Naming is unusual in SMB/CIFS networks. Although modern clients can use Internet domain names to refer to each other, older clients relied on a Microsoft-specific system known as the Windows Internet Name Service (WINS) server, or the NetBIOS Name Server (NBNS); the two terms are synonymous. Configuring Samba properly for name resolution is therefore important. So is configuring browsing, which is the mechanism by which servers learn what shares are available on specific servers.
Prerequisites

About the elective LPI-302 exam

Linux Professional Institute Certification (LPIC) is like many other certifications in that different levels are offered, with each level requiring more knowledge and experience than the previous one. The LPI-302 exam is an elective specialty exam in the third level of the LPIC hierarchy and requires an advanced level of Linux systems administration knowledge.

To get your LPIC-3 certification, you must pass the two first-level exams (101 and 102), the two second-level exams (201 and 202), and the LPIC-3 core exam (301). After you have achieved this level, you can take the elective specialty exams, such as LPI-302.

This article assumes that you have a working knowledge of Linux command-line functions and that you understand the basics of Samba configuration. You should be familiar with the overall structure of the smb.conf configuration file. You should also have a basic understanding of the Domain Name System (DNS) as a point of comparison for some of the topics covered in this article.

Understanding NetBIOS name resolution

Build your own feed

You can build a custom RSS, Atom, or HTML feed so you will be notified as we add new articles or update content. Go to developerWorks RSS feeds. Select Linux for the zone and Articles for the type, and type Linux Professional Institute for the keywords. Then, choose your preferred feed type.

For the most part, computers' names are likely to be the same regardless of the protocol you're using. In the case of the Server Message Block (SMB)/Common Internet File System (CIFS) protocols, though, some wrinkles exist because of the history of the protocol outside of the usual TCP/IP realm. Specifically, SMB/CIFS originated on networks that used the NetBIOS and NetBIOS Extended User Interface (NetBEUI) protocols, which have their own rules. When SMB/CIFS was modified to work with TCP/IP, some of the old NetBIOS naming conventions had to be adapted for TCP/IP use. Although you don't always have to follow the NetBIOS naming conventions with modern SMB/CIFS clients and servers, sometimes you do, so you should understand the rules, including the different ways NetBIOS resolves names and the role of a WINS server on your network.

Understanding NetBIOS

Figure 1 shows a representation of a pair of network stacks (for client and server) using the Open System Interconnection model. Think of SMB/CIFS as a presentation-level protocol stack in this model. In the original pre-TCP/IP implementation, SMB/CIFS relied on NetBIOS at the session level, which in turn relied on NetBEUI at the transport level.
Figure 1. NetBEUI and NetBIOS are session- and transport-layer protocols in the network protocol stack

In Linux, neither NetBIOS nor NetBEUI exists. Samba takes on the duties of both the SMB/CIFS protocols and the NetBIOS implementation. Instead of using NetBEUI, Samba communicates over TCP and User Datagram Protocol (UDP). The combination of NetBIOS over TCP/IP is sometimes referred to as **NBT**, whose duties include:

- **The name service.** In TCP/IP networking, computers have names used by people, but computers communicate using IP addresses and hardware addresses. In NetBEUI, IP addresses are not used—just names and hardware addresses. NBT therefore handles creating the IP addresses that are not required on SMB/CIFS over NetBIOS and NetBEUI networks. The first half of this article is devoted to this topic.

- **The datagram service.** This is a type of network communication that doesn't create a permanent connection. In TCP/IP, it's roughly equivalent to the UDP, so NBT uses UDP for features that require datagram services, such as the browser elections described in the second half of this article.

- **The session service.** A NetBIOS transaction forms a longer-term connection for most data-transfer requests, such as when a user mounts a file share. This type of service maps logically onto the TCP part of TCP/IP, so NBT uses TCP for such connections.

