The following paper was originally published in the
Proceedings of the Large Installation System Administration of Windows NT Conference
Seattle, Washington, August 5–8, 1998

NT 3.5 / 4.0 Domains for UNIX

Luke Kenneth Casson Leighton

For more information about USENIX Association contact:

1. Phone: 510 528-8649
2. FAX: 510 548-5738
3. Email: office@usenix.org
4. WWW URL: http://www.usenix.org/

NT 3.5 / 4.0 Domains for UNIX

Luke Kenneth Casson Leighton
lkcl@switchboard.net

Abstract

NT domain logins, and some experimental administrative capabilities, have been added to a development branch of SAMBA, the publicly available file/print share program that makes UNIX servers look like Microsoft windows NT server.

Further work is needed, but the goal is to make UNIX look like windows NT, over a network. This will include full UNIX command-line administrative capability as well.

The implications of this are that UNIX will be fully adminsterable by the standard NT server tools (e.g "user manager for domains"; "server manager for domains"), and both UNIX and NT will be fully administerable using HTML (cgi-bin wrappers around the smbclient program).

Some of this functionality (both client and server) is already available. The latest version can be obtained by following the instructions in http://samba.anu.edu/au/cvs.html.

At present, SAMBA and smbclient can only provide or obtain information using DCE/RPC: no capability has been added to administer domain servers. This can (should) only be possible to do by administrators. Adding or changing SAM user accounts or domain groups is encrypted. The "backup domain controller" and "inter-domain trust relationships" also needs to be researched.

Final point: anyone running windows NT who allows SMB access through their firewall (ports 137-139) is strongly advised to look up and enable the "RestrictAnonymous" registry key in the microsoft KB articles, and to look for information on the "red button" bug in NT.

1. DCE/RPC in NT (not Win95)

NT runs an implementation of DCE/RPC over an SMB inter-process communication pipe. It uses an IDL to describe the data structures of the various remote procedure calls. NT opens named pipes such as \PIPE\NETLOGON and \PIPE\ntlsa (to do NT domain logins); \PIPE\samr (to do SAM database replication and administration - e.g using SRVMGR.EXE and USRMGR.EXE); \PIPE\srvsvc (to check on the status of files / connections / shares, and to disconnect users or close files).

Basically, all of the administration tools that can select "domains" or "workstations" in NT server all use DCE/RPC.

Windows 95, and WfWg, do _not_ use DCE/RPC. Where similar functionality can be found, there are equivalent calls made to SMB servers on the SMBTrans2 pipe. Windows NT does not use the majority of these functions. Only occasionally, if a DCE/RPC call is not implemented, will an NT server or NT workstation "thunk" down to using SMB Trans2 calls.

The only attempts seen by Windows 95 to use DCE/RPC is simply to open \PIPE\srvsvc (or \PIPE\wkssvc) and then close it. This would appear to be a method to check whether a machine is, to all network intents and purposes, an NT server or NT workstation (from an SMB file access point of view, a Windows SMB client reacts differently depending on what it determines the server to be. The "determination" step is extraordinarily convoluted, and is a really good example of bad object orientated programming).

The DCE/RPC over-the-wire data is relatively simple to decode. Unfortunately, no-one except Microsoft knows what's being used: you can only infer the meaning of this data from clicking on various admin tools, and seeing what happens: classic network reverse-engineering.

The work in progress is a double-edged sword for Microsoft. Their proprietary system is being understood and implemented in a popular SMB client / server suite for UNIX (SAMBA). One of the medium to long term benefits to Microsoft is that bugs will (and are) being found in their code and their protocols. This can

only improve the reliability and security of a quite impressive implementation of remote server administration.

That this is the most widely used remote administration suite in the world (USRMGR.EXE and SRVMGR.EXE, amongst other tools) makes it all the more important that it be secure and reliable.

2. NT Domain Logins

These are documented in http://www.cb1.com/~lkcl/ntdom/cifsntdomain.txt, and implemented in a pre-alpha version of SAMBA. SAMBA is available under the GPL (Gnu Public License).

NT domain logons are what happens on NT when you press ctrl-alt-delete, type in a username and password, select an NT-domain, and press return. What happens under Win95 is completely different (it uses the SMBtrans2 NetUserGetInfo call, but has the same end result in a completely insecure manner).

