

[Figures are not included in this sample chapter]

Windows NT Domain Architecture

- 3 -

The SAM Database

This chapter will review:

- **SAM Contents**

The SAM database is stored as part of the Registry on domain controllers. It stores user, group, and computer account information for the entire domain.

- **SAM Size**

The size of the SAM database is a key factor in every domain design. Knowing the size calculations related to the SAM will help you optimize your domain design strategy.

- **SAM Synchronization**

The NetLogon service is responsible for SAM synchronization. The traffic associated with synchronization greatly affects your placement of domain controllers for load balancing and fault tolerance.

SAM Contents

The SAM database is the heart of the domain model. The user and group information stored in the SAM is used for authentication and controlling access to resources. The SAM stores the following information:

- User accounts
- Group accounts
- Computer accounts (for NT servers and workstations only)
- Trust relationship accounts

Windows 95 machines do not have computer accounts in the domain SAM and fail to utilize many of the advanced security features supported by Windows NT. Only Windows NT machines have computer accounts in the SAM. Unknown to many, each NT machine changes its computer account password every seven days. New passwords are randomly generated, and the change is initiated by the client machine.

AUTHOR'S NOTE

Several security features are employed to protect the SAM information. The user, group, and computer accounts are distinguished using security identifiers (SIDs), and each account has a corresponding password. User passwords are not stored in clear text; they are hashed using the RSA MD4 hashing algorithm. The SAM database is stored using triple DES encryption. The latest service pack uses a strong system key to further protect the SAM database; see Microsoft Knowledge Base article Q143475 for more information.

SAM Storage

The SAM database is a part of the NT Registry. The files that compose the Registry, called *hive files*, can be found in the WINNT\SYSTEM32\CONFIG directory. Two files make up the SAM database:

WINNT\SYSTEM32\CONFIG\SAM

WINNT\SYSTEM32\CONFIG\SECURITY

AUTHOR'S NOTE

In the WINNT\SYSTEM32\CONFIG directory are a SAM.LOG and a SECURITY.LOG file, which are used in a manner similar to a transaction log for a database. These transaction logs help maintain the integrity of the Registry when changes are made. A copy of the Registry hive files can be found in the WINNT\REPAIR directory. This directory is used during the initial installation to back up the initial Registry. Files in the REPAIR directory are updated when you create an Emergency Repair Disk (ERD).

Upon startup, the Registry files are loaded and exclusively locked by NT (that is, they can be directly accessed only by the NT system). The entire SAM database is loaded into RAM and cannot be paged out to the hard disk. Locating the SAM in fast, volatile memory should speed up the logon validation process.

The Registry editor can be used to view the structure of the SAM. However, before you can do this, you must change the default Registry permissions. The SAM is stored in two Registry keys:

HKEY_LOCAL_MACHINE\SAM

HKEY_LOCAL_MACHINE\SECURITY

The default Registry permissions for these keys are illustrated in Figure 3.1.

FIGURE 3.1 *The default permissions must be changed for you to see the structure of the SAM and SECURITY portions of the Registry.*

To view the structure of the SAM database you must modify permissions for the SAM and SECURITY registry subkeys. You modify the permissions by adding your user account with Full Control to each subkey. After doing so, you can expand the SAM and SECURITY keys to expose the structure of the SAM (see Figure 3.2).

FIGURE 3.2 *After changing the permissions, you can expose the structure of the SAM.*

SAM Structure

The SAM is composed of three separate sections:

- *SAM Accounts database*. Contains user, group, and computer accounts created by the administrator.
- *SAM Built-in database*. Contains built-in user and group accounts.
- *LSA database*. Contains the LSA (Local Security Authority) secrets used in trusts and domain controller account passwords. This also includes account policy settings.

SAM Size

The size of the SAM database is extremely important in enterprise environments because it must be synchronized among all domain controllers. Ideally, the SAM would store user, group, and computer accounts for the largest organization and mirror changes in real time. Realistically, you must live with the fact that much of the synchronization occurs over slower WAN links (instead of LAN links) and that the SAM has size limitations.