In practice, NBT isn't mentioned much; we typically speak of **NetBIOS names** or **NetBIOS name resolution** when describing the names used in SMB/CIFS networking and how they're mapped onto IP addresses. NetBIOS names come in two parts:

- **The machine name.** This name, like the machine name in DNS, refers to a specific computer. As a practical matter, it's best if this name is the same as the computer's DNS machine name. The machine name might not be unique in the world, but it should be unique on your network—at least in the computer's workgroup or domain.
• **The workgroup name.** This name applies to all the related computers on a network. In a simple (non-domain) SMB/CIFS configuration, the workgroup name is just a way of organizing computers for purposes of browsing, as described in the second half of this article. In a network that's configured as a domain, the workgroup name is given to the domain, which provides domain-level services, as described in "Learn Linux, 302 (Mixed environments): Domain control."

This two-tier naming system is one of the reasons that NetBIOS networking is local in scope, at least compared to the globe-spanning Internet that's based on TCP/IP. With only two naming levels, it's impractical to create enough names that are unique enough to identify literally billions of computers.

In NetBIOS networking, each computer is configured with its own name, which it announces to the network using a broadcast. This decentralized approach to naming makes setup easy for relatively inexperienced users and administrators—provided they don't try to use a name that's already in use.

NetBIOS names can be up to 15 characters long. They can contain letters, numbers, and the following symbols:

```
! @ # $ % ^ & ( ) - ' { } . ~
```

As a practical matter, it's usually best to restrict names to letters and perhaps numbers. Letters in names are case-insensitive. In this article, I use uppercase letters for NetBIOS names and lowercase letters in DNS names.

Although DOS, Windows, and IBM Operating System/2 (OS/2) all support NetBEUI, they have largely come to use NBT today, even on networks where all the computers can use NetBEUI. The basic principles of NBT operation and NetBIOS name resolution are the same on all of these platforms.

**Types of NetBIOS name resolution**

Suppose a user sits down at a workstation and wants to access the server called **SPEAKER**. The user launches a suitable client application and enters **SPEAKER** as the server's name. What happens? Four different methods of name resolution exist on an NBT network:

- **Broadcast resolution.** The client can send a broadcast message, which all the computers on the local network receive, asking for the IP address associated with the host name **SPEAKER**. That computer should then respond, and the client can initiate the connection for subsequent data transfers. This type of name resolution is sometimes called **B mode** name resolution, and a computer that uses this method exclusively is called a **B-node**.

- **Using a name server.** The client can contact an NBNS server, or WINS server. (The two terms are synonymous.) Like a DNS server, a WINS server maintains a mapping of IP addresses and host names. A WINS server, however, collects these mappings based on the broadcasts computers make when they boot rather than based on a locally maintained
configuration file and consultation with other servers. This method is sometimes referred to as P-mode name resolution.

- **Using lmhosts.** A configuration file called `lmhosts` (typically stored in `/etc/samba` on Samba systems or in `C:\WINDOWS\SYSTEM32\DRIVERS\ETC` on Windows systems) can store mappings of IP addresses to host names. This file is similar to the `/etc/hosts` file, which serves a similar purpose for TCP/IP host names.

- **Using DNS.** Although it’s not a part of NBT, you can configure Samba to use DNS for NetBIOS name resolution. (The same is true for modern versions of Windows.)

Many computers will be configured to use several of these name resolution methods, and in a particular order, to maximize the chance of finding the desired system with a minimum fuss. Old Microsoft Windows 9x or Windows Me systems used B-mode resolution by default, but most newer versions of Windows use DNS as the primary NetBIOS name resolution method.

A WINS server, whether it's a Linux computer running Samba or a Windows server, must listen for name broadcasts, maintain a mapping of IP addresses to NetBIOS names, and answer queries about these mappings from clients. As a practical matter, this means that the WINS server must be available at all times—or as close to that as you can manage. Frequently, a domain controller is an excellent candidate to function as a WINS server. If your computer hosts its own DNS server, that computer might be a good choice, too. You must explicitly configure your chosen computer as a WINS server—a task that’s described in the next section for Samba servers.

