is another arguably weak attempt at obfuscation.

Audit: Audit the access of global system objects Disabled by default, when enabled, this setting will set Windows to create system objects such as mutexes, events, and semaphores with auditing enabled. This is usually unnecessary in most environments, but it illustrates the level of security configuration granularity that Windows provides. Enabling this setting is a prerequisite for auditing system objects. You will then actually have to set an audit policy for “Audit object access” for events to show up in your logs.

Audit: Audit the use of Backup and Restore privilege Disabled by default, this setting requires that you have an audit policy set up for “Audit privilege use” before events will actually be written to your logs. Once you enable it, however, all user privileges will be audited, including the use of the backup and restore privileges.

Audit: Shut down system immediately if unable to log security audits This setting is disabled by default for good reason. You would only want to enable this in high-security settings where the system should not operate unless it can log security events. Once enabled, the system will actually crash with a Stop error BSOD (blue screen of death) when the security event log is full and cannot be written to.

Devices: Allow undock without having to log on This setting only works for laptops that cannot be mechanically undocked, since software cannot protect against that. When it is disabled (not the default), a user will actually have to log on to undock the computer and be granted the Remove Computer from Docking Station privilege. This works in conjunction with the “Remove computer from docking station” user right.

Devices: Allowed to format and eject removable media This control is granted to Administrators by default, but you can relax it if necessary by allowing Administrators and Power Users, or Administrators and Interactive Users, the right to format and eject removable media.

Devices: Prevent users from installing printer drivers This setting affects only network printers, not locally connected printers. If you enable this setting (it is disabled by default), then only Administrators and Power Users can install network printer drivers, unless a trusted path has been set up for the network printer drivers.

Devices: Restrict CD-ROM access to locally logged-on user only This setting is disabled by default. If you want to protect against network access to your CD-ROMs, then enable this setting. Keep in mind that it only applies when you are logged in interactively. Once you log out, the CD-ROM is still accessible from the network.

Devices: Restrict floppy access to locally logged-on user only If you want to protect against network access to your floppy drives, then enable this setting; it is disabled by default. Keep in mind that it only applies when you are logged in interactively. Once you log out, the floppy disk is still accessible from the network.

Devices: Unsigned driver installation behavior Installed drivers are loaded into kernel space and therefore have serious security implications. This setting controls how drivers are installed through the Setup API. By default, Windows will “Warn but allow installation.” If you want to be ultrasecure then set this to “Do not allow installation,” which will not stop a driver from being installed by other means, such as manually. Keep in mind that Administrators by default are the only ones with the “Load and unload device drivers” user right.

Domain controller: Allow server operators to schedule tasks Because the Schedule service can be used to launch programs in the context of the all-powerful SYSTEM account, it is recommended that this setting be disabled. Otherwise, a member of the server operators group could abuse this ability to elevate their own privileges.

Domain controller: LDAP server signing requirements Applicable to domain controllers only, this setting determines whether the LDAP server requires the LDAP client to sign the traffic. Setting to “not defined” (the default) or None does not require the LDAP client to sign. Setting this to “Require Signature” will require the LDAP client to negotiate signing methods, unless SSL/TLS is being used.

Domain controller: Refuse machine account password changes This domain controller-specific setting is not typically something you would want to enable. The reason is simply that it prevents a computer password from being changed on a domain controller. Since these passwords are randomly generated and strong, you should not let your users change them unless they know what they are doing.

Domain member: Digitally encrypt or sign secure channel data (always) This setting is disabled by default to allow for interoperability with clients and servers that cannot set up signed or encrypted channels. In large mixed environments, you will probably want to leave this disabled. However, in specific segments, organizational units, or networks with all Windows NT 4.0 SP4 and later OSs, you should consider enabling this.

