

16

Managing System Users

IN THIS CHAPTER, YOU WILL LEARN TO:

▶ **AUDIT USER ACCESS (Pages 320–328)**

- List the Policies (Page 321)
- Get a Policy (Page 322)
- Set a Policy (Page 326)
- Perform a Backup (Page 327)
- Perform a Restore (Page 327)
- Clear an Audit Policy (Page 328)
- Remove an Audit Policy (Page 328)

▶ **WORK WITH GROUP POLICIES (Pages 328–330)**

- Obtain Group Policy Results (Page 328)
- Manage Group Policies (Page 330)

▶ **OBTAIN SESSION STATUS INFORMATION (Pages 331–332)**

- Get Process Information (Page 331)
- Get Session Information (Page 331)
- Get User Information (Page 332)
- Get Terminal Server Information (Page 332)

▶ **GET THE USER'S IDENTITY (Pages 332–333)**

- Obtain User Logon Information (Page 333)
- Discover User Identity (Page 333)

The command line is helpful for automating common user management tasks. For example, configuring the auditing policies for a group of users is extremely easy at the command line. On the other hand, the command line can't easily perform some user management tasks. If you want to see the overall statistics of user productivity on a computer, that's best left to the GUI because seeing that kind of data is easier using charts and graphs. In short, the command line and GUI environments each have their particular uses. This chapter focuses on common tasks that you could probably automate to some extent.

NOTE As with many command line tasks, the tasks in this chapter usually require administrator privileges. However, because of the nature of the tasks performed in this chapter, the requests for administrator-level elevation become quite annoying after a while when working on Vista or Windows 7 systems. To avoid this problem, right-click the Command Prompt shortcut and choose Run as Administrator from the context menu. User Account Control will ask you whether you want to run the command processor as an administrator. Click Yes. You can now accomplish all of the required tasks without continuous interference from Windows.

Audit User Access

Auditing system activity is a necessary process in many situations. Of course, there's the obvious use of ensuring the system remains secure by thwarting any misguided user activity. However, auditing can help you do more than just check security. For example, careful auditing can often alert you to potentially damaging system activities or help you better understand why a system doesn't perform as well as it could. Checking object access can help you better define how a user interacts with a system so that you can make the system more efficient. A user's privilege use can help you locate security holes that occur when a user has too many rights, some of which aren't even used. The following sections discuss the `AuditPol` utility, the command line interface for auditing needs.

List the Policies

Before you can use audit policies, you need to know which policies are available and whom they affect. Windows applies categories of auditing policies to specific users, so you actually have two concerns when discovering the current auditing configuration. The `AuditPol /List` command makes it possible to check users, auditing categories, and auditing subcategories as described in the following sections.

List Audit Users

To discover which users are audited, type `AuditPol /List /User` and press Enter. The output of this command provides a list of which users are audited, but not how they're being audited. To discover how the user is being audited, you type `AuditPol /Get /User:UserName /Category:*` and press Enter, where *UserName* is the user's name (see the "Get a Policy" section of the chapter for additional information). If you also want to know the user's Security Identifier (SID), type `AuditPol /List /User /V` and press Enter. The SID comes in useful for a number of purposes and ensures that you can uniquely identify the user to the system.

List Audit Categories

Many of the `AuditPol` commands require that you know a category. If you want information for all categories, you simply use the asterisk (*), but often the wildcard search returns far too much information to be useful unless you limit the output in some other way. Consequently, knowing the precise category you want is important in many situations. To obtain a basic category listing, type `AuditPol /List /Category` and press Enter. In most cases, the basic listing is all you need. However, if you plan to work with the category at a detailed level or want to search for its entry in the registry, you need a Globally Unique Identifier (GUID) that precisely identifies the category to the system. To obtain this information, type `AuditPol /List /Category /V` and press Enter.

List Audit Subcategories

Categories are divided into subcategories. For example, the `Object Access` category contains a subcategory of `File System` (among other subcategories). You can choose to audit a user's access to the file system, without monitoring other kinds of `Object Access`, by specifying a subcategory. To obtain the subcategories of a specific category, type **`AuditPol /List /Subcategory:"CategoryName"`** and press Enter, where *CategoryName* is the name of any category you want to see.

