Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación

Curso de Integración de Sistemas Linux/Windows

Linux CentOS 5 y Windows Server 2003 R2

por
Fernando Ferrer García
Andrés Terrasa Barrena

Curso Académico 2009/2010
Valencia, 21 de mayo de 2010

Curso de Integración de Sistemas Linux/Windows

Indice

Lista de figuras

Lista de tablas

1

Administración de dominios Windows 2003

Indice

1.1. Introducción

Este capítulo introduce los conceptos fundamentales sobre dominios Windows 2003, suficientes para poder unificar y centralizar la administración de conjuntos de sistemas Windows 2003 en organizaciones de cualquier tamaño.

En concreto, se explicarán los conceptos fundamentales que soportan el Directorio Activo (*Active Directory*), así como la administración del mismo, incluyendo los principales objetos que pueden definirseen el mismo, la compartición de recursos entre sistemas de la organización y la delegación de tareas administrativas dentro de un dominio.

1.2. El Directorio Activo

1.2.1. Dominios Windows 2003 y el Directorio Activo

Hoy en día, los ordenadores existentes en cualquier organización se encuentran formando parte de redes de ordenadores, de forma que pueden intercambiar información. Desde el punto de vista de la administración de sistemas, la mejor forma de aprovechar esta característica es la creación de un *dominio* de sistemas, en donde la información administrativa y de seguridad se encuentra *centralizada* en uno o varios servidores, facilitando así la labor del administrador. Windows 2003 utiliza el concepto de **directorio** para implementar dominios de sistemas Windows 2003.

En el ámbito de las redes de ordenadores, el concepto de *directorio* (o almacén de datos) es una estructura jerárquica que almacena información sobre objetos en la red, normalmente implementada como una base de datos optimizada para operaciones de lectura y que soporta búsquedas de grandes datos de información y con capacidades de exploración.

Active Directory es el servicio de directorio de una red de Windows 2003. Este servicio de directorio es un servicio de red que almacena información acerca de los recursos de la red y permite el acceso de los usuarios y las aplicaciones a dichos recursos, de forma que se convierte en un medio de organizar, controlar y *administrar* centralizadamente el acceso a los recursos de la red.

Como veremos, al instalar el Directorio Activo en uno o varios sistemas Windows 2003 (Server) de nuestra red, convertimos a dichos ordenadores en los servidores del dominio, o más correctamente, en los denominados *Controladores de Dominio* (*Domain Controllers*) o simplemente "DCs". El resto de los equipos de la red pueden convertirse entonces en los *clientes* de dicho servicio de directorio, con lo que reciben toda la información almacenada en los controladores. Esta información incluye no sólo las cuentas de usuario, grupo, equipo, etc., sino también los perfiles de usuario y equipo, directivas de seguridad, servicios de red, etc. El Directorio Activo se convierte así en la herramienta fundamental de administración de toda la organización.

Una de las ventajas fundamentales del Directorio Activo es que separa la estructura *lógica* de la organización (dominios) de la estructura *física* (topología de red). Ello permite, por una

parte, independizar la estructuración de dominios de la organización de la topología de la(s) red(es) que interconectan los sistemas; y, por otra parte, permite administrar la estructura física explícitamente cuando es necesario, de forma independiente de la administración de los dominios. Más adelante en este capítulo se exponen ambas estructuras detalladamente.

1.2.2. Estándares relacionados

A diferencia de su antecesor NT 4.0, Windows 2003 proporciona compatibilidad con un buen número de protocolos y estándares existentes, ofreciendo interfaces de programación de aplicaciones que facilitan la comunicación con otros servicios de directorio. Entre ellos, podemos destacar los siguientes:

- DHCP (*Dynamic Host Configuration Protocol*). Protocolo de configuración dinámica de ordenadores, que permite la administración desatendida de direcciones de red.
- DNS (*Domain Name System*). Servicio de nombres de dominio que permite la administración de los nombres de ordenadores. Este servicio constituye el mecanismo de asignación y resolución de nombres (traducción de nombres simbólicos a direcciones IP) en Internet.
- SNTP (*Simple Network Time Protocol*). Protocolo simple de tiempo de red, que permite disponer de un servicio de tiempo distribuido.
- LDAP (*Lightweight Directory Access Protocol*). Protocolo ligero (o compacto) de acceso a directorio. Este es el protocolo mediante el cual las aplicaciones acceden y modifican la información existente en el directorio.
- Kerberos V5. Protocolo utilizado para la autenticación de usuarios y máquinas..
- Certificados X.509. Estándar que permite distribuir información a través de la red de una forma segura.

De entre todos ellos, es imprescindible que el administrador conozca en detalle la relación entre el Directorio Activo y DNS. A continuación se exponen los aspectos fundamentales de esta relación.

1.2.3. El Directorio Activo y DNS

El Directorio Activo y DNS son espacios de nombres. Podemos entender un espacio de nombres como un área delimitada en la cual un nombre puede ser resuelto. La resolución de nombres es el proceso de traducción de un nombre en un objeto o información que lo representa. Por ejemplo, el sistema de ficheros NTFS puede ser considerado un espacio de nombres en cual un fichero puede ser resuelto en el fichero propiamente dicho.

DNS es el sistema de nombres de facto para redes basadas en el protocolo TCP/IP y el servicio de nombres que se usa para localizar equipos en Internet. Windows 2003 utiliza DNS para localizar equipos y controladores de dominio. Una estación de trabajo o servidor miembro busca un controlador de dominio preguntando a DNS.