The sequence of events is as follows:

- Open \PIPE\ntlsa
- Issue a LsaQueryInfoPolicy, level 3
- Issue a LsaQueryInfoPolicy, level 5
- Close \PIPE\ntlsa
- Open \PIPE\NETLOGON
- Issue a LsaReqChallenge
- Issue a LsaAuth2
- Issue a LsaSamLogon, at the "interactive" level
- Leave the connection open until the user logs off
- Issue a LsaSamLogoff, or close the \PIPE\NETLOGON

The LsaQueryInfoPolicy calls determine whether the server is a member of a workgroup (negative answer to both calls), a member of a domain (positive answer only to the level 3 query), or a primary domain controller (positive answer to both calls). Only if the workstation determines that it is talking to a DC does it proceed further.

A session key is created from the server's response to the LsaReqChal and the client's LsaAuth2 Query, on _both_ the client and server. Each and every subsequent transaction is "signed" with a credential chain generated from this session key. [Unfortunately, the signature is not generated from the contents of the transaction itself, making it easy for someone to compromise Domain logons].

The response from the Domain Controller to the LsaSamLogon Query contains all the information that an administrator creates for a user with USRMGR.EXE for example, the user's profile location.

3. NT and UNIX Administration

The DCE/RPC commands, once understood and implemented, can be used to administer or be administered by, anything that also uses those commands. To the best of my knowledge, currently, only NT and its various ports to UNIX (by AT&T and SCO) use these commands. Oh, and SAMBA.

Stub functionality has been implemented in the SAMBA server code to enable "user manager for domains" (USRMGR.EXE) to view UNIX user accounts, and "server manager for domains" (SRVMGR.EXE) to view files, shares and sessions on a UNIX server.

The smbclient program has been updated to allow it to send DCE/RPC commands. It can be used, usually only with administrator priveleges, to provide exactly the same information as USRMGR.EXE and SRVMGR.EXE. This includes viewing files, shares, sessions, domain users, domain groups and domain aliases. Future versions will provide exactly the same administrative capabilities as these two programs, and more.

By ensuring that smbclient's output is in a machine-readable format, parsing scripts running from cgi-bins can be written that will allow NT (and UNIX servers running SAMBA) to be administered by HTML (WWW) clients.

[small side-note: the original purpose behind writing smbclient was as a "boot-strap", or "testing" tool for smbd, and it is still used for this purpose...]

4. NT to UNIX Mapping

This area is probably going to cause the most pain to administrators of mixed UNIX / NT systems, even though it's a pain at the moment. There are two areas of contention: users, and workstations. Through the use of SIDs NT supports, across all installations, unique (world-wide) user, group and workstation identification. (Adminstrators of NIS+ will be aware

that workstations are also similarly uniquely indentified).

The identification is subdivided into SIDs (security ids) for a domain, and RIDs (relative ids - relative to the SID, that is). A RID can be either a local user or a local group. Regardless, it must be unique within the domain (relative to the SID).

NT also has some common (well-known) RIDs and SIDs that have specific meanings (0x1f4 for the domain administrator's RID; S-1-1 for the World SID: see winnt.h or cifsntdomain.txt for some details).

The various UNIX flavours use 0x0 for the uid of the root superuser. They do not support the concept of a well-known guest account, like NT does: they certainly don't support, as standard, the concept of a "SID".

Probably the easiest way to deal with this is to first convert to using NIS+ (or an equivalent). The reason for this is that both NIS+ and NT support, in some form, the concept of "workstations" as individual users. Each domain must have a SID associated with it, and each user (including workstations) must have a user id. There are two other types of accounts, which are used for inter-domain trust relationships, and for primary / backup domain controller relationships.

SIDs and the well-known RIDs will need to be added to UNIX, somehow, in order to support NT domains. It is envisaged that SAMBA will provide this mapping, in such a way that the UNIX OS need know absolutely nothing about NT domains, and NT need know nothing about UNIX.

It actually doesn't matter what the scheme is, as long as it exists, whereby a set of UNIX accounts, workstations and UNIX groups are uniquely mapped into a SID/RID pair, making them world-wide unique.

Of even more contention at present is the issue of mapping existing NT accounts into UNIX ones. This would be best resolved by having a separate SID for the UNIX domain, and setting up an inter-domain trust relationship with the NT server.

