

Basic IIS Lockdown Using Scripts and Group Policy

Mark Squire 2003-08-12

Microsoft Active Directory and Group Policy have a feature-rich set of tools and processes to help save an administrator time and energy in maintaining security within the domain. Locking down a server requires many steps to complete, and depending on the extent to which the server is locked down, it can take up to several hours. This paper is primarily written for system administrators who want to make their life managing IIS easier using scripts with Active Directory and Group Policy.

Common scenario

Suppose your company needs you to set up an IIS server, possibly in a DMZ. Being security conscious, you want to implement all of the recommended procedures set forth by Microsoft. Oh and by the way, you also need that server built yesterday. As with anything else, business moves at blistering speeds. Often no thought is given to security because no time is allotted for it. This is where Active Directory and a little ingenuity can come in handy.

For example . . .

In Group Policy there is a nice little feature which you will become very familiar with in this series called Startup and Shutdown scripts. They are not too much different than the scripting that was used in the NT days, except that you can set these policies to apply to groups of computers if you choose.

The Startup script setting in Group Policy, as you can imagine, controls what is run during startup. These scripts run before the user ever gets the cntrl-alt-del logon screen, so they are executed with the privileges of the local system account. This gives us a lot of flexibility, and also a lot of power so be careful!

For all of the things the Group Policy can do, there is a lot that it cannot do. For all of the stuff it can't do, there is vbscript! Vbscript is great because it is designed to help an administrator automate many of the common tasks of systems administration, including (in this case) locking down a server. Team them up and you have a formidable force. For instance the following code:

```
Dim Site
Dim ServerName
Dim SiteIndex
ServerName = "LocalHost"
SiteIndex = "1"
Set Site = GetObject("IIS://" & ServerName & "/W3SVC/" & SiteIndex)
Site.LogExtFileDate = True
Site.LogExtFileTime = True
Site.LogExtFileClientIp = True
Site.LogExtFileUserName = True
Site.LogExtFileSiteName = False
Site.LogExtFileComputerName = False
Site.LogExtFileServerIp = True
Site.LogExtFileServerPort = True
Site.LogExtFileMethod = True
Site.LogExtFileUriStem = True
Site.LogExtFileUriQuery = False
Site.LogExtFileHttpStatus = False
Site.LogExtFileWin32Status = False
```

```
Site.LogExtFileBytesSent = False
Site.LogExtFileBytesRecv = False
Site.LogExtFileTimeTaken = False
Site.LogExtFileProtocolVersion = False
Site.LogExtFileUserAgent = True
Site.LogExtFileCookie = False
Site.LogExtFileReferer = False
Site.SetInfo
```

The above script tells IIS to set logging. Modifying this is as simple as changing the True and False values. Cut and paste this into a notepad session, and save it as "logging.vbs". Be sure to change the file type to all files so that notepad doesn't add a .txt at the end of it.

Now, go to the webserver in question, take a look at your logging settings in IISAdmin. Make careful note of them. Next, find a way to get that script onto the webserver somewhere (I recommend the desktop for the moment), and double click on it and marvel. "But nothing happened" you may say. Not true. Close out of the IISAdmin, then open it back up again, and check the logging settings. Viola! New settings, and all you did was double click on that one little script.