Cada dominio de Windows 2003 se identifica unívocamente mediante un nombre DNS (por ejemplo, `miempresa.com`) y cada equipo basado en Windows 2003 que forma parte de un dominio tiene un nombre DNS cuyo sufijo es precisamente el nombre DNS de dicho dominio (siguiendo con el ejemplo, `pc0100.miempresa.com`). De esta forma vemos que dominios y equipos se representan como objetos en Active Directory y como nodos en DNS. Por tanto resulta fácil confundir ambos espacios de nombres ya que comparten idénticos nombres de dominio. La diferencia es que aunque comparten la misma estructura, almacenan información diferente: DNS almacena zonas y registros de recursos y el Directorio Activo guarda dominios y objetos de dominio.

Como conclusión diremos que Directorio Activo *utiliza* DNS, para tres funciones principales:

1. **Resolución de nombres**: DNS permite realizar la resolución de nombres al convertir los nombres de host a direcciones IP.
2. **Definición del espacio de nombres**: Directorio Activo utiliza las convenciones de nomenclatura de DNS para asignar nombre a los dominios.
3. **Búsqueda de los componentes físicos de AD**: para iniciar una sesión de red y realizar consultas en Directorio Activo, un equipo con Windows 2003 debe encontrar primero un controlador de dominio o servidor de catálogo global para procesar la autenticación de inicio de sesión o la consulta. La base de datos DNS almacena información acerca de qué equipos realizan estas funciones para que se pueda atender la solicitud adecuadamente. En concreto, esto se lleva a cabo mediante registros de recursos `SRV` que especifican el servidor (o servidores) del dominio que proporcionan los servicios de directorio correspondientes.

1.2.4. Estructura lógica

La estructura lógica del Directorio Activo se centra en la administración de los *recursos* de la red organizativa, independientemente de la ubicación física de dichos recursos, y de la topología de las redes subyacentes. Como veremos, la estructura lógica de la organización se basa en el concepto de *dominio*, o unidad mínima de directorio, que internamente contiene información sobre los recursos (usuarios, grupos, directivas, etc.) disponibles para los ordenadores que forman parte de dicho dominio. Dentro de un dominio es posible subdividir lógicamente el directorio mediante el uso de *unidades organizativas*, que permiten una administración independiente sin la necesidad de crear múltiples dominios. Sin embargo, si la organización desea estructurarse en varios dominios, también puede hacerlo, mediante los conceptos de *árbol* y *bosque*; ambos son jerarquías de dominios a distintos niveles, en función de si los dominios comparten o no un espacio de nombres común. A continuación se presentan todos estos conceptos de forma más detallada.

1.2.4.1. Dominios

La unidad central de la estructura lógica del Directorio Activo es el dominio. Un dominio es

un conjunto de equipos que comparten una base de datos de directorio común. Dentro de una organización, el Directorio Activo se compone de uno o más dominios, cada uno de ellos soportado, al menos, por un controlador de dominio. Como hemos visto, cada dominio se identifica unívocamente por un nombre de dominio DNS, que debe ser el sufijo DNS principal de todos los ordenadores miembros del dominio, incluyendo el o los controladores.

El uso de dominios permite conseguir los siguientes objetivos:

- **Delimitar la seguridad**. Un dominio Windows 2003 define un límite de seguridad. Las directivas de seguridad, los derechos administrativos y las listas de control de acceso (*Access Control Lists*, ACLs) no se comparten entre los dominios. Active Directory puede incluir uno o más dominios, teniendo cada uno sus propias directivas de seguridad.
- **Replicar información**. Un dominio es una partición del directorio, las cuales son unidades de replicación. Cada dominio almacena solo la información sobre los objetos localizados en este dominio. Active Directory utiliza un modelo de replicación con varios maestros. Todos los controladores de dominio del dominio pueden recibir cambios realizados sobre los objetos, y pueden replicar estos cambios a todos los controladores de dominio en el dominio.
- **Aplicar Políticas (o Directivas) de Grupo**. Un dominio define un posible ámbito para las políticas. Al aplicar un objeto de política de grupo (GPO) al dominio, este establece como los recursos del dominio se configuran y se usan. Estas políticas se aplican dentro del dominio y no a través de los dominios.
- **Delegar permisos administrativos**. En las redes que ejecutan Windows 2003, se puede delegar a medida la autoridad administrativa tanto para unidades organizativas (OUs) individuales como a dominios individuales, lo cual reduce el número de administradores necesarios con amplios permisos administrativos. Ya que un dominio representa un límite de seguridad, los permisos administrativos se limitan al dominio.

1.2.4.2. Múltiples dominios en la misma organización

Existen muchos casos en los que es interesante disponer de varios dominios de ordenadores Windows 2003 en la misma organización (distribución geográfica o departamental, distintas empresas, etc.). El Directorio Activo permite almacenar y organizar la información de directorio de varios dominios de forma que, aunque la administración de cada uno sea independiente, dicha información esté disponible para todos los dominios.

Según los estándares de nombres DNS, los dominios de Active Directory se crean dentro de una estructura de árbol invertida, con la raíz en la parte superior. Además, esta jerarquía de dominios de Windows 2003 se basa en relaciones de confianza, es decir, los dominios se vinculan por relaciones de confianza entre dominios.

Cuando se instala el primer controlador de dominio en la organización se crea lo que se denomina el *dominio raíz* del bosque, el cual contiene la configuración y el esquema del bosque (compartido por todos los dominios de la organización). Más adelante, podemos agregar

dominios como subdominios de dicha raíz (**árbol de dominios**) o bien crear otros dominios "hermanos" de la raíz (**bosque de dominios**), debajo del cual podemos crear subdominios, y así sucesivamente.