If you want to see multiple categories, simply create a list separated by commas of category names. For example, to see the subcategories of the `Account Logon` and `Account Management` categories, you'd type **`AuditPol /List /Subcategory:"Account Logon","Account Management"`** and press Enter. To see all of the subcategories for every category, type **`AuditPol /List /Subcategory:*`** and press Enter. As with categories, subcategories have GUIDs. To see the GUIDs for the subcategories, add the `/V` command line switch.

Get a Policy

Listing a policy simply tells you that the policy exists but doesn't tell you the policy setting. Getting a policy won't tell you that the policy exists—you must already know that the policy exists. However, it does tell you how the policy is configured. Even though listing and getting may sound a lot alike, the two are completely different. The `AuditPol /Get` command is all about discovering the system settings.

It's also important to understand that audit policies are configured at two levels. First, you can configure an audit policy at the system level, which means that the policy affects everyone. Second, you can configure an audit policy for a specific user, which means that the policy affects only that user. The `AuditPol /Get /User` command tells you about specific user settings, while the `AuditPol /Get /Category` and `AuditPol /Get /Subcategory` commands tell you about system-level settings.

A special setting level affects the system directly when an audit event occurs. For example, the `CrashOnAuditFail` option causes the system to crash when the auditing system fails for some reason. This is a safety feature because it ensures that no one can turn off auditing and then continue to use the system unless they use the standard methods to do so and have the proper rights. The following sections describe all of these `AuditPol /Get` command scenarios.

Get Audit Users

The `AuditPol /Get /User` command obtains information about a specific user. In most cases, you want to know a user's full rights, so you'll type `AuditPol /Get /User:UserName /Category:*`, where *UserName* is the name of the user, and press Enter. However, you can specify a particular category to discover information about just that category or you can use the `/Subcategory` command line switch to be even more selective and discover information about just one setting. The output you see contains three columns: the name of the category or subcategory, the inclusive setting, and the exclusive setting.

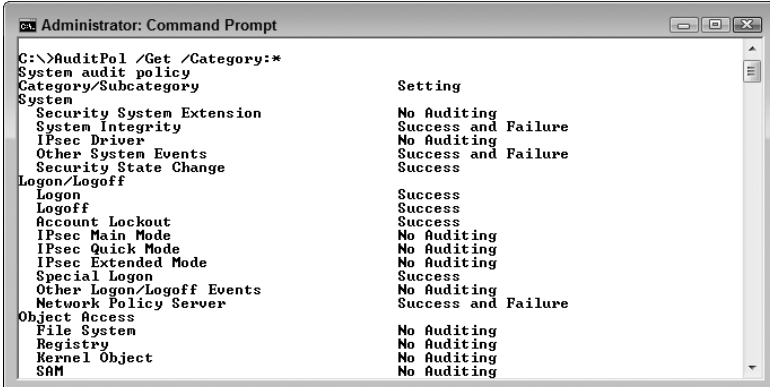
NOTE When you set a user audit policy, it's either inclusive or exclusive. An inclusive policy is one that adds to the system-level settings. For example, if you audit the user's failure to log on to the system, it's an inclusive policy because it's in addition to any system-level settings. However, if the system normally monitors logon failures, but you don't want to check a particular user, then you'd create an exclusive policy. Even though everyone else is monitored, this particular user is excluded from the policy. It's unusual to create exclusive policies—inclusive policies are far more common.

You may need to output the user settings in a form that you can import into a database. In this case, you'd add the `/R` command line switch to create Comma Separated Value (CSV) output. For example, if you need to retrieve the settings for user Jamal and put them in a CSV file, you'd type `AuditPol /Get /User:Jamal /Category:* /R > AuditPol.CSV` and press Enter.

Get Audit Categories

The `AuditPol /Get /Category` command obtains the system-wide settings for both categories and subcategories. For example, if you type `AuditPol /Get /Category:*` and press Enter, you see output similar to that shown in Figure 16.1 (which shows only a partial listing of the categories and subcategories). Of course, you can choose to obtain a specific category by using the category name in place of `*`. For example, to obtain the Logon/Logoff category, you type `AuditPol /Get /Category:"Logon/Logoff"` and press Enter. As with the user information, you can output the categories to CSV format using the `/R` command line switch.