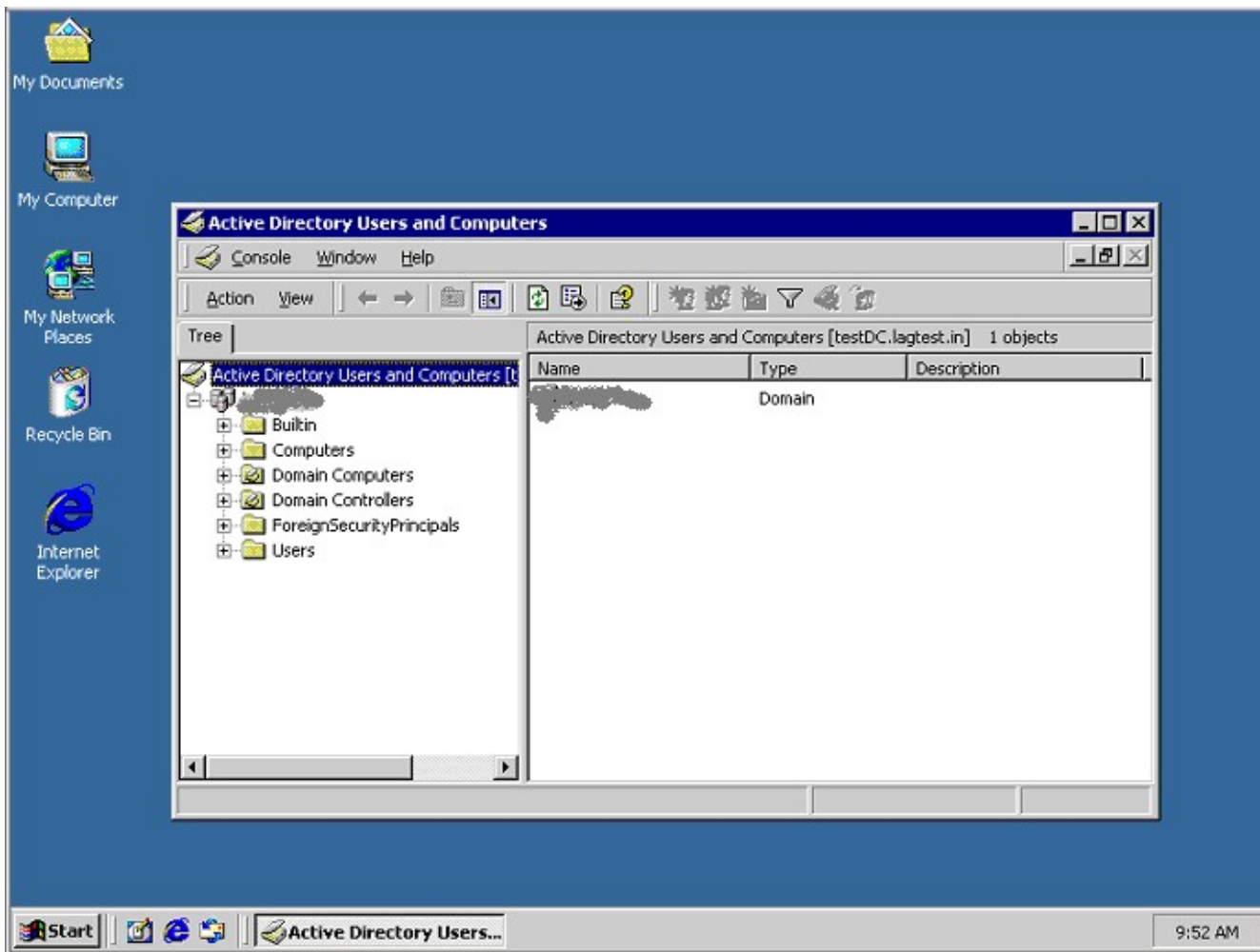
Applying this example

Let us step back and bask in the moment. With a simple double-click, you were able to completely automate the process of opening IISAdmin, clicking on the tree, the tabs, buttons, more buttons, all of the APPLYS and OKs and everything in between. Believe it or not, just to change those one or two settings requires some serious clicking, and when you have a stack of work to get done, the time and all the steps add up. Why keep doing the same thing over and over again in your job when you don't have to? Automate and leverage the system to do your work for you.