Arbol Un árbol es un conjunto de uno o más dominios que comparten un espacio de nombres contiguo. Si existe más de un dominio, estos se disponen en estructuras de árbol jerárquicas.

El primer dominio creado es el dominio raíz del primer árbol. Cuando se agrega un dominio a un árbol existente este pasa a ser un dominio secundario (o hijo). Un dominio inmediatamente por arriba de otro dominio en el mismo árbol de dominio es su padre. Todos los dominios que tengan un dominio raíz común se dice que forman un espacio de nombres contiguo.

Los dominios secundarios (hijos) pueden representar entidades geográficas (valencia, madrid, barcelona), entidades administrativas dentro de la organización (departamento de ventas, departamento de desarrollo ...), u otras delimitaciones específicas de una organización, según sus necesidades.

Los dominios que forman un árbol se enlazan mediante relaciones de confianza bidireccionales y transitivas. La relación padre-hijo entre dominios en un árbol de dominio es simplemente una relación de confianza. Los administradores de un dominio padre no son automáticamente administradores del dominio hijo y el conjunto de políticas de un dominio padre no se aplican automáticamente a los dominios hijo.

Por ejemplo, en la Universidad Politécnica de Valencia cuyo dominio actual de Active Directory es `upv.es` se crean dos nuevos departamentos: DSIC y DISCA. Con el fin de permitir la administración de los dominios por parte de los técnicos de los respectivos departamentos, se decide agregar dos nuevos dominios a su árbol de dominios existente en lugar de crear dos unidades organizativas en el dominio existente. Los dominios resultantes, `dsic.upv.es` y `disca.upv.es` forman un espacio de nombres contiguo, cuya raíz es `upv.es`. El administrador del dominio padre (`upv.es`) puede conceder permisos para recursos a cuentas de cualquiera de los tres dominios del árbol, pero por defecto no los puede administrar.

Bosque Un bosque es un grupo de árboles que no comparten un espacio de nombres contiguo, conectados a través de relaciones de confianza bidireccionales y transitivas. Un dominio único constituye un árbol de un dominio, y un árbol único constituye un bosque de un árbol. Los árboles de un bosque aunque no forman un espacio de nombres común, es decir, están basados en diferentes nombres de dominio raíz de DNS, comparten una configuración, un esquema de directorio común y el denominado catálogo global.

Es importante destacar que, aunque los diferentes árboles de un bosque no comparten un espacio de nombres contiguo, el bosque tiene siempre un único dominio raíz, llamado precisamente *dominio raíz del bosque*; dicho dominio raíz será siempre el primer dominio creado por la organización.

Añadir nuevos dominios a un bosque es fácil. Sin embargo, existen ciertas limitaciones que hemos de tener en cuenta al respecto:

- No se pueden mover dominios de Active Directory entre bosques.
- Solamente se podrán eliminar dominios de un bosque si este no tiene dominios hijo.
- Después de haber establecido el dominio raíz de un árbol, no se pueden añadir dominios con un nombre de nivel superior al bosque.
- No se puede crear un dominio padre de un dominio existente.

En general, la implementación de bosques y árboles de dominio permite mantener convenciones de nombres tanto contiguos como discontiguos, lo cual puede ser útil en organizaciones con divisiones independendientes que quieren mantener sus propios nombres DNS.

Finalmente, debemos relacionar estos conceptos con el procedimiento para **crear un dominio**. Esto se hace mediante la ejecución de un asistente denominado **dcpromo** en el sistema Windows 2003 Server que queramos *promocionar* a controlador de dominio. En concreto, este asistente nos permite elegir entre las siguientes opciones de instalación:

1. DC adicional de un dominio existente o DC para un dominio nuevo (creación de un dominio).

2. En el segundo caso, el dominio (nuevo) puede ser un dominio secundario de otro dominio existente (es decir, un subdominio en un árbol de dominios ya creado), o bien el dominio principal (raíz) de un nuevo árbol de dominios.

3. En este segundo caso, el dominio raíz puede ser de un bosque existente (agregamos una raíz nueva a un bosque) o de un nuevo bosque (creación del bosque). Por tanto, el primer dominio que creemos en una organización siempre será un dominio nuevo de un árbol nuevo de un bosque nuevo.

1.2.4.3. Niveles funcionales

La funcionalidad de los dominios de Windows 2003 es diferente de la que tenían sistemas anteriores de la misma familia (Windows NT4 y Windows 2000). Tanto Windows 2000 como Windows 2003 pueden configurarse en diferentes *niveles funcionales* ("modos de dominio" en Windows 2000) para que puedan ser compatibles con sistemas anteriores.

En concreto, Windows Server 2003 soporta cuatro niveles funcionales *de dominio* y tres niveles funcionales *de bosque*, explicados a continuación.

Un *dominio* Windows 2003 puede estar en cuatro niveles funcionales:

1. **Windows 2000 mixto**. Este es el nivel funcional por defecto cuando se crea un nuevo do-

minio (o cuando se actualiza de un sistema anterior). En este nivel, los DCs de Windows 2003 son compatibles dentro del mismo dominio con DCs Windows NT4 y Windows 2000. Por este motivo, un conjunto significativo de opciones de configuración no están disponibles (como por ejemplo el anidamiento de grupos, el cambio de ámbito de grupo y los grupos universales).

2. **Windows 2000 nativo**. En este nivel funcional, los DCs de Windows 2003 son compatibles dentro del mismo dominio con DCs que ejecuten Windows 2000 pero no Windows NT4. Se tiene una funcionalidad completa del Directorio Activo a nivel de Windows 2000, aunque se excluyen las nuevas opciones que Windows 2003 ha introducido en los dominios (entre las cuales destaca la posibilidad de cambiar de nombre a un DC sin necesidad de despromocionarlo previamente).

