

# CHAPTER 4

## ENUMERATION

Footprint

Scan

**Enumerate**

Penetrate

Escalate

Get interactive

Pillage

Expand influence

Cleanup

Assuming that footprinting and scanning haven't turned up any immediate avenues of conquest, an attacker will next turn to identifying more detailed information about prospective victims, including valid user account names or poorly protected resource shares. Many methods can be used to extract such information from Windows Server 2003, a process we call *enumeration*.

The key difference between previously discussed information-gathering techniques and enumeration is in the level of intrusiveness: Enumeration involves active connections to systems and directed queries. As such, they may (should!) be logged or otherwise noticed. We will show you what to look for and how to block it, if possible.

Much of the information gathered through enumeration may appear harmless at first glance. However, the information that leaks from the following holes can be your undoing, as we will try to illustrate throughout this chapter. In general, once a valid username or share is enumerated, it's usually only a matter of time before the intruder guesses the corresponding password or identifies some weakness associated with the resource-sharing protocol. By closing these easily fixed loopholes, you eliminate the first foothold of the malicious hacker.

Our discussion of Windows Server 2003 enumeration will focus on the following topics:

- ▼ NetBIOS Name Service enumeration
- Microsoft Remote Procedure Call (MSRPC) enumeration
- Server Message Block (SMB) enumeration
- Domain Name System (DNS) enumeration
- Simple Network Management Protocol (SNMP) enumeration
- ▲ Active Directory enumeration

First, let's review the information we've gathered so far to establish how we're going to proceed.

## PRELUDE: REVIEWING SCAN RESULTS

Enumeration techniques are mostly service specific and thus should be targeted using information gathered in Chapter 3 via port scanning. Table 4-1 lists the key services that will be sought out by attackers for enumeration purposes.

We will systematically attack these services in the upcoming sections, revealing information that will make you cringe—all with no authentication required!

### NetBIOS Names vs. IP Addresses

Remember that we can use information from ping sweeps (see Chapter 3) to substitute IP addresses for NetBIOS names of individual machines. IP address and NetBIOS

Port	Service
TCP 53	DNS zone transfer
TCP 135	Microsoft RPC Endpoint Mapper
UDP 137	NetBIOS Name Service (NBNS)
TCP 139	NetBIOS session service (SMB over NetBIOS)
TCP 445	SMB over TCP (Direct Host)
UDP 161	Simple Network Management Protocol (SNMP)
TCP/UDP 389	Lightweight Directory Access Protocol (LDAP)
TCP/UDP 3268	Global Catalog Service

**Table 4-1.** Windows Server 2003 Services Typically Targeted by Enumeration Attacks

names are mostly interchangeable (for example, `\\192.168.202.5` can be equivalent to `\\SERVER_NAME`). For convenience, attackers will often add the appropriate entries to their `%systemroot%\system32\drivers\etc\LMHOSTS` file, appended with the `#PRE` syntax, and then run `nbtstat -R` at a command line to reload the name table cache. They are then free to use the NetBIOS name in future attacks, and it will be mapped transparently to the IP address specified in LMHOSTS.

Beware when establishing sessions using NetBIOS names versus IP addresses. All subsequent commands must be launched against the original target. For example, if you establish a null session (see the next section) with `\\192.168.2.5` and then attempt to extract information via this null session using the NetBIOS name of the same system, you will not get a result. Windows remembers which name you specified, even if you don't!

## Disable and Block These Services!

It goes without saying that one countermeasure for every vulnerability mentioned in this chapter is to disable the services listed in Table 4-1. If you cannot disable them for technical or political reasons, we are going to show you in acute detail how vulnerable you are. We will also illustrate some specific countermeasures to mitigate the risk from running these services. However, if these services are running, especially SMB (over NetBIOS or TCP), you will *always* be exposed to some degree of risk.

Of course, it is also important to block access to these services at external network gateways. These services are mostly designed to exist in an unauthenticated local area network (LAN) environment. If they are available to the Internet, it will only be a matter of time before a compromise results—it's almost guaranteed.

Last but not least, use defense in depth. Also configure host-based defenses to block access to these services. The Internet Connection Firewall (ICF) that ships with Windows Server 2003 is a great host-based mechanism to achieve this.

## NETBIOS NAME SERVICE ENUMERATION

The first thing a remote attacker will try on a well-scouted NT family network is to get a sense of what exists on the wire. Since Windows Server 2003 is still dependent on NetBIOS Name Service (NBNS, UDP 137) by default, we sometimes call these activities “enumerating the NetBIOS wire.” The tools and techniques for peering along the NetBIOS wire are readily available—in fact, most are built into the various NT family operating systems! We will discuss those first and then move into some third-party tools. We save discussion of countermeasures until the end, since fixing all of this is rather simple and can be handled in one fell swoop.



### Enumerating Domains with net view

<i>Popularity:</i>	9
<i>Simplicity:</i>	10
<i>Impact:</i>	2
<i>Risk Rating:</i>	7

The `net view` command is a great example of a built-in enumeration tool. `net view` is an extraordinarily simple command-line utility that will list domains available on the network and then lay bare all machines in a domain. Here’s how to enumerate domains on the network using `net view`:

```
C:\>net view /domain
```

```
Domain
```

```
-----
CORLEONE
BARZINI_DOMAIN
TATAGGLIA_DOMAIN
BRAZZI
```

The command completed successfully.

Supplying an argument to the `/domain` switch will list computers in a particular domain, as shown next:

```
C:\>net view /domain:corleone
```

```
Server Name
```

```
Remark
```

```
-----
\\VITO                Make him an offer he can't refuse
\\MICHAEL             Nothing personal
\\SONNY               Badda bing badda boom
\\FREDO                I'm smart
\\CONNIE              Don't forget the cannoli
```

For the command-line challenged, the Network Neighborhood shows essentially the same information shown in these commands. However, because of the sluggishness of updates to the browse list, we think the command-line tools are snappier and more reliable.



## Dumping the NetBIOS Name Table with nbtstat and nbtscan

<i>Popularity:</i>	8
<i>Simplicity:</i>	9
<i>Impact:</i>	1
<i>Risk Rating:</i>	6

Another great built-in tool is nbtstat, which calls up the NetBIOS Name Table from a remote system. The Name Table contains a great deal of information, as seen in the following example:

```
C:\>nbtstat -A 192.168.202.33
Local Area Connection:
Node IpAddress: [192.168.234.244] Scope Id: []
      NetBIOS Remote Machine Name Table
Name                Type                Status
-----
CAESARS             <00>  UNIQUE             Registered
VEGAS2              <00>  GROUP              Registered
VEGAS2              <1C>  GROUP              Registered
CAESARS             <20>  UNIQUE             Registered
VEGAS2              <1B>  UNIQUE             Registered
VEGAS2              <1E>  GROUP              Registered
VEGAS2              <1D>  UNIQUE             Registered
.._MSBROWSE_.      <01>  GROUP              Registered
MAC Address = 00-01-03-27-93-8F
```

As illustrated, nbtstat extracts the system name (CAESARS), the domain or workgroup it's in (VEGAS2), and the MAC (Media Access Control) address. These entities can be identified by their NetBIOS suffix (the two-digit hexadecimal number to the right of the name), which are listed in Table 4-2.

What's interesting about Windows Server 2003 versus its predecessor Windows 2000 is the lack of information about any logged-on users in the nbtstat output. By default on Windows Server 2003, the Messenger service is disabled (see Chapter 16). As you can see in Table 4-2, logged on users would normally have an entry in the NetBIOS Name Table for the Messenger service (see the row beginning with <username>). Since this service is off by default in Windows Server 2003, the NetBIOS Name Table cannot be used to identify valid account names on the server.

NetBIOS Name	Suffix	Name Type	Service
<computer name>	00	U	Workstation
<computer name>	01	U	Messenger (for messages sent to this computer)
<_MS_BROWSE_>	01	G	Master Browser
<computer name>	03	U	Messenger
<computer name>	06	U	RAS Server
<computer name>	1F	U	NetDDE
<computer name>	20	U	Server
<computer name>	21	U	RAS Client
<computer name>	22	U	MS Exchange Interchange
<computer name>	23	U	MS Exchange Store
<computer name>	24	U	MS Exchange Directory
<computer name>	30	U	Modem Sharing Server
<computer name>	31	U	Modem Sharing Client
<computer name>	43	U	SMS Clients Remote Control
<computer name>	44	U	SMS Remote Control Tool
<computer name>	45	U	SMS Client Remote Chat
<computer name>	46	U	SMS Client Remote Transfer
<computer name>	4C	U	DEC Pathworks TCPIP
<computer name>	52	U	DEC Pathworks TCPIP
<computer name>	87	U	MS Exchange MTA
<computer name>	6A	U	Netmon Agent
<computer name>	BF	U	Netmon Application
<username>	03	U	Messenger Service (for messages sent to this user)
<domain name>	00	G	Domain Name
<domain name>	1B	U	Domain Master Browser
<domain name>	1C	G	Domain Controllers
<domain name>	1D	U	Master Browser
<domain name>	1E	G	Browser Service Elections

**Table 4-2.** NetBIOS Suffixes with Associated Name Types and Services

NetBIOS Name	Suffix	Name Type	Service
<INet~Services>	1C	G	IIS
<IS-computername>	00	U	IIS
<computername>	2B	U	Lotus Notes Server
IRISMULTICAST	2F	G	Lotus Notes
IRISNAMESERVER	33	G	Lotus Notes

**Table 4-2.** NetBIOS Suffixes with Associated Name Types and Services (*continued*)

This output also shows no information on running services. In Windows 2000, a system running IIS would typically show the INet~Services entry in its table. The output was taken from a Windows Server 2003 system running IIS, but this information does not appear. We're unsure what lies at the root of this behavior, but it's a welcome change security-wise, since it provides potential intruders with less information.

The Name Type column in Table 4-2 also has significance, as listed in Table 4-3.



### Scanning NetBIOS Name Tables with nbtscan

<i>Popularity:</i>	9
<i>Simplicity:</i>	10
<i>Impact:</i>	2
<i>Risk Rating:</i>	7

The nbtstat utility has two drawbacks: it is restricted to operating on a single host at a time, and it has rather inscrutable output. Both of those issues are addressed by the free

NetBIOS Name Type	Description
Unique (U)	The name might have only one IP address assigned to it.
Group (G)	A unique name, but it might exist with many IP addresses.
Multihomed (M)	The name is unique but may exist on multiple interfaces of the same computer.

