

[This article was originally written for publication by [the Open Group](#).]

Samba: An Introduction

Chris Hertel

[Samba Team](#)

[jCIFS Team](#)

Last updated: \$Date: 2001/11/27 21:50:29 \$ GMT

The story goes something like this:

Linus Torvalds, the creator of the Linux Kernel, was visiting his friend Andrew Tridgell, the creator of the Samba suite. They were walking through the Zoo in Canberra when, without warning, a huge flock of vampire attack penguins dove out of the sky and tried to carry Linus away. Fortunately, Andrew had an umbrella. Still, one of the birds was able to nip Linus' hand with its fanged beak. Rumor has it that on moonlit nights Linus still runs out into the darkness and jumps, stark naked, into icy water. Of course, he's Finnish and may always have done this. In any case, this is why the Penguin is the Linux Mascot.

Andrew says that the story has changed a bit since the actual event.

What Samba is All About

The commercialization of the Internet over the past few years has created something of a modern melting pot. It has brought business-folk and technologists closer together than was previously thought possible. As a side effect, Windows and Unix systems have been invading each others' turf, and people expect that they will not only play together nicely, but that they will **share**.

A lot of emphasis has been placed on peaceful coexistence between Unix and Windows. The [Usenix Association](#) has even created an annual conference ([LISA/NT-- July 14-17, 1999](#)) around this theme. Unfortunately, the two systems come from very different cultures and they have difficulty getting along without mediation. ...and that, of course, is Samba's job. [Samba](#) runs on Unix platforms, but speaks to Windows clients like a native. It allows a Unix system to move into a Windows "Network Neighborhood" without causing a stir. Windows users can happily access file and print services without knowing or caring that those services are being offered by a Unix host.

All of this is managed through a protocol suite which is currently known as the "Common Internet File System", or CIFS. This name was introduced by [Microsoft](#), and provides some insight into their hopes for the future. At the heart of CIFS is the latest incarnation of the Server Message Block (SMB) protocol, which has a long

and tedious history. Samba is an open source CIFS implementation, and is available for free from the <http://samba.org/> mirror sites.

Samba and Windows are not the only ones to provide CIFS networking. OS/2 supports SMB file and print sharing, and there are commercial CIFS products for Macintosh and other platforms (including several others for Unix). Samba has been ported to a variety of non-Unix operating systems, including VMS, AmigaOS, & NetWare. CIFS is also supported on dedicated file server platforms from a variety of vendors. In other words, this stuff is all over the place.

History - the (hopefully) Untedious Version

It started a long time ago, in the early days of the PC, when IBM and Sytec co-developed a simple networking system designed for building small LANs. The system included something called NetBIOS, or ***Network Basic Input Output System***. NetBIOS was a chunk of software that was loaded into memory to provide an interface between programs and the network hardware. It included an addressing scheme that used 16-byte names to identify workstations and network-enabled applications. Next, Microsoft added features to DOS that allowed disk I/O to be *redirected* to the NetBIOS interface, which made disk space sharable over the LAN. The file-sharing protocol that they used eventually became known as SMB, and now CIFS.

Lots of other software was also written to use the NetBIOS API (***Application Programmer's Interface***), which meant that it would never, ever, ever go away. Instead, the workings beneath the API were cleverly gutted and replaced. NetBEUI (***NetBIOS Enhanced User Interface***), introduced by IBM, provided a mechanism for passing NetBIOS packets over Token Ring and Ethernet. Others developed NetBIOS LAN emulation over higher-level protocols including DECnet, IPX/SPX and, of course, TCP/IP.

NetBIOS and TCP/IP made an interesting team. The latter could be routed between interconnected networks (internetworks), but NetBIOS was designed for isolated LANs. The trick was to map the 16-byte NetBIOS names to IP addresses so that messages could actually find their way through a routed IP network. A mechanism for doing just that was described in the Internet RFC1001 and RFC1002 documents. As Windows evolved, Microsoft added two additional pieces to the SMB package. These were service announcement, which is called "browsing", and a central authentication and authorization service known as Windows NT Domain Control.