3. **Windows Server 2003 provisional**. En este nivel funcional, los DCs de Windows 2003 son compatibles dentro del mismo dominio con DCs que ejecuten Windows NT4 pero no Windows 2000. Se trata de un nivel funcional reservado únicamente para migraciones directas de NT4 a Windows 2003, y se supone que es completamente transitorio.

4. **Windows Server 2003**. En este nivel funcional, los DCs de Windows 2003 son compatibles únicamente entre sí (sólo puede configurarse si todos los DCs del dominio son Windows Server 2003). Este nivel ofrece la funcionalidad completa de dominios, incluyendo todas las características definidas en Windows 2000 más las nuevas incluidas en Windows Server 2003.

Por otro lado, un *bosque de dominios* Windows 2003 puede estar en tres niveles funcionales:

- **Windows 2000**. Este es el nivel por defecto al crear un nuevo bosque (o actualizar desde un sistema anterior). En este nivel funcional, los DCs de Windows 2003 son compatibles dentro del bosque con DCs que ejecuten Windows 2000 o Windows NT4. Se tiene una funcionalidad completa del bosque a nivel de Windows 2000, aunque se excluyen las nuevas opciones que Windows 2003 ha introducido a este nivel (incluyendo mejoras en la replicación, las relaciones de confianza entre bosques y la posibilidad de renombrar dominios en lugar de eliminarlos y volverlos a crear).

- **Windows Server 2003 provisional**. En este nivel funcional, los DCs de Windows 2003 son compatibles dentro del bosque con DCs que ejecuten Windows NT4 pero no Windows 2000. Se trata de un nivel funcional reservado únicamente para la migración directa de NT4 a Windows 2003 en el primer dominio del bosque, y se supone que es completamente transitorio.

- **Windows Server 2003**. En este nivel funcional, los DCs de Windows 2003 son compatibles únicamente entre sí (sólo puede configurarse si todos los DCs *del bosque* son Windows Server 2003). Este nivel ofrece la funcionalidad completa para los bosques, incluyendo todas las características definidas en Windows 2000 más las nuevas incluidas en Windows Server 2003.

Por tanto, por defecto al crear un nuevo bosque, éste se sitúa en el nivel funcional "Windows 2000", y al crear un nuevo dominio, éste se sitúa en el nivel funcional "Windows 2000 mixto", manteniendo, en ambos casos, la compatibilidad completa con sistemas anteriores.

Tanto a nivel de dominio como de bosque, la transición entre niveles funcionales sólo es posible *elevando* el nivel actual, es decir, pasando a un nivel con mayor funcionalidad (a excepción de los niveles provisionales, reservados para casos poco frecuentes). La elevación de nivel funcional es, por tanto, un paso irreversible, y sólo debe hacerse cuando se está seguro de que no van a añadirse sistemas anteriores como DCs al dominio, o al bosque. La elevación del nivel funcional de dominio o de bosque se realiza desde la herramienta "Dominios y Confianzas de Active Directory".

1.2.4.4. Relaciones de confianza

Una relación de confianza es una relación establecida entre dos dominios de forma que permite a los usuarios de un dominio ser reconocidos por los DCs de otro dominio. Estas relaciones permiten a los usuarios acceder a los recursos de otro dominio y a los administradores definir los permisos y derechos de usuario para los usuarios del otro dominio.

Windows Server 2003 soporta varios tipos de relaciones de confianza, que veremos posteriormente. Al margen de su uso, los diferentes tipos de relaciones se diferencian en función de tres rasgos característicos:

- **Método de creación**: algunos tipos de relaciones de confianza se crean de forma automática (implícita) y otros de forma manual (explícita).
- **Dirección**: algunos tipos de relaciones son unidireccionales y otros bidireccionales. Si la relación es unidireccional, los usuarios del dominio A (de confianza) pueden utilizar los recursos del dominio B (que confía), pero no al revés. En una relación bidireccional, ambas acciones son posibles.
- **Transitividad**: algunos tipos de relaciones son transitivas y otras no. Una relación de confianza transitiva es aquella que permite que si un dominio A confía en otro B, y éste confía en un tercero C, entonces de forma automática, A confía en C. En las relaciones no transitivas, la confianza entre A y C tendría que añadirse explícitamente.

Después de ver las características de las relaciones de confianza, se explican a continuación los tipos de relaciones de confianza válidos en dominios y bosques Windows Server 2003:

- **Confianza raíz de árbol**. Esta relación se establece de forma automática entre los dominios raíz del mismo bosque. Es bidireccional y transitiva.
- **Confianza principal-secundario**. Esta relación se establece de forma automática entre un dominio dado y cada uno de sus subdominios (o dominios secundarios). Es bidireccional y transitiva.

- **Confianza de acceso directo**. Este tipo de relación debe establecerse de forma manual, y tiene como objetivo mejorar la eficiencia en los inicios de sesión remotos. Si los usuarios de un dominio A necesitan acceder frecuentemente a los recursos de un dominio B, y ambos dominios se encuentran "lejos" entre sí (con muchos dominios intermedios), la confianza permite una relación directa que acorta el tiempo necesario para la autentificación de los usuarios. Es transitiva y unidireccional (si se necesita en ambos sentidos, deben crearse dos relaciones de confianza).

- **Confianza externa** . Este tipo de relación se crea manualmente y permite a usuarios de un dominio Windows 2003 acceder a recursos ubicados en dominios de otro bosque, o bien dominios Windows NT4. Es unidireccional e intransitiva.