AUTHOR'S NOTE

When you run RDISK.EXE to create an ERD, the utility does not update the SAM and SECURITY files. Because the SAM database can grow to a size of several megabytes, RDISK.EXE is not designed to capture the SAM. However, if the SAM and SECURITY files are small enough to fit on the ERD, you can run RDISK -S to update them.

The theoretical size limitation of the SAM database is 40MB, although a more practical size is around 20MB. You can calculate the size of the SAM based on the user and group structure of your NT domain, as follows:

| | |
|----------------------|--|
| User Account | 1024 bytes |
| Computer Account | 512 bytes |
| Global Group Account | 512 bytes for the group plus 12 bytes per member |
| Local Group Account | 512 bytes for the group plus 36 bytes per user |

TIP

Windows Terminal Server increases the size of each user account in the SAM from 1KB to 2KB per user. The extra 1KB is used to store information describing the user's last terminal session and the user's terminal session configuration. For large organizations, the introduction of Windows Terminal Server can dramatically increase the size of the SAM. Consult the Windows Terminal Server documentation for more information.

Calculating the SAM Size

XYZ Corporation has a domain with 4,000 global domain users, 2,000 NT Workstations, and 100 NT

servers. There are 500 global groups with an average of 120 users each. There are only 20 local groups with an average of 8 members. What is the estimated size of the domain SAM?

| | | |
|--|---|------------|
| Number of user accounts | (A) | 4000 |
| Number of global groups | (B) | 500 |
| Average number of global group members | (C) | 120 |
| Number of local groups | (D) | 20 |
| Average number of local group members | (E) | 8 |
| NT Workstation and server domain members | (F) | 2000 + 100 |
| Domain SAM Size = | A×1024 + | |
| | B×512 + | |
| | C×B × 12 + | |
| | D×512 + | |
| | E×D×36 + | |
| | F×512 | |
| | =6,162,200 bytes, or approximately 6.16MB | |

AUTHOR'S NOTE

If you delete a large number of users or groups from the SAM, it does not shrink in size. Although Windows NT provides no mechanism to compress the size of the SAM, it does reclaim the space when users and groups are added. Microsoft proposes three recommended solutions to the problem. To obtain further information on the problem and the solutions, see Microsoft Knowledge Base Article Q140380.

SAM Synchronization

The primary domain controller (PDC) contains a master copy of the SAM database, and all backup domain controllers (BDCs) synchronize their SAM with the PDC. The SAM on the PDC is the only one that can be modified; the BDC SAM is read-only. When you make a change to the SAM database, it is committed to the master SAM on the PDC and replicated to all BDCs.

Changes to the SAM are recorded in a *change log*. The default size of the change log in Windows NT is 64K. Because each change entry is approximately 32 bytes, the log typically holds about 2,000 changes. The PDC checks for changes to its SAM on a regular interval (every five minutes). When the PDC discovers one or more changes, it informs all BDCs of the change. However, not all BDCs are informed at the same time; this prevents overloading the PDC with numerous simultaneous requests for changes.

When a BDC requests changes from the PDC, it informs the PDC of the last change it received. Thus, the PDC is able to track which BDCs have been updated. The synchronization process uses UDP port

138 (NetBIOS Datagram Service), and communication takes place using mailslot messages. The sequence of events is outlined in the following list:

1. The PDC discovers a change to its SAM.
2. The PDC announces the change to a BDC.
3. The BDC connects to IPC\$ of the PDC.
4. The BDC establishes a secure channel to the PDC and uses the NetLogon service to verify the SAM.
5. The BDC uses server message blocks (SMB) or Remote Procedure Calls (RPC) to transfer the updated data (depending on the size of the update).

AUTHOR'S NOTE

When you add a user account or reset a password, the change is made to the master copy of the SAM located on the PDC. Because the PDC checks for changes every five minutes, this change can take as much as five minutes to replicate to a BDC.