### Setting Samba name resolution options

With the needs and methods of NetBIOS name resolution in mind, you can proceed to configuring Samba's name resolution options. These fall into two broad classes: Samba client options and options that control Samba's behavior as a WINS server.

### Setting Samba client options

You may want to adjust a number of smb.conf options that can affect how Samba presents itself to the network or how it resolves NetBIOS names:

- **netbios name.** This option sets the NetBIOS name that the server claims. If you omit this option, it defaults to the machine name portion of your DNS host name, such as `TUNESMITH` if the DNS host name is `tunesmith.example.com`.

- **netbios aliases.** A single computer can claim more than one NetBIOS name. To do so, you use this option, as in `netbios aliases = PRILL HALRLOPRILLALAR`. These names are claimed *in addition to* any set via the `netbios name` option.

- **workgroup.** You set the workgroup or domain name with this option. For browsing and domain features to work correctly, you must set this option to the correct value. The default is `WORKGROUP`.

- **name resolve order.** This parameter takes one or more of four options to control which of the four name resolution methods the programs in the Samba suite use. Although it has little practical effect on the server side, this option does affect how tools such as `smbclient` resolve names.
• **wins server.** You can specify the host names or IP addresses of computers configured to function as WINS servers with this option. Note that specifying this option does not guarantee that these servers will be consulted; the name resolve order option determines that detail.

Ordinarily, the default netbios name option works well, and there's no need for a netbios aliases option; however, you might want to use one or both of these options under certain circumstances. For instance, if you configure a Samba server to take over the duties of other servers that you're retiring, you might want to use netbios aliases so that the Samba server will respond to queries directed at the retired computers' names. You might use netbios name or netbios aliases if you want to use a different name for NetBIOS addressing than you do for other purposes.

To use name resolve order, you specify one or more of four options in a space-separated list:

- **lmhosts.** This option refers to the /etc/samba/lmhosts file, which is described in more detail shortly.
- **host.** This option refers to normal TCP/IP name resolution, which can include a DNS server, an /etc/hosts file, or other methods, depending on your local TCP/IP configuration.
- **wins.** This option refers to name resolution via a WINS server. You specify the server via the wins server option.
- **bcast.** If you include this option, Samba clients will use NetBIOS broadcast name resolution.

The default name resolve order configuration uses the four sources in the order just specified. In many cases, this default setting is fine; however, you might want to tweak it. For instance, if you don't maintain an lmhosts file and want to favor NetBIOS methods over DNS methods of name resolution, you might change it as follows:

```bash
name resolve order = wins bcast host
```

If you want to use a WINS server, you should be sure to include the wins server parameter in your configuration. If you don't, Samba will have no way to locate the WINS server, and the practical effect will be the same as if you had omitted wins from the name resolve order line.

If you want to maintain an lmhosts file, you should understand its file format, which is similar to that of /etc/hosts. Each computer has an entry that occupies a single line beginning with the computer's IP address and ending with its NetBIOS name. Comment lines begin with a hash mark (#). The result might resemble the following:

```plaintext
# Edit this file when adding or removing machines
192.168.7.5  NESSUS
192.168.7.8  PRILL
192.168.7.56 TUNESMITH
```

### Configuring Samba as a WINS server

In addition to Samba options that affect how it resolves NetBIOS names, you can set options that enable a Samba server to function as a WINS server:
WINS replication

Windows WINS servers can use a process known as WINS replication to exchange data between WINS servers on different subnets. Unfortunately, Samba version 3 doesn't support WINS replication, although support for this feature is planned for Samba version 4. For a time, a tool called wrep1 provided support for WINS replication; however, this program never made it out of the experimental stage and has since been abandoned.

As a practical matter, use of options such as \texttt{dns proxy} and \texttt{wins proxy}, along with the fact that most clients now use DNS for most name resolution purposes, makes the lack of WINS replication support unimportant for most installations.