- **Confianza de bosque** . Este tipo de relación debe crearse de forma manual entre los dominios raíz de dos bosques distintos, y permite a los usuarios de cualquier dominio de un bosque acceder a los recursos de cualquier dominio del otro bosque. Es unidireccional y sólo es transitiva entre dos bosques. Este tipo de relaciones sólo están disponibles si ambos bosques se sitúan en el nivel funcional "Windows Server 2003".

- **Confianza de territorio** . Este tipo de relación debe crearse de forma manual entre un dominio Windows 2003 y un territorio (*realm*) Kerberos (versión 5) que no sea Windows, y permite interoperabilidad entre ambos. Es unidireccional y puede ser transitiva o no.

Por tanto, las relaciones de confianza automáticas (implícitas) se crean por defecto al ir añadiendo dominios al bosque, y mantienen relacionados todos esos dominios de forma bidireccional y transitiva. El efecto de estas relaciones es que de forma automática, los usuarios de cualquier dominio del bosque son conocidos (y pueden acceder a los recursos) en todos los dominios de dicho bosque. Las relaciones de confianza manuales (explícitas) están reservadas para casos en donde se busca mejorar la eficiencia o permitir interactuar con otros bosques o con dominios que no son Windows 2003.

1.2.4.5. Unidades Organizativas

Una Unidad Organizativa (*Organizational Unit*, OU) es un objeto del Directorio Activo que puede contener a otros objetos del directorio. Es decir, es un *contenedor* de otros objetos, de forma análoga a una carpeta o directorio en un sistema de archivos tradicional. En concreto, dentro de una unidad de este tipo pueden crearse cuentas de usuario, de grupo, de equipo, de recurso compartido, de impresora compartida, etc., además de *otras* unidades organizativas. Es decir, mediante unidades organizativas podemos crear una *jerarquía* de objetos en el directorio (lo cual se asemeja otra vez a un sistema de archivos típico de Windows). Los objetos ubicados dentro de una unidad organizativa pueden moverse más tarde a otra, si fuera necesario. Sin embargo, un objeto no puede *copiarse*: cada objeto es único en el directorio, y su existencia es independiente de la unidad organizativa a la que pertenece.

Por tanto, el objetivo de las unidades organizativas es *estructurar* u organizar el conjunto de los objetos del directorio, agrupándolos de foma coherente. En el Directorio Activo, las unidades organizativas permiten:

1. **Delegar la administración**. Cada unidad organizativa puede administrarse de forma independiente. En concreto, se puede otorgar la administración total o parcial de una unidad organizativa a un usuario o grupo de usuarios cualquiera. Esto permite *delegar* la administración de subconjuntos estancos del dominio a ciertos usuarios que posean el nivel de responsabilidad adecuada.

2. **Establecer de forma centralizada comportamientos distintos a usuarios y equipos**. A cada unidad organizativa pueden vincularse políticas de grupo, que aplican comportamientos (generalmente en forma de restricciones) a los usuarios y equipos cuyas cuentas se ubican en dicha unidad. De esta forma, podemos aplicar restricciones distintas a subconjuntos de usuarios y equipos del dominio, en función exclusivamente de la unidad organizativa donde se ubican. Por ejemplo, podemos limitar a los usuarios del departamento de contabilidad para que sólo puedan utilizar ciertas aplicaciones, pero que esto no se aplique a los usuarios del departamento de informática.

En muchos sentidos, el concepto de unidad organizativa se puede utilizar en Windows 2003 de la misma forma que se entendía el concepto de dominio en versiones anteriores de Windows NT, es decir, conjunto de usuarios, equipos y recursos administrados independientemente. En realidad, en Windows 2003 el concepto de dominio viene más bien asociado a la distribución de los sitios (topología de red) y a la implementación de DNS que exista (o quiera crearse) en la empresa.

De este modo, en muchas organizaciones de pequeño o medio tamaño resulta más adecuado implementar un modelo de dominio único con múltiples unidades organizativas que un modelo de múltiples dominios. Si es necesario, cada unidad puede administrarse independientemente, con uno o varios administradores delegados y comportamientos (políticas) diferentes.

1.2.5. Estructura física

En Active Directory, la estructura lógica está separada de la estructura física. La estructura lógica se utiliza para organizar los recursos de red mientras que la estructura física se utiliza para configurar y administrar el tráfico de red. En concreto, la estructura física de Active Directory se compone de sitios y controladores de dominio.

La estructura física de Active Directory define dónde y cuándo se producen el tráfico de replicación y de inicio de sesión. Una buena comprensión de los componentes físicos de Active Directory permite optimizar el tráfico de red y el proceso de inicio de sesión, así como solventar problemas de replicación.

1.2.5.1. Sitios

Un sitio es una combinación de una o varias subredes IP que están conectadas por un vínculo de alta velocidad. Definir sitios permite configurar la topología de replicación y acceso a Active Directory de forma que Windows 2003 utilice los vínculos y programas más efectivos para el tráfico de inicio de sesión y replicación.

Normalmente los sitios se crean por dos razones principalmente:

- Para optimizar el tráfico de replicación.
- Para permitir que los usuarios se conecten a un controlador de dominio mediante una conexión confiable de alta velocidad.

Es decir, los sitios definen la estructura física de la red, mientras que los dominios definen la estructura lógica de la organización.

1.2.5.2. Controladores de dominio

Un controlador de dominio (*Domain Controller*, DC) es un equipo donde se ejecuta Windows 2003 Server y que almacena una replica del directorio. Los controladores de dominio ejecutan el servicio KDC, que es responsable de autenticar inicios de sesión de usuario.