TIP

You can use the ADDUSERS.EXE utility from the NT Resource Kit to export the user and group information from the SAM to a text file. You can also use the utility to add a large number of accounts to the SAM. This is best done by listing all the new accounts in a spreadsheet, exporting the spreadsheet to a comma-delimited text file, and importing the file into the SAM using the ADDUSERS utility.

You can extract user and group information from the SAM by issuing the following command:

```
addusers \\computername /d <filename>
```

By default, the command creates a comma-separated text file that can be easily imported into spreadsheet programs. For more information on ADDUSERS.EXE, consult the documentation included with the NT Resource Kit.

You should be very cautious of adding a large number of accounts at once. If replication of the SAM occurs over a WAN link, you might saturate the link with a large number of changes.

Synchronization Overview

There are two types of SAM synchronization:

- *Partial synchronization.* The timed replication in which all BDCs are notified of SAM changes that have occurred since the last synchronization.
- *Full synchronization.* Copying the entire SAM to a BDC. Full synchronization events are dangerous because they can saturate slow network links. According to Microsoft, on a 56Kbps point-to-point circuit it would take about 24 hours to replicate a 30,000-user SAM.

A full synchronization typically occurs when

- A new BDC is installed
- The change log fills
- An error occurs during a partial synchronization event

When you install a BDC, you must be sure it has network connectivity to the PDC. During the installation, the BDC performs a full synchronization with the PDC to establish its initial copy of the SAM. It is also possible for a BDC to do a full synchronization if it has been offline for an extended period and the change log has filled during that time; the change log simply wraps to overwrite older changes with newer ones. When this happens, the only way for the BDC to get an up-to-date copy is to pull the entire SAM from the PDC. This is a good reason to be cautious if you are using tools that add a large number of user accounts to the SAM instantaneously.

TIP

Suppose you must distribute a BDC to a remote site. If possible, you should connect the machine to the same LAN as the PDC, install Windows NT, and then distribute the machine to its remote location. Installing on the same LAN as the PDC enables a fast, full synchronization of the domain SAM, which is unlike pulling a large SAM database across a slow WAN link.

Calculating SAM Replication Traffic

XYZ Corporation has a domain with 4,000 global domain users, 2,000 NT workstations, and 100 NT servers. The password expiration policy states that passwords expire every 60 days.

| | | |
|---|-----------------|------|
| Number of user accounts | (A) | 4000 |
| Password expires in how many days | (B) | 60 |
| Number of machine accounts | (C) | 2100 |
| *How many user accounts will you add per month? | (D) | 200 |
| *How many computer accounts will you add per month? | (E) | 200 |
| *Estimate five percent of A if unknown. | | |
| Number of changes to SAM per day = | A / B + | |
| | C / 7 + | |
| | D / 30 + | |
| | E / 30 | |
| | =380 | |
| | changes per day | |

NOTE: This formula assumes there are 30 days per month.

AUTHOR'S NOTE

This formula gives a good estimate but assumes that user password changes are spread over the 60-day period. In reality, most users do not change their password until Windows NT sends a warning to them stating that there are 14 days left until expiration. In most cases, user accounts on the NT system are all added at about the same time. This means that all the user account passwords will expire at nearly the same time, so there will be a large number of changes after the warnings are sent.

Synchronization Registry Parameters

The synchronization process is controlled via several Registry parameters found in the following location:

```
\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ Netlogon\Parameters
```

The following sections describe each of the parameters you will find there.

Pulse

Pulse defines the pulse frequency, in seconds. All changes made to the user account database since the last pulse are collected. Then, after the Pulse time expires, a pulse is sent to each BDC needing the changes (no pulse is sent to a BDC that is up-to-date). Default value: 300 (5 minutes); value range: 60 (1 minute)-3,600 (1 hour).