- \textbf{wins support}. Setting this Boolean option to \texttt{Yes} causes it to function as a WINS server. The default value is \texttt{No}.
- \textbf{wins proxy}. This Boolean option determines whether Samba will respond to broadcast requests asking about names other than its own. Setting this option to \texttt{Yes} (the default is \texttt{No}) can improve NetBIOS name resolution reliability if your network spans multiple subnets or if some computers are configured to use broadcasts and others to use a WINS server.
- \textbf{dns proxy}. If this Boolean option is set to \texttt{Yes} (as it is by default) and your Samba WINS server doesn't have a NetBIOS name on file that matches a lookup request, it performs a DNS lookup using the locally configured DNS settings to try to find a name.
- \textbf{max wins ttl}. When working as a WINS server, Samba keeps information on names it has collected for a limited period of time, as determined by the client's request and this parameter. You should specify a time in seconds. The default value is 518400 (6 days).
- \textbf{min wins ttl}. This option works much like \texttt{min wins ttl}, except that it sets a minimum time to store the name data. The default value is 21600 (6 hours).

Ordinarily, setting \texttt{wins server = Yes} is sufficient to configure Samba as a WINS server—at least on the Samba end. There's no automatic method for WINS clients to locate servers, though, so you'll need to configure your clients to tell them about the WINS server. The easiest way to do this, at least for Windows clients, is to adjust your Dynamic Host Configuration Protocol (DHCP) server configuration. The popular named server can be configured through its dhcpd.conf file, which is normally in /etc. You should set two options, which may already be present near the top of the configuration file:

\begin{verbatim}
option netbios-name-servers 192.168.7.2;
option netbios-node-type 8;
\end{verbatim}

The first of these lines identifies the IP address of your Samba WINS server. The second sets the \textit{node type}, which determines which name resolution methods are used. The values range from 1 to 8, but only four are meaningful:

- \textbf{1}. Clients should use broadcasts alone; that is, they function as B-nodes.
- \textbf{2}. Clients should use the WINS server alone; that is, they function as P-nodes.
- \textbf{4}. Clients should use broadcasts first, but, if that fails, use the WINS server. Such a client is sometimes called an \textit{M-node}.
- \textbf{8}. Clients should use the WINS server first, but, if that fails, use broadcasts. Such a client is sometimes called an \textit{H-node}.
As a general rule, it makes sense to configure clients as H-nodes, so passing a NetBIOS node type of 8 via the DHCP server is appropriate.

**Understanding browsing**

In computer circles, the term *browsing* typically applies to the Web: We use web browsers to read websites, as you're doing now. In the context of SMB/CIFS, though, *browsing* means something else (although it's similar): It refers to the process of discovering what resources are available on your network. Samba, naturally, provides configuration options that affect how it interacts with clients as they perform browsing operations. To adjust these settings, though, you must understand some key browsing concepts, such as those of a master browser and of browser elections.

**An overview of browsing**

If you've used Windows' file-sharing features, chances are you've used browsing, which is illustrated by Figure 2. This figure shows the Computer window in Windows. In the left pane, near the bottom, a Network item reveals the computers accessible on the network—HARK, NESSUS, and WEMBLETH, in the case of Figure 2. You can click one of these computers to see the shares it's offering, click the share icons to see the files they hold, and so on.

*Figure 2. Browsing enables users to quickly locate servers and shares*

From the user's point of view, all of this happens automatically; but of course there's considerable effort behind the scenes to make it all work. Browsing relies on one computer, known as a master browser, to compile the lists of computers. When a client browses the network, it contacts the master browser to obtain this list. The master browser is, of course, a server. This server might not need much configuration, but if you're to avoid browsing disasters, such as browse lists that suddenly empty for no obvious reason, you must understand a bit about how browsing works.
Officially, two types of master browsers exist:

- **Local master browsers.** These master browsers serve a single subnet. They listen for the broadcast messages that computers use to announce themselves and claim a NetBIOS name and maintain their browse lists from these broadcasts. They also respond to broadcast queries intended to locate master browsers. Local master browsers are designated by a process known as an *election*, so in principle any computer can become a local master browser. (Recall that SMB/CIFS was designed for use in peer-to-peer configurations, in which all the computers function as both clients and servers.)