**Table 4-3.** NetBIOS Name Types

tool nbtscan from Alla Bezroutchko. nbtscan will “nbtstat” an entire network with blistering speed and format the output nicely:

```
C:\>nbtscan 192.168.234.0/24
Doing NBT name scan for addresses from 192.168.234.0/24
```

IP address	NetBIOS Name	Server	User	MAC address
192.168.234.31	PRNTRSRV	<server>	PRINT	00-50-da-30-1e-0f
192.168.234.34	LAPTOP	<server>	<unknown>	00-b0-d0-56-bf-d4
192.168.234.43	LUXOR	<server>	<unknown>	00-01-03-24-05-7e
192.168.234.44	LUXOR	<server>	<unknown>	00-02-b3-16-db-2e
192.168.234.46	CAESARS	<server>	<unknown>	00-d0-b7-1f-e8-b0

Note in this output that only the server PRNTRSRV indicates a logged-on user. This is the only Windows 2000 machine listed in the output, highlighting our earlier point that account names will not show up in Windows Server 2003 NetBIOS Name Tables by default. In any case, nbtscan is a great way to flush out hosts running Windows on a network. Try running it against your favorite Class C-sized network, and you’ll see what we mean. You may achieve erratic results running it across the Internet due to the vagaries of NBNS over the Internet.



## Enumerating Windows Domain Controllers

<i>Popularity:</i>	9
<i>Simplicity:</i>	10
<i>Impact:</i>	2
<i>Risk Rating:</i>	7

To dig a little deeper into the Windows Server 2003 network structure, we’ll need to use a tool from the Windows Server 2003 Support Tools. (Install these from the \support\tools directory on the Windows Server 2003 CD-ROM.) In the next example, you’ll see how the tool called nltest identifies the domain controllers (the keepers of Windows Server 2003 network authentication credentials) in a Windows Server 2003 domain:

```
C:\>nltest /dclist:vegas2
Get list of DCs in domain 'vegas2' from '\\CAESARS'.
You don't have access to DsBind to vegas2 (\\CAESARS)
(Trying NetServerEnum).
List of DCs in Domain vegas2
    \\CAESARS (PDC)
The command completed successfully
```

## NetBIOS Network Enumeration Countermeasures

<i>Vendor Bulletin:</i>	NA
<i>Bugtraq ID:</i>	NA
<i>Fixed in SP:</i>	NA
<i>Log Signature:</i>	N

All the preceding techniques operate over the NetBIOS Name Service, UDP 137. (Note that the `n1test` command will also try directory-related services such as LDAP.) The best way to prevent these activities is by blocking access to these ports using a router, firewall, or other network gatekeeper. At the host level, configure IPSec filters (see Chapter 16) or install some other host-based firewall functionality.

If you must allow access to NBNS, the only way to prevent user data from appearing in NetBIOS Name Table dumps is to disable the Alerter and Messenger services on individual hosts. The startup behavior for these services can be configured through the Services Control Panel. As we've noted earlier, these services are disabled by default on Windows Server 2003.

## RPC ENUMERATION

Near and dear to NetBIOS Name Service in the pantheon of Windows services susceptible to enumeration is Microsoft's RPC Endpoint Mapper on TCP port 135. We'll level with you right up front and note that the information gathered via MSRPC is not on par with that gathered from SMB (see the section "SMB Enumeration" later in this chapter), but this service is almost always found on NT family networks and may even be exposed on the Internet for such applications as Exchange.



### RPC Enumeration

<i>Popularity:</i>	7
<i>Simplicity:</i>	8
<i>Impact:</i>	1
<i>Risk Rating:</i>	5

Querying the RPC portmapper services on UNIX machines has traditionally been a time-tested hacking technique. On Windows, the portmapper is called the RPC Endpoint Mapper, and although the output is a lot messier than the UNIX equivalent, the concept is the same. The `epdump` tool queries the RPC Endpoint Mapper and shows RPC service interfaces bound to IP addresses and port numbers (albeit in a very crude form). This tool

has been around for so long that we're not sure of its origins anymore, but it's still effective (we've truncated the following output significantly to highlight key points):

```
C:\>epdump servername
binding is 'ncacn_ip_tcp:servername'
int 12345678-1234-abcd-ef00-0123456789ab v1.0
    binding 0000@ncacn_ip_tcp:192.168.234.43[1025]
    annot 'IPSec Policy agent endpoint'
int 3473dd4d-2e88-4006-9cba-22570909dd10 v5.1
    binding 0000@ncalrpc:[LRPC0000061c.00000001]
    annot 'WinHttp Auto-Proxy Service'
int 1ff70682-0a51-30e8-076d-740be8cee98b v1.0
    binding 0000@ncacn_ip_tcp:192.168.234.43[1026]
    annot ''
```

The key things to note in this output are the `int` items, which specify RPC interfaces, and each subsequent `binding` and `annot` entry. The `binding` specifies the IP address and port number on which the RPC endpoint is listening (for example, `192.168.234.43[1025]`), and the annotation often lists the common name of the endpoint (for example, “IPSec Policy agent endpoint”).

More recent tools for dumping MSRPC endpoints include `rpcdump`. Several versions of `rpcdump.exe` are floating around. Don't be confused by the `rpcdump` from David Litchfield (written circa 1999), which is a tool for querying the UNIX portmapper on TCP 111. The other two versions of `rpcdump` are used to query MSRPC—one from the Resource Kit and another that was written by Todd Sabin and comes as part of his RPC Tools suite. Sabin's `rpcdump` adds the ability to query each registered RPC server for all the interfaces it supports via the `RpcMgmtInqIflds` API call, so it can report more than just the interfaces a server has registered. Sabin's tool is a lot like `epdump`, listing each endpoint in sequence. `Rpcdump` from the Resource Kit categorizes its output into interface types, which can help differentiate local RPC interfaces from network (again, we've severely truncated the output here to highlight relevant information):

```
C:\>rpcdump /s servername
Querying Endpoint Mapper Database...
31 registered endpoints found.

ncacn_np(Connection-oriented named pipes)
  \\SERVERNAME[\PIPE\protected_storage] [12345678]
  IPSec Policy agent endpoint :NOT_PINGED

ncalrpc(Local Rpc)
  [dsrole] [12345678] IPSec Policy agent endpoint
  :NOT_PINGED
```

```
ncacn_ip_tcp(Connection-oriented TCP/IP)
  192.168.234.44[1025] [12345778] :NOT_PINGED
  192.168.234.44[1026] [0a74ef1c] :NOT_PINGED
  192.168.234.44[1026] [378e52b0] :NOT_PINGED
  192.168.234.44[1026] [1ff70682] :NOT_PINGED
  192.168.234.44[1025] [12345678] IPSec Policy agent
endpoint :NOT_PINGED
```

rpcdump completed successfully after 1 seconds

You'll note that none of the information disclosed in the output is overwhelmingly useful to an attacker. Depending on the RPC endpoints available, further manipulation could be possible. Typically, the most useful information in this output is the internal IP address of multihomed systems, as well as virtual IP addresses hosted on the same server, which appear as RPC interface bindings. This data can give potential intruders a better idea of what kind of system they are dealing with, including RPC applications that are running, but that's about it.

## RPC Enumeration Countermeasures

<i>Vendor Bulletin:</i>	<i>NA</i>
<i>Bugtraq ID:</i>	<i>NA</i>
<i>Fixed in SP:</i>	<i>NA</i>
<i>Log Signature:</i>	<i>N</i>

Despite Microsoft's tightening of default services in Windows Server 2003 (see Chapter 16), the RPC Endpoint Mapper is still available by default and is still susceptible to anonymous dumping of RPC endpoints. Thus, the best defense against RPC enumeration is to block access to TCP/UDP 135. This can prove challenging to organizations that publish MSRPC-based applications on the Internet, the primary example being Exchange, which must have TCP 135 accessible for Messaging Application Programming Interface (MAPI) clients. Some workarounds to this situation include using Outlook Web Access (OWA) rather than MAPI or using RPC over HTTP (TCP 593). You could also consider using a firewall or virtual private network (VPN) to preauthenticate access to RPC.

To get more granular control over what named pipes can be accessed by anonymous users, you could remove the EPMAPPER entry from the "Network access: Named pipes that can be accessed anonymously" setting that can be accessed via Security Policy.

Don't forget that the Endpoint Mapper only redirects clients to the appropriate RPC port for an application—remember to lock down access to those ports as well. See the "References and Further Reading" section at the end of this chapter for a link to more information on restricting the dynamic allocation of RPC service endpoints.

## SMB ENUMERATION

Next, we will discuss the most widely enumerated Windows interface, Server Message Block (SMB), which forms the basis for Microsoft's File & Print Sharing services. In our discussion of SMB enumeration, we will demonstrate the *null session*, which is an all-time classic enumeration technique. The null session allows an anonymous attacker to extract a great deal of information about a system—most importantly, account names.



### SMB Enumeration: Null Sessions

<i>Popularity:</i>	8
<i>Simplicity:</i>	10
<i>Impact:</i>	8
<i>Risk Rating:</i>	9

One of the NT family's most serious Achilles' heels has traditionally been its default reliance on the Common Internet File System/Server Message Block (CIFS/SMB; hereafter, just SMB) networking protocols. The SMB specs include APIs that return rich information about a machine via TCP ports 139 and 445, even to unauthenticated users. The first step in accessing these APIs remotely is creating just such an unauthenticated connection to a Windows Server 2003 system by using the so-called "null session" command, assuming TCP port 139 or 445 is shown listening by a previous port scan:

```
C:\>net use \\192.168.202.33\IPC$ "" /u:""
```

The command completed successfully.

This syntax connects to the hidden interprocess communications "share" (IPC\$) at IP address 192.168.202.33 as the built-in anonymous user (/u: " ") with a null (" ") password. If successful, the attacker now has an open channel over which to attempt all the various techniques outlined in the rest of this section to pillage as much information as possible from the target: network information, shares, users, groups, Registry keys, and so on.

Almost all the information-gathering techniques described in this section on host enumeration take advantage of this one out-of-the-box security failing of the NT family. Whether you've heard it called the "Red Button" vulnerability, null session connections, or anonymous logon, it can be the single most devastating network foothold sought by intruders.

We will discuss the various attacks that can be performed over null sessions, followed up with a discussion of countermeasures at the end of this section. The great news is that Windows XP and Windows Server 2003 have finally taken large steps toward making SMB enumeration a thing of the past, as we will see throughout our discussion.

**Enumerating Shares** With a null session established, we can also fall back on good ol' net view to enumerate shares on remote systems:

```
C:\>net view \\vito
```

```
Shared resources at \\192.168.7.45
```

```
VITO
```

```
Share name      Type           Used as      Comment
```

```
-----
NETLOGON       Disk           Logon server share
Test           Disk           Public access
Finance        Disk           Transaction records
Web            Disk           Webroot for acme.com
```

```
The command completed successfully.
```

Three other good share-enumeration tools from the Resource Kit are `rmtshare`, `srvcheck`, and `srvinfo` (using the `-s` switch). `rmtshare` generates output similar to `net view`. `srvcheck` displays shares and authorized users, including hidden shares, but it requires privileged access to the remote system to enumerate users and hidden shares. `srvinfo`'s `-s` parameter lists shares along with a lot of other potentially revealing information.

