Unified Logons between Windows NT and UNIX using Winbind

Tim Potter
(tpot@linuxcare.com.au)
Andrew Tridgell
(tridge@linuxcare.com.au)

October 16, 2000

Abstract

Integration of UNIX and Microsoft Windows NT through a unified logon has been considered a "holy grail" in heterogeneous computing environments for a long time. We present winbind*, a component of the Samba suite of programs as a solution to the unified logon problem. Winbind uses a UNIX implementation of Microsoft RPC calls, Pluggable Authentication Modules, and the Name Service Switch to allow Windows NT domain users to appear and operate as UNIX users on a UNIX machine. This paper describes the winbind system, explaining the functionality it provides, how it is configured and how it works internally.*

1 Introduction

It is well known that UNIX and Microsoft Windows NT have different models for representing user and group information and use different technologies for implementing them. This fact has made it difficult to integrate the two systems in a satisfactory manner.

One common solution in use today has been to create identically named user accounts on both the UNIX and Windows systems and use the Samba suite of programs to provide file and print services between the two. This solution is far from perfect however, as adding and deleting users on both sets of machines becomes a chore and two sets of passwords are required both of which which can lead to synchronization problems between the UNIX and Windows systems and confusion for users.

We divide the unified logon problem for UNIX machines into three smaller problems:

- Obtaining Windows NT user and group information
- Authenticating Windows NT users
- Password changing for Windows NT users

Ideally, a prospective solution to the unified logon problem would satisfy all the above components without duplication of information on the UNIX machines and without creating additional tasks for the system administrator when maintaining users and groups on either system. The winbind system provides a simple and elegant solution to all three components of the unified logon problem.

2 What Winbind provides

Winbind unifies UNIX and NT account management by allowing a UNIX box to become a full member of a NT domain. Once this is done the UNIX box will see NT users and groups as if they were native UNIX users and groups, allowing the NT domain to be used in much the same manner that NIS+ is used within UNIX-only environments.

The end result is that whenever any program on the UNIX machine asks the operating system to lookup a user or group name the query will be resolved by asking the NT domain controller for the specified domain to do the lookup. Because Winbind hooks into the operating system at a low level (via the NSS name resolution modules in the C library) this redirection to the NT domain controller is completely transparent.

Users on the UNIX machine can then use NT user and group names as they would use "native" UNIX names. They can chown files so that they are owned by NT domain users or even login to the UNIX machine and run a UNIX X-Window session as a domain user.

The only obvious indication that Winbind is being used is that user and group names take the form DOMAIN\user and DOMAIN\group. This is necessary as it allows Winbind to determine that redirection to a domain controller is wanted for a particular lookup and which trusted domain is being referenced.

Additionally, Winbind provides a authentication service that hooks into the Pluggable Authentication Modules (PAM) system to provide authentication via a NT domain to any PAM enabled applications. This capability solves the problem of synchronizing passwords between systems as all passwords are stored in a single location (on the domain controller).

2.1 Target uses

Winbind is targeted at organizations that have an existing NT based domain infrastructure into which they wish to put UNIX workstations or servers. Winbind will allow these organizations to deploy UNIX workstations without having to maintain a separate account infrastructure. This greatly simplifies the administrative overhead of deploying UNIX workstations into a NT based organization.

Another interesting way in which we expect Winbind to be used is as a central part of UNIX based appliances. Appliances that provide file and print services to Microsoft based networks will be able to use Winbind to provide seamless integration of the appliance into the domain.

3 How Winbind Works

The winbind system is designed around a client/server architecture. A long-running winbind daemon listens on a UNIX domain socket waiting for requests to arrive. These requests are generated by the NSS and PAM clients and processed sequentially.

The technologies used to implement winbind are described in detail below.

3.1 Microsoft Remote Procedure Calls

Over the last two years, efforts have been underway by various Samba Team members to decode various aspects of the Microsoft Remote Procedure Call (MSRPC) system. This system is used for most network related operations between Windows machines including remote management, user authentication and NT print spooling. Although initially this work was done to aid the implementation of Primary Domain Controller (PDC) functionality in Samba, it has also yielded a body of code which can be used for other purposes.

Winbind uses various MSRPC calls to enumerate domain users and groups and to obtain detailed information about individual users or groups. Other MSRPC calls can be used to authenticate NT domain users and to change user passwords. By directly querying a Windows PDC for user and group information, winbind maps the NT account information onto UNIX user and group names.

3.2 Name Service Switch

The Name Service Switch, or NSS, is a feature that is present in many UNIX operating systems. It allows system information such as hostnames, mail aliases and user information to be resolved from different sources. For example a standalone UNIX workstation may resolve system information from a series of flat files stored on the local filesystem. A networked workstation may first attempt to resolve system information from local files, then consult a NIS database for user information or a DNS server for hostname information.

The NSS application programming interface allows winbind to present itself as a source of system information when resolving UNIX usernames and groups. Winbind uses this interface, and information obtained from a Windows NT server using MSRPC calls to provide a new source of account enumeration. Using standard UNIX library calls, one can enumerate the users and groups on a UNIX machine running winbind and see all users and groups in a NT domain plus any trusted domain as though they were local users and groups.

