

Configuring Windows Security

In this chapter:

Setting Permissions for Keys	204
Mapping Default Permissions	208
Taking Ownership of Keys	215
Auditing Registry Access	215
Preventing Local Registry Access	217
Restricting Remote Registry Access	217
Deploying Security Templates	218
Configuring New Security Features	226
Internet Explorer Privacy Settings	228
Internet Explorer Security Zones	229

Security is not the most interesting registry-related topic, nor is it the most popular. It is one of the most important topics facing IT today, however.

There are hundreds of facets of security, but this chapter focuses on just one: the registry. You can change a key's access control list (ACL). You can audit keys. You can also take ownership of keys. You can't do any of these things with individual values, though, like you can with individual files. Power users generally won't care much about registry security, but IT professionals often have no choice.

Just because you can edit keys' ACLs doesn't mean you should, however. Changing your registry's security is not a good idea unless you have a specific reason to do so. At best, you will make a change that's irrelevant, but at worst, you could prevent Microsoft Windows XP and Windows Server 2003 (Windows) from working properly. So why am I including registry security in this book at all? There are cases in which IT professionals must change the registry's default permissions to deploy software. That is a totally different story than tinkering with your registry's security out of curiosity. For example, you might have an application that users can run only when they log on to the operating system as a member of the Administrators group. Ouch. In a corporate environment, you don't want to dump all your users in this group. The

solution is to deploy Windows with custom permissions so that users can run those programs as a member of the Power Users or Users group. This is the most common scenario, and it's the primary focus of this chapter.

You have two methods of deploying custom permissions. First, you can do it manually. For the sake of completeness, I show you how to change a key's permissions in Registry Editor (Regedit). Second, you can build a security template, complete with custom registry permissions, and then apply that template to a computer manually. You wouldn't run around from desktop to desktop applying the template, though; you'd apply that template to your disk images before deployment. The second method is by using Group Policy. You create a Group Policy Object (GPO) and then import a security template into it to create a security policy for your network. Windows automatically applies the custom permissions in your template to the computer and user if that GPO is in scope. I don't talk about Group Policy a lot in this book, but the last section in Chapter 7, "Using Registry-Based Policy," points out a lot of good, free resources for learning more about it.

Windows XP Service Pack 2 (SP2) and Windows Server 2003 Service Pack 1 (SP1) provide a number of new security features. For example, the Windows Security Center helps users configure security for maximum protection. Windows Firewall prevents unwanted access to computers so that using the Internet and opening e-mail attachments are safer. This chapter doesn't discuss those features in detail; instead, it describes how to use the registry to customize these features. For more information about the security features in Windows XP SP2, see <http://www.microsoft.com/windowsxp/sp2/default.aspx>. For more information about the security features in Windows Server 2003, see <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/servicepack/default.aspx>.

Setting Permissions for Keys

Registry security is similar to file system security except that you can set permissions for keys only, not values. Other than that, the dialog boxes look similar, the permissions are similar, and so on. If you don't understand basic security concepts, take a moment and review them in Help and Support Center before tinkering with permissions. I don't include the basic concepts in this chapter because I assume that you're an IT professional and already understand the basics of security.

If you have full control of or own a registry key, you can edit its permissions for users and groups in the key's ACL:

1. In Regedit, click the key with the ACL that you want to edit.
2. On the Edit menu, click Permissions. (See Figure 8-1.)

3. In the Group Or User Names list, click the user or the group for whom you want to edit permissions, and then select the check box in the Allow or Deny column to allow or deny the following permissions:
 - ❑ **Full Control.** Grants the user or the group permission to open, edit, and take ownership of the key. This permission literally gives full control of the key.
 - ❑ **Read.** Grants the user or the group permission to read the key's contents but not to save changes made to it. Read this as *read-only*.
 - ❑ **Special Permissions.** Grants the user or the group a special combination of permissions. To grant special permissions, click Advanced. You learn more about this permission setting in the section “Assigning Special Permissions,” later in this chapter.

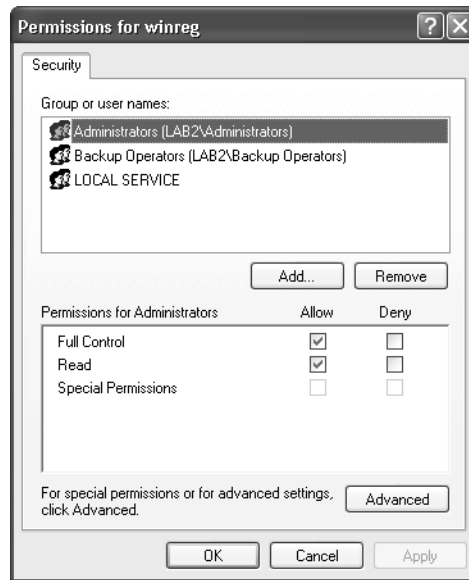


Figure 8-1 This dialog box is almost identical to the dialog box for file system security.

Sometimes the check boxes in the Permissions For *Name* area are shaded. You can't change them. The reason is that the key inherits that permission from the parent key. You can prevent a key from inheriting permissions, and you learn how to do that later in this chapter in the section “Assigning Special Permissions.”



Tip OK, you had your fun. You tinkered with your registry's security and satisfied your curiosity; but now what? You can easily restore the original permissions by applying the Setup Security template. You learn how to apply this template in the section “Modifying a Computer's Configuration,” later in this chapter.

Adding Users to ACLs

You can add users or groups to a key's existing ACL:

1. In Regedit, click the key with the ACL that you want to edit.
2. On the Edit menu, click Permissions, and then click Add.
3. In the Select Users, Computers, Or Groups dialog box, click Locations, and then click the computer, the domain, or the organizational unit in which you want to look for the user or the group that you want to add to the key's ACL.
4. In the Enter The Object Names To Select box, type the name of the user or the group that you want to add to the key's ACL, and then click OK.
5. In the Permissions For *Name* list, configure the permissions that you want to give the user or the group by selecting the Allow or Deny check box.



Tip In step 4, you type all or part of the user or the group name that you want to add to the key's ACL. If you don't know what the name is, you can search for it. First, if possible, narrow your search by choosing a location as I described in step 3. Then click Advanced, and click Find Now. Click the name of the user or the group that you want to add, and click OK. You can further narrow the results by clicking Object Types and then clearing the Built-In Security Principals check box.

The only real-world scenario I can think of for adding users to a key's ACL is allowing a group to access a computer's registry over the network, which you learn how to do in "Restricting Remote Registry Access," later in this chapter. Otherwise, adding a user or a group to a key's ACL is sometimes useful as a quick fix when an application can't access the settings it needs when users run it. Generally speaking, adding users or groups to a key's ACL does little harm, but if you're not careful, you can open holes in the security of Windows so wide that users and hackers can walk through them. And if the edit you're making will be required on more than one computer or user, consider deploying it as a security template. (See "Deploying Security Templates," later in this chapter.)

Removing Users from ACLs

Here's how to remove a user or a group from a key's ACL:

1. In Regedit, click the key with the ACL that you want to edit.
2. On the Edit menu, click Permissions.
3. Click the user or the group that you want to remove, and click Remove.