Another scheme is to dedicate an entire UNIX server to be a SAMBA DC. Under these circumstances, ordinary UNIX access would be completely denied, and UNIX uids could then be allocated arbitrarily. This would be ideal for migrating from NT to UNIX.

SAMBA supports NT encrypted passwords through a password database API (see http://samba.anu.edu.au/listproc/samba-technical, thread named "password API needed"). If the NT user gives a correct NT password (fully documented in cifs6.txt), SAMBA allows the user access. The UNIX password is not involved in this process, and the UNIX password database still has to be maintained if the users are to be allowed access to the standard UNIX resources.

An alternative scheme to resolve the UNIX / NT username issue is to have a unique mapping for domain RIDs for users within a UNIX domain, but to have a non-monotonic mapping between those RID and the UNIX uids. For example:

NT user	**NT RID**	**UNIX user**	**uid**
Administrator	0x1f4	root	0
root	0x1000	root	0
sales_usr1	0x1001	salsusr	521
sales_usr2	0x1002	salesusr	521
foouser	0x1003	foouser	522
faruser	0x1004	faruser	523

Each NT username / NT RID is unique, yet the sales_usr 1 and 2 map to the same UNIX user and uid. likewise with root and administrator. This would be implemented by having a database which is maintained in parallel with the UNIX password file, which provides you with the mapping between UNIX uids and NT RIDs.

Identical consideration must be given for UNIX to NT group mapping, bearing in mind also, the fact that NT groups are like users: they can own resources such as files and directories.

5. Short-Term Plans

To research the "user manager for domains" functionality, finishing off the "read-only" side. this will allow viewing of UNIX users and groups with

USRMGR.EXE or smbclient. At present, the user profile information is available through smbd, but the domain group and domain alias information is either stub-code or incomplete.

To research the "server manager for domains" functionality, again finishing off the "read-only" side. This will allow SRVMGR.EXE or smbclient to view open files, sessions, shares and users on the server. The DCE/RPC side of this has been implemented in SAMBA. smbd currently only provides stub information; smbclient fully implements the client side. There are two buttons missing: "replication" and "alerts". These are on two separate DCE/RPC pipes, which I have not yet examined, and intend to.

To publicly encourage the discussion and resolution of the UNIX <-> NT SID and RID issue, and to implement an example mapping in SAMBA [see http://samba.anu.edu.au/listproc/samba-technical].

To rework the DCE/RPC code currently written so that it can be used in a general way, not just in a SAMBA-specific way (use of higher order - sometimes known as callback – functions where appropriate, for some of the enumeration containers. e.g. the "NetrFileEnum" and "SamrQueryDisplayInfo" functions).

To write a PAM (plug-in authentication module) which will allow users of Solaris 2.5 and Linux (the only systems that support PAMs, at the moment, to the best of my knowledge) to log in to (and out of!) NT Domains [see http://www.cb1.com/~lkcl/pam_ntdom].

6. Long-Term Plans

To document the full set of DCE/RPC administrative capabilities currently available in NT server, and to see them implemented in SAMBA (client and server) as a means to test and prove their usability and worth (making UNIX, from a network point of view, exactly like Windows NT 3.51 / 4.0 server).

This will include:

- Domain "primary / backup" relationships and inter-domain relationships
- Adding / creating accounts (first implementation to be SAM accounts)
- Closing of files / disconnecting of users / adding or removing shares.
- Changing SAM passwords from NT workstation.

Once this has been achieved, to then suggest the possibility of extending these calls for UNIX-specific (or other o/s, e.g MacOS) needs, as has been done with the CIFS file access protocol (cifs-unix by SCO - www.sco.com; mac extensions by Thursby - www.thursby.com).

Because the work already done by Microsoft in this field is very comprehensive, and appears to include some long-term planning and some degree of protocol independence, it is not expected that there be any re-design of, or significant additions to the DCE/RPC pipes protocols, for use in UNIX <-> UNIX administration.

7. Strategic Aim

To ensure that the administration and operation of "domain" systems (SAMBA, NT 4.0, NT 5.0, etc.) are reliable and secure. This is to be achieved by ensuring that all critical protocols are fully documented, and available for public review.