**Enumerating Trusted Domains** Once a null session is set up to one of the machines in the enumerated domain, the `nltest /server:<server_name> /domain_trusts` syntax can be used to learn about further Windows domains with trust relationships to the first. This information will come in handy when we discuss Local Security Authority (LSA) secrets in Chapter 8.

**Enumerating Users** In the good ol' days of hacking, before Windows Server 2003, NT family machines would cough up account information just about as easily as they revealed shares. Some key changes to the default configuration around null session access in Windows XP and Windows Server 2003 have put a stop to all that.

#### CAUTION

When a Windows Server 2003 system is configured as a domain controller, null session restrictions are relaxed.

For this reason, the following examples were run against a Windows Server 2003 domain controller—this command would be denied against a default stand-alone or member server configuration.

A few Resource Kit tools can provide more information about users via null sessions, such as the `usrstat`, `showgrps`, `local`, and `global` utilities. We typically use the `local` utility to dump the members of the local Administrators group on a target server:

```
C:\>local administrators \\caesars
Administrator
```

```
Enterprise Admins
Domain Admins
backadmin
```

Note that the RID 500 account is always listed first in this output, and that additional administrative accounts (such as backadmin) are listed after groups.

The global tool can be used in the same way to find the members of the Domain Admins:

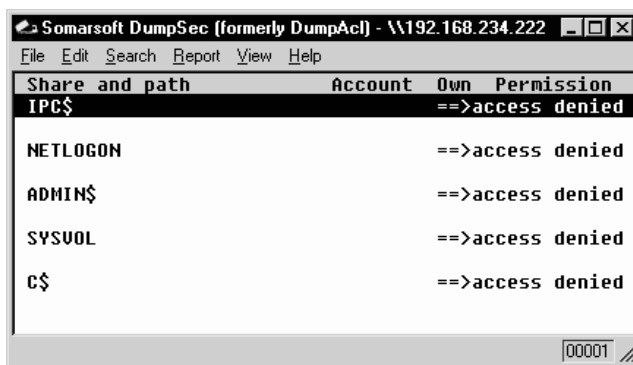
```
C:\>global "domain admins" \\caesars
Administrator
backadmin
```

In the next section, we will discuss some all-in-one enumeration tools that also do a great job of enumerating users, in addition to shares, trusts, and other tantalizing information.

**All-in-One Enumeration Tools** The tools we've shown you so far are all single-purposed. In the following paragraphs, we'll introduce some all-purpose enumeration tools that perform all of the SMB enumeration techniques we've seen so far—and then some!

One of the best tools for enumerating NT family systems is DumpSec (formerly DumpACL) from Somarsoft. Few tools deserve their place in the Windows security administrator's toolbox more than DumpSec—it audits everything from file system permissions to services available on remote systems. DumpSec has an easy-to-use graphical interface, or it can be run from the command line, making for easy automation and scripting.

To use DumpSec anonymously, first set up a null session to a remote system. Then, in DumpSec, choose Report | Select Computer and type in the name of the remote system. (Make sure to use the exact name you used to create the null session, or you will get an error.) Then select whatever report you want to run from the Reports menu. In Figure 4-1, we show



**Figure 4-1.** DumpSec reveals all shares over a null session.

DumpSec being used to dump share information from a remote computer by choosing Report | Dump Permissions For Shares. Note that this displays both hidden and non-hidden shares.

Remember that dumping shares over a null session is still possible by default on Windows Server 2003. DumpSec can also dump user account information, but only if the target system has been configured to permit release of such information over a null session (some might say *mis*-configured). Windows Server 2003 domain controllers will permit this activity by default, so the following examples were run against a Windows Server 2003 domain controller. In this example, we use DumpSec from the command line to generate a file containing user information from the remote computer (remember that DumpSec requires a null session with the target computer to operate):

```
C:\>dumpsec /computer=\\caesars /rpt=usersonly
      /saveas=tsv /outfile=c:\temp\users.txt
C:\>cat c:\temp\users.txt
5/26/2003 3:39 PM - Somarsoft DumpSec (formerly DumpAcl) - \\caesars
UserName          FullName          Comment
Administrator
Built-in account for administering the computer/domain
backadmin         backadmin
Guest
Built-in account for guest access to the computer/domain
IUSR_CAESARS
Internet Guest Account Built-in account for anonymous access to
Internet Information Services
IWAM_CAESARS      Launch IIS Process Account
Built-in account for Internet
Information Services to start out of process applications
krbtgt           Key Distribution Center Service Account
SUPPORT_388945a0  CN=Microsoft Corporation,L=Redmond,S=Washington,C=US
This is a vendor's account for the Help and Support Service
```

Using the DumpSec GUI, many more information fields can be included in the report, but the format shown here usually ferrets out troublemakers. For example, we once came across a server that stored the password for the renamed Administrator account in the FullName field!

DumpSec is also capable of gathering policies, user rights, and services over a null session, but these items are restricted by default on Windows Server 2003.

It took the Razor team from Bindview to throw just about every SMB enumeration feature into one tool, and then some. They called it enum—fittingly enough for this chapter. The following listing of the available command-line switches for this tool demonstrates how comprehensive it is.

```
C:\>enum
usage: enum [switches] [hostname|ip]
      -U: get userlist
```

```
-M: get machine list
-N: get namelist dump (different from -U|-M)
-S: get sharelist
-P: get password policy information
-G: get group and member list
-L: get LSA policy information
-D: dictionary crack, needs -u and -f
-d: be detailed, applies to -U and -S
-c: don't cancel sessions
-u: specify username to use (default "")
-p: specify password to use (default "")
-f: specify dictfile to use (wants -D)
```

enum even automates the setup and teardown of null sessions. Of particular note is the password policy enumeration switch, `-P`, which tells remote attackers whether they can remotely guess user account passwords (using `-D`, `-u`, and `-f`) until they find a weak one. The following example has been edited for brevity to show enum in action against a Windows Server 2003 domain controller:

```
C:\>enum -U -d -P -L -c caesars
server: caesars
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
  netbios: VEGAS2
  domain: VEGAS2
quota:
  paged pool limit: 33554432
  non paged pool limit: 1048576
  min work set size: 65536
  max work set size: 251658240
  pagefile limit: 0
  time limit: 458672
trusted domains:
  indeterminate
netlogon done by a PDC server
```

```

getting user list (pass 1, index 0)... success, got 7.
Administrator (Built-in account for administering the
computer/domain)
  attributes:
  backadmin  attributes: disabled
Guest (Built-in account for guest access to the computer/domain)
  attributes: disabled no_passwd
IUSR_CAESARS
(Built-in account for anonymous access to
Internet Information Services)
  attributes: no_passwd
IWAM_CAESARS
(Built-in account for Internet Information Services to start out
of process applications)
  attributes: no_passwd
krbtgt (Key Distribution Center Service Account)
  attributes: disabled
SUPPORT_388945a0 (This is a vendor's account for the
Help and Support Service)
  attributes: disabled

```

enum will also perform remote password guessing one user at a time using the `-D -u <username> -f <dictfile>` arguments.

Another great enumeration tool written by Sir Dystic, called nete (NetE), will extract a wealth of information from a null session connection. We like to use the `/0` switch to perform all checks, but here's the command syntax for nete to give some idea of the comprehensive information it can retrieve via null session:

```

C:\>nete
NetE v.96  Questions, comments, etc. to sirdystic@cultdeadcow.com

```

```

Usage: NetE [Options] \\MachinenameOrIP

```

```

Options:
/0 - All NULL session operations
/A - All operations
/B - Get PDC name
/C - Connections
/D - Date and time
/E - Exports
/F - Files
/G - Groups
/I - Statistics
/J - Scheduled jobs
/K - Disks

```

```

/L - Local groups
/M - Machines
/N - Message names
/Q - Platform specific info
/P - Printer ports and info
/R - Replicated directories
/S - Sessions
/T - Transports
/U - Users
/V - Services
/W - RAS ports
/X - Uses
/Y - Remote registry trees
/Z - Trusted domains

```

**Bypassing RestrictAnonymous = 1** Before we discuss countermeasures, we thought it appropriate to discuss a security setting that predates Windows Server 2003 to illuminate some other enumeration tools that have not been discussed yet.

Following the release of NT 4 Service Pack 3, Microsoft attempted to defend against the null session enumeration vulnerability by creating the infamous RestrictAnonymous Registry value:

```
HKLM\SYSTEM\CurrentControlSet\Control\LSA\RestrictAnonymous
```

With Windows 2000, Microsoft exposed this setting via the Security Policy MMC snap-in (see Chapter 16), which provided a GUI to the many arcane security-related Registry settings like RestrictAnonymous that needed to be configured manually under NT 4. The setting was called “Additional Restrictions for Anonymous Connections” in Windows 2000 policy, and it introduced a third value called “No Access Without Explicit Anonymous Permissions.” (This is equivalent to setting the RestrictAnonymous Registry value equal to 2; see Table 4-4.) This setting is no longer available in the Windows XP and Windows Server 2003 Security Policy interface, but the Registry value persists.

RestrictAnonymous is a REG\_DWORD and can be set to one of three possible values: 0, 1, or 2. These values are described in Table 4-4.

Value	Security Level
0	None. Rely on default permissions
1	Does not allow enumeration of SAM accounts and names
2	No access without explicit anonymous permissions

**Table 4-4.** RestrictAnonymous Values

Interestingly, setting `RestrictAnonymous` to 1 does not actually block anonymous connections. However, it does prevent most of the information leaks available over the null session, primarily enumeration of user accounts and shares.

Setting `RestrictAnonymous` to 2 prevents the special `Everyone` identity from being included in anonymous access tokens. This setting may cause undesirable connectivity problems for third-party products and/or older Windows platforms. It effectively blocks null sessions from being created:

```
C:\>net use \\mgmgrand\ipc$ "" /u:""  
System error 5 has occurred.
```

Access is denied.

Some enumeration tools and techniques will still extract sensitive data from remote systems, even if `RestrictAnonymous` is set to 1. We'll discuss some of these tools next.

**Bypassing `RestrictAnonymous=1`** Two extremely powerful NT family enumeration tools are `sid2user` and `user2sid` by Evgenii Rudnyi. They are command-line tools that look up NT family SIDs from username input and vice versa. (SIDs are introduced and described in Chapter 2.) To use them remotely requires null session access to the target machine. The following techniques will work even if `RestrictAnonymous = 1`.