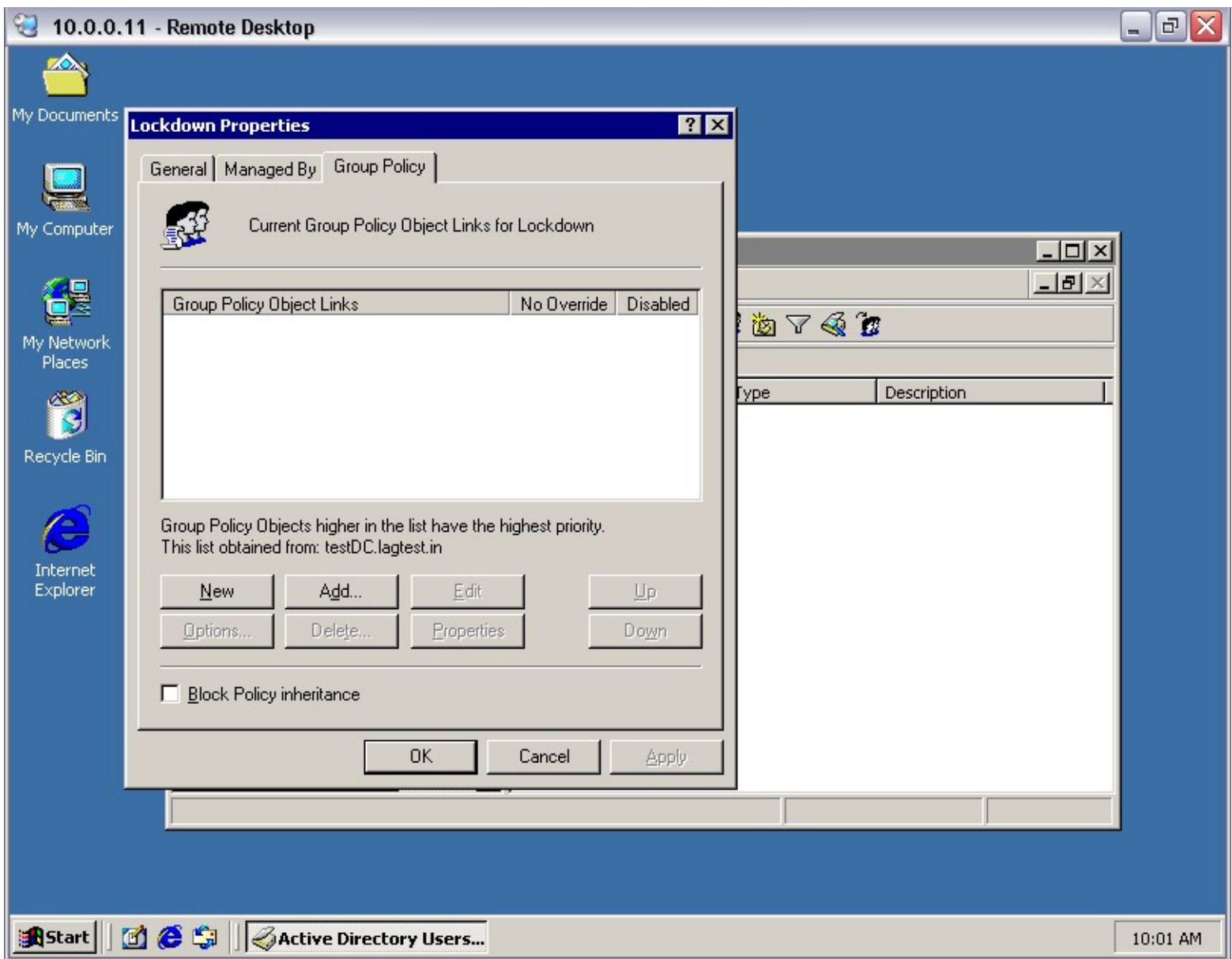
If you had an army of these scripts, you could automate nearly everything that needs to be done to a machine after the OS has been installed. It is only limited by your imagination and your craftiness with vbscript. Do you have to be a genius with VBScript to do this though? No, and that is the beauty of it. First, much is available on a simple Google search for "vbs." Second there are newsgroups with extremely helpful people. Third I am going to provide you with a host of scripts that will help you do the most basic lockdown procedures set forth by Microsoft for IIS 5.0 and Windows 2000 that aren't covered in group policy or any security templates.

So here is where we are going with this. First, change the logging settings manually in IISAdmin so they are completely different than what we just set (it doesn't matter what those settings are, but just make sure they are different, and you take note of them). We will see why later. Remember those startup scripts in Group Policy we mentioned earlier? We are going to use that setting to tell Windows to run that script above by using the startup scripts in Group Policy. Copy that script onto a share somewhere on the network. Being a security guy, I have to tell you to set appropriate permissions on that particular share.