When it is enabled, the Windows XP client will always attempt to encrypt or sign the secure channel used for communication with a supportive domain controller (Windows NT 4.0 SP4 and later). If an encrypted or signed channel cannot be set up, then communication will fail altogether. When disabled, a secure channel can still be set up, but the signing and encryption parameters are negotiated, and not required. Secure channel communication with a domain controller usually involves NTLM passthrough authentication, SID/Name lookups, and other types of domain authentication traffic.

NOTE Interoperability is a recurring theme throughout this book, and is something you will always want to keep in mind. Microsoft is providing such granular settings as “always” and “when possible” to give you the flexibility to either require security or allow it to be negotiated more transparently.

Domain member: Digitally encrypt secure channel data (when possible) This setting should always be enabled (as it is by default) so that encrypted communication with a domain controller is preferred to unencrypted. An encrypted, secure channel will be negotiated, but is not required, so that interoperability with incapable systems still works.

Domain member: Digitally sign secure channel data (when possible) This setting should always be enabled (as it is by default) so that signed communication with a domain controller is preferred to nonsigned.

Domain member: Disable machine account password changes This setting is disabled by default for good reason. Leave it that way unless you have some pressing need to change machine account passwords. These passwords are managed transparently by Windows and the domain controllers and should not require human intervention. If you need to reset machine account passwords, you can do so through the Active Directory Users and Computers snap-in by right-clicking a machine name and selecting “Reset Account.” By default these passwords are automatically changed every 30 days, as specified in the next setting, “Maximum machine account password age.”

Domain member: Maximum machine account password age Defaulting to 30 days, this setting specifies how often a machine account password is to be automatically reset by Windows.

Domain member: Require strong (Windows 2000 or later) session key If your domain controllers are all Windows 2000 or later, and you require that certain computers communicate with them only over secure channels with strong, 128-bit encryption, then put those computers in an organizational unit and enable this setting. Communication with a DC usually involves NTLM passthrough authentication, SID/Name lookups, and other types of domain authentication traffic. This setting is not related to SMB traffic (Windows networking with Server Message Block), which is covered in upcoming text. It is disabled by default, which means Windows XP hosts will tolerate weaker encryption keys to set up secure channels.

Interactive logon: Do not display last user name While this setting is still disabled by default, you should enable it to prevent the last logged-on user’s name from appearing at the login screen. This typically enforced setting usually gets enabled through a GPO for all computers in the domain.

Interactive logon: Do not require CTRL+ALT+DEL The familiar CTRL-ALT-DEL sequence that precedes a Windows 2000 login is not the default in stand-alone Windows XP Professional computers. Instead, Fast User Switching is enabled, which doesn’t invoke the secure logon channel that pressing CTRL-ALT-DEL does. Once a Windows XP Professional machine joins a domain, however, this setting is automatically disabled, and the secure CTRL-ALT-DEL sequence is required.

Interactive logon: Message text for users attempting to log on This is another setting typically enforced through a GPO at the domain level, because an organizational login message is normally required by policy. If you are not already using this feature, you should enable it for all your network computers. In some legal cases, message text at the logon prompt can be required to show that users were warned not to attempt unauthorized access to a system.

Interactive logon: Message title for users attempting to log on This is the text for the title bar that accompanies the message text at logon.

Interactive logon: Number of previous logons to cache (in case domain controller is not available)

The past ten unique logon credentials will be cached by default. This is considered insecure, because these credentials are stored in a protected part of the system registry where they could possibly be retrieved by someone with Administrator privileges. The problem may not seem obvious at first, but suppose that some of your users have Administrator rights on their personal machines, and one day your domain administrator has to log into one of those machines to do some work. Are you comfortable knowing that the domain administrator's credentials have just been cached in the registry of that computer?

The most paranoid mind will set this to zero, so that no domain logon credentials are cached. However, this would mean that domain accounts will not be able to log into the machine if a domain controller is not available. For laptops and traveling employees, you should probably set this to at least 1, so that the last logon is cached. This will ensure that they can still log on to their computer when disconnected from the network (provided they were the last ones to log on). Configure this setting wisely.