Caution Be wary of removing groups from keys' ACLs. Generally, the ACLs you see in Windows after installing it (Setup Security) are the bare minimum required for users to start and use the operating system. If you remove the Users or the Power Users group from a key, users in those groups can't read the key's values, and this is likely going to mangle the operating system or an application. If you dare remove the Administrators group from a key, you might not be able to manage the computer at all. Removing individual users from a key's ACL isn't necessarily a bad thing, however. Windows doesn't assign permissions to individual users, so those permissions might have gotten there by devious means. You should never remove users from their profile hives' ACLs, though. Doing so prevents them from accessing their own settings, of which they should have full control.

Assigning Special Permissions

Special permissions give you more granular control of a key's ACL than the basic Full Control and Read permissions. You can allow or deny users the ability to create subkeys, set values, read values, and so on. You can get very detailed. Here's how:

1. In Regedit, click the key with the ACL that you want to edit.
2. On the Edit menu, click Permissions.
3. In the Group Or User Names list, click the user or the group for whom you want to edit permissions. Add the user or the group if necessary. Then click Advanced.
4. Double-click the user or the group to whom you want to give special permissions. You see the Permission Entry For *Name* dialog box shown in Figure 8-2.

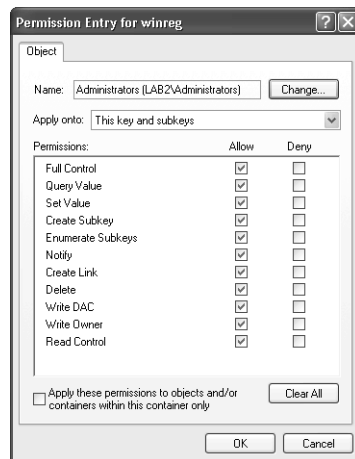


Figure 8-2 Special permissions give you finer control of a user or group's permissions to use a key, but assigning special permissions is generally unnecessary.

5. In the Apply Onto drop-down list, click one of the following:
 - This Key Only.** Applies the permissions to the selected key only.
 - This Key And Subkeys.** Applies the permissions to the selected key and all its subkeys. In other words, it applies them to the entire branch.
 - Subkeys Only.** Applies the permissions to all the key's subkeys but not to the key itself.

6. In the Permissions list, select the Allow or Deny check box for each permission that you want to allow or deny:
 - Full Control.** All the following permissions.
 - Query Value.** Read a value from the key.
 - Set Value.** Set a value in the key.
 - Create Subkey.** Create subkeys in the key.
 - Enumerate Subkeys.** Identify the key's subkeys.
 - Notify.** Receive notification events from the key.
 - Create Link.** Create symbolic links in the key.
 - Delete.** Delete the key or its values.
 - Write DAC.** Write the key's discretionary access control list.
 - Write Owner.** Change the key's owner.
 - Read Control.** Read the key's discretionary access control list.

A word about inheritance is necessary here. With inheritance enabled, subkeys inherit the permissions of their parent keys. In other words, if a key gives a group full control, all the key's subkeys also give that group full control. In fact, when you view the subkeys' ACLs, the Allow check box next to Full Control is shaded for that group because you can't change inherited permissions. There are a couple of actions that you can take to configure inheritance. First, you can prevent a subkey from inheriting its parent key's permissions: in the Advanced Security Settings For Key dialog box, clear the Inheritable Permission check box. Second, you can replace the ACLs of a key's subkeys, effectively resetting an entire branch to match a key's ACL: select the Replace Permission Entries On All Child Objects With Entries Shown Here That Apply To Child Objects check box.

Mapping Default Permissions

Understanding the registry's default permissions is useful if you're an IT professional deploying software. Knowing whether members of the Users group can change a particular setting helps you test applications prior to deployment and determine if the

application works with default permissions. If you determine that an application does work properly with the default permissions, then it's ready to deploy. If you determine that an application doesn't work properly with the default permissions, you must either fix the program or change the offending key's permissions. The easiest way to do that, of course, is by using security templates.

First you must understand the three fundamental groups in Windows: Users, Power Users, and Administrators. Through these groups, Windows provides different levels of access depending on each group's needs:

- **Users.** This group has the highest security because the default permissions given to it don't allow its members to change operating system data or other users' settings. Generally, users in this group can't change per-computer operating system and application settings. They can usually include programs certified for Windows that administrators deploy to their computers. Also, this group gives its members full control over everything in their user profiles, including their profile hives (HKCU). What frequently keeps IT professionals from assigning users to this group is that members can't usually run legacy applications. Rather than assign users to another group, deal with this problem by applying a compatible security template, which you learn how to do in the section titled "Deploying Security Templates," later in this chapter.
- **Power Users.** This group provides backward compatibility for running programs that aren't certified for Windows. The default permissions give this group the ability to change many per-computer operating system and program settings. Generally, if you have legacy applications that users can't run as members of the Users group and you're not going to use security templates, adding those users to the Power Users group allows the applications to run. However, this group does have enough permissions to install most applications; members can't change operating system files or install services. The permissions given to the Power Users group is somewhere in the middle of the Users and Administrators groups. It's similar to the Users group in Microsoft Windows NT 4.0. And no, members of this group can't add themselves to the Administrators group.
- **Administrators.** This group provides full control of the entire computer. Its members can change all operating system and application files. They can change all settings in the registry. Also, they can take ownership of keys and change a key's ACL. IT professionals are often tempted to add users to this group to avoid having trouble deploying applications that are otherwise difficult to install or run. Don't. Because users in this group can install anything they like or change any setting they like, viruses are free to do their damage and users are free to subject their configurations to the inevitable bout of human error. To secure your enterprise's desktops and reduce downtime, reserve this group for actual administrators. Even if you're an administrator, use your computer as a power

user for the same reasons. Instead, when you need to perform an administrative task, use a secondary logon to start a program as Administrator: hold down the SHIFT key while you right-click the program's shortcut, click Run As, and then type the account name and password that you want to use to run the program.

Table 8-1 describes the registry's default permissions after a fresh installation of Windows. (These permissions don't apply to Windows Server 2003 domain controllers.) Keep in mind that the resulting permissions are different if you upgrade from an earlier version of Windows to Windows XP or Windows Server 2003. I got these permissions from the security template that you use to restore Windows to *out of box* security. I've focused on the Users and Power Users groups because these are the primary issue. In most of these cases, the Administrators group has full control, as do the Creator Owner and System built-in accounts. In most cases—but not all—each key's permissions replace all subkeys' permissions. This is through the magic of inheritance, which you learned about in the preceding section.

When you see the word *Special* in the Power Users column, it means the group has special permissions on that key (and subkeys in most cases), and that permissions is usually the ability to modify values. The Power Users group doesn't ever get the Full Control, Create Link, Change Permissions, or Take Ownership permission for any key in the registry, though. The interesting thing about this table is that Windows gives the Users group Read permission and the Power Users group special permissions for all of HKLM\SOFTWARE. The remaining entries in the table are exceptions to this rule that limit access to specific keys in HKLM\SOFTWARE.