- **Domain master browsers.** Because local master browsers compile their server lists by listening for broadcast messages, they can't work across subnets—at least, not without some extra help. That's where domain master browsers come in; they communicate with local master browsers on multiple subnets and exchange browse lists, enabling computers on these different subnets to "see" each other. (This approach does create a delay in the appearance and disappearance of servers, though.) Domain master browsers must be explicitly configured as such; they aren't designated by elections. Normally, domain controllers function as domain master browsers.

Samba provides options, described in [Setting Samba browsing options](#), that enable a local master browser to exchange browse lists with master browsers on another subnet. This can be handy if you have a multi-segment network but don't want to run a domain configuration.

Local master browsers can also have understudies, known as *backup browsers*. Backup browsers maintain browse lists but aren't normally called by clients. The intent is that they can take over master browser duties with current browse lists if the master browser becomes unavailable.

### Master browser elections

Elections begin when a computer calls for one, which typically occurs when the former master browser isn't responding to queries; but some systems (such as Samba) can call for elections whenever they boot up or at other times. Elections typically take a few seconds to finish, during which time browsing won't work, so you shouldn't configure Samba to be too aggressive in demanding elections. The election begins when the computer that called for it broadcasts its election criteria. These criteria can be ranked, as described shortly. Any computer that has a criteria rank that beats the first one broadcasts its criteria in response. A computer that believes it's winning the election will continue to broadcast its criteria every 200-800 milliseconds until it's done so four times, at which point it claims victory and becomes the local master browser.

The criteria used in a master browser election, in order of decreasing priority, are as follows:

1. **Election protocol version.** Multiple election protocols are theoretically possible, but to date, only version 1 has been used. Thus, although this feature could be important, in practice it's not.
2. **Operating system level.** Each operating system has a ranking from 0 to 255 for preference. Higher operating system levels are preferred over lower levels. Most Microsoft operating systems have values between 1 and 32. It's possible to adjust Samba's operating system level, as described next, under [Setting Samba browsing options](#).
3. **Primary domain controller (PDC) status.** If a computer is currently a PDC, it wins over any other computer at the same operating system level and election protocol version.

4. **WINS server status.** A computer that's currently functioning as a WINS server will win an election if other factors haven't already determined the outcome.

5. **Preferred master browser.** A computer can be designated as a *preferred master browser*, which gives it a boost in the election.

6. **Current master browser.** If the computer is currently functioning as a master browser, it receives a boost in the election.

7. **Former master browser.** If the computer used to function as a master browser but lost a recent election and was demoted to backup browser status, it gets a small advantage in the election.

8. **Running backup browser.** If the computer is currently the backup browser, it gets a slight edge in the election.

If an election is called and the results are a tie, two additional tie-breaking criteria may be used:

- **Up time.** If one computer has been running longer than the other, the longer-running machine wins.
- **NetBIOS name.** The computer with the first name, when sorted alphabetically, wins the election if all other criteria result in a tie.

### Setting Samba browsing options

Naturally, Samba provides many options that affect its performance in browser elections and its capacity to function as a domain master browser. I describe options you can use to "rig" an election, those used to enable cross-subnet browsing, and those required to acquire domain master browser status. I also describe options you can set on Samba servers (even those that are not master browsers) to adjust the shares that users see when they browse to your computer.