Interactive logon: Prompt user to change password before expiration By default, users are prompted 14 days prior to a required password change. Set this to the number of days' advance notice you want users to have that a password change will be required.

Interactive logon: Require Domain Controller authentication to unlock workstation Once enabled, a workstation cannot be unlocked with a domain account unless a domain controller is present. If you have a group of traveling employees with laptops, you do not want to enable this for them, and you might consider placing them in a separate organizational unit with their own GPO.

Interactive logon: Require smart card (Windows.NET AD only) In Windows .NET Active Directory, smart card authentication can be required for user interactive logons. This setting is disabled by default, but it can be enabled and applied to a GPO if desired to require an organizational unit or domain to use smart cards for logon. Smart cards provide some of the strongest authentication methods possible today.

Interactive logon: Smart card removal behavior When a smart card is removed for a logged-on user, one of three things can be configured to happen. Either no action is taken, the workstation is locked, or the user is forcefully logged off. In most cases, setting this to "Lock Workstation" will provide a good security measure.

Microsoft network client: Digitally sign communications (always) When both a client and a server sign their packets, message integrity can be achieved and man-in-the-middle attacks can be prevented. This is called *mutual authentication* and provides the most security. The type of communications that this setting is concerned with is a client using Server Message Block (SMB) to connect to a server. This setting is disabled by default, because in most mixed OS environments, some legacy systems will not be capable of signing SMB communications. If your network is purely Windows 2000 and higher, then you should definitely enable this setting for maximum security. The server

requirements defined in “Microsoft network server: Digitally sign communications (always)” must be enabled to provide the “mutual authentication” mentioned.

NOTE Digitally signing packets does incur a CPU performance penalty on the client and server.

Microsoft network client: Digitally sign communications (if server agrees) This setting is enabled by default and should remain so. It attempts to negotiate secure, signed communications, but will not fail the communication if the other computer is not capable of signing.

Microsoft network client: Send unencrypted password to third-party SMB servers This setting is disabled by default and should stay that way. It prevents the client from sending plaintext passwords to SMB servers that do not support encryption during the authentication process. If you have some specific need to enable this setting, you should create an organizational unit for only the computers that need it, and warn everyone of the consequences. If someone can put a sniffer on the network or trick a machine into sending its plaintext credentials, then those credentials can be reused on the network.

Microsoft network server: Amount of idle time required before suspending session You can set a timeout for SMB communications. This setting is undefined for Windows XP workstations by default, and set to 15 minutes for servers. For example, if an established SMB connection with a server has been idle for 15 minutes, the server will disconnect it.

Microsoft network server: Digitally sign communications (always) In a perfectly networked world, digital communications would always be signed, so that the identity of each party could be validated. However, in the real world of backward compatibility, hybrid environments, and incapable clients (and even servers), you can only enable this setting when you have absolute certainty that every computer is capable of signing. This is not to say that once this setting is enabled (it’s disabled by default) you will be more secure, but your Windows 2000/XP/.NET computers will at least require that all SMB communications with their Server service be signed by the remote client computer.

Microsoft network server: Digitally sign communications (if client agrees) This setting is disabled on workstations and enabled on servers, but it won’t hurt to enable this on your Windows 2000 and XP workstations. In fact, it’s a good idea. Once you do, the SMB server of the workstation will attempt to set up signed communications with whatever remote computer is attempting connection. If the remote computer does not agree to sign its communications, they will still be allowed.

Microsoft network server: Disconnect clients when logon hours expire When this setting is enabled, remote sessions with the workstation’s local SMB server will be forcibly disconnected once the logon hours of the account have expired. While it is undefined by default, you should enable it if you want logon hours to be enforced on your network.

Network access: Allow anonymous SID/Name translation New to Windows XP/.NET, this setting is disabled on workstations and enabled on servers. Certain aspects of authentication with a domain controller require that anonymous SID-to-name translation be allowed, but it is certainly not necessary for all servers. You should leave this setting disabled on workstations to prevent remote, anonymous users from being able to illicit SIDs by username, or usernames by SID. SID/Name translation is a common technique used by hackers to enumerate administrator (and other users) account names and SIDs on remote machines.

