Inside SMB

This tutorial will discuss the guts of the SMB protocol. It isn't pretty.

Be prepared for the worst!

Please ask questions at any time!

SMB vs NFS

Samba implements the SMB protocol and a number of associated protocols. It is sometimes useful to compare SMB to NFS

- SMB has scores of top level commands some with dozens of subcommands and subsubcommands
- NFS has around a dozen commands in total
- SMB is connection oriented with a client-maintains-state architecture to handle reconnect
- NFS is stateless
- SMB has sophisticated command chaining, client caching and locking capabilities.
- NFS is well documented and was designed before being

implemented

- SMB is badly documented with the documentation being written retrospectively from the existing implementations

The anatomy of SMB

The SMB protocol can work over many transports, but the most interesting these days is TCP/IP. SMB on TCP/IP uses three ports:

- UDP/137 is used for name resolution and registration
- UDP/138 is used for browsing
- TCP/139 is used for the main file and print sharing transactions

This is a simplification, but it is close, at least until NT5 when everything might change.

NBT on UDP/137

UDP/137 carries name registrations and name queries. When the queries are unicast this protocol if often referred to as WINS (or Windows Internet Name Server). It is a inappropriate name as WINS does not work at all well on large WANs like the Internet.

We will now observe a client starting up and watch the traffic on UDP/137. Note particularly the differences in the startup for WINS and non-WINS.

NBT on UDP/138

UDP/138 carries browsing traffic. This is mostly local broadcast traffic with the occasional message between subnets for cross subnet browsing.

We will take a look at the following situations:

- a browse client boots up
- a browse server restarts

One thing to note about these packets is that they actually have a mixed byte order! It is a rare thing for a internet protocol to mix byte orders. Can you work out how it happened?

SMB on TCP/139

TCP/139 is where the main action happens with the SMB protocol. All file and printer sharing happens on this port. It is a complex protocol but its structure is fairly easy to understand. We will look at a number of situations:

- connecting to a SMB server
- copying a file
- obtaining and breaking oplocks
- printing a file

SMB authentication

SMB authentication is a thorny issue. There are two main authentication models available but unfortunately many people want something that is a combination of the two.

We will examine the authentication process for the two security models:

- user level security - where the client first authenticates then chooses a share to access
- share level security - where the client authenticates as part of accessing a share

Domain logons

The SMB protocol (or more accurately the MS implementations) support two quite different methods for “network logons”, one being used by Windows95 and the other by WindowsNT.

The Windows95 domain logon protocol is fairly simple but is a useful way of allowing users to move between workstations while maintaining a single point of logon and single place where profiles are stored.

Win95 Domain logons

A Windows95 domain logon works by:

- The client doing a broadcast on UDP/138 to ask for a logon server
- The logon server responding with its netbios name
- The client doing a SMB connect to the logon server
- The client asking the logon server for information on its domain logon configuration, such as logon scripts, home directories and profile paths.

WinNT Domain logons

A WindowsNT domain logon works by:

- The client doing a broadcast on UDP/138 to ask for a NT domain server
- The domain server responding with its netbios name
- The client doing a SMB connect to the logon server
- The client establishes an encrypted secure pipe using a shared secret key and a RC4 encryption stream
- The client asks the logon server for information on its domain logon configuration, such as logon scripts, home directories and profile paths.

The NT domain protocol may look a lot more secure, but it actually has some major flaws. The initialisation of the secret key is particularly bad.

Locking and client caching

One area where SMB is particularly rich is in the range of locking capabilities available. They are divided into 3 types of locks:

- byte range locking - for locking records in a file
- share modes - for specifying what other users can do with a file at the same time
- opportunistic locks - not really locking at all, oplocks are a safe client caching mechanism

SMB Extensions in Samba

The CIFS/SMB protocol has been notorious for each implementation doing their own little extensions to add features or work around problems in other implementations. Samba is no exception!

WINS and *1B

Cross-subnet browsing has always been difficult with SMB. It can be made to work with the protocols as they exist in the spec but not without a lot of trouble and only if your network layout is “just right”.

After looking at this we found that the core piece of missing technology is some way for a host to ask the WINS server for a complete list of Domain Master Browsers that it knows about. Given that information it is easy to construct enterprise wide browse lists.

We solved this by extending the Samba WINS server to accept name queries queries for the name *1B. The query will return the current list of known Domain Master Browser IP addresses.

Browse Synchronization

To use the *1B feature we modified our DMB implementation to periodically send *1B queries to the WINS server (if present). If the query is successful then a node status request is sent to any of the resulting IPs which are unknown to the DMB. The node status request is used to determine the workgroup name. Resulting workgroups are scheduled for browse synchronization via the normal mechanisms.

Inter-DMB synchronization

Our *1B browse synchronization solves the cross-subnet browsing issue when all DMBs are using the same WINS server, but does nothing to help the case where some DMBs use a different WINS server or use broadcast registration.

To solve this problem we added inter-DMB synchronization. Each Samba DMB will perform a workgroup browse sync with all currently known DMBs by choosing a random DMB to sync with at regular intervals. The frequency of these sync operations is fixed, regardless of the number of DMBs, thus preventing a N^2 explosion of network traffic.

These sync operations allow a Samba DMB to build a much more complete list of workgroups. The disadvantage is that “dead” workgroups can survive for long after they have no members. We are thinking about a solution to that problem, which is caused by lack of TTL information if the NetServerEnum response.

Remote announce

When the above mechanisms fail, such as in very disjoint networks without a WINS server, it is sometimes useful to tell Samba to send host announcements to a remote subnet where they will be picked up by the local master browser. You can do this using a “remote announce” option.

This option allows you to force the appearance of a server in any workgroup listing you like as long as you know the broadcast address of the remote network. Even over an international link.

Netbios aliases

Samba can announce and register itself as any number of simultaneous netbios names on the network. This can be useful when the server has multiple roles which you may want to split onto separate machines at a later date.

Associated with each name can be a Samba config file, so you can configure each of the names to behave quite differently. For example one may be a user level security file server and another might be a share level security print server.

Encryption conversion

A constant headache for Unix SMB server implementations is the handling of the encrypted SMB password database and its synchronization with the standard unix password systems.

To try to reduce the pain a little Samba can auto-add users to the encrypted database as they login. This allows you to convert from non-encrypted to encrypted with a minimum of fuss.

sync options

Windows programmers don't seem to know the difference between sync() and flush(). We see quite a few programs (Explorer being one of them!) using O_SYNC when opening files in a quite inappropriate manner. This has an enormous performance penalty.

For this reason we give the administrator the option of ignoring sync requests (this can be set on a per share basis).

smbwrapper

For Samba 2.0 we have implemented a library of posix-like filesystem functions that can be preloaded by a unix loader, allowing existing unix binaries to see a virtual SMB filesystem and network neighborhood.

This provides a SMB client filesystem on a wide range of Unix flavors which finally allows unix systems to become true members of a SMB workgroup.

The system (called smbwrapper) also shows printer shares as directories, with printing via cp, print queue listing via ls and print queue deletion via rm.

SSL

Samba 2.0 has SSL support in both the client and server (contributed by Christian Starkjohann). This allows SMB sessions to be established over an underlying secure transport, giving a fully encrypted network filesystem.

This is particularly useful in conjunction with smbwrapper.