# Samba Future Development Plans


## By Jeremy Allison

Samba Team

email: jeremy@netcom.com

# What is the state of Samba Today (1.9.18p4 release) ?

- Samba today can replace Windows NT server for all sites supporting Windows 95 clients, and for some sites using Windows NT workstation.
- Samba provides robust file and print services for SMB clients, rendering PC-NFS unneeded.
- Samba is more scalable than Windows NT servers (leveraging UNIX scalability).
- Samba is faster under load than Windows NT servers.
- Source-code available and under the GNU GPL, Samba can be locally modified for client needs.

# Making UNIX talk to Windows: Issues and Solutions.

- In most cases, Windows clients need to use Microsoft password hashing.
  - This is very insecure compared with UNIX techniques (see http://www.l0pht.com/l0phtcrack/ for details).
  - Requirement for different password hashes means two password databases - with possibility for drift.
  - In my opinion, the refusal of Microsoft to use standard authentication mechanisms is an attempt to force NT servers to be installed (whoever owns the enterprise user database owns the master servers).
- The SMB/CIFS protocol is vulnerable to man-in-the middle and other attacks.

# Making UNIX talk to Windows: Issues and Solutions (continued).

- Having partial security in the protocol is worse than no security (NFS) - it gives a false sense of security.
  - Microsoft have attempted to address this with protocol extensions (packet signing) in NT4.0 SP3. These are not widely understood or used by MS customers.
- Backwards compatibility needs in the protocol (originally 'DOS-on-the-wire') mean an ugly, complex protocol. It is obvious that SMB/CIFS grew, and was not designed.
- Implementing such features as case insensitivity in user space on UNIX is expensive.

# Making UNIX talk to Windows: Issues and Solutions (continued).

- Windows locking semantics are very different from UNIX semantics.
- Locks are mandatory rather than advisory.
- 'DENY MODES' are a foreign concept to UNIX. Samba implements this with shared memory (2 different ways) or files.
  - No interaction between 'DENY MODES' and local unix processes.
- Opportunistic locks - allowing client side caching.
  - Needed for speed (for NT comparison).
  - Has potential for data loss.

# Making UNIX talk to Windows : Issues and Solutions (continued).

- Samba implements 'op-locks' via shared memory.
  - No interaction with native UNIX processes.
  - UNIX kernel level changes needed to allow this.
- Internationalization issues:
  - Non-NT Windows uses DOS 'code pages'.
  - NT uses 16 bit UNICODE natively.
  - Samba implements loadable 'code pages' - used to provide case insensitivity, and also allows filename translation for UNIX compatible storage on disk (ISO-8859-1 and EUC for example).

# Making UNIX talk to Windows: Issues and Solutions (continued).

- Windows NetBIOS naming protocols and the 'browsing concept' are one of the worst parts of the protocol. (Block ports 137/138 at the firewall).
  - Not scalable. Even with WINS servers.
  - Too complex. Very few people understand it.
  - Insecure. Dynamic with no authentication.
- Browsing/NetBIOS naming is so complex it took Microsoft several attempts to get it 'right'. It took the Samba team 3 attempts (over 4 years).
- Samba implements all browsing and NetBIOS naming features except 'backup browsers'.

# Solving the Windows NT Server problem.

- Currently, to fully support Windows NT clients, a Windows NT server is needed.
- This is due to the Windows NT Domain protocols, that NT clients use to communicate with servers.
  - These protocols can be compared to YP/NIS, but are closer to NIS+.
  - They are completely undocumented. MS regards these protocols as their 'leverage' to force customers to buy NT servers if they wish to support NT clients.
- Without UNIX integration into the NT Domain protocols - an integrated user database is not possible.

# Solving the Windows NT server problem (continued).

- Replacing the authentication mechanism on NT clients is not yet possible (although NIS/GINA is a valiant attempt).
  - For details see : http://www.dcs.qmw.ac.uk/~williams/nisgina-current/
- Samba is attempting to solve this by adding the NT Domain controller protocols to the main Samba daemon.
- Current pre-alpha code allows Samba servers to serve NT client logons - more details on this later.

# Windows NT Domain controller protocols.

- Domain protocols are based on DCE/RPC over SMB/CIFS named pipes.
- Although undocumented, the Microsoft Network monitor program will dump the format of many of these packets.
- Another source of information is the Windows NT header files in the MSDN developers kit.
  - Most of the Domain controller protocols are just linearizations of the local calls into DCE/RPC calls.
  - Few of the calls are encrypted over the wire. Microsoft will 'fix' this for NT 5.0.

# Windows NT Domain controller protocols (continued).

- Domain protocols depend on a 'machine' account with password to create a shared secret between client and domain controller.
  - This is similar to Kerberos and other 'realm' based security mechanisms.
  - Adding a machine to the domain is insecure as it starts with a fixed (known) password.
- Certain parts of the client to domain controller protocols are encrypted with a session key, generated using this shared secret.
- The main function of these protocols is to create a distributed account database.

# Windows NT Domain controller protocols (continued).

- The NT distributed account database is one of the well designed areas of NT. It is difficult to emulate this under UNIX.
- NT accounts (security ID, or SID) consist of an arbitrary length 'Domain' component (different for each Domain), followed by a 32 bit 'relative' ID within the Domain (the RID).
  - There are also several 'well known' SIDs that are domain ID independent.
  - Group IDs are also created in this space.
- Samba needs to map these large numbers into the smaller UNIX uid_t and gid_t space.