Table 8-1 Default Windows Installation Registry Permissions

Branch	Users	Power Users
hk1m\software	Read	Special
hk1m\software\classes	Read	Special
hk1m\software\classes\help	Read	Read
hk1m\software\classes\helpfile	Read	Read
hk1m\software\microsoft\ads\providers\ldap\extensions	Read	Read
hk1m\software\microsoft\ads\providers\nds	Read	Read
hk1m\software\microsoft\ads\providers\nwcompat	Read	Read
hk1m\software\microsoft\ads\providers\winnt	Read	Read
hk1m\software\microsoft\command processor	Read	Read
hk1m\software\microsoft\cryptography	Read	Read
hk1m\software\microsoft\cryptography\calais	None	None
hk1m\software\microsoft\driver signing	Read	Read
hk1m\software\microsoft\enterprisecertificates	Read	Read
hk1m\software\microsoft\msdtc	None	None

Table 8-1 Default Windows Installation Registry Permissions

Branch	Users	Power Users
hk1m\software\microsoft\netdde	None	None
hk1m\software\microsoft\non-driver signing	Read	Read
hk1m\software\microsoft\ole	Read	Read
hk1m\software\microsoft\protected storage system provider	None	None
hk1m\software\microsoft\rpc	Read	Read
hk1m\software\microsoft\secure	Read	Read
hk1m\software\microsoft\systemcertificates	Read	Read
hk1m\software\microsoft\upnp device host	Read	None
hk1m\software\microsoft\windows nt\currentversion\accessibility	Read	Read
hk1m\software\microsoft\windows nt\currentversion\aedebug	Read	Read
hk1m\software\microsoft\windows nt\currentversion\asr\commands	Read	Read
hk1m\software\microsoft\windows nt\currentversion\classes	Read	Read
hk1m\software\microsoft\windows nt\currentversion\drivers32	Read	Read
hk1m\software\microsoft\windows nt\currentversion\efs	Read	Read
hk1m\software\microsoft\windows nt\currentversion\font drivers	Read	Read
hk1m\software\microsoft\windows nt\currentversion\fontmapper	Read	Read
hk1m\software\microsoft\windows nt\currentversion\image file execution options	Read	Read
hk1m\software\microsoft\windows nt\currentversion\inifilemapping	Read	Read
hk1m\software\microsoft\windows nt\currentversion\perflib	None	None
hk1m\software\microsoft\windows nt\currentversion\perflib\009	None	None
hk1m\software\microsoft\windows nt\currentversion\profilelist	Read	Read
hk1m\software\microsoft\windows nt\currentversion\secdit	Read	Read
hk1m\software\microsoft\windows nt\currentversion\setup\recoveryconsole	Read	Read
hk1m\software\microsoft\windows nt\currentversion\svchost	Read	Read

Table 8-1 Default Windows Installation Registry Permissions

Branch	Users	Power Users
hklm\software\microsoft\windows nt\currentversion\terminal server\install\software\microsoft\windows\currentversion\runonce	Read	Read
hklm\software\microsoft\windows nt\currentversion\time zones	Read	Read
hklm\software\microsoft\windows nt\currentversion\windows	Read	Read
hklm\software\microsoft\windows nt\currentversion\winlogon	Read	Read
hklm\software\microsoft\windows\currentversion\explorer\user shell folders	Read	Read
hklm\software\microsoft\windows\currentversion\group policy	None	None
hklm\software\microsoft\windows\currentversion\installer	None	None
hklm\software\microsoft\windows\currentversion\policies	None	None
hklm\software\microsoft\windows\currentversion\reliability	Read	Read
hklm\software\microsoft\windows\currentversion\runonce	Read	Read
hklm\software\microsoft\windows\currentversion\runonceex	Read	Read
hklm\software\microsoft\windows\currentversion\telephony	Read	Special
hklm\software\policies	Read	Read
hklm\system	Read	Read
hklm\system\clone	None	None
hklm\system\controlset001	None	None
hklm\system\controlset001\services\dhcp\configurations	Read	Read
hklm\system\controlset001\services\dhcp\parameters	Read	Read
hklm\system\controlset001\services\dhcp\parameters\options	Read	Read
hklm\system\controlset001\services\dns cache\parameters	Read	Read
hklm\system\controlset001\services\mrxdav\encrypteddirectories	None	None
hklm\system\controlset001\services\netbt\parameters	Read	Read
hklm\system\controlset001\services\netbt\parameters\interfaces	Read	Read

Table 8-1 Default Windows Installation Registry Permissions

Branch	Users	Power Users
hk1m\system\controlset001\services\tcpip\linkage	Read	Read
hk1m\system\controlset001\services\tcpip\parameters	Read	Read
hk1m\system\controlset001\services\tcpip\parameters\adapters	Read	Read
hk1m\system\controlset001\services\tcpip\parameters\interfaces	Read	Read
hk1m\system\controlset002	None	None
hk1m\system\controlset003	None	None
hk1m\system\controlset004	None	None
hk1m\system\controlset005	None	None
hk1m\system\controlset006	None	None
hk1m\system\controlset007	None	None
hk1m\system\controlset008	None	None
hk1m\system\controlset009	None	None
hk1m\system\controlset010	None	None
hk1m\system\currentcontrolset\control\class	None	None
hk1m\system\currentcontrolset\control\keyboard layout	Read	Read
hk1m\system\currentcontrolset\control\keyboard layouts	Read	Read
hk1m\system\currentcontrolset\control\network	Read	Read
hk1m\system\currentcontrolset\control\securepipeservers\winreg	None	None
hk1m\system\currentcontrolset\control\session manager\executive	None	Special
hk1m\system\currentcontrolset\control\timezoneinformation	None	Special
hk1m\system\currentcontrolset\control\wmi\security	None	None
hk1m\system\currentcontrolset\enum	None	None
hk1m\system\currentcontrolset\hardware profiles	None	None
hk1m\system\currentcontrolset\services\apmgmt\security	None	None
hk1m\system\currentcontrolset\services\clipsrv\security	None	None
hk1m\system\currentcontrolset\services\cryptsvc\security	None	None
hk1m\system\currentcontrolset\services\dns cache	Read	Read
hk1m\system\currentcontrolset\services\ersvc\security	None	None
hk1m\system\currentcontrolset\services\eventlog\security	None	None

Table 8-1 Default Windows Installation Registry Permissions

Branch	Users	Power Users
hk1m\system\currentcontrolset\services\irenum\security	None	None
hk1m\system\currentcontrolset\services\netbt	Read	Read
hk1m\system\currentcontrolset\services\netdde\security	None	None
hk1m\system\currentcontrolset\services\netddedsdm\security	None	None
hk1m\system\currentcontrolset\services\remoteaccess	Read	Read
hk1m\system\currentcontrolset\services\rpcss\security	None	None
hk1m\system\currentcontrolset\services\samss\security	None	None
hk1m\system\currentcontrolset\services\scarddrv\security	None	None
hk1m\system\currentcontrolset\services\scardsvr\security	None	None
hk1m\system\currentcontrolset\services\stisvc\security	None	None
hk1m\system\currentcontrolset\services\sysmonlog\log queries	None	None
hk1m\system\currentcontrolset\services\tapisrv\security	None	None
hk1m\system\currentcontrolset\services\tcpip	Read	Read
hk1m\system\currentcontrolset\services\w32time\security	None	None
hk1m\system\currentcontrolset\services\wmi\security	None	None
hku\.default	Read	Read
hku\.default\software\microsoft\netdde	None	None
hku\.default\software\microsoft\protected storage system provider	None	None
hku\.default\software\microsoft\systemcertificates\root\protectedroots	None	None

Figuring out which keys an application uses is part science but mostly art. Sometimes I simply open the program's binary file in a text editor and look for strings that look like keys. Most often, I use a tool such as Winternals Registry Monitor (Regmon), which you learn how to use in Chapter 10, "Finding Registry Settings," to monitor registry activity while I run the program I'm putting through its paces. Then I record the different keys that the program references and check to see whether the Users or Power Users groups have the required permissions for those keys. Last, well-behaved applications report errors when they can't read or write a value in the registry. I wouldn't count on this behavior, however, because ill-behaved programs just bounce along happily even after encountering a registry error.