First, we extract a domain SID using `user2sid`:

```
C:\>user2sid \\192.168.202.33 "domain users"  
  
S-1-5-21-8915387-1645822062-1819828000-513
```

```
Number of subauthorities is 5  
Domain is WINDOWSNT  
Length of SID in memory is 28 bytes  
Type of SID is SidTypeGroup
```

This tells us the SID for the machine; the string of numbers that begins with *S-1* separated by hyphens in the first line of output above.

As we saw in Chapter 2, the numeric string following the last hyphen is called the *Relative Identifier* (RID), and it is predefined for built-in NT Family users and groups like Administrator or Guest. For example, the Administrator user's RID is always 500, and the Guest user's RID is 501. Armed with this tidbit, a hacker can use `sid2user` and the known SID string appended with an RID of 500 to find the name of the Administrator's account (even if it's been renamed):

```
C:\>sid2user \\192.168.2.33 5 21 8915387 1645822062 18198280005 500  
  
Name is godzilla  
Domain is WINDOWSNT  
Type of SID is SidTypeUser
```

Note that the *S-1* and hyphens are omitted. Another interesting factoid is that the first account created on any NT/2000 local system or domain is assigned an RID of 1000, and each subsequent object gets the next sequential number after that (1001, 1002, 1003, and so on—RIDs are not reused on the current installation). Thus, once the SID is known, a hacker can basically enumerate every user and group on an NT/2000 system, past and present.

Here's a simple example of how to script `user2sid/sid2user` to loop through all of the available user accounts on a system. Before running this script, we first determine the SID for the target system using `user2sid` over a null session, as shown previously. Recalling that NT/2000 assigns new accounts an RID beginning with 1000, we then execute the following loop using the NT/2000 shell command `FOR` and the `sid2user` tool (see earlier) to enumerate up to 50 accounts on a target:

```
C:\>for /L %i IN (1000,1,1050) DO sid2user \\acmepdc1 5 21 1915163094
1258472701648912389 %I >>>> users.txt
C:\>cat users.txt
```

```
Name is IUSR_ACMEPDC1
Domain is ACME
Type of SID is SidTypeUser
```

```
Name is MTS Trusted Impersonators
Domain is ACME
Type of SID is SidTypeAlias
```

```
. . .
```

This raw output could be sanitized by piping it through a filter to leave just a list of usernames. Of course, the scripting environment is not limited to the NT shell—Perl, VBScript, or whatever is handy will do. As one last reminder before we move on, realize that this example will successfully dump users as long as TCP port 139 or 445 is open on the target, `RestrictAnonymous = 1` notwithstanding.

---

**NOTE**

The `UserDump` tool, discussed shortly, automates this “SID walking” enumeration technique.

---

**TIP**

Configure the Security Policy setting “Network Access: Allow Anonymous SID/Name Translation” to Disabled in Windows XP and Server 2003 to prevent this attack.

The `UserInfo` tool from Tim Mullen (Thor@hammerofgod.com) will enumerate user information over a null session even if `RestrictAnonymous` is set to 1. By querying `NetUserGetInfo` API call at Level 3, `UserInfo` accesses the same sensitive information as other tools like `DumpSec` that are stymied by `RestrictAnonymous = 1`. Here's `UserInfo` enumerating the Administrator account on a remote system with `RestrictAnonymous = 1`:



```
Thursday      111111111111111111111111111111111111
Friday       111111111111111111111111111111111111
Saturday     111111111111111111111111111111111111
```

Get hammered at HammerofGod.com!

A related tool from Tim Mullen is UserDump. It enumerates the remote system SID and then “walks” expected RID values to gather all user account names. UserDump takes the name of a known user or group and iterates a user-specified number of times through SIDs 1001 and up. UserDump will always get RID 500 (Administrator) first, and it then begins at RID 1001 plus the maximum number of queries specified. (A MaxQueries setting of 0 or blank returns SID 500 and 1001.) Here’s a sample of UserDump in action:

```
C:\>userdump \\mgmgrand guest 10
```

```
UserDump v1.11 - thor@hammerofgod.com
```

```
Querying Controller \\mgmgrand
```

```
USER INFO
```

```
Username:      Administrator
```

```
Full Name:
```

```
Comment:      Built-in account for
               administering the computer/domain
```

```
User Comment:
```

```
User ID:      500
```

```
Primary Grp:  513
```

```
Privs:       Admin Privs
```

```
OperatorPrivs: No explicit OP Privs
```

```
[snip]
```

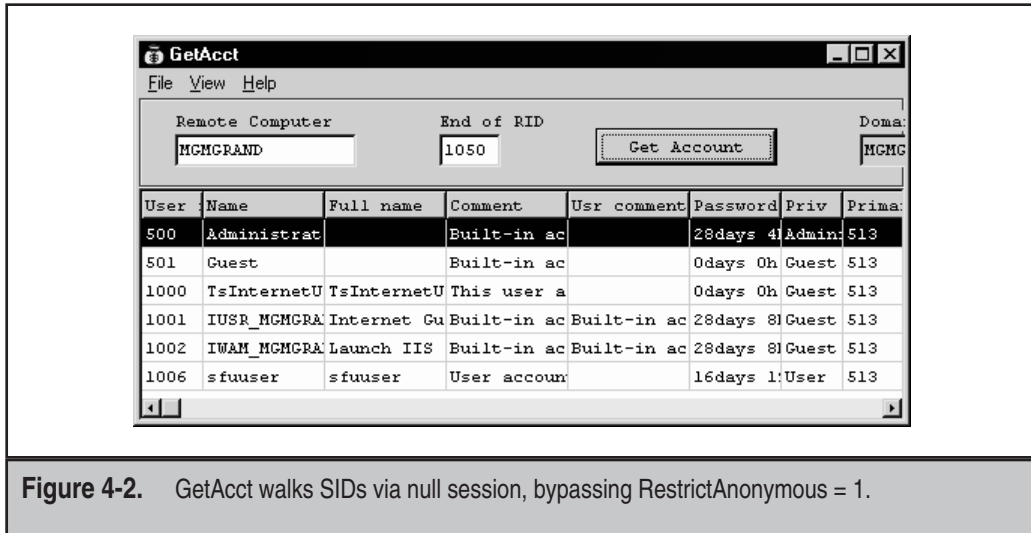
```
LookupAccountSid failed: 1007 does not exist...
```

```
LookupAccountSid failed: 1008 does not exist...
```

```
LookupAccountSid failed: 1009 does not exist...
```

Get hammered at HammerofGod.Com!

Another tool called GetAcct by Uurity performs this same SID walking technique. GetAcct has a graphical interface and can export results to a comma-separated file for later analysis. It does not require the presence of an Administrator or Guest account on the target server. GetAcct is shown in Figure 4-2, obtaining user account information from a system with RestrictAnonymous = 1.



**Figure 4-2.** GetAcct walks SIDs via null session, bypassing RestrictAnonymous = 1.

walksam, one of three RPCTools from Todd Sabin, also walks the Security Accounts Manager (SAM) database and dumps out information about each user found. It supports both the “traditional” method of doing this via Named Pipes and the additional mechanisms that are used by Windows Server 2003 domain controllers. It can bypass RestrictAnonymous = 1 if null sessions are feasible. Here’s an abbreviated example of walksam in action (note that a null session already exists with the target server):

```
C:\rpctools>walksam 192.168.234.44
rid 500: user Administrator
Userid: Administrator
Full Name:
Home Dir:
Home Drive:
Logon Script:
Profile:
Description: Built-in account for administering the computer/domain
Workstations:
Profile:
User Comment:
Last Logon: 7/21/2001 5:39:58.975
Last Logoff: never
```

```

Last Passwd Change: 12/3/2000 5:11:14.655
Acct. Expires: never
Allowed Passwd Change: 12/3/2000 5:11:14.655
Rid: 500
Primary Group Rid: 513
Flags: 0x210
Fields Present: 0xffffffff
Bad Password Count: 0
Num Logons: 88

rid 501: user Guest
Userid: Guest
[etc.]

```

We hope you enjoyed this little stroll down memory lane. Next, we're going to discuss some major improvements to Windows XP and Windows Server 2003 that essentially eliminate the need to worry about RestrictAnonymous.



## SMB Enumeration Countermeasures

<i>Vendor Bulletin:</i>	<i>NA</i>
<i>Bugtraq ID:</i>	<i>NA</i>
<i>Fixed in SP:</i>	<i>NA</i>
<i>Log Signature:</i>	<i>N</i>

Blocking or restricting the damage feasible via Windows Server 2003 SMB enumeration can be accomplished in several ways:

- ▼ Block access to TCP ports 139 and 445 at the network or host level.
- Disable SMB services.
- ▲ Set Network Access settings in Security Policy appropriately.

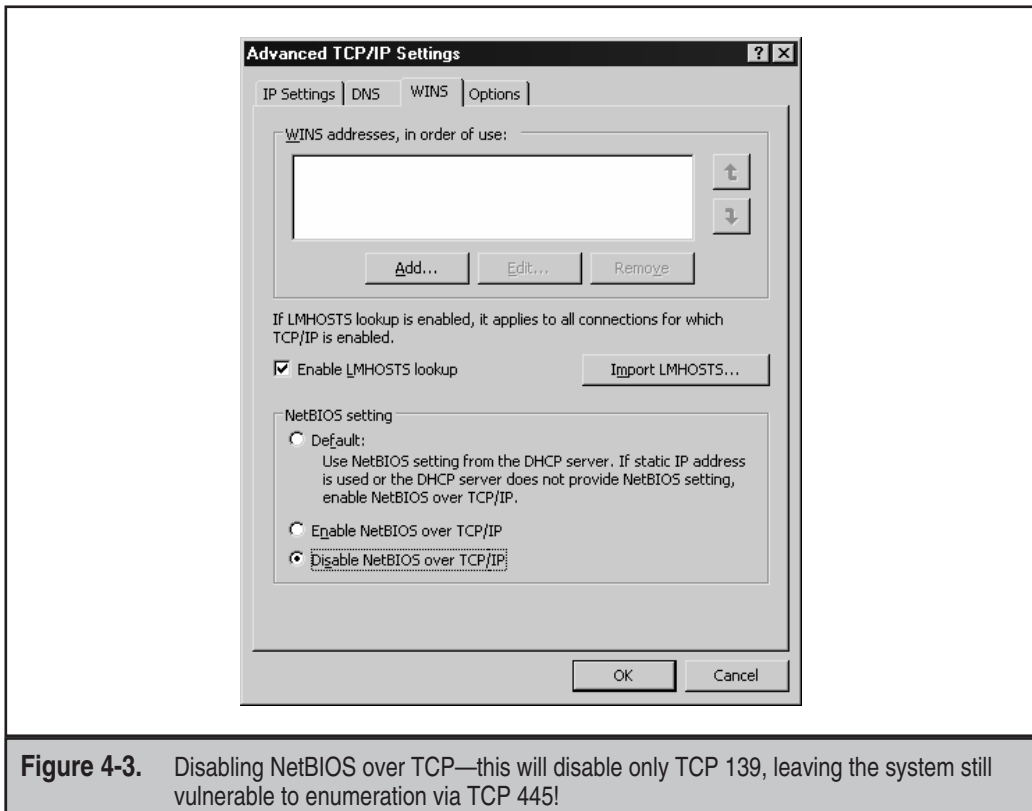
The best way, of course, is to limit untrusted access to these services using a network firewall, which is why we've listed this option first. Also consider the use of filters such as ICF on individual hosts to restrict SMB access (see Chapter 16) and for "defense-in-depth," in case the firewall is penetrated.