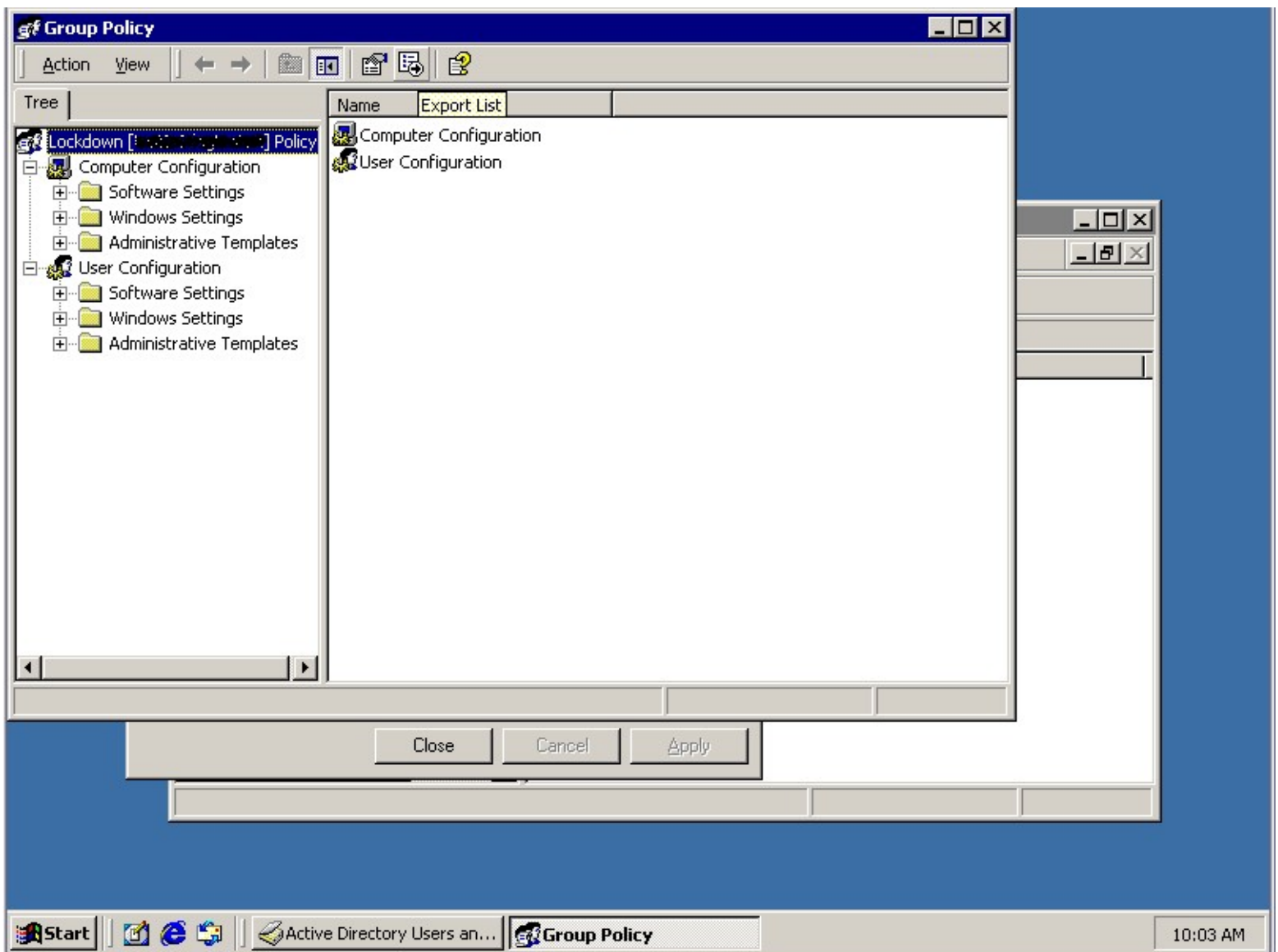
Okay, that is out of the way, now you have shared out the directory to your script, be sure to make note of its network path. Lets get into group policy. Start -> Programs -> Administrative Tools -> Active Directory Users and Computers. Your screen should look something like this:



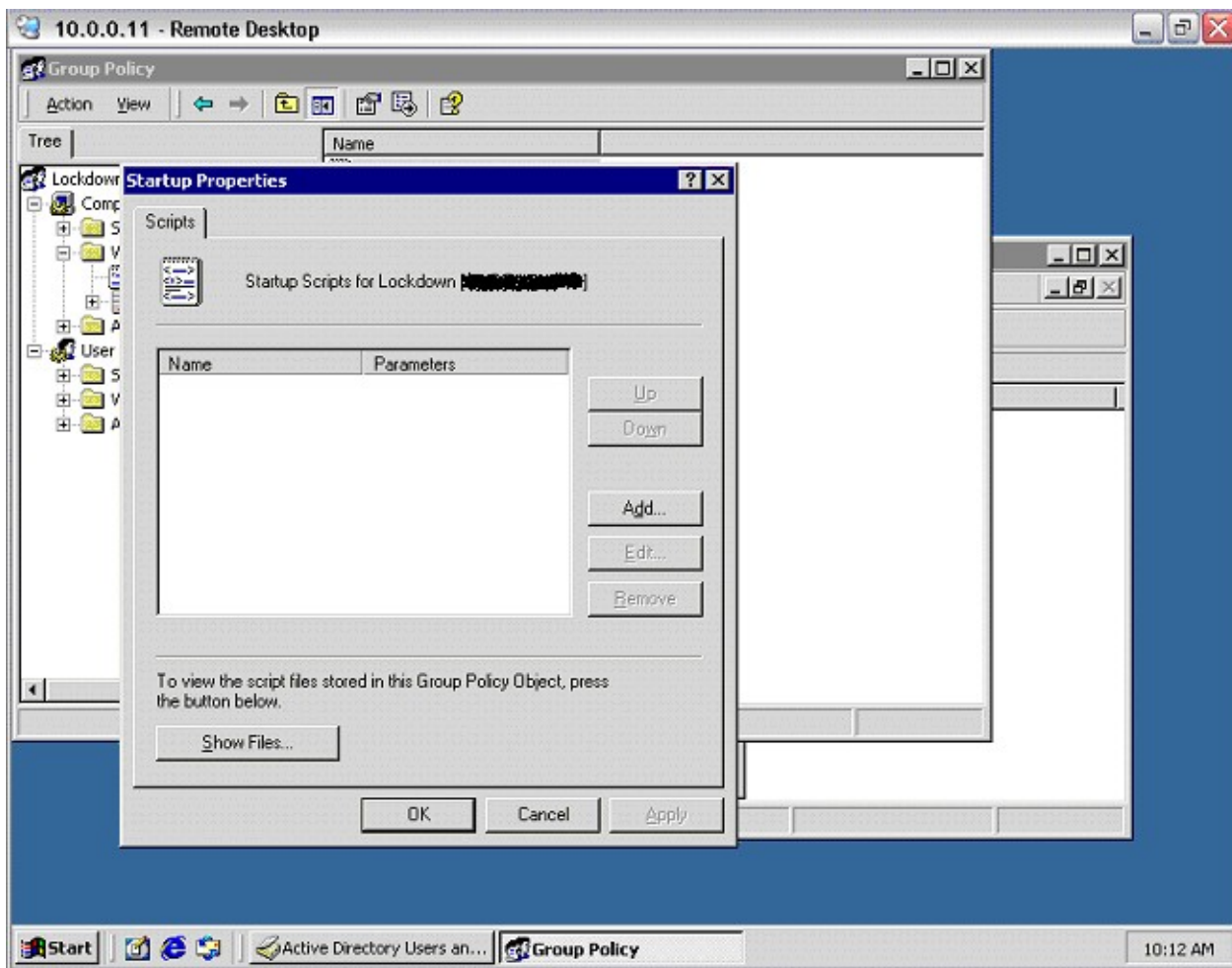
Keep in mind that I have blanked out the domain name, and I have also added Domain Computers as an Organizational Unit (OU). Your display won't show that. Create a new OU called "Bob." Well maybe "Lockdown" would be more appropriate. Now right-click on the "Lockdown" OU, and choose properties and on the screen that pops up, choose the Group Policy tab.