Network access: Do not allow anonymous enumeration of SAM accounts RestrictAnonymous has been much improved in Windows XP to include more granular control of anonymous access without the past Windows 2000 problems of breaking domain functionality. The old RestrictAnonymous setting of Windows 2000 is now split into several separate settings. This option corresponds to a new registry value named RestrictAnonymousSam and, as it is enabled by default, will prevent user account information from being enumerated by an anonymous user.

NOTE RestrictAnonymous has been newly designed for Windows XP. In Windows 2000, setting RestrictAnonymous=2 prevents null users from even connecting to the IPC\$ share, which ends up killing down-level client access and trusted domain enumeration. This isn't the case in Windows XP/.NET, however, which gives you more control over restricting anonymous access by providing the following finely tuned options:

- Network access: Do not allow anonymous enumeration of SAM accounts
- Network access: Do not allow anonymous enumeration of SAM accounts and shares
- Network access: Allow anonymous SID/Name translation
- Network access: Let Everyone permissions apply to anonymous users

In its own way, each setting contributes to the permissions that an anonymous user has on a system. Remember, too, that these are in addition to other settings that control anonymous access to named pipes and registry keys.

Network access: Do not allow anonymous enumeration of SAM accounts and shares Enabling this setting is somewhat similar to setting RestrictAnonymous to 1 in Windows 2000, although you should remember that the functionality has changed in Windows XP/.NET. In Windows XP/.NET, this setting still allows an anonymous connection to IPC\$ as null, but it won't allow user account and share enumeration.

It is disabled by default, which means that while user account information can be enumerated, share information cannot. You should enable this setting on any Windows XP/.NET computer that does not need to allow anonymous users to remotely enumerate both user accounts and shares. Note that enabling this may break some applications, so test first.

CHALLENGE

You have a development department that needs a higher level of network security that can only be configured in a purely Windows 2000 SP2 and later network. Luckily, you just finished a complete desktop upgrade to Windows XP for this department, and you know that all their domain controllers are at least Windows 2000 SP2. The file and print servers in this department have also recently been upgraded from Windows NT 4.0 to Windows 2000 SP2.

So what are the strongest security settings you can enable that will be supported by both Windows 2000 and Windows XP?

You can actually enable a lot of security in a network where only Windows XP and Windows 2000 computers exist. By configuring the following settings for Windows XP and the corresponding settings for Windows 2000, you will increase security for network operations such as domain authentication, SMB file shares, and anonymous user access:

- **Domain member: Digitally encrypt or sign secure channel data (always)** Enabled
- **Domain member: Require strong (Windows 2000 or later) session key** Enabled
- **Microsoft network client: Digitally sign communications (always)** Enabled
- **Microsoft network server: Digitally sign communications (always)** Enabled
- **Network access: Allow anonymous SID/Name translation** Disabled
- **Network access: Do not allow anonymous enumeration of SAM accounts and shares** Enabled
- **Network security: Do not store LAN Manager hash value on next password change** Enabled
- **Network security: LAN Manager authentication level** Send NTLMv2 response only\refuse LM. (Ideally, if you can set this value to "Send NTLMv2 response only\refuse LM & NTLM" then you should do so. Be sure to test this, however, as we have seen this break functionality in different environments.)
- **Network security: Minimum session security for NTLM SSP based (including secure RPC) clients** Require each message integrity, message confidentiality, NTLMv2 session security, and 128-bit encryption

CHALLENGE (continued)

- **Network security: Minimum session security for NTLM SSP based (including secure RPC) servers** Require each message integrity, message confidentiality, NTLMv2 session security, and 128-bit encryption
- **System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing** Enabled
- **System objects: Default owner for objects created by members of the Administrators group** Administrators Group

Note that a few questions arise here. First, how do we intend to apply Windows XP–specific security settings through a Windows 2000 domain controller, when these settings don’t even show up in the Group Policy MMC snap-in on the DC?