Let's discuss the other options in more depth.

**Disabling SMB** Disabling SMB on Windows Server 2003 can actually be quite confusing. First, identify the network connection you want to configure in the Network Connections Control Panel (the connections with “Local Area Connection” in their names are typically the primary LAN connections for the system—you may have to spend some time figuring out which one is plugged into the network on which you want to disable SMB). Right-click the one you want, and select “Properties.” On the Properties sheet, click on “Internet Protocol (TCP/IP),” hit the “Properties” button, and in the ensuing dialog box, click the Advanced button, then navigate to the WINS tab, and locate the setting called “Disable NetBIOS Of TCP/IP,” as shown in Figure 4-3.

Most users assume that by disabling NetBIOS over TCP/IP, they have successfully disabled SMB access to their machines. *This is incorrect.* This setting disables only the NetBIOS Session Service, TCP 139.

In contrast to NT 4, Windows Server 2003 runs another SMB listener on TCP 445. This port will remain active even if NetBIOS over TCP/IP is disabled. Windows SMB client



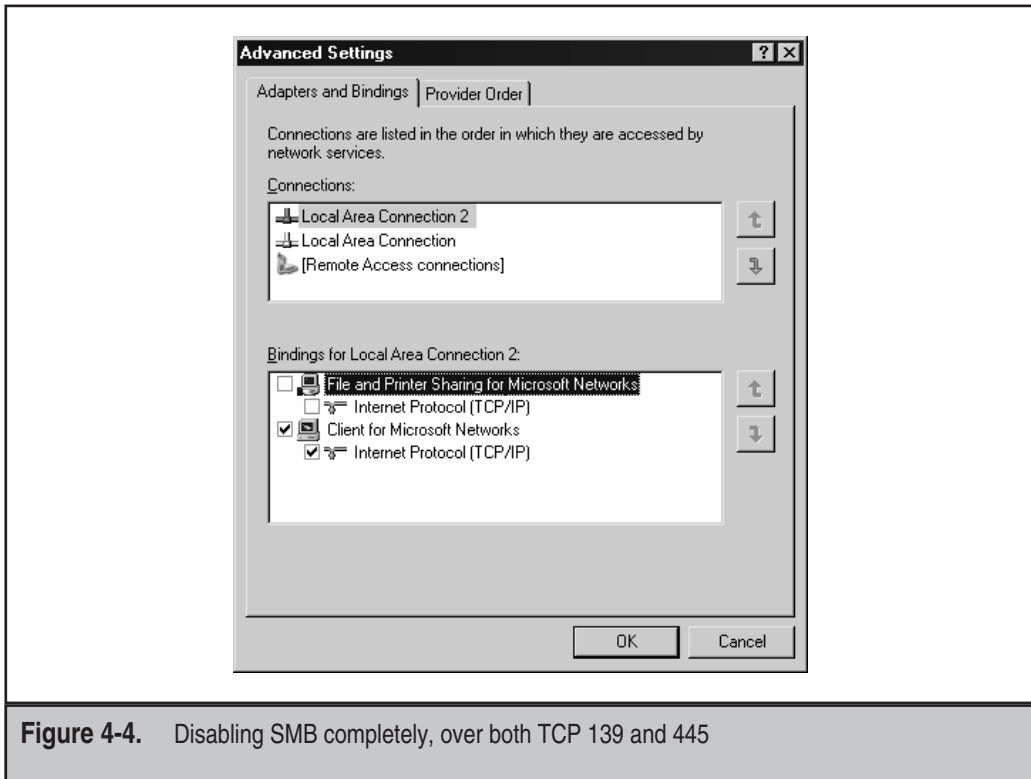
**Figure 4-3.** Disabling NetBIOS over TCP—this will disable only TCP 139, leaving the system still vulnerable to enumeration via TCP 445!

versions later than NT 4 Service Pack 6a will automatically fail over to TCP 445 if a connection to TCP 139 fails, so null sessions can still be established by up-to-date clients even if TCP 139 is disabled or blocked. To disable SMB on TCP 445, open the Network Connections applet in Control Panel, pull down the “Advanced” menu at the top of the window, select “Advanced Settings”; then deselect “File And Printer Sharing For Microsoft Networks” on the appropriate adapter, as shown in Figure 4-4.

With File And Printer Sharing disabled, null sessions will not be possible over 139 and 445 (along with File And Printer Sharing, obviously). No reboot is required for this change to take effect. TCP 139 will still appear in port scans, but no connectivity will be possible.

**TIP**

Another way to prevent access to SMB-based services is to disable the Server service via the Services Administrative Tool (services.msc), which turns off File And Print Sharing, provides access to Named Pipes over the network, and disables the IPC\$ share.



**Figure 4-4.** Disabling SMB completely, over both TCP 139 and 445

**Configuring “Network Access” in Security Policy** If you need to provide access to SMB (say, for a domain controller), disabling SMB is not an option. We also saw that Microsoft’s first cut at fixing the null session problem, RestrictAnonymous, presented users with some extreme options that essentially broke key functionality. For example, setting RestrictAnonymous to its most secure setting (2) has the deleterious effect of preventing down-level client access and trusted domain enumeration. (Windows 95 clients can be updated with the dsclient utility to alleviate some of this; see Microsoft KB article Q246261 for more details.) To address these issues, the interface to control anonymous access has been redesigned in Windows XP and Windows Server 2003 to provide more granularity and better out-of-the-box security.

The most immediate change visible in the Security Policy’s Security Options node is that the option “Additional Restrictions for Anonymous Connections” (which configured RestrictAnonymous Windows 2000) is gone. Under Windows XP and Windows Server 2003, all settings under Security Options have been organized into categories. The settings relevant to restricting anonymous access fall under the category with the prefix “Network Access.” Table 4-5 shows the new settings and our recommended configurations.

Looking at Table 4-5, it’s clear that the main additional advantage gained by Windows XP and Windows Server 2003 is more granular control over resources that are accessible via null sessions. Providing more options is always better, but we still liked the elegant simplicity of Windows 2000’s RestrictAnonymous = 2, because null sessions simply were not possible. Of course, compatibility suffered, but hey, we’re security guys, okay? Simple always beats complex when it comes to security. At any rate, we were unable to penetrate the settings outlined in Table 4-5 using the tools discussed in this chapter.

Even better, the settings in Table 4-5 can be applied at the organizational unit (OU), site, or domain level so they can be inherited by all child objects in Active Directory if applied from a Windows Server 2003 domain controller. This requires the Group Policy snap-in. (See Chapter 16 for more information about Group Policy.)

---

**CAUTION**

By default, Windows Server 2003 domain controllers relax some of the settings that prevent SMB enumeration—see Table 4-5.

---

**NOTE**

Don’t forget to make sure Security Policy is applied, either by right-clicking the Security Settings node in the MMC and selecting Reload or by refreshing Group Policy on a domain.

## WINDOWS DNS ENUMERATION

As we saw in Chapter 3, one of the primary sources of footprinting information is the Domain Name System (DNS), the Internet standard protocol for matching host IP addresses

Windows XP/Server 2003 Setting	Recommended Configuration
<b>Network Access</b> Allow anonymous SID/Name translation	<b>Disabled</b> Blocks user2sid and similar tools (this is enabled on DCs).
<b>Network Access</b> Do not allow anonymous enumeration of SAM accounts	<b>Enabled</b> Blocks tools that bypass RestrictAnonymous = 1.
<b>Network Access</b> Do not allow anonymous enumeration of SAM accounts and shares	<b>Enabled</b> Blocks tools that bypass RestrictAnonymous = 1 (this is disabled on DCs).
<b>Network Access</b> Let Everyone permissions apply to anonymous users	<b>Disabled</b> Although this looks like RestrictAnonymous = 2, null sessions are still possible.
<b>Network Access</b> Named Pipes that can be accessed anonymously	Depends on system role. You may consider removing SQL\QUERY and EPMAPPER to block SQL and MSRPC enumeration, respectively.
<b>Network Access</b> Remotely accessible Registry paths and subpaths	Depends on system role. Most secure is to leave this empty.
<b>Network Access</b> Restrict anonymous access to named pipes and shares	<b>Enabled</b>
<b>Network Access</b> Shares that can be accessed anonymously	Depends on system role. Empty is most secure; the default is COMCFG, DFS\$.

**Table 4-5.** Anonymous Access Settings on Windows XP and Server 2003

with human-friendly names like amazon.com. With the advent of Active Directory (AD) in Windows 2000, which bases its namespace on DNS, Microsoft revamped its DNS server implementation to accommodate the needs of AD and vice versa.

Active Directory relies on the DNS SRV record (RFC 2052), which allows servers to be located by service type (for example, Global Catalog, Kerberos, and LDAP) and protocol (for example, TCP). Thus, a simple zone transfer can enumerate a lot of interesting network information, as shown next.