### Configuring Samba to win an election

Options that can affect Samba's ability to win a local master browser election include the following:

- **local master.** When this option is set to `Yes` (its default), Samba participates in local master browser elections; when set to `No`, it doesn't participate at all and so cannot win an election.
- **os level.** Samba's default `os_level` value is 20, which is sufficient to win elections against most non-server Microsoft operating systems but not against server operating systems, which tend to have operating system levels of 32. If you want to be absolutely positive that a Samba server wins an election, you can set this value to 255, which will win against anything but another similarly configured Samba server. Don't set such an extreme option routinely, though; in a network with multiple Samba servers, one should be "rigged" to win the election.
- **domain logons.** As described in the article on LPIC objective 312.4, this Boolean option configures a Samba server to function as a PDC, and so it also affects the server's chances in a browser election.
- **wins support.** Because WINS server status is a factor in elections, this Boolean option can affect the outcome of elections.
**preferred master.** Setting this Boolean option to `Yes` tells Samba to call an election whenever it starts up. It also sets the preferred master browser flag used in elections. The default value is `No`.

**browse list.** This Boolean parameter, whose default value is `Yes`, determines whether the server maintains a browse list. If it does, it can become a backup or master browser. (If you set this option to `No`, Samba won't participate in elections.)

Most Samba servers work well with the default settings; they will defer to Windows Server operating systems or Samba servers that have been configured to win elections, but they'll win elections in workgroup configurations that don't contain Windows Server operating systems.

If you're experiencing browsing problems, you may want to investigate your Samba server's settings. Problems most often result from either of two conditions:

- **Unreliable local master browsers.** If a computer that functions as a master browser is shut down or disconnected from the network, the result is a disruption in browsing. This might happen in a peer-to-peer network with many computers at the same operating system level if users shut down their computers from time to time. In such a situation, a good solution is to configure a reliable Samba server to win the election by boosting its operating system level and perhaps changing other options.

- **Too many preferred master browsers.** If more than one computer is configured as a preferred master browser, the result can be repeated calls for elections, each of which disrupts browsing for a brief period. Ensure that at most one Samba server per network segment is configured as a preferred master browser. If a Windows computer is a preferred master (as when it's configured as a domain controller), ensure that no Samba server is so configured.

**Enabling cross-subnet browsing in a workgroup**

If your network uses a workgroup configuration but spans multiple subnets, name resolution and browsing might not work properly across the subnets. WINS server options, as described in the first half of this article, can overcome the name resolution issues. (So can using DNS for name resolution.) Browsing, though, may require using some unusual Samba-only options. One possibility is to configure a Samba WINS server as a domain master browser. You can do this in Samba without configuring a full domain, because Samba splits the relevant features into independent options. If you do this, though, you must configure a single Samba server as both the WINS server and the domain master browser; if you don't, the domain master browser won't have information on all the host names it needs to do its job.

If you can't configure one computer as both the WINS server and the domain master browser, or if you find it difficult to coax all your computers to register themselves with the WINS server, a combination of one or more of three additional parameters can help you span multiple subnets:

- **remote browse sync.** You can specify the IP addresses of Samba servers or entire networks to this option, and Samba will exchange browse lists with those Samba servers or with the master browsers it finds on the specified networks. An important limitation of this option is that
it works only with Samba servers; Samba can't exchange browser lists with Windows local master browsers using this option.

- **remote announce.** Samba can announce its presence to a remote network by passing an IP address (including a broadcast address) and an optional workgroup name, as in 192.168.8.255/PAK, where it announces itself as a local master browser for the PAK workgroup on the 192.168.8.0/24 subnet. Unlike remote browse sync, this option works to exchange browse lists with any type of master browser, not just Samba servers.

- **enhanced browsing.** This Boolean option, which defaults to **Yes**, causes Samba to search for and exchange browse lists with domain master browsers it finds via a WINS server or in any other way. This option can make cross-subnet browsing more reliable, but it can also keep defunct workgroups from disappearing from browse lists, so you may want to set it to **No** if you experience that problem.