Meanwhile, on the Other Side of the Planet...

Andrew Tridgell, who is both tall and Australian, had a bit of a problem. He needed to mount disk space from a Unix server on his DOS PC. Actually, this wasn't the problem at all because he had an NFS (***Network File System***) client for DOS and it worked just fine. Unfortunately, he also had an application that required the NetBIOS interface. Anyone who has ever tried to run multiple protocols under DOS

knows that it can be...er...quirky.

So Andrew chose the obvious solution. He wrote a packet sniffer, reverse engineered the SMB protocol, and implemented it on the Unix box. Thus, he made the Unix system appear to be a PC file server, which allowed him to mount shared filesystems from the Unix server while concurrently running NetBIOS applications. Andrew published his code in early 1992. There was a quick, but short succession of bug-fix releases, and then he put the project aside. Occasionally he would get E'mail about it, but he otherwise ignored it. Then one day, almost two years later, he decided to link his wife's Windows PC with his own Linux system. Lacking any better options, he used his own server code. He was actually surprised when it worked.

Through his E'mail contacts, Andrew discovered that NetBIOS and SMB were actually (though nominally) documented. With this new information at his fingertips he set to work again, but soon ran into another problem. He was contacted by a company claiming trademark on the name that he had chosen for his server software. Rather than cause a fuss, Andrew did a quick scan against a spell-checker dictionary, looking for words containing the letters "smb". "Samba" was in the list. Curiously, that same word is not in the dictionary file that he uses today. (Perhaps they know it's been taken.)

The Samba project has grown mightily since then. Andrew now has a whole team of programmers, scattered around the world, to help with Samba development. When a new release is announced, thousands of copies are downloaded within days. Commercial systems vendors, including Silicon Graphics, bundle Samba with their products. There are even Samba T-shirts available. Perhaps one of the best measures of the success of Samba is that it was listed in the "Halloween Documents", a pair of internal Microsoft memos that were leaked to the Open Source community. These memos list Open Source products which Microsoft considers to be competitive threats. The absolutely best measure of success, though, is that Andrew can still share the printer with his wife.

What Samba Does

Samba consists of two key programs, plus a bunch of other stuff that we'll get to later. The two key programs are `smbd` and `nmbd`. Their job is to implement the four basic modern-day CIFS services, which are:

- File & print services
- Authentication and Authorization
- Name resolution
- Service announcement (browsing)

File and print services are, of course, the cornerstone of the CIFS suite. These are provided by `smbd`, the SMB Daemon. `smbd` also handles "share mode" and "user mode" authentication and authorization. That is, you can protect shared file and

print services by requiring passwords. In share mode, the simplest and least recommended scheme, a password can be assigned to a shared directory or printer (simply called a "share"). This single password is then given to everyone who is allowed to use the share. With user mode authentication, each user has their own username and password and the System Administrator can grant or deny access on an individual basis.

The Windows NT Domain system provides a further level of authentication refinement for CIFS. The basic idea is that a user should only have to log in once to have access to all of the authorized services on the network. The NT Domain system handles this with an authentication server, called a Domain Controller. An NT Domain (which should *not* be confused with a *Domain Name System* (DNS) Domain) is basically a group of machines which share the same Domain Controller.

The NT Domain system deserves special mention because, until the release of Samba version 2, only Microsoft owned code to implement the NT Domain authentication protocols. With version 2, Samba introduced the first non-Microsoft-derived NT Domain authentication code. The eventual goal, of course, it to completely mimic a Windows NT Domain Controller.

The other two CIFS pieces, name resolution and browsing, are handled by nmbd. These two services basically involve the management and distribution of lists of NetBIOS names.

Name resolution takes two forms: broadcast and point-to-point. A machine may use either or both of these methods, depending upon its configuration. Broadcast resolution is the closest to the original NetBIOS mechanism. Basically, a client looking for a service named Trillian will call out "Yo! Trillian! Where are you?", and wait for the machine with that name to answer with an IP address. This can generate a bit of broadcast traffic (a lot of shouting in the streets), but it is restricted to the local LAN so it doesn't cause too much trouble.