Taking Ownership of Keys

By default, Windows assigns ownership to the **HKLM** and **HKCU** as follows:

- Administrators own each subkey in **HKLM**.
- Users own each subkey in their profile hives, **HKCU**.

If you have full control of a key (and administrators usually do), you can take ownership of it if you're not already the owner by following these steps:

1. In Regedit, click the key for which you want to take ownership.
2. On the Edit menu, click Permissions; then click Advanced.
3. On the Owner tab, select the new owner, and then click OK.

Auditing Registry Access

Auditing registry access is a great way to track down registry settings, and it's one of the methods that I discuss in Chapter 10, "Finding Registry Settings." It's also a reasonable way to monitor access to sensitive settings. The problem with auditing the registry is that you must either get very specific about which key you're auditing or pay a severe performance penalty by auditing too much of the registry. It's a fine line between getting the information you need and grinding the computer to a halt.

Auditing a key is a three-step process. First you must enable Audit Policy. You can do that on the network using Group Policy, but that seems silly considering the scope of the performance impact. If you're using auditing as a troubleshooting tool or to track down a setting, turn on Audit Policy locally. In Control Panel, in Classic view, open the Administrative Tools folder, and launch Local Security Policy. You won't find Local Security Policy on a domain controller. In the left pane, under Local Policies, click Audit Policy. In the right pane, double-click Audit Object Access, and then select the Success and Failure check boxes. After you've enabled Audit Policy, use Regedit to audit individual keys, as follows:

1. In Regedit, click the key that you want to audit.
2. On the Edit menu, click Permissions; then click Advanced.
3. On the Auditing tab, shown in Figure 8-3, click Add.
4. In the Select Users, Computers, Or Groups dialog box, click Locations, and then click the computer, the domain, or the organizational unit in which you want to look for the user or the group that you want to audit.
5. In the Enter The Object Names To Select box, type the name of the user or the group that you want to add to the key's audit list, and then click OK.

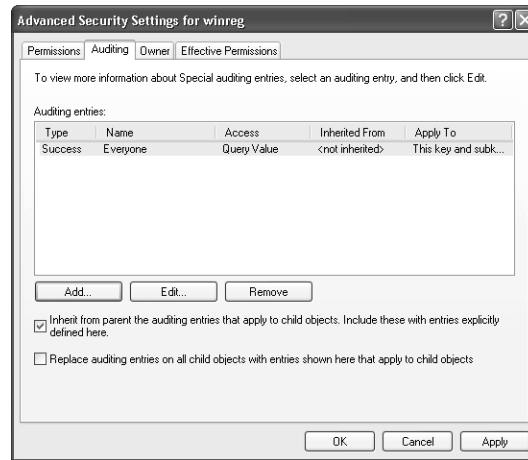


Figure 8-3 Audit keys sparingly because doing so can significantly impact performance.

6. In the Auditing Entry For *Name* dialog box, in the Access list, select both the Successful and Failed check boxes next to the activities for which you want to audit successful and failed attempts. These correspond to the permissions you learned about in the section “Assigning Special Permissions” earlier in this chapter:
 - Full Control
 - Query Value
 - Set Value
 - Create Subkey
 - Enumerate Subkeys
 - Notify
 - Create Link
 - Delete
 - Write DAC
 - Write Owner
 - Read Control

After enabling Audit Policy and auditing specific keys, check the results using Event Viewer. To open Event Viewer, in Control Panel, in Classic view, open the Administrative Tools folder, and launch Event Viewer. In Event Viewer’s left pane, click Security. You see each entry in the right pane, and the most recent entries are at the top of the list. Double-click any entry to see more details. The Event Properties dialog box tells you what type of access Windows detected, the object type, and the process that

accessed the key or the value. Chapter 10, “Finding Registry Settings,” shows you how to use this information to figure out where Windows or a program stores certain settings in the registry.

Preventing Local Registry Access

Whenever I bring up registry security, the inevitable question is always how to prevent users from accessing the registry. You can't. Remember that the registry contains settings that the user must be able to read for Windows to work properly. Users also must have full control of their profile hives for the operating system and applications to save their preferences. You can't prevent access—nor do you want to prevent it. The best you should hope for is limiting users' ability to edit the registry using Regedit or other registry editors.

The most elegant way to prevent access to Regedit is by enabling the **Prevent access to registry editing tools** policy. When users start Regedit, all they see is an error message that says “Registry editing has been disabled by your administrator.” The problem with this policy is that not all registry editors honor this policy. Nothing prevents a determined user from downloading a shareware registry editor, of which there are plenty, and using it. Another possibility is using Software Restriction Policies, which you can learn more about in Help and Support Center. Even this doesn't prevent users from running shareware registry editors unless you use Software Restriction Policies to completely restrict them to a short list of acceptable applications.

Restricting Remote Registry Access

Securing local access to the Windows registry is one thing; securing remote access is another. Windows gives members of the local Administrators and Backup Operators groups remote access to the registry. Because the Domain Admins group is a member of each computer's local Administrators group, all domain administrators can connect the registry of any computer that's joined to the domain. Also, Windows now limits remote access to the registry more than earlier versions of Windows.

There might be limited scenarios in which you want to open remote access to computers' registries. For example, in Active Directory, you might create an administrators group for each organizational unit and want to give it the ability to edit computers' registries if they belong to the organizational unit. To enable that group to remotely edit a computer's registry, add that group to the ACL of the key `HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg`. The problem you're going to run into is that although adding a group to `winreg` allows remote access, each key's ACL still determines which keys the group can change. So to allow a remote user or group to change a setting on the computer, add that user or group to the local Users, Power Users, or Administrators group.



Caution Don't open each computer's registry to security threats by haphazardly adding groups to the winreg key's ACL. Doing so creates a hole large enough for many Trojan viruses to infect Windows and invites predators to hack away at your infrastructure. The best practice is to limit remote registry access to domain administrators.

Deploying Security Templates

You use security templates to create a security policy for your computer or network. Rather than using the techniques that you learned about in this chapter to hunt-and-peck security on a computer, security templates give you a single place to configure a range of security settings and then deploy those settings to numerous computers. It's a little used, often misunderstood tool that organizes many of the available security settings in one place to make managing security a far easier job. It saddens me when administrators tell me their security woes and yet they've never heard of security templates, which would deal with most of their problems admirably. Security templates are an IT professional's best friend. Interested yet? I hope so.

You use a variety of tools to create and apply templates. First you use security templates to create and edit templates. Then you use either the Security Configuration And Analysis or Group Policy console to apply templates. This section walks you through the process of using these tools, starting with creating the Microsoft Management Console (MMC) that you'll use to edit templates, and ending with deploying templates on a network.

To begin with, here's an explanation of the different security settings in a template. The following list shows the different categories of settings you see in a security template. Following each category is a description of the settings that you can define within it.