## Windows 2000 DNS Zone Transfers

<i>Popularity:</i>	5
<i>Simplicity:</i>	9
<i>Impact:</i>	2
<i>Risk Rating:</i>	5

Performing zone transfers is easy using the built-in nslookup tool. In the following example, a zone transfer is executed against the Windows 2000 domain labfarce.org (edited for brevity and line-wrapped for legibility):

```
C:\>nslookup
Default Server: corp-dc.labfarce.org
Address: 192.168.234.110
>> ls -d labfarce.org
[[192.168.234.110]]
labfarce.org. SOA corp-dc.labfarce.org admin.
labfarce.org. A 192.168.234.110
labfarce.org. NS corp-dc.labfarce.org
. . .
_gc._tcp SRV priority=0, weight=100, port=3268, corp-dc.labfarce.org
_kerberos._tcp SRV priority=0, weight=100, port=88, corp-dc.labfarce.org
_kpasswd._tcp SRV priority=0, weight=100, port=464, corp-dc.labfarce.org
_ldap._tcp SRV priority=0, weight=100, port=389, corp-dc.labfarce.org
```

Per RFC 2052, the format for SRV records is

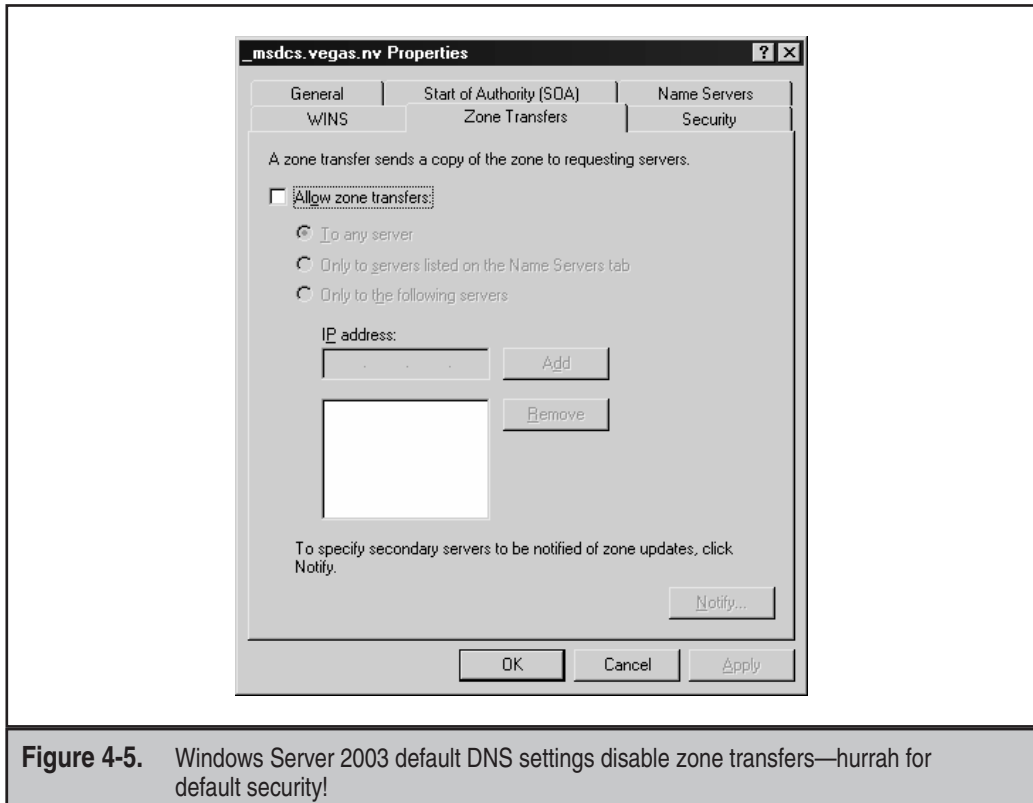
```
Service.Proto.Name TTL Class SRV Priority Weight Port Target
```

Some simple observations an attacker could gather from this file would be the location of the domain's global catalogue service (\_gc.\_tcp), domain controllers using Kerberos authentication (\_kerberos.\_tcp), Lightweight Directory Access Protocol (LDAP) servers (\_ldap.\_tcp), and their associated port numbers (only TCP incarnations are shown here).

## Blocking Windows DNS Zone Transfers

<i>Vendor Bulletin:</i>	NA
<i>Bugtraq ID:</i>	NA
<i>Fixed in SP:</i>	NA
<i>Log Signature:</i>	N

By default—you guessed it—Windows 2000 comes configured to allow zone transfers to any server. Fortunately, Windows Server 2003 restricts zone transfers by default, as shown in Figure 4-5. This screen opens when the Properties option for a forward lookup zone (in this case, labfarce.org) is selected from within the DNS Management console (dnsmgmt.msc).



**Figure 4-5.** Windows Server 2003 default DNS settings disable zone transfers—hurrah for default security!

**NOTE**

Although we recommend the settings shown in Figure 4-5, it is probably more realistic to assume that backup DNS servers will need to be kept up to date on zone file changes, so we'll note that permitting zone transfers to authorized servers is also OK.

Kudos to Microsoft for disabling zone transfers by default in Windows Server 2003!

**TIP**

Although it won't work against Windows's DNS implementation, the following command will determine the version of a server running BIND DNS:

```
nslookup -q=txt -class=CHAOS version.bind.
```

## SNMP ENUMERATION

One of our favorite pen-testing anecdotes concerns the stubborn sysadmin at a client (target) site who insisted that his Windows NT 4 systems couldn't be broken into. "I've locked down SMB, and there's no way you can enumerate user account names on my Windows systems. That'll stop you cold."

Sure enough, access to TCP 139 and 445 was blocked or the SMB service was disabled. However, an earlier port scan showed that something just as juicy was available: the Simple Network Management Protocol (SNMP) agent service, UDP 161. SNMP is not installed by default on the NT family, but it is easily added via Add/Remove Programs in Windows 2000 and later. Many organizations manage their networks with SNMP, so it is commonly found.

In Windows 2000 and earlier, the default installation of SNMP used "public" as the READ community string (the community string is the rough equivalent of a password for the service). Even worse, the information that can be extracted from the Windows SNMP agent is just as damaging as everything we have discussed so far in this chapter. Boy, was this sysadmin disappointed. Read on to see what we did to his machines—to ensure that you don't make the same mistake he did.

**NOTE**

Windows Server 2003 makes significant changes to the default installation of SNMP that prevents all of the following attacks. Unless noted otherwise, the following descriptions apply to Windows 2000 only.



### SNMP Enumeration with `snmputil`

<i>Popularity:</i>	8
<i>Simplicity:</i>	7
<i>Impact:</i>	5
<i>Risk Rating:</i>	7

If an easily guessable read community string has been set on the victim system, enumerating Windows accounts via SNMP is a cakewalk using the Resource Kit `snmputil`

tool. The next example shows `snmputil` reading the LAN Manager Management Information Base (MIB) from a remote Windows 2000 machine using the commonly used read community string “public”:

```
C:\>snmputil walk 192.168.202.33 public .1.3.6.1.4.1.77.1.2.25
Variable = .iso.org.dod.internet.private.enterprises.lanmanager.
          lanmgr-2.server.svUserTable.svUserEntry.svUserName.5.
          71.117.101.115.116
Value    = OCTET STRING - Guest

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.
          lanmgr-2.server.svUserTable.svUserEntry.svUserName.13.
          65.100.109.105.110.105.115.116.114.97.116.111.114
Value    = OCTET STRING - Administrator

End of MIB subtree.
```

The last variable in the preceding `snmputil` syntax, `.1.3.6.1.4.1.77.1.2.25`, is the *object identifier* (OID) that specifies a specific branch of the Microsoft enterprise MIB, as defined in SNMP. The MIB is a hierarchical namespace, so walking “up” the tree (that is, using a less specific number, like `.1.3.6.1.4.1.77`) will dump larger and larger amounts of information. Remembering all those numbers is clunky, so an intruder will use the text string equivalent. Table 4-6 lists some segments of the MIB that yield the juicy stuff.

<b>SNMP MIB (Append This to .iso.org.dod.internet.private.enterprises.lanmanager.lanmgr2)</b>	<b>Enumerated Information</b>
.server.svSvcTable.svSvcEntry.svSvcName	Running services
.server.svShareTable.svShareEntry.svShareName	Share names
.server.svShareTable.svShareEntry.svSharePath	Share paths
.server.svShareTable.svShareEntry.svShareComment	Comments on shares
.server.svUserTable.svUserEntry.svUserName	Usernames
.domain.domPrimaryDomain	Domain name

**Table 4-6.** OIDs from the Microsoft Enterprise SNMP MIB that can be used to Enumerate Sensitive Information

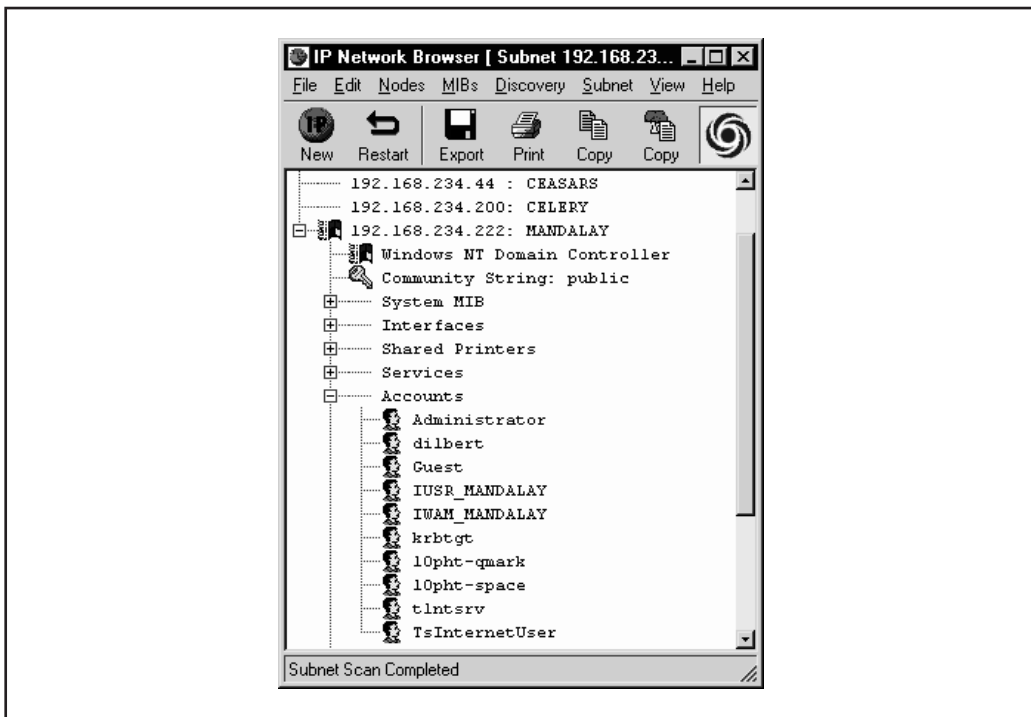


## SNMP Enumeration with SolarWinds Tools

Popularity:	8
Simplicity:	7
Impact:	5
Risk Rating:	7

Of course, to avoid all this typing, you could just download the excellent graphical SNMP browser called IP Network Browser, one of the many great tools included in SolarWinds' Professional Plus Toolset (see "References and Further Reading" for a link). The Professional Plus suite costs \$695, but it's worth it for the numerous tools included in the package.

IP Network Browser enables an attacker to see all this information displayed in living color. Figure 4-6 shows IP Network Browser examining a machine running the Windows 2000 SNMP agent with a default read community string of public.