The primary control file for NSS is /etc/nsswitch.conf. When a UNIX application makes a request to do a lookup the C library looks in /etc/nsswitch.conf for a line which matches the service type being requested, for example the "passwd" service type is used when user or group names are looked up. This config line specifies which implementations of that service should be tried and in what order. If the passwd config line is:

```
passwd: files example
```

then the C library will first load a module called /lib/libnss_files.so followed by the module /lib/libnss_example.so. The C library will dynamically load each of these modules in turn and call resolver functions within the modules to try to resolve the request. Once the request is resolved the C library returns the result to the application[1].

This NSS interface provides a very easy way for Winbind to hook into the operating system. All that needs to be done is to put libnss_winbind.so in /lib/ then add "winbind" into /etc/nsswitch.conf at the appropriate place. The C library will then call Winbind to resolve user and group names.

3.3 Pluggable Authentication Modules

Pluggable Authentication Modules, also known as PAM, is a system for abstracting authentication and authorization technologies. With a PAM module it is possible to specify different authentication methods for different system applications without having to recompile these applications. PAM is also useful for implementing a particular policy for authorization. For example a system administrator may only allow console logins from users stored in the local password file but only allow users resolved from a NIS database to log in over the network.

Winbind uses the authentication management and password management PAM interface to integrate Windows NT users into a UNIX system. This allows Windows NT users to log in to a UNIX machine and be authenticated against a suitable Primary Domain Controller. These users can also change their passwords and have this change take effect directly on the Primary Domain Controller.

PAM is configured by providing control files in the directory /etc/pam.d/ for each of the services that require authentication. When a authentication request is made by an application the PAM code in the C library looks up this control file to determine what modules to load to do the authentication check and in what order. This interface makes adding a new authentication service for Winbind very easy, all that needs to be done is that the pam_winbind.so module is copied to /lib/security/ and the pam control files for relevant services are updated to allow authentication via winbind. See the PAM documentation for more details[2].

3.4 User and Group ID Allocation

When a user or group is created under Windows NT is it allocated a numerical relative identifier (RID). This is slightly different to UNIX which has a range of numbers which are used to identify users, and the same range in which to identify groups. It is winbind's job to convert RIDs to UNIX id numbers and vice versa.

When winbind is configured it is given part of the UNIX user id space and a part of the UNIX group id space in which to store Windows NT users and

[1] For more details see the `nsswitch.conf(5)` man page

[2] On most Linux systems you will find detailed PAM documentation in /usr/doc/pam*/

groups. If a Windows NT user is resolved for the first time, it is allocated the next UNIX id from the range. The same process applies for Windows NT groups. Over time, winbind will have mapped all Windows NT users and groups to UNIX user ids and group ids.

The results of this mapping are stored persistently in a ID mapping database (held in a tdb database). This ensures that RIDs are mapped to UNIX IDs in a consistent way[3].

3.5 Result Caching

A active system can generate a lot of user and group name lookups. To reduce the network cost of these lookups winbind uses a caching scheme based on the SAM sequence number supplied by NT domain controllers.

User or group information returned by a PDC is cached by winbind along with a sequence number also returned by the PDC. This sequence number is incremented by Windows NT whenever any user or group information is modified. If a cached entry has expired, the sequence number is requested from the PDC and compared against the sequence number of the cached entry. If the sequence numbers do not match, then the cached information is discarded and up to date information is requested directly from the PDC.

4 Installation and Configuration

The easiest way to install winbind is by using the packages provided in the pub/samba/appliance/ directory on your nearest Samba mirror. These packages provide snapshots of the Samba source code and binaries already setup to provide the full functionality of winbind. This setup is a little more complex than a normal Samba build as winbind needs a small amount of functionality from a development code branch called SAMBA_TNG[4].

Once you have installed the packages you should read the winbindd man page which will provide you with configuration information and give you sample configuration files. You may also wish to update the main Samba daemons (smbd and nmbd) with a more recent development release, such as the recently announced Samba 2.2 alpha release[5].

5 Limitations

Winbind has a number of limitations in its current released version which we hope to overcome in future releases

[3]On UNIX systems with a 32 bit uid/gid space it would be simpler to just use a linear algorithmic mapping. With the release of the Linux 2.4 kernel Linux systems will be ready for 32 bit UIDs. At that time we expect to release an update to winbind to use a linear mapping and dispense with the mapping database

[4]We are working on removing this requirement for a future release

[5]The appliance releases are based on earlier development versions of Samba 2.2

- Winbind is currently only available for the Linux operating system, although ports to other operating systems are certainly possible. For such ports to be feasible, we require the C library of the target operating system to support the Name Service Switch and Pluggable Authentication Modules systems. This is becoming more common as NSS and PAM gain support among UNIX vendors.

- The mappings of Windows NT RIDs to UNIX ids is not made algorithmically and depends on the order in which unmapped users or groups are seen by winbind. It may be difficult to recover the mappings of rid to UNIX id mapping if the file containing this information is corrupted or destroyed.

- Currently the winbind PAM module does not take into account possible workstation and logon time restrictions that may be been set for Windows NT users.

- Building winbind from source is currently quite tedious as it requires combining source code from two Samba branches. Work is underway to solve this by providing all the necessary functionality in the main Samba code branch.

6 Conclusion

The winbind system, through the use of the Name Service Switch, Pluggable Authentication Modules, and appropriate Microsoft RPC calls have allowed us to provide seamless integration of Microsoft Windows NT domain users on a UNIX system. The result is a great reduction in the administrative cost of running a mixed UNIX and NT network.