- **Account Policies.** Password Policy, Account Lockout Policy, and Kerberos Policy
- **Local Policies.** Audit Policy, User Rights Assignment, and Security Options
- **Event Log.** Application, System, and Security Event Log settings
- **Restricted Groups.** Membership of security-sensitive groups
- **System Services.** Startup and permissions for system services
- **Registry.** Permissions for registry keys (the topic of this section)
- **File System.** Permissions for files and folders

Security templates are nothing more than text files that have the *.inf* extension. You can copy them, edit them, and so on. The file looks much like an INI file. You can create your own security templates from scratch, which I don't recommend because it's too much work with so much risk, or you can customize one of the predefined templates that come with Windows. Customizing a predefined template is definitely the way to go because most of the work is already done for you. Keep in mind that because only the Administrators group has permissions to change the default security template folder, %SystemRoot%\Security\Templates, only administrators can edit and apply security templates.



More Info Regini.exe is a tool that ships with Windows that you can use to script changes to registry security. It's simple to use and sometimes useful for changing keys' ACLs from logon scripts. It is a legacy tool that's superseded by more robust security features in Windows, however. For more information about using Regini.exe, see <http://support.microsoft.com/kb/264584> and <http://support.microsoft.com/?kbid=237607>.

Creating a Security Management Console

To make your job easier, create an MMC that includes all the tools you'll need for editing, analyzing, and applying security templates:

1. Click Start, Run; then type **mmc**, and click OK.
2. On the File menu, click Add/Remove Snap-in.
3. In the Add/Remove Snap-in dialog box, click Add.
4. In the Add Standalone Snap-In dialog box, select Security Templates, and then click Add.
5. Select Security Configuration And Analysis, and click Add.
6. Click Close, and then click OK.

After creating your console, save it to a file for quick access. On the File menu, click Save. I like to name the file *Templates.msc*. MMC saves your file in your Administrative Tools folder. To open it again quickly, click Start, All Programs, Administrative Tools, and then Templates (or whatever you called it). Figure 8-4 shows the console that I created as described in this section.

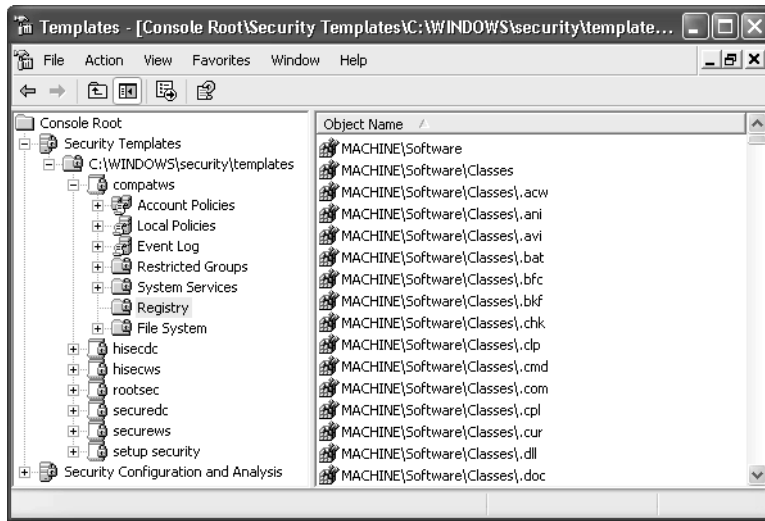


Figure 8-4 You build templates with SecurityTemplates, and you analyze and apply templates using Security Configuration And Analysis.

Choosing a Predefined Security Template

Windows comes with several predefined security templates. You almost never need to create a new template because you can usually just customize one of these predefined templates and save it to a different file. They provide starting points for applying security policies in different scenarios, whether those scenarios include one, one hundred, or thousands of computers. The following predefined security policies are in %SystemRoot%\Security\Templates by default:

- **Default security (Setup security.inf).** This template contains the default security settings that setup applies when you install Windows. It includes file system and registry permissions, too. If you need information about the operating system's default permissions, you'll find that information here. You can use this template to restore a computer to the original Windows security settings by applying it with Security Configuration And Analysis, but don't deploy it using Group Policy.
- **Compatible (Compatws.inf).** This template contains security settings that relax restrictions on the Users group enough to allow legacy applications to run. This is preferable to moving users from the Users group to the Power Users or, oh my, the Administrators groups. Specifically, this template changes the file system and registry permissions granted to the Users group so that they're consistent with legacy and other applications that aren't certified for Windows. This template also assumes that the administrator doesn't want users in the Power Users group, so it moves users from Power Users to the Users group. This template applies to workstations only, and you shouldn't apply it to servers.

- **DC Security (DC Security.inf).** This template is created when a server is promoted to a domain controller. It reflects file, registry, and system service default security settings. If you reapply this template, these settings are set to the default values. However, the template may overwrite permissions on new files, registry keys, and system services created by other programs.
- **Secure (Secure*.inf).** These templates tighten security settings that are least likely to affect application compatibility. Securedc.inf is for domain controllers, and Securews.inf is for workstations. It applies strong password, lockout, and audit settings, for example. It also limits the user of LAN Manager and Windows NT LAN Manager (NTLM) authentication protocols by configuring Windows to send only NTLM version 2 (NTLMv2) responses and configuring servers to refuse LAN Manager responses. Last, this template restricts anonymous users by preventing them from enumerating account names, enumerating shares, and translating Security Identifiers (SIDs). (See Chapter 1, “Learning the Basics.”) Test this template carefully before deploying it.
- **Highly Secure (Hisec*.inf).** These templates are supersets of the previous templates, and they apply even more restrictions. Hisecdc.inf is for domain controllers, and Hisecws.inf is for workstations. For example, this template sets the levels of encryption and signing that Windows requires for authentication and for data moving over secure channels. It requires strong encrypting and signing. Last, it removes all members of the Power Users groups and makes sure that only the Domain Admins group and the local Administrator are members of the local Administrators group. Test these templates to ensure compatibility with your infrastructure and applications because only certified applications are likely to run after applying this template.
- **System root security (Rootsec.inf).** This template defines root permissions for the Windows file system. It contains no registry permissions. It does apply permissions for the root of %SystemDrive%. You can apply this template to a computer to restore these permissions to the root of the system drive or to apply the same permissions to additional volumes.
- **No Terminal Server user SID (Notssid.inf).** This template removes unnecessary Terminal Server SIDs from the file system and registry when running Terminal Server in application compatibility mode. If possible, run Terminal Server in full security mode instead, a mode in which the Terminal Server SID isn’t used at all.

Most of these security templates are incremental. They modify the default or existing security settings if those settings are already configured on the computer. Other than the Setup Security template, they don’t configure the default security settings before changing the computer’s security configuration. Also, you can’t use security templates to secure Windows when you use the FAT file system.

You can view these templates in your MMC. In the console's left pane, double-click a security template to open it. By default, the templates are under `C:\Windows\Security\Templates`, as shown under your console's Security Templates node. You can add a new path, however. Right-click Security Templates, and then click New Template Search Path. You'll see both the previous and new paths listed under Security Templates. If you want to remove a path from Security Templates, right-click it, and then click Delete.

Building a Custom Security Template

The hard way to create a custom security template is to start from scratch:

1. In Security Templates, right-click the folder in which you want to create the new template, and then click New Template.
2. In the Template Name box, type the name of the new template, and in the Description box, type a brief but useful description of your new template, and click OK.
3. In the left pane, double-click the new security template to open it. Select a security area, such as Registry, in the left pane, and configure that area's security settings in the right pane.