La información almacenada en cada controlador de dominio se divide en tres categorías (particiones): dominio, esquema y datos de configuración. Estas particiones del directorio son las unidades de replicación:

1. **Partición del directorio de esquema**: contiene todos los tipos de objetos y atributos que pueden ser creados en Active Directory. Estos datos son comunes a todos los dominios en el bosque. Por tanto los datos del esquema se replican a todos los controladores de dominio del bosque.
2. **Partición de directorio de configuración**: contiene la estructura de los dominios y la topología de replicación. Estos datos son comunes a todos los dominios en el bosque, y se replican a todos los controladores de dominio en el bosque.
3. **Partición de directorio de dominio**: contiene todos los objetos del directorio para este dominio. Dichos datos se replican a todos los controladores de ese dominio, pero no a otros dominios.
4. **Partición de directorio de aplicaciones**: contiene datos específicos de aplicación. Estos datos pueden ser de cualquier tipo excepto *principales de seguridad* (usuarios, grupos y equipos). En este caso, se tiene un control fino sobre el ámbito de la replicación y la ubicación de las réplicas. Este tipo de partición es nuevo de Windows Server 2003.

Además de estas cuatro particiones de directorio de escritura, existe una cuarta categoría de información almacenada en un controlador de dominio: el catálogo global. Un catálogo global es un controlador de dominio que almacena las particiones de directorio de escritura, así como copias parciales de sólo lectura de todas las demás particiones de directorio de dominio del bosque.

1.2.5.3. Funciones de los controladores de dominio

Las versiones anteriores de Windows NT Server usaban múltiples controladores de dominio y sólo se permitía que uno de ellos actualizase la base de datos del directorio. Este esquema de maestro único exigía que todos los cambios se replicasen desde el controlador de dominio principal (*Primary Domain Controller*, PDC) a los controladores de dominio secundarios o de reserva (*Backup Domain Controllers*, BDCs).

En Windows 2003, todos los controladores de dominio admiten cambios, y estos cambios se replican a todos los controladores de dominio. Las operaciones de administración de usuarios, grupos y equipos son operaciones típicas de múltiples maestros. Sin embargo no es práctico que algunos cambios se realicen en múltiples maestros debido al tráfico de replicación y a los posibles conflictos en las operaciones básicas. Por estas razones, las funciones especiales, como la de servidor de catálogo global y operaciones de maestro único, se asignan sólo a determinados controladores de dominio. A continuación veremos estas funciones.

1.2.5.4. Servidor de catálogo global

El *catálogo global* es un depósito de información que contiene un subconjunto de atributos para todos los objetos de Active Directory (partición de directorio de dominio). Los atributos que se almacenan en el catálogo global son los que se utilizan con más frecuencia en las consultas. El catálogo global contiene la información necesaria para determinar la ubicación de cualquier objeto del directorio.

Un servidor de catálogo global es un controlador de dominio que almacena una copia del catálogo y procesa las consultas al mismo. El primer controlador de dominio que se crea en Active Directory es un servidor de catálogo global. Se pueden configurar controladores de dominio adicionales para que sean servidores de catálogo global con el fin de equilibrar el tráfico de autenticación de inicios de sesión y la transferencia de consultas.

El catálogo global cumple dos funciones importantes en el directorio:

- Permite que un usuario inicie una sesión en la red mediante el suministro de la información de pertenencia a grupos universales a un controlador de dominio cuando inicia un proceso de sesión.
- Permite que un usuario busque información de directorio en todo el bosque, independiente de la ubicación de los datos.

1.2.5.5. Operaciones de maestro único

Un maestro de operaciones es un controlador de dominio al que se le ha asignado una o varias funciones de maestro único en un dominio o bosque de Active Directory. Los controladores de dominio a los que se les asignan estas funciones realizan operaciones que no pueden ocurrir simultáneamente en otros controladores de dominio de la red. La propiedad de estas operaciones de maestro único puede ser transferida a otros controladores de dominio.

Todos los bosques de Active Directory deben tener controladores de dominio que cumplan dos de las cinco funciones de operaciones de maestro único. Las funciones para todo el bosque son:

- **Maestro de esquema**. El controlador de dominio maestro de esquema controla todas las actualizaciones y modificaciones del esquema. Para actualizar el esquema de un bosque, debe tener acceso al maestro de esquema.
- **Maestro de nombres de dominio**. El controlador de dominio maestro de esquema controla las operaciones de agregar o quitar dominios del bosque, asegurando que los nombres de dominio sean únicos en el bosque.

Todos los dominios de Active Directory deben tener controladores de dominio que cumplan tres de las cinco funciones de operaciones de maestro único:

- **Maestro de identificadores relativos (RID)**. El controlador de dominio maestro de identificadores relativos (RID) asigna secuencias de identificadores relativos a cada uno de los distintos controladores de su dominio.

  Cuando un controlador de dominio crea un objeto de usuario, grupo o equipo, asigna al objeto un identificador de seguridad único (SID). Este identificador está formado por un identificador de seguridad de dominio, que es el mismo para todos los que se crean en el dominio, y un identificador relativo que es único para cada identificador de seguridad que se crea en el dominio.
- **Emulador de controlador de dominio principal (PDC)**. Para mantener la compatibilidad con servidores basados en Windows NT que puedan funcionar como controladores de dominio de reserva (BDC) en dominios de Windows 2003 en modo mixto, pero todavía requieren un controlador principal de dominio (PDC), se asigna a un controlador de dominio específico basado en Windows 2003, la función de emular a un PDC. A este controlador de dominio lo ven los servidores basados en NT como un PDC.
- **Maestro de infraestructuras**. Cuando los objetos se mueven o se eliminan, un controlador de dominio de cada dominio, el maestro de infraestructura, es el responsable de actualizar los identificadores de seguridad y nombres completos en las referencias de objetos de dominio cruzado de ese dominio.