# Domain User ID mapping.

- There are two scenarios Samba must work within to map user ids between NT and UNIX:
  - Samba domain controller, Windows NT domain members
  - Windows NT domain controller, Samba domain members.
- Having a Samba domain controller is the easiest in terms of account mapping, as the NT domain members will just believe the RIDs given to them by the Samba PDC.
- This is the most difficult to code, however, as it means implementing all the PDC function calls.

# Domain User ID mapping (continued).

- Making Samba a member of a Windows NT hosted domain is easier, in terms of the code needed, but gives the following problems:
  - Accounts created on the NT PDC may not exist on the UNIX box. Currently, Samba ensures that a given user exists in the UNIX password database before allowing access as a non-guest user.
  - A solution to the above would be to use NT RIDs directly as UNIX uid_t and gid_ts. I am against this solution (Samba has always been subordinate to the normal UNIX security policies).
  - A better solution would be for UNIX admins in such a site to create non-login UNIX users to match the NT users.

# Domain User ID mapping (continued).

- Current plan is to use 4 bits of RID to specify account type (user, group, machine etc.). Use 28 bits minus the space for the well known RIDs - (2^28 - 0x1000) to map the uid_t and gid_ts into the RID space.
- Expand the use of the 'username map' parameter in Samba, add a 'groupname map' to allow arbitrary mapping of the users and groups between the systems (eg. map 'Administrators' to group 'wheel' etc.).
- This only gives trouble on UNIXs with 32 bit uid_ts where high numbers have been used.

# Domain User ID mapping (continued).

- One remaining problem is differentiating between 'local' accounts and 'domain' accounts. On UNIX this has traditionally been ignored (depends on the placement of the '+:::' line in the /etc/passwd file).
- All solutions will have some problems - the challenge is to pick the 'best' from the possibilities.
- To get involved with the decisions, joing the Samba NT Domains mailing list at by mailing: listproc@samba.anu.edu.au

# File Ownership and Permissions.

- Current level of SMB protocol supported by Samba doesn't have support for file ownership or the changing of file permissions.
- This is only important for Windows NT clients.
- Unfortunately, in order to support this correctly the NT Domain protocols must be implemented as well.
  - This is due to the fact that username to SID mapping is done via DCE/RPC calls over an SMB named pipe.
  - This was done by accident by MS - there is no reason why this could not be done by a published mechanism, it was just easier for them to use the Domain code.

# File Ownership and Permissions (continued).

- Once the NT SID to uid mapping is in place, reporting file ownership will be easy.
- Changing file ownership will be more difficult, what if the incoming SID has no mapping in the UNIX space (just fail the rename probably) ?
- Likewise, reporting file permissions is also straightforward. NT ACLs are a superset of standard UNIX permissions and so can easily be mapped.
- Changing file permissions is more difficult.

# Changing File permissions.

- As the NT ACLs are a superset of standard UNIX permissions, some data may be lost when mapping NT ACLs to UNIX 'user-group-world'.
  - As a side effect, this may also affect the DOS 'attributes' Samba stores, as these are mapped onto the UNIX permission bits in various, user-configurable ways.
  - Storing the atttibutes or full ACLs in a separate place is possible (as AT&T Advanced Server for UNIX does), but I would strongly resist such a change (keep meta-data in the filesystem, as good design intended). It would also slow Samba down by an order of magnitude. Really.

# Changing File permissions (continued).

- Depending on vendor enthusiasm, and the availability of POSIX ACLs in the Free UNIXs, another option is to map NT ACLs onto POSIX ACLs.
  - I have done less thinking about this, as few UNIX installations seem to use ACLs in any great amount, and the Free UNIXs don't have much support for adding them.
- In terms of accuracy of representing NT ACLs, this would be the best solution, but only time will tell if this is really important for Samba to implement.

# Other Planned Samba enhancements - near term.

- Samba profiling to increase speed.
- Support for Windows NT specific SMB calls (additional performance increase is hoped for).
- ChangeNotify support.
- Integration with other PC filesharing mechanisms (op-lock library for AppleShare, NetWare).
- GUI Admin tool (choose one from many).
- Add blocking lock semantics.

# Longer Term

- UNICODE support over the wire.
- Windows NT Domain Contoller Support:
  - Current code is 'proof of concept'.
  - First step - integration of Samba machines into existing domain structure.
  - Second step - no need for NT Domain controllers.
  - This code will NOT support NT5.x - Microsoft are almost certainly doing a complete re-write.
- I18N Samba internals.

# Longer Term

- Samba WINS replication - heirarchical NetBIOS naming.
- Track changes to SMB/CIFS protocols. Add Kerberos 5 integration (awaiting specs).
- Introduce SMB remote administration protocol. Attempt to reduce need for NT server manager.
  - Make protocol ONC/RPC based.
  - Create NT Service to be run on Primary Domain Controller - gateway to their proprietary APIs.
  - Publish code under BSD license - attract support from other SMB/CIFS vendors.
  - Manage both NT & UNIX from one GUI tool.

# Samba references and URLs.

- Main Samba site :
  - http://samba.anu.edu.au/samba
- List of Samba commercial support providers :
  - ftp://samba.anu.edu.au/pub/samba/Support.txt
- Samba survey (please register your site if you are a Samba user - it's the only way we get interest from UNIX vendors, when they see some of their biggest customers there :-).
  - http://samba.anu.edu.au/samba/survey/