That's the hard way, and it's definitely not the way I recommend. First, it's too labor-intensive. Second, it's error-prone. The best way to create a security template is to start with one of the predefined templates, save it to a new file, and then edit it—carefully. Most of the times I've done this, I started with the `Compatws.inf` template file and customized it as necessary to give a legacy application enough room to work. Here's how:

1. In Security Templates, double-click `C:\Windows\Security\Templates`.
2. Right-click the predefined template that you want to customize, click Save As, type a new file name for the security template, and click Save.
3. In the left pane, double-click the new security template to open it. Select a security area, such as Registry, in the left pane, and configure that area's security settings in the right pane.

Because this is a registry book, I'll give you a little more detail about configuring registry security in a template. In the left pane of Security Templates, double-click your template, and then click Registry. You'll see a list of registry keys in the right pane. To add a key to the list, right-click Registry, and then click Add Key. Because the list already covers all of `HKLM`, add exceptions to the settings that the template defines for

HKLM\SOFTWARE and HKLM\SYSTEM. To edit a key's settings, double-click it, and then select one of the following options:

- **Configure This Key Then.** After selecting this option, select one of the following:
 - **Propagate Inheritable Permissions To All Subkeys.** The key's subkey inherits the key's security settings, assuming that the subkeys' security settings don't block inheritance. In case of a conflict, the subkey's explicit permissions overwrite the permissions that they inherit from the parent key.
 - **Replace Existing Permissions On All Subkeys With Inheritable Permissions.** The key's permissions overwrite all its subkeys' permissions. In other words, each subkey's permissions will be identical to the parent key's permissions. If you select this option and apply the template, the change is permanent unless you change it by applying a different template to the registry.
- **Do Not Allow Permissions On This Key To Be Replaced.** Select this option if you don't want to configure the key or its subkey's permissions.

To edit the actual permissions that you want the template to apply to a key, click Edit Security. You do this in the same Security For *Name* dialog box that you saw earlier in this chapter. You can add and remove groups. You can allow or deny permissions for different users and groups to perform various tasks. You can audit users' and groups' access to a key. You can also change ownership of a key. When you apply the template to a computer or deploy the template through Group Policy, the key receives the permissions that you define in this dialog box.

Analyzing a Computer's Configuration

With your custom template in hand, you can use it to analyze a computer's security configuration. Security Configuration And Analysis enables you to compare the current state of the computer's security configuration to the settings defined in the template. You can use this tool to make immediate changes to the computer's configuration, such as when troubleshooting a problem. You can also use it to track and ensure a certain level of security as part of your enterprise risk management program, detecting flaws in security as they occur over time.

Here's how to analyze a computer's security using Security Configuration And Analysis:

1. Right-click Security Configuration And Analysis, which you added to your console in the section titled "Creating a Security Management Console," earlier in this chapter, and then click Open Database.

2. In the Open Database dialog box, do one of the following:
 - ❑ To create a new analysis database, type the name of your new database in the File Name box, and click Open (you don't have a database initially). Then in the Import Template dialog box, select a template and click Open.
 - ❑ To open an existing analysis database, type the name of an existing database in the File Name box, and click Open.
3. Right-click Security Configuration And Analysis, click Analyze Computer Now, and then accept the default log file path or specify a new one.

Security Configuration And Analysis compares the computer's current security against the analysis database. If you import multiple templates into the database, which you can do by right-clicking Security Configuration And Analysis and then clicking Import Template, the tool merges the templates together to create one template. If it detects a conflict, the last template that you loaded has precedence (last in, first out). After Security Configuration And Analysis analyzes the computer, it displays results that you can browse. The organization of these results is the same as in security templates. The difference is that Security Configuration And Analysis displays the following indicators that show whether a current setting matches or is inconsistent with a setting defined in the template:

- **Red X.** The setting is in the analysis database and on the computer, but the two versions don't match. The trick is to drill down through settings that have a red X next to them until you isolate the specific problem.
- **Green Check Mark.** The setting is in the analysis database and on the computer, and the two match.
- **Question Mark.** The setting is not in the analysis database and was not analyzed. This might also mean that the user who ran Security Configuration And Analysis didn't have the permissions necessary to do so.
- **Exclamation Point.** The setting is in the analysis database but not on the computer. A registry key might exist in the database but not on the computer.
- **No Indicator.** The setting is not in the database or on the computer.

What do you do with any discrepancies you find between the analysis database and the computer's settings? First you can update the database by double-clicking the troublesome registry setting and clicking Edit Security. (See Figure 8-5.) This updates the database but not the template, however. Also, it doesn't change the computer's settings. To do that, see the next section. You can also import a more appropriate template for that computer or an updated template into the database and then analyze it again. To avoid problems that result from merging templates, consider creating a new database if you use a new or updated template.

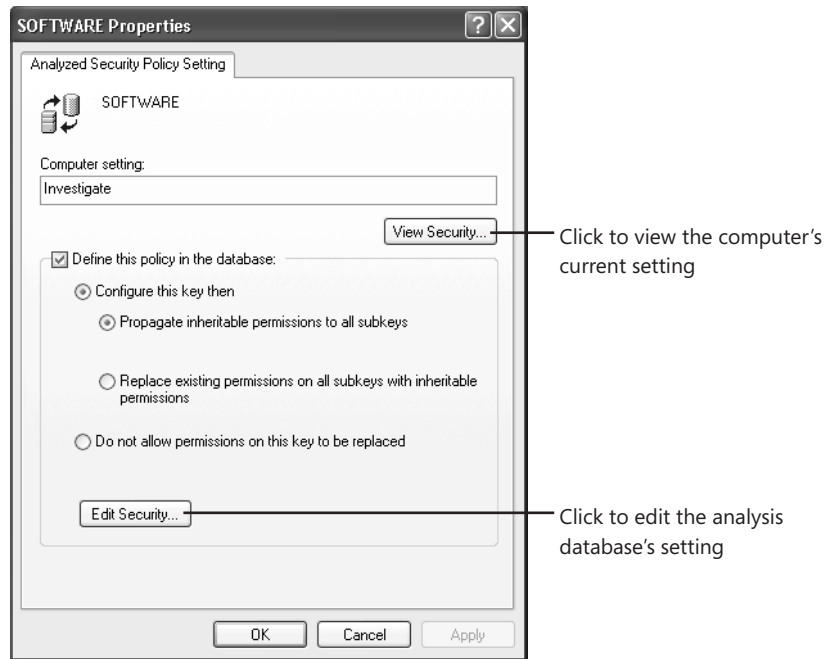


Figure 8-5 You can view and edit settings in the registry setting's Properties dialog box.

Modifying a Computer's Configuration

After you've created a security template and verified it by analyzing computers using Security Configuration And Analysis, you're ready to apply it to the computer:

1. Right-click Security Configuration And Analysis, and then click Open Database.
2. In the Open Database dialog box, do one of the following:
 - ❑ To create a new database, type the name of your new database in the File Name box, and click Open. Then in the Import Template dialog box, click a template, and click Open.
 - ❑ To open an existing database, type the name of an existing database in the File Name box, and click Open. If you modified a database without updating the template on which it's based, make sure you open the existing database.
3. Right-click Security Configuration And Analysis, click Configure Computer Now, and then accept the default log file path or specify a new one.