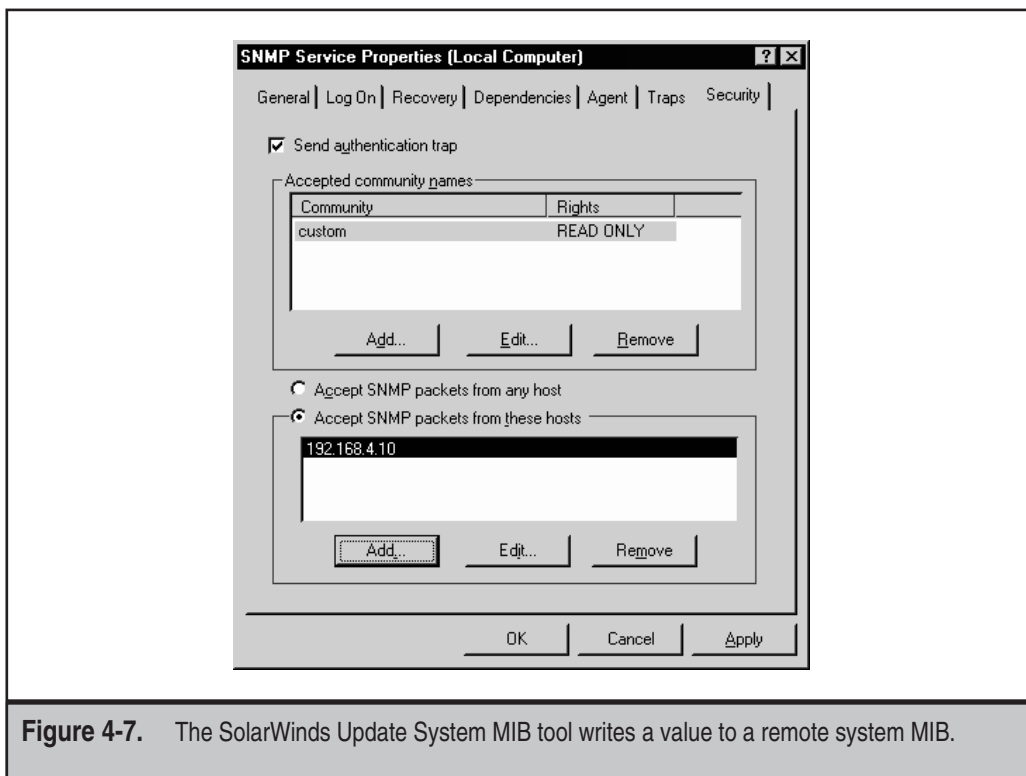
**Figure 4-6.** SolarWinds' IP Network Browser expands information available on systems running the Windows SNMP agent when provided with the correct community string. The community string shown here is Windows 2000's default, "public."

Things get even worse if you identify a write community string via IP Network Browser. Using the Update System MIB tool from the SolarWinds Professional Plus Toolset, you can write values to the System MIB if you supply the proper write string, including system name, location, and contact info. Figure 4-7 shows the Update System MIB tool.

## — SNMP Enumeration Countermeasures

The simplest way to prevent enumeration activity is to remove the SNMP agent or to turn off the SNMP service in the Services Control Panel (services.msc).

If shutting off SNMP is not an option, you should at least ensure that it is properly configured with unique community names (not the default “public” used on Windows 2000) so that it responds only to specific IP addresses. This is a typical configuration in environments that use a single management workstation to poll all devices for SNMP

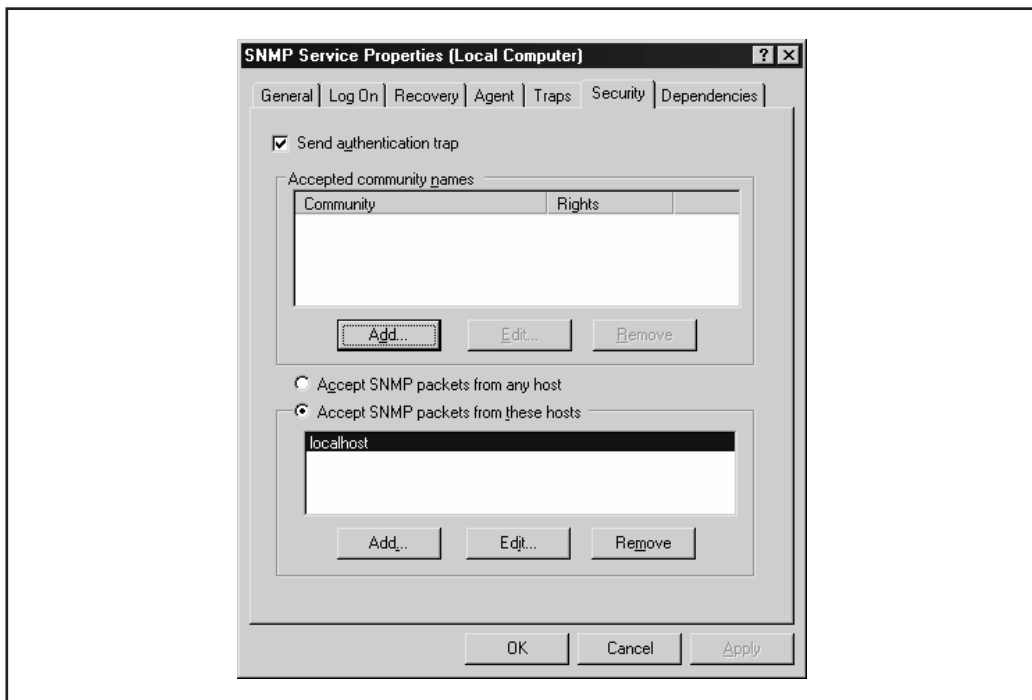


**Figure 4-7.** The SolarWinds Update System MIB tool writes a value to a remote system MIB.

data. To specify these configurations, open the Services Control Panel, select properties of the SNMP Service, click on the Security tab, and change the following values:

Accepted Community Names	Specify unique (nondefault), difficult to guess community strings
Accept SNMP Packets From These Hosts	Specify the IP address of your SNMP management workstation(s)

Figure 4-8 shows these settings in the default Windows Server 2003 SNMP agent configuration. We are happy to report that the default configuration specifies no valid community strings and restricts access to the SNMP agent to the local host only—another shining example of Microsoft’s Trustworthy Computing initiative’s “Secure by Default” mantra. Of course, most administrators will have to make changes to these values to make the SNMP service useful, but at least it’s locked down out-of-the-box.



**Figure 4-8.** The Windows Server 2003 SNMP agent’s default configuration specifies no valid community strings and locks down access to localhost only.

Of course, if you're using SNMP to manage your network, make sure to block access to TCP and UDP ports 161 (SNMP GET/SET) at all perimeter network access devices. As you will see later in this chapter and in others, allowing internal SNMP info to leak onto public networks is a definite no-no.

For more advanced administrators, you can also configure the Windows Server 2003 SNMP service to permit only approved access to the SNMP Community Name and to prevent Windows account information from being sent. To do this, open `regedt32` and go to `HKLM\System\CurrentControlSet\Services\SNMP\Parameters\ValidCommunities`. Choose Security | Permissions, and then set them to permit only approved users access. Next, navigate to `HKLM\System\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents`, delete the value that contains the "LANManagerMIB2Agent" string, and then rename the remaining entries to update the sequence. For example, if the deleted value was 1, then rename 2, 3, and so on, until the sequence begins with 1 and ends with the total number of values in the list.

## ACTIVE DIRECTORY ENUMERATION

The most fundamental change introduced by Windows 2000 was the addition of a Lightweight Directory Access Protocol (LDAP)-based directory service that Microsoft calls Active Directory (AD). AD is designed to contain a unified, logical representation of all the objects relevant to the corporate technology infrastructure, and thus, from an enumeration perspective, it is potentially a prime source of information leakage. Windows Server 2003's AD is largely identical to its predecessor and thus can be accessed by LDAP query tools, as shown in the next example.



### Active Directory Enumeration with ldp

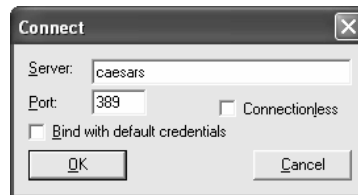
Popularity:	2
Simplicity:	2
Impact:	5
Risk Rating:	3

The Windows Server 2003 Support Tools (available on the Server install CD in the `Support\Tools` folder) includes a simple LDAP client called `ldp.exe` that connects to an AD server and browses the contents of the directory.

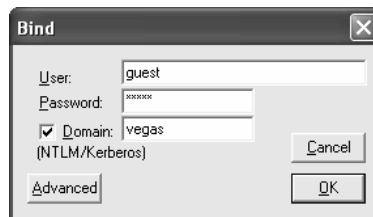
While analyzing the security of Windows 2000 release candidates during the summer of 1999, the authors of this book found that by simply pointing `ldp` at a Windows 2000 domain controller, *all of the existing users and groups could be enumerated with a simple LDAP query*. The only task required to perform this enumeration is to create an authenticated session via LDAP. If an attacker has already compromised an existing account on the target via other means, LDAP can provide an alternative mechanism to enumerate users if SMB ports are blocked or otherwise unavailable.

We illustrate enumeration of users and groups using `ldp` in the following example, which targets the Windows Server 2003 domain controller `caesars.vegas.nv`, whose AD root context is `DC=vegas,DC=nv`. We will assume that we have already compromised the Guest account on `caesars`—it has a password of “`guest`.”

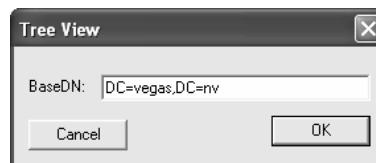
1. Connect to the target using `ldp`. Choose `Connection | Connect`, and enter the IP address or DNS name of the target server. This creates an unauthenticated connection to the directory. You can connect to the default LDAP port 389 or use the AD Global Catalog port 3268 or the UDP versions of either of these services (“connectionless”). TCP port 389 is shown in the following illustration:



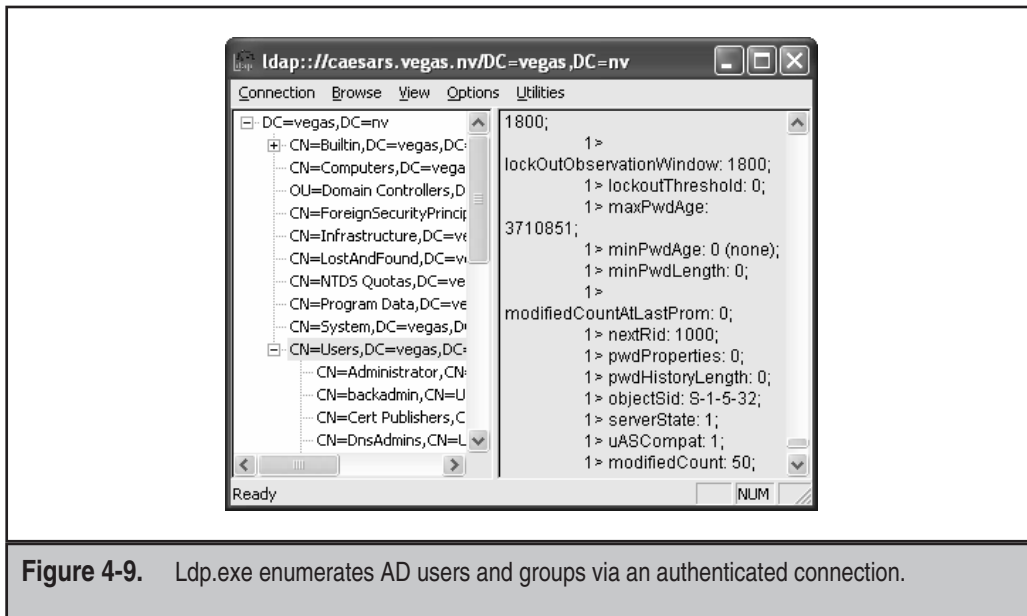
2. The null connection reveals some information about the directory, but we can authenticate as our compromised Guest user and get even more. This is done by choosing `Connections | Bind`, making sure the `Domain` check box is selected with the proper domain name, and entering Guest's credentials, as shown next:



3. You should see output reading “`Authenticated as dn:'guest'.`” Now that an authenticated LDAP session is established, we can actually enumerate Users and Groups. Choose `View | Tree` and enter the root context in the ensuing dialog box. (For example, `DC=vegas,DC=nv` is shown here.)