As a general rule, using Samba local master browsers on each subnet and configuring each to use remote browse sync and point to the other Samba servers works well to exchange browse lists across subnets. If you don't know the exact IP address of the remote local master browser, you can use a broadcast address, as in 192.168.8.255 to refer to the 192.168.8.0/24 network.

**Enabling domain master browser options**

Normally, Samba assumes domain master browser status when you configure it as a domain controller. You can force the issue by setting the domain master option to **Yes**. This parameter accepts Boolean values plus its default of **Auto**, which causes the domain master browser configuration to mirror that of the domain logins parameter, which is described in "Learn Linux, 302 (Mixed environments): Domain control."

Once Samba is configured as a domain controller and domain master browser, domain members should have no problems using the server as a domain master browser. Local master browsers in other subnets can locate the domain master browser by using domain name lookups; such lookups contain codes to identify the domain master browser. You can use the remote browse sync parameter on Samba local master browsers to further guarantee correct operation.

Be aware that browse list propagation across subnets is not instantaneous. Local master browsers need time to collect their browse lists, and they synchronize browse lists with the domain master browser every once in a while. All told, it may be several minutes before all your computers appear in the browse lists on all your subnets.

**Setting local browsing options**

Most of the browsing features described in this article relate to master browsers, which maintain lists of computers. For browsing to do any good, though, individual servers have to make information available on the shares they provide. The default settings normally work well; they make most shares appear in browse lists. The HOMES share is a partial exception; this share definition creates one share for each user of the computer, and that share normally appears in browse lists only for the user to whom the share belongs. You can set several options on Samba file and printer servers, even if they aren't master browsers, to adjust how local share information is shared:
• **preload.** You can specify a list of shares that you want to be available in browse lists even if they wouldn't ordinarily be available. For instance, you might include specific users' home shares, as in `preload = LOUIS` to display that user's home share to all comers. This option is also called *auto services.*

• **load printers.** This Boolean option, which defaults to `Yes`, determines whether printer shares created by the `PRINTERS` definition are made available.

• **browseable.** This Boolean option defaults to `Yes` except for a few special share types. When set to `Yes`, it makes a share visible in browse lists. Setting the value to `No` hides shares but does not make them inaccessible to users who know the share's name.

### Testing NetBIOS and WINS features

Once you've configured a Samba server as a WINS server or master browser, or even if you haven't done so, you may want to test NetBIOS name resolution and browsing features. Samba provides a number of tools that can help you do so. The most notable of these tools are `findsmb` and `smbtree`. (Both of these tools rely on lower-level tools, such as `nmblookup` and `smbclient`, to do the "grunt work."

The `findsmb` program is a tool that sends out a broadcast query for NetBIOS computers on the network and reports the results. It's normally called with no options, although you can pass it a subnet address to query a non-local subnet, and the `-r` option corrects for a bug that's specific to Microsoft Windows 95. The program's results resemble Listing 1.

**Listing 1. Example output from findsmb**

```
$ findsmb

* = DMB
+ = LMB

<table>
<thead>
<tr>
<th>IP ADDR</th>
<th>NETBIOS NAME</th>
<th>WORKGROUP/OS/VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.7.1</td>
<td>CISCOC8231</td>
<td>+[                   ]</td>
</tr>
<tr>
<td>192.168.7.5</td>
<td>NESSUS</td>
<td>*[RINGWORLD] [Unix] [Samba 3.4.12]</td>
</tr>
<tr>
<td>192.168.7.9</td>
<td>HARK</td>
<td>[RINGWORLD] [Unix] [Samba 3.5.7]</td>
</tr>
<tr>
<td>192.168.7.56</td>
<td>TUNESMITH</td>
<td>*[RINGWORLD] [Unix] [Samba 3.0.28a-apple]</td>
</tr>
</tbody>
</table>
```

This output includes not just the names and IP addresses of local computers but information on their workgroups, operating systems, and Samba versions being run.