Deploying Security Templates on the Network

In the preceding section, “Modifying a Computer’s Configuration,” you learned how to apply a security template to a computer manually. This is fine for one-off scenarios, but it’s not the way to deploy security templates to multiple computers on the network. To deploy templates on a network, use Group Policy: create a new GPO, and then edit it. In the Group Policy Editor, right-click Security Settings, and then click Import Policy. Click the template that you want to apply, and then click Open.

It’s so simple, but I don’t want to make light of this: deploying security templates on your network requires careful planning. You must first identify the templates that your network requires. Then you must identify which organizational units get which security templates. For example, if the sales department uses a legacy application that requires the Users group to have full control of certain registry keys, document and test the security template, and then import the template into a GPO that you assign to the sales department’s organizational unit. Ideally, you’ll account for security templates early in the deployment planning process. What really ends up happening, unless they planned carefully, is that IT professionals use security templates as a big fire hose to put out fires created by lack of foresight and planning.

Configuring New Security Features

New enhancements are available in Microsoft Windows XP SP2 for improving the manageability and visibility of key security capabilities in personal computers. New enhancements include the following:

- The new Windows Security Center feature tells you the status of three major security components: Windows Firewall, Automatic Updates, and Virus Protection.
- Windows Security Center indicates whether key security capabilities are turned on and up to date. Windows Security Center notifies you when updates are required or when you must take additional steps to help make your computer secure.
- You can manage Windows Security Center by using Active Directory Group Policy settings. By default, Windows Security Center is turned off in domain environments.

The following sections describe how you can configure Windows XP SP2 and Windows Server 2003 SP1 security features. These features include the new Windows Security Center (Windows XP) and Windows Firewall. The question I’ve been most frequently asked since the release of SP2 is how to configure these two features.

Security Center Alerts

The Windows Security Center displays alerts in popup balloons when the firewall, the virus scanner, or Automatic Updates is not configured properly or out of date. You see these alerts in the system tray. You can disable these alerts by using the registry. Table 8-2 describes the `REG_DWORD` values for each type of alert. You set these values in `HKLM\SOFTWARE\Microsoft\Security Center`. (Create the key and settings if they don't already exist.) For example, to prevent Windows Security Center from displaying alerts when the Windows Firewall is not enabled (a configuration that Microsoft recommends against), set `FirewallDisableNotify` to `0x01`.

Table 8-2 Security Center Settings

Name	Type	Values
<code>AntiVirusDisableNotify</code>	<code>REG_DWORD</code>	0x00—Disable AntiVirus alerts. 0x01—Display AntiVirus alerts.
<code>AntiVirusOverride</code>	<code>REG_DWORD</code>	0x00—Windows Security Center monitors AntiVirus. 0x01—Windows Security Center doesn't monitor AntiVirus.
<code>FirewallDisableNotify</code>	<code>REG_DWORD</code>	0x00—Disable firewall alerts. 0x01—Display firewall alerts.
<code>FirewallOverride</code>	<code>REG_DWORD</code>	0x00—Windows Security Center monitors the firewall. 0x01—Windows Security Center doesn't monitor the firewall.
<code>UpdatesDisableNotify</code>	<code>REG_DWORD</code>	0x00—Disable Automatic Update alerts. 0x01—Display Automatic Update alerts.

Windows Firewall

Windows XP SP2 and Windows Server 2003 SP1 include the new Windows Firewall. Most companies and many enthusiasts will want to customize the Windows Firewall during installation. Microsoft provides three methods of doing so. The best way to manage Windows Firewall settings in a business environment is to use the new Windows Firewall Group Policy settings. This method requires the use of Active Directory with either Windows 2000 or Windows Server 2003 domain controllers. For more information, see <http://www.microsoft.com/technet/prodtechnol/winxp/dep/depfwset/wfsp2wgp.msp>.

The following list describes methods that don't require Group Policy:

- **Unattended-setup answer file.** The unattended-setup answer file (unattend.txt) for Windows XP SP2 has options to configure Windows Firewall settings when running an unattended setup of Windows XP SP2.
- **Netfw.inf.** The Netfw.inf file for Windows XP SP2 can configure the Windows Firewall by specifying a set of registry settings equivalent to the options available from the Windows Firewall component in Control Panel and through Windows Firewall Group Policy settings when a user is performing an interactive setup of Windows XP SP2.
- **Netsh script.** To configure computers running Windows XP with SP2 after SP2 has been installed, you can have your users run a script file, such as a .BAT or a .CMD file, that contains the series of Netsh.exe commands to configure the Windows Firewall operational mode, allowed programs, allowed ports, etc.
- **Custom configuration programs.** To configure computers running Windows XP with SP2 after Windows XP SP2 has been installed, you can have your users run a custom configuration program that uses the new Windows Firewall configuration APIs to configure the Windows Firewall for operation mode, allowed programs, allowed ports, and other settings.

For more information about using these options, see <http://www.microsoft.com/technet/prodtechnol/winxppro/deploy/depfwset/wfsp2ngp.msp>.

You can disable Windows Firewall by using the registry. The settings are in `HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall`. (Create the key and values if they don't already exist.) First, there are two subkeys: `DomainProfile` and `StandardProfile`. The settings in `DomainProfile` apply when the computer is currently connected to the domain. The settings in the `StandardProfile` apply when the computer isn't currently connected to the domain (a disconnected laptop computer, for example). Within each of those two subkeys, create the value `EnableFirewall`. Set this value to `0x00` to disable the firewall in that scenario, or set it to `0x01` to enable it.

Internet Explorer Privacy Settings

Microsoft Internet Explorer 6 added a Privacy tab to give users more control over cookies. There are different levels of privacy on the Internet zone, and they are stored in the registry at the same location as the security zones.

You can also add a site to allow or to block cookies based on the site, regardless of the privacy policy on the Web site. Those registry keys are stored in `HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\P3P\History`. In this key,

you see the domains that have been added as a managed site. These domains are set to one of the following values:

- `0x00000005`. Always Block
- `0x00000001`. Always Allow

Internet Explorer Security Zones

Internet Explorer security zones settings are stored in `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings` and `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings`. By default, security zones settings are stored in `HKCU`. The settings for one user do not affect the settings for another. The `Internet Settings` key has the following subkeys:

- `TemplatePolicies`
- `ZoneMap`
- `Zones`

If the `Security Zones: Use only machine settings` setting in Group Policy is enabled, or if the `Security_HKLM_only` `REG_DWORD` value is present and has a value of 1 in `HKLM\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings`, only local computer settings are used and all users have the same security settings. With the `Security_HKLM_only` policy enabled, `HKLM` values will be used by Internet Explorer, but the `HKCU` values will still be displayed in the zone settings on the Security tab in Internet Explorer. This is by design and there are no plans to change this functionality. If the `Security Zones: Use only machine settings` setting is not enabled in Group Policy, or if the `Security_HKLM_only` `REG_DWORD` value does not exist or is set to 0, computer settings are used along with user settings. However, only user settings appear in Internet Options. For example, when this `REG_DWORD` value does not exist or is set to 0, `HKLM` settings are read along with `HKCU` settings, but only `HKCU` settings appear in the Internet Options.

TemplatePolicies

The `TemplatePolicies` key determines the settings of the default security zone levels (Low, Medium Low, Medium, and High). You can change the security level settings from the default settings. However, you cannot add additional security levels. The keys contain values that determine the setting for the security zone. Each key contains a `Description` string value and a `Display Name` string value that determine the text that appears on the Security tab for each security level.