Chapter 11 will answer this question, when it describes how you can use your Windows XP Professional computer (or Windows .NET member server) to manage the Active Directory and GPOs.

Another question is, how in the world do these Windows XP–specific settings map over to the Windows 2000 computers, and vice versa? Or more importantly, how can a single GPO be created that applies these settings to both Windows 2000 and Windows XP?

The answers are that they do map over (but not in the way that you would expect), and that you cannot create a single GPO for both operating systems (unless you want to wreak havoc on your network). The section on managing Windows XP GPOs in Chapter 11 will answer these questions as well.

Network access: Do not allow storage of credentials or .NET Passports for network authentication

This setting is disabled by default. Credentials and .NET passports used for network connections will be stored in the new Credential Manager. They will be used transparently when an integrated authentication package (Kerberos, NTLM, etc.) requires them. The Credential Manager acts like a key ring. It can be managed through the User Accounts control panel applet by clicking the “Manage my network passwords” link.

Network access: Let Everyone permissions apply to anonymous users A brand new and powerful setting in Windows XP/.NET lets you control whether or not Everyone permissions apply to the Anonymous user token. When this setting is disabled, the Anonymous account (security principal) will not be considered a member of the Everyone group, and its token therefore will not carry the SID for the Everyone group.

NOTE A security token is a collection of SIDs that define the access permissions for an account, as discussed further in Chapter 6.

Without being a member of the Everyone group, the Anonymous user is even more limited than it was Windows 2000 and Windows NT. Because this may break some applications, you can easily enable this setting for troubleshooting.

Network access: Named Pipes that can be accessed anonymously Named pipes are communications channels between computers on a network. Services and applications will set up and use these channels. The list here can also be found in the registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters` under the `NullSessionPipes` value, where the configuration ultimately resides.

This list represents the anonymously accessible named pipes, but there can exist many other named pipes that are not accessible with null credentials, and hence do not appear on this list. You probably do not need all the default pipes here, but deleting them is something you should only do in a test lab, one by one, to figure out which ones are needed for the applications you run. For example, a print server will require that the `SPOOLSS` exist. Most other systems will require the `EPMAPPER`, `LLSRPC`, and `LOCATOR` for most networking functions.

Network access: Remotely accessible registry paths As described further in Chapter 3, on registry security, the remotely accessible registry paths correspond to the `HKLM\SYSTEM\CCS\Control\SecurePipeServers\winreg\AllowedPaths` key values. The paths specified here can, more precisely, be accessed remotely by Anonymous users. The best method for removing them is one by one in a test lab, figuring out which ones do and don't break your applications and network management needs. For instance, if the host is not serving as a print server or a terminal server, then there is most likely no need for the `System\CurrentControlSet\Control\Print\Printers` and the `System\CurrentControlSet\Control\Terminal Server` paths.

Network access: Restrict anonymous access to Named Pipes and Shares (.NET AD only) Enabled by default on a Windows .NET domain controller, this setting effectively overrides both the "Network access: Named Pipes that can be accessed anonymously" and the "Network access: Shares that can be accessed anonymously" settings. Enabling this setting restricts anonymous access for named pipes and shares specified in the two mentioned settings.

Network access: Shares that can be accessed anonymously Most shares in Windows require some form of authentication, unless you specifically add the Anonymous SID. In this case the shares listed here have that SID, and while they are not created by default on a Windows XP host, they will allow anonymous access once a share with that name is

created. However, the Anonymous user will not have any permissions unless specifically defined. For maximum security, you should remove the shares listed here (COMCFG, DFS\$) unless your applications (typically COM+ and DFS) require them.