Click on New, which creates a new group policy, then change the name of that policy to "Lockdown." Edit the new policy, which will bring up the following window:



Under "Computer Configuration" click to expand "Windows Settings" and click on Scripts (Startup/Shutdown), then haul off and double-click on Startup. This brings up the following screen:



Add a new script. You don't need to worry about typing anything into script parameters in this case because there are no parameters. So click on the Browse button, and browse on over to the network share you left your script, and select it, and click ok. It should now appear in your "Script Name" field. Exit everything including the Group Policy window, and then add the name of your web server to the "Lockdown" organizational unit. Once finished force replication of your domain controllers (assuming you have more than one), then reboot your web server. When it boots back up you should see something about it running the startup scripts. Log in, and check your logging settings. Voila! They have been automatically set for you! Every time it boots, these scripts will change those settings thereby enforcing the policy. Even if another administrator changes it later, it doesn't matter, policy will change it back upon restart. Keep that in the back of your mind when troubleshooting as well as it's easy to make changes, reboot, and then wonder why your changes didn't hold.

More scripts

That is just one procedure recommended by Microsoft to lock down your server, but if your Windows installation is a default install there is much more to be done. I have [included some scripts](#) that separately do most of the procedures outlined in Microsoft's hardening documents plus a few extras. One of the more curious ones is the following:

```

Const HKEY_CLASSES_ROOT = &H80000000
Const HKEY_CURRENT_USER = &H80000001
Const HKEY_LOCAL_MACHINE = &H80000002
Const HKEY_USERS = &H80000003
Const HKEY_CURRENT_CONFIG = &H80000005
Const HKEY_DYN_DATA = &H80000006

```

```

Const REG_SZ = 1
Const REG_EXPAND_SZ = 2
Const REG_BINARY = 3
Const REG_DWORD = 4
Const REG_MULTI_SZ = 7

Set objRegNT = GetObject( "winmgmts://localhost/root/default:StdRegProv" )
Set WshShell = WScript.CreateObject("WScript.Shell")
Set objRegistry = GetObject("winmgmts:" & Computer & "root\default:StdRegProv")
sNetworkCards = "Software\Microsoft\Windows NT\CurrentVersion\NetworkCards"
RC = objRegistry.EnumKey(HKEY_LOCAL_MACHINE, sNetworkCards, sCardNumbers)

If (RC = 0) And (Err.Number = 0) Then

for q = lbound(sCardNumbers) to ubound(sCardNumbers)
sNetCard = "Software\Microsoft\Windows NT\CurrentVersion\NetworkCards\" & sCardNumbers(q)
RCB = objRegistry.GetStringValue(HKEY_LOCAL_MACHINE, sNetCard, "ServiceName", sService)
If (RCB > 0) then
msgbox "Error Number: " & err.number & vbCrLf & "Description: " & err.description, "Error "

Else

myCardObj = sService

SubKey = "SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\" & myCardObj

' This sets the ports that are allowed to be open. You can totally modify these values. 3389 is
for Terminal Server.

TCPports = Array( "3389", "80" )
RawIPAllowed = Array( "53", "" )

ChangeTCPValues = objRegNT.SetMultiStringValue(HKEY_LOCAL_MACHINE, Subkey,
"TCPAllowedPorts", TCPports)
ChangeProtoValues = objRegNT.SetMultiStringValue(HKEY_LOCAL_MACHINE,
Subkey, "RawIPAllowedProtocols", RawIPAllowed)

End If
next
End if

WshShell.RegWrite "HKLM\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces\Tcpip_
& myCardObj & "\NetbiosOptions", "2", "REG_DWORD"
WshShell.RegWrite "HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
\EnableSecurityFilters", "1", "REG_DWORD"

```

This script basically takes an array of ports (3389, 80, 53), and tells Windows NT, or 2000, to allow traffic only to those ports, but block everything else. Windows 2000 has an IPSecurity editor that accomplishes more granular control over these ports, however that mechanism isn't available on NT, so the above works on both. It

can be thought of as a very basic host-based firewall. Microsoft recommends that servers be placed on a protected DMZ, however taking this step offers yet another level of protection. Be careful though. Once this is applied, the domain relationship between the server and the domain is blocked, because the necessary ports are blocked. You can obviously add or remove ports as necessary to correct that however.