Figure 16.1: Getting the categories also obtains the subcategory information.



```

Administrator: Command Prompt
C:\>AuditPol /Get /Category:*
System audit policy
Category/Subcategory      Setting
System
  Security System Extension  No Auditing
  System Integrity           Success and Failure
  IPsec Driver                No Auditing
  Other System Events        Success and Failure
  Security State Change      Success
Logon/Logoff
  Logon                       Success
  Logoff                       Success
  Account Lockout             Success
  IPsec Main Mode             No Auditing
  IPsec Quick Mode            No Auditing
  IPsec Extended Mode         No Auditing
  Special Logon               Success
  Other Logon/Logoff Events   No Auditing
  Network Policy Server       Success and Failure
Object Access
  File System                 No Auditing
  Registry                    No Auditing
  Kernel Object               No Auditing
  SAM                         No Auditing

```

Get Audit Subcategories

Use the `AuditPol /Get /Subcategory` command when you need to obtain the system-wide setting for a single subcategory. For example, to retrieve the status of the Logon subcategory, you'd type `AuditPol /Get /Subcategory:"Logon"` and press Enter. Unlike the `/Category` command line switch, you can't use `*` with the `/Subcategory` command line switch.

Get Audit Options

The `AuditPol /Get /Option` command retrieves audit policy settings that affect the system as a whole when certain audit policy events occur. The following list describes each of these options:

- **CrashOnAuditFail:** When you enable this setting, it forces the system to crash should the auditing system become unable to log events. The advantage to this setting is that it forces everyone to use the auditing policies you set. However, the disadvantage is that an outsider could use this option to force the server to crash and cause an apparent Distributed Denial of Service (DDoS) attack. You need to use this setting with care. After this event occurs, only administrators can log on to the system. The administrator must fix whatever caused the crash before the system will allow anyone to log back on. This setting is generally useful on client systems, but not recommended for servers.
- **FullPrivilegeAuditing:** When you tell the system to audit privileges, it normally does so for most privileges, but it leaves out a few

commonly used privileges to keep the event log from quickly overflowing, such as the following privileges:

- Generate security audit (SeAuditPrivilege)
- Bypass traverse checking (SeChangeNotifyPrivilege) debug programs (SeDebugPrivilege)
- Create a token object (SeCreateTokenPrivilege)
- Replace process-level token (SeAssignPrimaryTokenPrivilege)
- Generate security audits (SeAuditPrivilege)
- Back up files and directories (SeBackupPrivilege)
- Restore files and directories (SeRestorePrivilege)

Enabling this setting forces the system to audit all privilege changes except SeAuditPrivilege. You can't audit the SeAuditPrivilege because it would cause an endless loop—every access to the audit system generates this privilege and therefore every entry to the log would generate another SeAuditPrivilege event.

- **AuditBaseObjects and AuditBaseDirectories:** Kernel objects come in two forms: container objects and base objects. The AuditBaseObjects policy affects base objects, those that can't contain object objects such as semaphores and mutexes. The AuditBaseDirectories policy affects container objects, those that can contain other objects, such as directories. Many kernel objects are unnamed and rely only on a handle that's accessible to just the process that created the object for access. Unnamed kernel objects are secure, but they don't allow interprocess communication, which is often necessary in applications. Named kernel objects do allow interprocess communication, but they present security risks because another process (other than those that should use the named process) can interact with the kernel object should it discover the object's name. Setting either of these options forces the operating system to assign a System Access Control List (SACL) to the named objects so that the auditing system can monitor them. The normal use for these settings is to detect and thwart squatting attacks (see the article at http://en.wikipedia.org/wiki/Squatting_attack for details). A problem with these settings is that you normally must reboot the system before the changes you make take effect.

You use these options individually. For example, to obtain the status of the `CrashOnAuditFail`, you type `AuditPol /Get /Option:CrashOnAuditFail` and press Enter. Unlike other audit policy settings, options are either enabled or disabled.

Set a Policy

Setting a policy is the act of creating a new entry for the system or a particular user. These settings work as stated in the “Get a Policy” section of the chapter. When you create a new policy, the user or the system as a whole is monitored for the success or failure of certain actions. You can also enable or disable audit options that perform a task based on an audit event (such as crashing the system when someone tries to override the audit system). The following sections describe how to set an audit policy.

Set Audit Users

The `AuditPol /Set /User` command controls settings made to a specific user. When working with users, you must remember that you can create inclusive settings that add to the system-level settings or exclusive settings that remove auditing from the system-level settings. Audits can affect failures and successes. You can also enable or disable a setting. For example, to set a user account to add (inclusive) failure auditing to the `Object Access` category, you’d type `AuditPol /Set /User:Username /Category:"Object Access" /Include /Failure:Enable`, where *Username* is the name of the user, and press Enter.