ZoneMap

The `ZoneMap` key contains the following keys:

- **Domains.** The `Domains` key contains domains and protocols that have been added to change their behavior from the default behavior. When a domain is added, a key is added to the `Domains` key. Subdomains appear as keys under the domain where they belong. Each key that lists a domain contains a `REG_DWORD` with a value name of the affected protocol. The value of the `REG_DWORD` is the same as the numeric value of the security zone where the domain is added.
- **ProtocolDefaults.** The `ProtocolDefaults` key specifies the default security zone that is used for a particular protocol (ftp, http, or https). To change the default setting, you can either add a protocol to a security zone by clicking Sites on the Security tab, or you can add a `REG_DWORD` value under the `Domains` key. The name of the `REG_DWORD` value must match the protocol name, and it must not contain any colons (:) or slashes (/).

The `ProtocolDefaults` key also contains `REG_DWORD` values that specify the default security zones where a protocol is used. You cannot use the controls on the Security tab to change these values. This setting is used when a particular Web site does not fall in a security zone.

- **Ranges.** The `Ranges` key contains ranges of TCP/IP addresses. Each TCP/IP range that you specify appears in an arbitrarily named key. This key contains a string value (`:Range`) that contains the specified TCP/IP range. For each protocol, a `REG_DWORD` value is added that contains the numeric value of the security zone for the specified IP range.

When the `Urlmon.dll` file uses the `MapUrlToZone` public function to resolve a particular URL to a security zone, it uses one of the following methods:

- If the URL contains a fully qualified domain name (FQDN), the `Domains` key is processed. In this method, an exact site match overwrites a random match.
- If the URL contains an IP address, the `Ranges` key is processed. The IP address of the URL is compared to the `:Range` value that is contained in each of the arbitrarily named keys under the `Ranges` key.



Note Because arbitrarily named keys are processed in the order that they were added to the registry, this method might find a random match before it finds an exact match. If so, the URL might be executed in a different security zone than the zone where it is typically assigned. This behavior is by design.

Zones

The **zones** key contains keys that represent each security zone that is defined for the computer. By default, the following five zones are defined (numbered zero through four):

- 0. My Computer
- 1. Local Intranet Zone
- 2. Trusted Sites Zone
- 3. Internet Zone
- 4. Restricted Sites Zone



Note By default, My Computer does not appear in the Zone box on the Security tab.

Each of these keys contains the following **REG_DWORD** values that represent corresponding settings on the custom Security tab:

- 1001. Download signed ActiveX controls
- 1004. Download unsigned ActiveX controls
- 1200. Run ActiveX controls and plug-ins
Run ActiveX controls and plug-ins (1200) has an extra setting named **Administrator approved**. When this setting is turned on, the **REG_DWORD** value is **0x00010000**, and **HKCU\Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\AllowedControls** is checked for a list of approved controls.
- 1201. Initialize and script ActiveX controls not marked as safe
- 1206. Allow scripting of Internet Explorer Webbrowser control
- 1400. Active scripting
- 1402. Scripting of Java applets
- 1405. Script ActiveX controls marked as safe for scripting
- 1406. Access data sources across domains
- 1407. Allow paste operations via script
- 1601. Submit non-encrypted form data

- 1604. Font download
- 1605. Run Java
- 1606. Userdata persistence
- 1607. Navigate sub-frames across different domains
- 1608. Allow META REFRESH
- 1609. Display mixed content
- 1800. Installation of desktop items
- 1802. Drag and drop or copy and paste files
- 1803. File Download

There is no prompt setting for File Download (1803) because it is either allowed or not allowed.

- 1804. Launching programs and files in an IFRAME
- 1805. Launching programs and files in webview
- 1806. Launching applications and unsafe files
- 1807. Reserved
- 1808. Reserved
- 1809. Use Pop-up Blocker
- 1A00. Logon

Logon setting (1A00) may have any one of the following values:

- 0x00000000. Automatically logon with current username and password
- 0x00010000. Prompt for user name and password
- 0x00020000. Automatic logon only in the Intranet zone
- 0x00030000. Anonymous logon
- 1A02. Allow persistent cookies that are stored on your computer
- 1A03. Allow per-session cookies (not stored)
- 1A04. Don't prompt for client certificate selection when no certificates or only one certificate exists
- 1A05. Allow 3rd party persistent cookies
- 1A06. Allow 3rd party session cookies

- 1A10. Privacy Settings

Privacy Settings (1A10) is used by the Privacy tab slider. The `REG_DWORD` values are in the following list:

- 00000003. Block All Cookies
 - 00000001. High
 - 00000001. Medium High
 - 00000001. Medium
 - 00000001. Low
- 00000000. Accept All Cookies

- 1C00. Java permissions

The Java Permissions setting (1C00) has the following five possible `REG_BINARY` values (binary):

- 00 00 00 00. Disable Java
- 00 00 01 00. High safety
- 00 00 02 00. Medium safety
- 00 00 03 00. Low safety
- 00 00 80 00. Custom

- 1E05. Software channel permissions

Software channel permissions (1E05) has three different values:

- 00010000. High
- 00020000. Medium
- 00030000. Low

- 1F00. Reserved

- 2000. Binary and script behaviors

- 2001. Run .NET components signed with Authenticode

- 2004. Run .NET components not signed with Authenticode

- 2100. Open files based on content, not file extension

- 2101. Web sites in less privileged Web content zone can navigate into this zone

- 2102. Allow script-initiated windows without size or position constraints

- 2200. Automatic prompting for file downloads
- 2201. Automatic prompting for ActiveX controls
- 2300. Allow Web pages to use restricted protocols for active content
- {AEBA21FA-782A-4A90-978D-B72164C80120}First Party Cookie
- {A8A88C49-5EB2-4990-A1A2-0876022C854F}Third Party Cookie

Unless stated otherwise, each **REG_DWORD** value is equal to zero, one, or three. Typically, a setting of zero sets a specific action as permitted, a setting of one causes a prompt to appear, and a setting of three does not allow the specific action.

Each security zone also contains the **Description** string value and the **Display Name** string value. The text of these values appears on the Security tab when you click a zone in the Zone box. There is also an **Icon** string value that sets the icon that appears for each zone. Except for the My Computer zone, each zone contains a **CurrentLevel**, a **MinLevel**, and a **RecommendedLevel** **REG_DWORD** value. The **MinLevel** value sets the lowest setting that can be used before you receive a warning message, **CurrentLevel** is the current setting for the zone, and **RecommendedLevel** is the recommended level for the zone. The following list describes the settings for these values:

- **0x00010000**. Low Security
- **0x00010500**. Medium Low Security
- **0x00011000**. Medium Security
- **0x00012000**. High Security

The **Flags** **REG_DWORD** value determines the ability of the user to modify the security zone's properties. To determine the **Flags** value, add the numbers of the appropriate settings together. The following **Flags** values are available:

- 1. Allow changes to custom settings
- 2. Allow users to add Web sites to this zone
- 4. Require verified Web sites (https protocol)
- 8. Include Web sites that bypass the proxy server
- 16. Include Web sites not listed in other zones
- 32. Do not show security zone in Internet Properties (default setting for My Computer)
- 64. Show the Requires Server Verification dialog box
- 128. Treat Universal Naming Connections (UNCs) as intranet connections