Network access: Sharing and security model for local accounts Touted as one of Windows XP's new features, this setting controls whether or not remote network connections to the XP computer will be forced to authenticate as a guest account or another account. This is irrelevant in domain environments, where this setting is forced to "Classic—local users authenticate as themselves," which essentially means any local account can be used for authentication from a remote machine. This setting was mainly added to help protect the networked user, by allowing (the default) remote connections to authenticate with only minimal Guest privileges.

Network security: Do not store LAN Manager hash value on next password change As described further in Chapter 3, this setting was first introduced in Windows 2000 SP2 as a registry hack. Now in Windows XP it is a Security Policy setting. You should definitely enable it (it is disabled by default), as it will prevent older LAN Manager-style password hashes, which are more easily crackable, from being stored on the computer. Be careful if you are using applications that require the LAN Manager (LM) hash, and test this setting in a lab. Also remember that once you flip it to enabled, you must make a password change before the old LM hash disappears.

NOTE You can achieve this same setting in both Windows 2000 and Windows XP by creating a password 15 characters or longer, for which a LM hash will not be generated.

Network security: Force logoff when logon hours expire Logon hours are defined by user either locally or in Active Directory, by domain or organizational unit. This setting is disabled by default, but if you are serious about your network's logon hours, then you should enable it. It applies only to client SMB connections, meaning it will only affect users logged in across the network. To see current SMB connections type **net session** from the command line.

Network security: LAN Manager authentication level This is another setting that allows you to balance between backward compatibility and network security. Figure 1-2 illustrates the options you can choose from. Although Kerberos authentication is at the center of a Windows 2000/XP/.NET domain, LAN Manager and NTLM will still be used with down-level clients (or servers) and when Kerberos authentication fails.

Windows 2000/XP/.NET all default to "Send LM & NTLM responses," which happens to be the weakest but most compatible stance. If someone can capture, or *sniff*, traffic on your network and get LM hashes, the hashes can be easily cracked with tools such as L0phtCrack to gain the username and password. NTLM is stronger but has its own set of problems, although it is not as readily criticizable. At a *minimum*, you should override the default setting in your default domain policy and pick the second one: "Send LM & NTLM—use NTLMv2 session security if negotiated." This will ensure backward compatibility while providing an option for stronger NTLMv2. If your

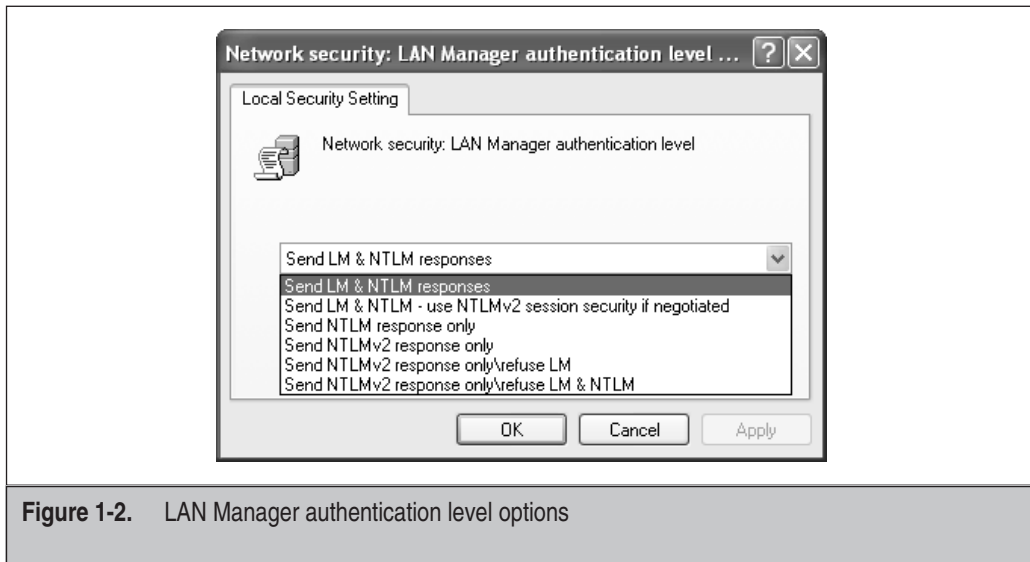


Figure 1-2. LAN Manager authentication level options

network is all Windows NT 4.0 and later, then you could choose “Send NTLM response only.” If it is all Windows 2000 and later, then you could choose “Send NTLMv2 response only\refuse LM & NTLM,” as we did in this chapter’s Challenge.