All user-level settings follow this same pattern. You provide the username, a category or subcategory, whether the setting is inclusive or exclusive, whether the auditing is for a success or failure, and whether the setting is enabled or disabled. As another example, let’s say you want to create an exclusion for a user for `Logon` subcategory auditing for both success and failure. In this case, you’d type `AuditPol /Set /User:Username /Subcategory:"Logon" /Exclude /Failure:Enable /Success:Enable` and press Enter.

Set Audit Categories

The `AuditPol /Set /Category` command controls settings made to the system as a whole. Unlike user-level settings, you simply set the policy to monitor success or failure. There isn’t any concept of inclusion or

exclusion. For example, to audit Account Logon failures, you'd type **AuditPol /Set /Category:"Account Logon" /Failure:Enable** and press Enter. **AuditPol** sets all of the subcategories for the entire Account Logon category to audit failures.

Set Audit Subcategories

The **AuditPol /Set /Subcategory** command controls settings made to the system as a whole, just like the category-level command. However, this command lets you set the individual subcategory entries, rather than an entire category. For example, you might want to failure audit the **Credential Validation** subcategory of the Account Logon category. To perform this task, you type **AuditPol /Set /Subcategory:"Credential Validation" /Failure:Enable** and press Enter.

Set Audit Options

The **AuditPol /Set /Option** command controls the audit policy options described in the “Get Audit Options” section of the chapter. You either enable or disable these options. For example, to enable the **CrashOnAuditFail** option, you type **AuditPol /Set /Option:CrashOnAuditFail /Value:Enable** and press Enter.

Perform a Backup

If you have a complex audit policy setup, you'll want to create a backup of it occasionally to ensure you don't lose the settings and have to make them all over again. You can also create a backup so that you can move the settings to another machine. No matter what reason you have for making the backup, type **AuditPol /Backup /File:Filename**, where *Filename* is the name of the backup file you want to use, and press Enter to create the backup.

Perform a Restore

Restoring a backup will overwrite all of the current settings for the target system. It's important to understand that restoring a backup is the same as making all of the settings changes by hand. To perform this task, type **AuditPol /Restore /File:Filename**, where *Filename* is the name of the backup file you want to use, and press Enter.

Clear an Audit Policy

You made a mistake. In fact, you made a really big mistake and the audit logs are filling up faster than you can clear them. The system is completely messed up and you don't know what to do about it. To correct this problem, type **AuditPol /Clear** and press Enter. This command essentially resets all of the audit policies to the state they were in when you installed Windows. Of course, you'll lose any good changes you made, but you'll also get rid of any incorrect settings you made as well.

Remove an Audit Policy

There are times when you simply want to remove the existing audit policies for a user or for all users on a system. A new company policy may define system-level settings that everyone should use, rather than rely on special settings for individual users. Alternatively, you might have monitored a particular user for a while, but decided the monitoring is no longer necessary and want to remove all of the auditing with one command. No matter what reason you have to make the change, you can type **AuditPol /Remove /User:Username**, where *Username* is a particular user's name, and press Enter to remove the audit policies for a specific user. To remove the audit policies for all users, type **AuditPol /Remove /AllUsers** and press Enter.

Work with Group Policies

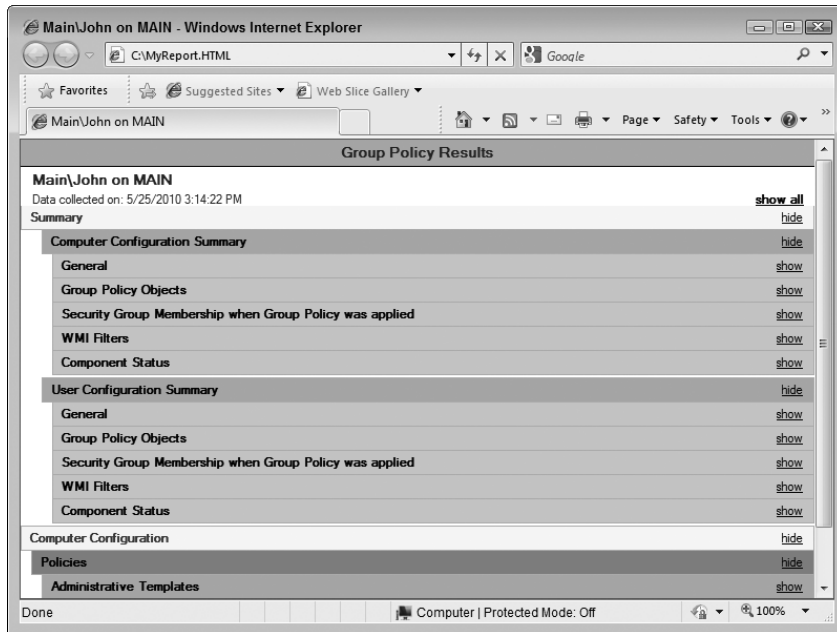
Group policies make it possible to create security and usage configurations for users without having to set every user's configuration individually. Setting the group policy is one task, working with it is another. The following sections describe how to work with group policies from the user's perspective.