1.3. Objetos que administra un dominio

El Directorio Activo, tal como se ha visto en capítulos anteriores, es en realidad una base de datos jerárquica de *objetos*, que representan las entidades que pueden administrarse en una red de ordenadores, o, más correctamente en nuestro caso, en un *dominio* de sistemas Windows 2003. Esta base de datos de objetos de administración es compartida, para consulta, por todos los ordenadores miembros del dominio y, para modificación, por todos los controladores del dominio (o DC, *Domain Controllers*).

Por tanto, en Windows 2003, la gestión de un dominio puede realizarse de forma centralizada, administrando únicamente el Directorio Activo. En este contexto, "administrar" significa crear y configurar adecuadamente los objetos del directorio que representan a las entidades o *recursos* que existen en el dominio (recursos como usuarios, grupos, equipos, etc.).

Este apartado expone con detalle los principales tipos de objetos que pueden crearse en el Directorio Activo de Windows 2003, planteando en cada caso sus opciones de configuración y su utilidad dentro de la administración del dominio.

1.3.1. Usuarios globales

En la administración de sistemas Windows 2003 independientes (administración local), se crean en los sistemas cuentas de usuario y de grupo que sirven para:

1. identificar y autentificar a las personas (usuarios) que deben poder acceder al sistema, y
2. administrar los permisos y derechos que permitirán aplicar el control de acceso adecuado a dichos usuarios en el sistema.

Por lo tanto, utilizando únicamente protección local, si una persona debe trabajar en varios ordenadores, necesita poseer una cuenta de usuario en cada uno de ellos. A continuación explicaremos una alternativa a esto.

En un dominio Windows 2003, cualquier servidor que actúa como DC puede crear cuentas de *usuario global*. En este caso, el término "global" debe interpretarse como *global al dominio*. Los datos de una cuenta de usuario global se almacenan en el Directorio Activo y por tanto son conocidos por todos los ordenadores del dominio (en realidad, por todos los ordenadores de *bosque*). Em otras palabras, no es que se cree una cuenta para ese usuario en cada ordenador miembro, sino que existe una *única* cuenta (con un único SID) que es visible en todos los ordenadores del dominio. En este caso, cuando una persona se conecta a cualquiera de dichos ordenadores utilizando para ello su cuenta de usuario global, el ordenador en cuestión realiza una consulta al Directorio Activo (i.e., a alguno de los DCs) para que se validen las credenciales del usuario. El resultado de la validación es enviado al ordenador miembro (y de éste al usuario), concediendo o rechazando la conexión.

Los ordenadores miembros de un dominio que no sean DCs, además de conocer a los usuarios globales del dominio, pueden crear también sus propios usuarios *locales*. En este caso, estos usuarios son únicamente visibles en el ordenador en el que han sido creados. Cuando una persona desea entrar en el sistema utilizando una cuenta local, dicha cuenta se valida contra la base de datos local de ese ordenador. Además, es importante resaltar que a dicho usuario local no se le pueden asignar permisos sobre recursos que residan en otro sistema Windows 2003 (puesto que allí no existe). Por el contrario, a un usuario global se le pueden conceder permisos sobre cualquier recurso (archivo, directorio, impresora, etc.) de cualquier ordenador miembro del dominio, puesto que es visible (y posee el mismo SID) en todos ellos.

1.3.2. Grupos

De forma análoga a los usuarios globales, existen *grupos* que son almacenados en el Directorio Activo y que por tanto son visibles desde todos los ordenadores del dominio (y, en algunos casos, también de otros dominios del bosque). En el directorio pueden crearse dos tipos de grupos: grupos de distribución y grupos de seguridad. Los primeros se utilizan exclusivamente para crear listas de distribución de correo electrónico, mientras que los segundos son los que se utilizan con fines administrativos. Por este motivo, a partir de ahora nos referiremos exclusivamente a los grupos de seguridad.

En concreto, en dominios Windows 2003 se definen tres clases de grupos de seguridad (o, de forma más precisa, se pueden definir grupos de tres *ámbitos* distintos):

1. **Grupos locales del dominio**. En un dominio en nivel funcional Windows 2000 mixto, pueden contener cuentas de usuario y grupo globales de cualquier dominio del bosque. En un dominio en nivel Windows 2000 nativo o Windows Server 2003, pueden contener, además, grupos universales y otros grupos locales del dominio. Sólo son visibles en el dominio en que se crean, y suelen utilizarse para conceder permisos y derechos en cualquiera de los ordenadores del dominio (nótese que en modo mixto, sólo son visibles por los DCs del dominio, y por tanto sólo se pueden utilizar para administrar permisos y derechos en esos ordenadores).

2. **Grupos globales**. En un dominio en nivel funcional Windows 2000 mixto, pueden contener cuentas de usuario globales del mismo dominio. En un dominio en nivel Windows 2000 nativo o Windows Server 2003, pueden contener, además, otros grupos globales del mismo dominio. Son visibles en todos los dominios del bosque, y suelen utilizarse para clasificar a los usuarios en función de las labores que realizan.

3. **Grupos universales**. Sólo están disponibles en dominios en nivel funcional Windows 2000 nativo o Windows Server 2003 nativo. Pueden contener cuentas de usuario y grupos globales, así como otros grupos universales, de cualquier dominio del bosque. Son visibles en todo el bosque.