Another interesting script is the following:

```
On Error Resume Next
Set objNetwork = WScript.CreateObject("WScript.Network")
strComputerName = objNetwork.ComputerName

'This part changes the OS user name
strNewUser = "IUSR_ACCT"
strOldUser = ("IUSR_" & strComputerName)

'This part tells IIS what that new username is so it doesn't get confused
Set oComputer = GetObject("WinNT://" & strComputerName)
Set oUser = GetObject("WinNT://" & strComputerName & "/" & strOldUser & ",user")
Set NewUser = oComputer.MoveHere(oUser.ADsPath, strNewUser)
Dim WebServerObj
Set WebServerObj = GetObject("IIS://localhost/W3SVC")
WebServerObj.Put "AnonymousUserName", "IUSR_ACCT"
WebServerObj.Put "AnonymousPasswordSync", TRUE
WebServerObj.SetInfo
```

This script does a couple of different things, but first a little background information is necessary. IIS uses the IUSR_<machinename> account, where <machinename> is the name of your computer. This username is highly guessable if you are able to get the host name of the machine. It also means that if an attacker were challenged for a password, the attacker could simply use that account name (IUSR_<machinename>), and repeatedly enter in bad passwords until the account is locked out. Depending on your lockout policy this could cause a denial of service under certain conditions. To fix this, you need to change the value for the IUSR_<machinename> account on the machine, then change the username in the IISAdmin tool. This is time consuming, especially if you have to set up multiple servers. However this script and group policy takes care of it in one step.

You can add more scripts and other Group Policy settings to lock the server down further. Notice that everything described here is broken up into separate scripts? It is more modular that way. You can choose what procedures are done by adding in a script/policy. I have to give credit to [Russ Cooper and Cullen Johnson](#) for their work with vbscript and IIS. They created a script that locks down IIS very thoroughly. The scripts I have just broke it into pieces, and added some extra OS-specific procedures. Please [check here](#) for those scripts.

Final thoughts

You might ask why would someone expose an Internet facing machine and give it a domain relationship? There are basically two answers to this question. First, you can join it to the domain temporarily so the changes are applied, then disjoin it later, or if you have a separate domain for the DMZ, you can just leave it joined there. In the first scenario, you would just need to be sure that no domain remnants are left on the server, and realize that there is no mechanism there to enforce the policy. The second scenario is more palatable for that reason because each time the server is rebooted, policies are applied. If you are bitterly concerned about

domain relationships, you can simply execute each of those scripts in succession on your local machine. This is not quite as automated but it's certainly effective and far less time consuming than doing it manually. Also, you could create a local policy that enforces these changes by using the MMC, and choosing the Group Policy snap-in. Also, as pointed out earlier, these scripts won't cover all of the recommended procedures. For instance, Microsoft recommends that the data for your website be placed on a separate partition from the system partition. These scripts won't check for/correct that issue. As well, auditing and other functions aren't set by these scripts. These are better accomplished using security templates, and/or group policy for Windows 2000. Finally, you may find that these settings break certain applications, or otherwise cause problems. Never fear though, you can restore the default configuration using the same methods. Notice I didn't say that you could restore the previous settings. These scripts won't do that. Scripts to undo the changes made by the lockdown scripts are also included in the scripts [provided](#) with this article. Also, as with anything else, try these out in a test environment first! So as you can see, locking down a server is just as easy as installing the OS, adding it to the appropriate OU, then rebooting.

[Privacy Statement](#)

Copyright 2006, SecurityFocus