4. A node appears in the left pane, and we click the plus symbol to unfold it to reveal the base objects under the root of the directory.
5. Finally, we double-click both the `CN=Users` and `CN=Builtin` containers. They will unfold to enumerate all the users and all the built-in groups on the server, respectively. The Users container is displayed in Figure 4-9.



**Figure 4-9.** Ldp.exe enumerates AD users and groups via an authenticated connection.

How is this possible with a simple user connection? Certain legacy NT 4 services, such as Remote Access Service (RAS) and SQL Server, must be able to query user and group objects within AD. The AD installation routine (dcpromo) prompts whether the user wants to relax access permissions on the directory to allow legacy servers to perform these lookups. If the relaxed permissions are selected at installation, user and group objects are accessible to enumeration via LDAP. Note that the default installation will relax the permissions over AD.

## ➤ Active Directory Enumeration Countermeasures

First and foremost, filter access to TCP ports 389 and 3268 at the network border. Unless you plan on exporting AD to the world, no one should have unauthenticated access to the directory.

To prevent this information from leaking out to unauthorized parties on internal semi-trusted networks, permissions on AD will need to be restricted. The difference between legacy-compatible mode (read: “less secure”) and native Windows Server 2003 essentially boils down to the membership of the built-in local group Pre-Windows 2000 Compatible Access. The Pre-Windows 2000 Compatible Access group has the default access permission to the directory shown in Table 4-7.

The Active Directory Installation Wizard automatically adds Everyone and the ANONYMOUS LOGON identity to the Pre-Windows 2000 Compatible Access group if

Object	Permission
Domain password and lockout policies	Read
Other domain parameters	Read
Directory root (and all children)	List contents
User objects	List Contents, Read All Properties, Read Permissions
Group objects	List Contents, Read All Properties, Read Permissions
InetOrgPerson objects	List Contents, Read All Properties, Read Permissions

**Table 4-7.** Permissions on Active Directory Objects Related to the Pre-Windows 2000 Compatible Access Group

you select Pre-Windows Server 2003 Compatible during dcpromo. These special identities include authenticated sessions with *anyone*, including null sessions (see Chapter 2). By removing the Everyone and ANONYMOUS LOGON groups from Pre-Windows 2000 Compatible Access (and then rebooting the domain controllers), the domain operates with the greater security. If you need to downgrade security again for some reason, these groups can be re-added by running the following command at a command prompt:

```
net localgroup "Pre-Windows 2000 Compatible Access" everyone /add
net localgroup "Pre-Windows 2000 Compatible Access" "ANONYMOUS LOGON" /add
```

The access control dictated by membership in the Pre-Windows 2000 Compatible Access group also applies to queries run over NetBIOS null sessions against a domain controller. To illustrate this point, consider the two uses of the enum tool (described previously) in the following example. The first time it is run against a Windows Server 2003 Advanced Server with Everyone and ANONYMOUS LOGON as a member of Pre-Windows 2000 Compatible Access group.

```
C:\>enum -U caesars
server: caesars
setting up session... success.
getting user list (pass 1, index 0)... success, got 8.
Administrator backadmin Guest guest2 IUSR_CAESARS IWAM_CAESARS
krbtgt SUPPORT_388945a0
cleaning up... success.
```

Now we remove Everyone and ANONYMOUS LOGON from the Pre-Windows 2000 Compatible Access group, reboot, and run the same enum query again:

```
C:\>enum -U caesars
server: caesars
setting up session... success.
getting user list (pass 1, index 0)... fail
return 5, Access is denied.
cleaning up... success.
```

**TIP**

Seriously consider upgrading all RAS, Routing and Remote Access Service (RRAS), and SQL Servers in your organization to at least Windows 2000 before the migration to AD so that casual browsing of account information can be blocked.

## SUMMARY

Using the information presented in this chapter, an attacker can now turn to active Windows Server 2003 system penetration, as we describe next in Chapter 5. Here is a short review of the countermeasures presented in this chapter that will restrict malicious hackers from getting at this information:

- ▼ Restrict network access to all of the services discussed in this chapter using network- and host-based firewalls (such as ICF). Disable these services if they are not being used. If you do enable these services, configure them to prevent disclosure of sensitive system information to unauthorized parties according to the following advice.
- Protect the SMB service (TCP/UDP 139 and 445). Disable it if possible by shutting off File And Print Sharing For Microsoft Networks as discussed in this chapter. If you enable SMB, use Security Policy to prevent anonymous access. Windows Server 2003 default settings are sufficient, but beware that the default domain controller settings are relaxed and permit enumeration of accounts. You can push these settings out to all domain computers using Group Policy (see Chapter 16).
- Access to the NetBIOS Name Service (NBNS, UDP 137) should be blocked at network gateways (recognize that blocking UDP 137 will interfere with Windows naming services).
- Disable the Alerter and Messenger services on NetBIOS-aware hosts. This prevents user account information from appearing in remote NetBIOS Name Table dumps. This setting can be propagated throughout a domain using Group Policy (see Chapter 16). These services are disabled by default on Windows Server 2003.

- Configure Windows Server 2003 DNS servers to restrict zone transfers to explicitly defined hosts, or disable zone transfers entirely. Zone transfers are disabled by default in Windows Server 2003.
- If you enable the optional SNMP Service, restrict access to valid SNMP management console machines and specify non-default, hard-to-guess community strings. The Windows Server 2003 SNMP Service restricts access to the local host and specifies no valid community strings by default.
- Heavily restrict access to the AD-specific services, TCP/UDP 389 and 3268. Use network firewalls, Windows Server 2003 ICF or IPSec filters, or any other mechanism available. Note that if you use IPSec filters, set the NoDefaultExempt Registry value to 1 so that the filters cannot be trivially bypassed by source port 88 attacks (see Chapter 16).
- ▲ Remove the Everyone identity from the Pre-Windows 2000 Compatible Access group on Windows Server 2003 domain controllers if possible. This is a backward compatibility mode to allow NT RAS and SQL services to access user objects in the directory. If you don't require this legacy compatibility, turn it off. Plan your migration to Active Directory so that RAS and SQL servers are upgraded first and you do not need to run in backward compatibility mode.

## REFERENCES AND FURTHER READING

References	Link
<b><i>Relevant Microsoft Bulletins, KB Articles, and Hotfixes</i></b>	
Q224196, "Restricting Active Directory Replication Traffic to a Specific Port" covers static allocation of RPC endpoints.	<a href="http://support.microsoft.com/?kbid=224196">http://support.microsoft.com/?kbid=224196</a>
Q143474, "Restricting Information Available to Anonymous Logon Users" covers the RestrictAnonymous Registry key.	<a href="http://support.microsoft.com/?kbid=143474">http://support.microsoft.com/?kbid=143474</a>
Q246261, "How to Use the RestrictAnonymous Registry Value in Windows 2000"	<a href="http://support.microsoft.com/?kbid=246261">http://support.microsoft.com/?kbid=246261</a>
Q240855, "Using Windows NT 4.0 RAS Servers in a Windows 2000 Domain" covers the Pre-Windows 2000 Compatible Access group.	<a href="http://support.microsoft.com/?kbid=240855">http://support.microsoft.com/?kbid=240855</a>
<b><i>Freeware Tools</i></b>	
nbtscan by Alla Bezroutchko	<a href="http://www.inetcat.org/software/nbtscan.html">http://www.inetcat.org/software/nbtscan.html</a>

References	Link
epdump	<a href="http://www.security-solutions.net/download/index.html">http://www.security-solutions.net/download/index.html</a>
rpcdump, part of the RPCTools by Todd Sabin	<a href="http://razor.bindview.com">http://razor.bindview.com</a>
Winfo by Arne Vidstrom	<a href="http://www.ntsecurity.nu">http://www.ntsecurity.nu</a>
nbtDump by David Litchfield	<a href="http://www.atstake.com/research/tools/info_gathering/">http://www.atstake.com/research/tools/info_gathering/</a>
DumpSec by Somarsoft	<a href="http://www.somarsoft.com">http://www.somarsoft.com</a>
enum	<a href="http://razor.bindview.com">http://razor.bindview.com</a>
nete	<a href="http://www.webhackingexposed.com/tools.html">http://www.webhackingexposed.com/tools.html</a>
sid2user/user2sid by Evgenii Rudnyi	<a href="http://www.chem.msu.su:8080/~rudnyi/NT/sid.txt">http://www.chem.msu.su:8080/~rudnyi/NT/sid.txt</a>
UserInfo and UserDump from Thor	<a href="http://www.hammerofgod.com/download.htm">http://www.hammerofgod.com/download.htm</a>
GetAcct by Urity	<a href="http://www.securityfriday.com">http://www.securityfriday.com</a>
walksam, part of the RPCTools by Todd Sabin	<a href="http://razor.bindview.com">http://razor.bindview.com</a>
<b>Commercial Tools</b>	
SolarWinds Professional Plus Edition Toolset	<a href="http://www.solarwinds.net">http://www.solarwinds.net</a>
<b>General References</b>	
“CIFS: Common Insecurities Fail Scrutiny” by Hobbit, the original SMB hacker’s technical reference	<a href="http://www.securityfocus.com/data/library/cifs.txt">http://www.securityfocus.com/data/library/cifs.txt</a>
RFCs 1001 and 1002, which describe the NetBIOS over TCP/UDP transport specifications	<a href="http://www.rfc-editor.org">http://www.rfc-editor.org</a>
RFCs for SNMP	<a href="http://www.rfc-editor.org">http://www.rfc-editor.org</a>