The other type of name resolution involves the use of an NBNS (*NetBIOS Name Service*) server. (Microsoft called their NBNS implementation WINS, for *Windows Internet Name Service*, and that acronym is more commonly used today.) The NBNS works something like the wall of an old fashioned telephone booth. (Remember those?) Machines can leave their name and number (IP address) for others to see.

Hi, I'm node Voomba. Call me for a good time! 192.168.100.101

It works like this: The clients send their NetBIOS names & IP addresses to the NBNS server, which keeps the information in a simple database. When a client wants to talk to another client, it sends the other client's name to the NBNS server. If the name is on the list, the NBNS hands back an IP address. You've got the name, look up the number.

Clients on different subnets can all share the same NBNS server so, unlike broadcast, the point-to-point mechanism is not limited to the local LAN. In many

ways the NBNS is similar to the DNS, but the NBNS name list is almost completely dynamic and there are few controls to ensure that only authorized clients can register names. Conflicts can, and do, occur fairly easily.

Finally, there's browsing. This is a whole 'nother kettle of worms, but Samba's nmbd handles it anyway. This is not the web browsing we know and love, but a browsable list of services (file and print shares) offered by the computers on a network.

On a LAN, the participating computers hold an election to decide which of them will become the Local Master Browser (LMB). The "winner" then identifies itself by claiming a special NetBIOS name (in addition to any other names it may have). The LMBs job is to keep a list of available services, and it is this list that appears when you click on the Windows "Network Neighborhood" icon.

In addition to LMBs, there are *Domain* Master Browsers (DMBs). DMBs coordinate browse lists across NT Domains, even on routed networks. Using the NBNS, an LMB will locate its DMB to exchange and combine browse lists. Thus, the browse list is propagated to all hosts in the NT Domain. Unfortunately, the synchronization times are spread apart a bit. It can take more than an hour for a change on a remote subnet to appear in the Network Neighborhood.

Other Stuff

Samba comes with a variety of utilities. The most commonly used are:

smbclient

A simple SMB client, with an interface similar to that of the FTP utility. It can be used from a Unix system to connect to a remote SMB share, transfer files, and send files to remote print shares (printers).

nmblookup

A NetBIOS name service client. Nmblookup can be used to find NetBIOS names on a network, lookup their IP addresses, and query a remote machine for the list of names the machine believes it owns.

swat

The *Samba Web Administration Tool*. Swat allows you to configure Samba remotely, using a web browser.

There are more, of course, but describing them would require explaining even more bits and pieces of CIFS, SMB, and Samba. That's where things really get tedious, so we'll leave it alone for now.

SMB Filesystems for Linux

One of the cool things that you can do with a Windows box is use an SMB file share as if it were a hard disk on your own machine. The N: drive can look, smell, feel, and act like your own disk space, but it's really disk space on some other

computer somewhere else on the network.

Linux systems can do this too, using the `smbfs` filesystem. Built from Samba code, `smbfs` (which stands for **SMB Filesystem**) allows Linux to map a remote SMB share into its directory structure. So, for example, the `/mnt/zarquon` directory might actually be an SMB share, yet you can read, write, edit, delete, and copy the files in that directory just as you would local files.

The `smbfs` is nifty, but it only works with Linux. In fact, it's not even part of the Samba suite. It is distributed with Samba as a courtesy and convenience. A more general solution is the new `smbsh` (**SMB shell**, which is still under development at the time of this writing). This is a cool gadget. It is run like a Unix shell, but it does some funky fiddling with calls to Unix libraries. By intercepting these calls, `smbsh` can make it look as though SMB shares are mounted. All of the read, write, etc. operations are available to the `smbsh` user. Another feature of `smbsh` is that it works on a per-user, per shell basis, while mounting a filesystem is a system-wide operation. This allows for much finer-grained access controls.