PulseConcurrency

PulseConcurrency defines the maximum number of BDCs that the PDC will notify at one time. The NetLogon Service sends pulses to individual BDCs, which causes the BDCs to respond by requesting any database changes. To control the maximum load these responses place on the PDC, the PDC must simultaneously handle only the same number of responses as the number of pulses specified under PulseConcurrency. Increasing PulseConcurrency increases the load on the PDC; decreasing PulseConcurrency increases the time it takes for a domain with a large number of BDCs to send a change to all of them. Default value: 20; value range: 1-500.

PulseMaximum

PulseMaximum defines the maximum time between pulses, in seconds. Every BDC will be sent at least one pulse this often, regardless of whether its user account database is up-to-date. Default value: 7,200 (2 hours); value range: 60 (1 minute)-86,400 (1 day).

PulseTimeout1

PulseTimeout1 defines how long, in seconds, the PDC will wait for a nonresponsive BDC. When a BDC is sent a pulse, it must respond within this time period; if it does not, it is considered to be nonresponsive. A nonresponsive BDC is not counted against the PulseConcurrency limit, thereby allowing the PDC to send a pulse to another BDC in the domain.

If this number is too large, a domain with a large number of nonresponsive BDCs will take a long time to complete a partial synchronization. If this number is too small, a slow BDC might be falsely accused of being nonresponsive. When the BDC finally does respond, it receives a partial synchronization from the PDC, which can increase the load on the PDC. Default value: 5 (5 seconds); value range: 1 (1 second)-120 (2 minutes).

PulseTimeout2

PulseTimeout2 defines how long, in seconds, a PDC will wait for a BDC to complete partial synchronization. Even after a BDC initially responds to a pulse (as required by PulseTimeout1), it must continue the synchronization process or else it will be considered nonresponsive. Each time the BDC calls the PDC, the BDC again must respond in the amount of time defined by PulseTimeout2.

If this number is too large, a slow BDC (or one that has its ReplicationGovernor rate artificially governed) will consume one of the PulseConcurrency slots. If this number is too small, the load on the PDC will be unduly increased because of the large number of BDCs doing a partial sync. Default value: 300 (5 minutes); value range: 60 (1 minute)-3,600 (1 hour).

Randomize

Randomize specifies the BDC backoff period, in seconds. When the BDC receives a pulse, it backs off between zero and the number of Randomize seconds before calling the PDC. Randomize should always be smaller than the PulseTimeout1. Default value: 1 (1 second); value range: 0-120 (2 minutes).

Consider that the time needed to synchronize a change to all the BDCs in a domain will be greater than the following:

$$\left(\frac{\text{Randomize}}{2} \right) \times \text{Number of BDCs in domain} / \text{PulseConcurrency}$$

The Replication Governor

Windows NT uses a 128KB buffer for the synchronization process. You can throttle the replication process by decreasing the size of this buffer. The buffer size is changed by adding the following Registry value:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ Netlogon\Parameters

Value Name: ReplicationGovernor

Date Type: REG_DWORD

Value: 0-100

The ReplicationGovernor can be set to a value between 0 and 100, which describes a percentage of the buffer used. For example, a value of 50 causes the BDC to use a 64KB buffer. A setting of 0 prevents replication; any setting below 25 is highly discouraged.

Forcing Synchronization

Rather than waiting for the PDC to check its SAM database for changes, you can force a synchronization of the SAM database. To synchronize a single BDC, open Server Manager, highlight the BDC, and select Synchronize with Primary Domain Controller from the Computer menu (see Figure 3.3). If you want to immediately propagate the SAM changes to all BDCs in the domain, highlight the PDC and select Synchronize Entire Domain from the Computer menu.

In Windows NT 3.51, synchronizing a single BDC in Server Manager forces a full synchronization; in Windows NT 4.0, it is possible only to trigger a partial synchronization. To force a full synchronization of a BDC, you have two choices:

- Open a command window on the BDC and type

```
net accounts /sync
```

- Use NLTEST from the NT Resource Kit as follows:

```
nltest /sync /server:<name of BD
```

FIGURE 3.3 *You can force a single BDC to synchronize its SAM with the PDC.*