NOTE Windows NT 4.0 SP4 brought support for NTLMv2, although it needs to be turned on through the registry. Also, Windows 95/98 clients can support NTLMv2 if you install the Directory Services Client from the Windows 2000 installation CD and enable the LMCompatibility registry key setting. See Microsoft Knowledge Base article Q239869 for more details.

Network security: LDAP client signing requirements “Negotiate signing” is the middle of the road and the default choice. In this way, network security will be achieved by the LDAP client requesting signed communications with the server. If the server agrees, then signed communications will commence; otherwise, communications will go by unsigned. The exception to this is if TLS/SSL is being used, in which case signing will not even be requested.

Network security: Minimum session security for NTLM SSP based (including secure RPC) clients This setting affects the behavior of the computer when it acts as a network client, and was also available in Windows 2000 and Windows 9x clients using the dsclient software. By default, there are no minimum security requirements set on application-to-application SSP (Security Service Provider)–based communications. As administrator, you can require none or any combination of the following: message integrity, message confidentiality, NTLMv2 session security, 128-bit encryption.

Network security: Minimum session security for NTLM SSP based (including secure RPC) servers

This setting affects the behavior of the computer when it acts as a network server. As administrator, you can require none or any combination of the following: message integrity, message confidentiality, NTLMv2 session security, 128-bit encryption.

Recovery console: Allow automatic administrative logon This setting should always be disabled unless you have some aggravating needs that require you to completely jeopardize security. By allowing automatic administrative logon to the recovery console, no logon information (such as username and password) is required for administrative access to the system. Anyone with physical access to the server can shut it down and boot straight to the recovery console with full admin rights.

Recovery console: Allow floppy copy and access to all drives and all folders By default, the recovery console allows access only to the system partition and a limited set of commands. Enabling this setting will provide access to all local drives, including the floppy, as well as open up some additional commands. Typically, default access to the system partition is enough to perform most troubleshooting or maintenance tasks that the recovery console is designed to do.

NOTE The recovery console is not installed by default, even though these settings exist. To install it, insert the setup CD and run `\i386\winnt32.exe /cmdcons`.

Shutdown: Allow system to be shut down without having to log on By default, Windows workstations can be shut down from the initial logon screen, while servers cannot. By disabling this setting, the logon screen option to shut down will be grayed out and a user will have to log on before it is enabled.

Shutdown: Clear virtual memory pagefile Since the paging file contains data swapped from physical memory, it can contain sensitive information such as passwords and encryption keys. While the operating system does a good job of protecting the pagefile while the system is running, there is nothing to stop someone with physical access from powering off a machine and rebooting to an alternate OS to access the data stored in the pagefile. After all, it is just a file on disk that simulates RAM. If this setting is enabled, be prepared for a slight performance hit; it will take longer to shut down and reboot.

System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing

This is a new crypto setting for Windows XP/.NET that allows you to force stronger encryption algorithms when it is enabled. For TLS/SSL, encryption algorithms are forced to use Triple DES, rather than their default DESX. Also, the RSA public key algorithm is forced for the key exchange and authentication, and only the SHA-1 hashing algorithm will be used for hashing negotiations. This setting also applies to EFS encryption algorithms and will force Triple DES rather than DESX.