Setup and Management

Samba is configured using the `smb.conf` file. This is a simple text file designed to look a lot like those `*.ini` files used in Windows. The goal, of course, is to give network administrators familiar with Windows something comfortable to play with. Over time, though, the number of things that can be configured in Samba has grown, and the percentage of Network Admins willing to edit a Windows `*.ini` file has shrunk. For some people, that makes managing the `smb.conf` file a bit daunting.

Still, learning the ins and outs of `smb.conf` is a worth-while penance. Each of the `smb.conf` variables has a purpose, and a lot of fine tuning can be accomplished. The file structure contents are fully documented, so as to give administrators a running head start, and `smb.conf` can be manipulated using `swat`, which at least makes it nicer to look at.

The Present

Samba 2.0 was released in January 1999. One of the most significant and cool features of the 2.0 release was improved speed. Ziff-Davis Publishing used their Netbench software to benchmark Samba 2.0 on Linux against Windows NT4. They ran all of their tests on the same PC hardware, and their results showed Samba's throughput under load to be at least twice that of NT. Samba is shipped with all major Linux distributions, and Ziff-Davis tested three of those.

Another milestone was reached when Silicon Graphics (SGI) became the first commercial Unix vendor to support Samba. In their December 1998 press release, they claimed that their Origin series servers running Samba 2.0 were the most

powerful line of file servers for Windows clients available. SGI now offers commercial support for Samba as do several other providers, many of which are listed on the Samba web site (see <http://samba.org/>). Traditional Internet support is, of course, still available via the `comp.protocols.smb` newsgroup and the `samba@samba.org` mailing list.

The Samba Team continues to work on new goodies. Current interests include NT ACLs (*Access Control Lists*), support for LDAP (*the Lightweight Directory Access Protocol*), NT Domain Control, and Microsoft's DFS (*Distributed File System*).

The Future

Windows 2000 looms on the horizon like a lazy animal peeking its head over the edge of its burrow while trying to decide whether or not to come out. No one is exactly sure about the kind of animal it will be when it does appear, but folks are fairly certain that it will have teeth.

Because of their dominance on the desktop, Microsoft gets to decide how CIFS will grow. Windows 2000, like previous major operating system releases, will give us a whole new critter to study. Based on the beta copies and the things that Microsoft has said, here are some things to watch for:

CIFS Without NetBIOS

Microsoft will attempt to decouple CIFS and NetBIOS. NetBIOS won't go away, mind you, but it won't be *required* for CIFS networking either. Instead, the SMB protocol will be carried natively over TCP/IP. Name lookups will occur via the DNS.

Dynamic DNS

Microsoft will implement Dynamic DNS, a still-evolving system designed by the IETF (*Internet Engineering Task Force*). Dynamic DNS allows names to be added to a DNS server on-the-fly.

Kerberos V

Microsoft has plans to use Kerberos V. The Microsoft K5 tickets are supposed to contain a [*Privilege Attribute Certificate \(PAC\)*](#), which will include user and group ID information from the Active Directory. Servers will be looking for this PAC when they grant access to the services that they provide. Thus, Kerberos may be used for both authentication and authorization.

Active Directory

The Active Directory appears to be at the heart of Windows 2000 networking. It is likely that legacy NetBIOS services will register their names in the Active Directory.

Hierarchical NT Domains

Instead of isolated Domain Controllers, the NT Domain system will become hierarchical. The naming system will change to one that is remarkably similar to that of the DNS.

One certainty is that W2K (as it is often called) is, and will be, under close

scrutiny. Windows has already attracted the attention of some of the Internet Wonderland's more curious inhabitants, including security analysts, standards groups, crackers dens, and general all-purpose geeks. The business world, which has finally gotten a taste of the freedom of Open Source Software, may be reluctant to return to the world of proprietary, single-vendor solutions. Having the code in your hands is both reassuring and empowering.

Whatever the next Windows animal looks like, it will be Samba's job to help it get along with its peers in the diverse world of the Internet. The Samba Team, a microcosm of the Internet community, are among those watching W2K to see how it develops. Watching does not go hand-in-hand with waiting, though, and Samba is an on-going and open effort. Visit the Samba web site, join the mailing lists, and see what's going on.

Participate in the future.