Obtain Group Policy Results

Use the **GPResult** command to obtain the Resultant Set of Policy (RSOP) for a particular user on a system. This command considers all of the security settings for both the computer and the user and creates a resultant policy—the policy that actually affects the user's security setup on the system. The **GPResult** command is one of the few commands where

typing `GPREsult` and pressing Enter displays the help information. If you want to see the RSoP for the user and computer, type `GPREsult /R` and press Enter. This report can be a little long, so `GPREsult` makes it possible to create an HTML report out of the information. Simply type `GPREsult /H MyReport.HTML` and press Enter. Figure 16.2 shows typical output from this command. You can likewise use the `/X Filename` command line switch (where *Filename* is the name of the file you want to use) to output the report in XML format suitable for import into a database.

Figure 16.2: `GPREsult` makes it possible to create a report from the RSoP information.



NOTE Microsoft provides a wealth of articles on RSoP. For example, you can see how RSoP affects Internet Protocol Security (IPSec) assignments at <http://technet2.microsoft.com/windowsserver/en/library/35675107-c728-47cd-8ad9-bfd-2d5e7fe0a1033.aspx>. You'll also find an excellent article on planning and logging RSoP at http://www.windowsnetworking.com/articles_tutorials/Resultant-Set-Policy-Planning-Logging.html.

You may decide you need something other than the default output. For example, if you need just the computer or the user information, you can type **GPResult /R /Scope:Computer** or **GPResult /R /Scope:User** and press Enter. Use the **/H** command line switch in place of the **/R** command line switch when you need an HTML report in place of the on-screen report. If you want even more information than the default report supplies, use the **/V** command line switch. For example, type **GPResult /R /V /Scope:Computer** if you want to discover detailed information about the computer. The **/Z** command line switch provides even more information for those who need it.

The **GPResult** utility works with the current user by default. However, you can use the **/User** command line switch to obtain information about other users. For example, if you want to find information about user Samantha, you'd type **GPResult /R /User Samantha** and press Enter. As with many command line utilities and commands, you can use the **/S Server**, **/U Username**, and **/P Password** command line switches to access information on remote systems.

Manage Group Policies

The Group Policy Update (**GPUpdate**) utility lets you update the group policies on a computer. Use this utility as a replacement for the now obsolete **/refreshpolicy** command line switch for the **SecEdit** utility. Using this utility ensures that essential group policy changes appear on a computer, especially systems that are on 24 hours per day. To ensure that the host system is updated after you make group policy changes, type **GPUpdate** and press Enter. If you want to apply only the computer changes, type **GPUpdate /Computer** and press Enter. Likewise, if you want to apply only the user changes, type **GPUpdate /User** and press Enter.

Foreground policy changes won't actually affect the user immediately in most cases. For example, if you give the user additional rights, the user won't actually see the change until the user logs back on to the system after a reboot or a logoff. To ensure foreground changes actually take effect, type **GPUpdate /Boot** or **GPUpdate /Logoff** and press Enter. Using the **/Logoff** command line switch is fine for all soft settings, such as security changes. However, if the policy update affects the hardware in some way, it's normally better to use the **/Boot** command line switch to force a reset of the hardware.

Obtain Session Status Information

The Query utility helps you see how users are employing resources on the current machine. You can learn about the processes users have started, the sessions that the machine is supporting, information about the users themselves, and basic Terminal Server information as well. The following sections describe this utility in more detail.

Get Process Information

Windows provides a number of ways to display the active processes. (A process isn't necessarily just an application—services also create processes.) For example, if you want to use the GUI approach, you can rely on Task Manager. The Query Process command provides a quick way of obtaining a list of running processes from the command line. If you want to discover the processes started by the current user, type **Query Process** and press Enter.