Unlike `findsmb`, which takes few options, the `smbtree` program takes several. This program is used to retrieve browsing information on your network. In its simplest form, it displays a browse list similar to what you can find in a Windows file browser (as in Figure 2) but in textual form, as shown in Listing 2. Note that `smbtree` asks for your user password; this password is used to gain access to the share information, which many servers won't divulge until after a successful login.
Listing 2. Example output from smbtree

```
$ smbtree
Enter rodsmith's password:
RINGWORLD
\WEMBLETH     wembleth server (Samba, Ubuntu)
  \WEMBLETH\IPC$  IPC Service (wembleth server (Samba, Ubuntu))
  \WEMBLETH\programs  User programs
\SEEKER        seeker server (Samba, Ubuntu)
  \SEEKER\rodsmith   Home Directories
  \SEEKER\IPC$       IPC Service (seeker server (Samba, Ubuntu))
  \SEEKER\MYHTV      Home Directories
\NESSUS        Nessus
  \NESSUS\rodsmith   Home Directories
  \NESSUS\hp4000     HP4000 via Ethernet
  \NESSUS\Epson_RX500 EPSON Stylus Photo RX500
  \NESSUS\IPC$      IPC Service (Nessus)
  \NESSUS\cf        Epson RX500 CF port
  \NESSUS\floppy     Floppy Drive
```

Most of the output from `smbtree` is self-explanatory: It identifies the workgroup or domain (RINGWORLD in this example), the servers (WEMBLETH, SEEKER, and NESSUS in this example), and the browseable shares on each server (along with a description, which can be set via the `comment` share option). The `IPC$` share is used to perform background tasks and is normally hidden from users, but `smbtree` exposes it.

You can add a variety of options to `smbtree` to alter its behavior. Table 1 presents the most useful of these options. Consult the man page for `smbtree` for details on more obscure options.

### Table 1. Options to the smbtree command

<table>
<thead>
<tr>
<th>Option</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-b</code> or <code>--broadcast</code></td>
<td>Makes queries by using broadcasts rather than by using the local master browser to obtain a browse list.</td>
</tr>
<tr>
<td><code>-D</code> or <code>--domains</code></td>
<td>Displays a list of known domains; does not display data on available servers and shares.</td>
</tr>
<tr>
<td><code>-S</code> or <code>--servers</code></td>
<td>Displays a list of domains and servers; does not display data on shares.</td>
</tr>
<tr>
<td><code>-d level</code> or <code>--debuglevel=level</code></td>
<td>Sets the level of detail to be logged to log files; the default is <code>0</code>, meaning little detail. Increase the level to help debug network problems.</td>
</tr>
<tr>
<td><code>-N</code> or <code>--no-pass</code></td>
<td>Suppresses the password prompt, which may also restrict the available information.</td>
</tr>
<tr>
<td><code>-U username</code> or <code>--user=username</code></td>
<td>Sets the user name used to access the share. You can also append a percent sign (%) and password, as in <code>teela%lucky</code> to use <code>teela</code> as the user name and <code>lucky</code> as the password. (Note that <code>lucky</code> is a very poor password; Teela must not worry about such things.)</td>
</tr>
<tr>
<td><code>-S</code> or <code>--servers</code></td>
<td>Displays a list of domains and servers; does not display data on shares.</td>
</tr>
</tbody>
</table>
Related topics

- Review configuring Samba as a domain controller in Rod's article "Learn Linux, 302 (Mixed environments): Domain control" (developerWorks, August 2011).
- Consult network browsing and WINS server configuration options in the Network Browsing chapter of the Samba HOWTO collection.
- Review the entire LPI exam prep series on developerWorks to learn Linux fundamentals and prepare for systems administrator certification based on LPI exam objectives prior to April 2009.
- In the developerWorks Linux zone, find hundreds of how-to articles and tutorials as well as downloads, discussion forums, and a wealth of other resources for Linux developers and administrators.
- Follow developerWorks on Twitter, or subscribe to a feed of Linux tweets on developerWorks.