System objects: Default owner for objects created by members of the Administrators group On Windows XP Professional, this setting defaults to “Object Creator,” meaning that if a member of the Administrators group creates an object (such as a new file or folder), their user SID will be designated the owner of that object. This is as opposed to the SID for the Administrators group being designated the owner of the object. On Windows .NET servers this setting defaults to “Administrators Group,” which is the more secure choice. This setting requires a reboot in order to take effect.



SECURITY ALERT A serious threat to system integrity can be caused by setting this to “Object Creator.” Imagine a scenario where Lisa is a member of the Administrators group, and she creates some new top secret folders and files for use by Administrators on one of the administrative workstations. If a few months later she takes a new position and is removed from the Administrators group, she will still have ownership rights to those files and folders, which means she can give herself full control and remove other user’s permissions.

System objects: Require case insensitivity for non-Windows subsystems (Windows .NET only)

The Win32 subsystem is case-insensitive by default, a feature that cannot be changed. Win32 will however support case sensitivity in subsystems such as POSIX. By default this setting is enabled, and case insensitivity is enforced for all subsystems except Win32.

System objects: Strengthen default permissions of internal system objects (e.g. Symbolic Links)

Enabled by default, this setting tightens security by setting stronger DACLs on shared system objects such as mutexes, semaphores, and DOS device names. Instead of carelessly giving out full control to everyone, these objects will be created so that nonadministrative users can read but cannot modify them. There is no good reason to disable this setting.

TOOLS OF THE TRADE

Several tools are used to implement security policy. Security settings can be managed locally by system or domainwide through Group Policy. In addition, the security settings can be exported to an .inf template file with the Security Configuration and Analysis tool, and imported either locally or as a part of a group policy object.

Local Security Policy (Secpol.msc) Renamed Local Security Settings in Windows XP/.NET, this is the interface for configuring some of the most important “security settings” related to the operating system. These settings are included as a subset of Group Policy, and contain the relevant configurations for Password Policy, Audit Policy, Security Options, Software Restriction Policies, and IP Security Policies. Configurations set through Local Security Policy are only applied to the local machine, regardless of the logged-on user.

Group Policy (Gpedit.msc) This is the MMC snap-in that is used to edit and apply Group Policy Objects. It is accessible either by entering `gpedit.msc` from the command line or by right-clicking a Domain or Organizational Unit in Active Directory and selecting Properties → Group Policy. You use it to apply the same computer and user configurations to a group of computers or users in Active Directory. On a local machine, Group Policy can also be used to apply local configurations.

Security Configuration and Analysis This powerful toolbox provides a means to configure all the security settings related to Local Security Policy, as well as many more related to the file system, registry, restricted groups, and services. Access the Security Configuration and Analysis snap-in by opening the MMC and selecting File → Add/Remove Snap-in.

CHECKLIST: SECURITY SETTINGS

The security settings, which provide some of the most important security configurations, have changed significantly from Windows 2000 to Windows XP. Configuring these wisely can drastically improve OS security on your network. To successfully implement them, you need to understand the effects of each setting and the effects of combining multiple settings. Do not plan your Windows XP rollouts without designing a solid baseline of security settings.

The following checklist highlights some of the key points made throughout this chapter:

- Local Policy is applied for a single machine, to all users on the machine.
- Group Policy is more powerful, allowing you to apply policy to sites, domains, and organizational units in Active Directory on a per user/computer or per group basis.
- Several interfaces exist to accomplish similar goals. Get familiar with the MMC, Local Security Policy (Secpol.msc), Group Policy (Gpedit.msc), and the Security Configuration and Analysis MMC snap-in.
- Plan a security policy baseline that can be applied across your organization.
- When planning your baseline, consider the need for interoperability with older Windows clients.
- Configure your account policies so they can be applied at the domain or organizational unit level, through a group policy object.
- Enable Auditing, which is disabled by default.
- Tighten up user rights assignments to match your own needs; do not give everyone rights that they do not need.
- Understand each of the security options available. Remember that some options that talk of signing or encrypting traffic are repeated for different types of traffic, such as SMB, LDAP, and session security.