In some cases, you need to know more than the current user. To see all of the processes started by anyone, type **Query Process *** and press Enter. If you want to see the processes started by a specific user, but not the current user, type **Query Process Username** and press Enter, where *Username* is the name of the user that started the process. For example, if you want to see all of the system processes, type **Query Process System** and press Enter. It's also possible to see processes based on a session type. For example, administrators often need to know which services are running. To see this information, type **Query Process Services** and press Enter.

Get Session Information

The Query Session command helps you discover information about Remote Desktop sessions on the current machine. Actually, the utility also shows information about the user that's currently logged on to the machine and the services session (session 0 on most systems) as well. To see the basic session information, type **Query Session** and press Enter. If you want session statistics (such as the number of sessions the machine has created), type **Query Session /Counter** and press Enter.

Get User Information

The `Query User` command displays information about users logged on to the machine. To see all of the users logged on to the system, type **Query User** and press Enter. If you want to see a specific user's information, type **Query User *Username*** and press Enter, where *Username* is the name of the user you want to see. You can also type the session name to see all of the users logged on under a particular session or the session identifier. No matter how you request the information, you see the username, session name, session ID, state, idle time, and logon time as output.

Get Terminal Server Information

The `Query TermServer` command locates any Remote Desktop Session Host servers on the domain. It's important to stress domain in this case because the command doesn't appear to work with workgroups. You have to have Active Directory set up and the whole domain configured for this command to work (as is the case for a few other commands). To obtain a list of all of the Remote Desktop Session Host servers on the network, type **Query TermServer** and press Enter. If you want to locate a particular server, type **Query TermServer *ServerName***, where *ServerName* is the name of the server you want to locate, and press Enter.

Get the User's Identity

Batch and script files often require that you know the user's identity in order to perform certain tasks. For example, a user might not have the rights required to perform the entire task, so you can modify the batch or script file execution to take this issue into account. Knowing the user's name (which is part of their identity, along with the user's security identifier and other elements) can also help make the batch file or script friendlier because you can use the user's name in prompts. Finally, you sometimes need to know the user's name to accomplish the task, such as when you need to set user-level auditing. The following sections describe two ways you can use to detect the user's identity.

Obtain User Logon Information

The `QUser` utility is a very simple way to find the user's identity. If you type `QUser` and press Enter, you see the username, session name, session ID, state, idle time, and logon time for every user logged on to the system. In fact, you see all of these statistics whenever you use the `QUser` utility, but you can ask for a single user's information.

If you're interested in a particular user, then you can use the `QUser Name` command, where *Name* is the user's name. You might be interested in a particular kind of session, such as a console session. In this case, you use the `QUser SessionName` command, where *SessionName* is the name of the session. You'll see every user logged on using that session type. Finally, you can see which user is logged on by a session identifier. For example, if you want to see which user is logged on session 1, then you'd type `QUser 1` and press Enter. The first session, session 1, is normally the local user.

Discover User Identity

The `WhoAmI` utility is a utility that you used to download as part of the Windows Resource Kit (see <http://www.microsoft.com/downloads/details.aspx?FamilyID=3e89879d-6c0b-4f92-96c4-1016c187d429>). However, starting with Windows XP Professional and Windows 2003, you started getting this utility as part of the operating system. When you type `WhoAmI` and press Enter, you see just the logon name of the user. The logon name consists of the domain name/username. If the user isn't part of a domain, then you see machine name/username instead.

Sometimes you need more than just the user's logon name. To obtain the user's User Principal Name (UPN), type `WhoAmI /UPN` and press Enter. Likewise, to obtain the user's Fully Qualified Domain Name (FQDN), type `WhoAmI /FQDN` and press Enter. There are a few situations where you need the user's logon identifier. Type `WhoAmI /LoginID` and press Enter to obtain this information. Don't confuse the logon identifier with the user's Security Identifier (SID).

You can use `WhoAmI` to obtain more information about the user. For example, if you want to find the user's SID, type `WhoAmI /User` and press Enter. If you need to know the user's group affiliations, type `WhoAmI /Groups` and press Enter. Acquire the user's privileges by typing `WhoAmI /Priv` and press Enter. Finally, if you need everything that `WhoAmI` can tell you about the user, type `WhoAmI /All` and press Enter.