En un ordenador miembro de un dominio también se pueden definir grupos locales. Los grupos locales pueden estar formados por cuentas de usuario locales y usuarios y grupos globales de cualquier dominio del bosque (en modo mixto) y además por grupos universales (en modo nativo). Un grupo local no puede ser miembro de otro grupo local. Los grupos locales pueden utilizarse para conceder permisos y derechos en el equipo en que son creados.

Por tanto, la administración de la protección en cada ordenador del dominio puede realizarse mediante grupos locales del dominio o grupos locales del equipo en que reside el recurso a administrar. Por tanto, la recomendación que se hacía en la protección local respecto a la asignación de permisos en base a grupos locales sigue siendo válida. En el caso más general, la regla que recomienda Windows 2003 es la siguiente:

1. Asignar usuarios globales a grupos globales, según las labores que desempeñen en la organización.

2. Incluir (usuarios y/o) grupos globales en grupos locales (del equipo o del dominio) según el nivel de acceso que vayan a tener.

3. Asignar permisos y derechos únicamente a estos grupos locales (del equipo o del dominio).

La utilización de grupos universales tiene sentido sólo cuando un mismo conjunto de usuarios (y/o grupos) de varios dominios deben recibir permisos/derechos en varios dominios simultáneamente. En Windows 2000, el uso de este tipo de grupo estaba muy desaconsejado, por dos motivos: primero, porque estos grupos y sus miembros deben replicarse en todos los catálogos globales del bosque, y segundo, porque cualquier incio de sesión de un usuario debe consultar a un catálogo global para determinar la pertenencia de dicho usuario a posibles grupos universales. Windows 2003 ha suavizado el primero de los problemas, mejorado la eficiencia de esa replicación (sólo si el bosque está en nivel funcional Windows Server 2003), de forma que su uso no está ahora tan desaconsejado como lo estaba en Windows 2000. En cualquier caso, el uso de un grupo universal siempre puede simularse con la combinación adecuada de grupos globales y locales del dominio. Se recomienda al lector reflexionar sobre cómo podría hacerse.

En relación con esto, es importante saber que cuando un ordenador pasa a ser miembro de un dominio, el grupo global `Administradores del dominio` se incluye automáticamente en el grupo local `Administradores` de dicho ordenador. De igual forma, el grupo global `Usuarios del dominio` se incluye dentro del grupo local `Usuarios`. De esta forma, los administradores y usuarios normales del dominio tienen en cada miembro los mismos derechos y permisos que los que tengan ya definidos los administradores y usuarios locales, respectivamente. El administrador local puede, si lo desea, invalidar esta acción automática, extrayendo posteriormente los grupos globales de los locales.

1.3.3. Equipos

Como hemos visto, en el Directorio Activo de un dominio se conserva toda la información relativa a cuentas de usuarios y grupos globales. Esta misma base de datos de directoio recoge también una *cuenta de equipo* por cada uno de los ordenadores miembro de un dominio.

Entre otras informaciones, en cada una de estas cuentas se almacena el nombre del ordenador, así como un identificador único y privado que lo identifica unívocamente. Este identificador es análogo al SID de cada cuenta de usuario o grupo, y sólo lo conocen los DCs y el propio ordenador miembro. Es por tanto, un dato interno del sistema operativo, y ni siquiera el administrador puede cambiarlo. Es precisamente este dato, propio de las cuentas de usuario, grupo y equipo, lo que permite asignar permisos y derechos en los sistemas a estos tres tipos de cuentas. Por este motivo, se denominan *principales de seguridad* (*security principals*). Por tanto, la asignación de derechos y permisos (NTFS) a cuentas de equipo es posible, pero se limita a situaciones muy poco frecuentes y está fuera del ámbito de este texto.

Windows 2003 puede utilizar distintos protocolos de comunicaciones seguros entre los ordenadores miembro de un dominio y los DCs. Entre ellos los más importantes son NTLM (el protocolo utilizado por versiones anteriores de Windows NT, que se mantiene por com-

patibilidad hacia atrás) y Kerberos V5. Kerberos presenta numerosas ventajas respecto a NTLM, pero sólo es viable en la práctica si todas las máquinas del dominio son Windows 2000, Windows XP o Windows Server 2003. Estos protocolos se utilizan siempre que información relativa a aspectos de seguridad se intercambia entre sistemas pertenecientes a algún dominio y, en concreto, para autenticar usuarios (como se ha explicado arriba).

1.3.4. Unidades Organizativas

Como hemos visto en Sección 1.2.4.5, "Unidades Organizativas", las unidades organizativas son objetos del directorio que a su vez, pueden contener otros objetos. El uso fundamental de las OUs es delegar la administración de sus objetos a otros usuarios distintos del administrador del dominio, y personalizar el comportamiento de los usuarios y/o equipos mediante la aplicación de directivas de grupo (GPOs) específicas a la unidad.

1.4. Compartición de recursos

Cuando un sistema Windows 2003 participa en una red (grupo de trabajo o dominio), puede compartir sus recursos con el resto de ordenadores. En este contexto, sólo vamos a considerar como recursos a compartir las *carpetas* o directorios que existen en un sistema Windows. La compartición de otros recursos (tales como impresoras, por ejemplo) queda fuera del ámbito de este texto.

1.4.1. Permisos y derechos

Cualquier sistema Windows 2003 puede compartir carpetas, tanto si es un servidor como si es una estación de trabajo. Para poder compartir una carpeta basta con desplegar su menú contextual desde una ventana o desde el explorador de archivos, y seleccionar `Compartir...`. En la ventana asociada a esta opción se determina el nombre que tendrá el recurso (que no tiene por qué coincidir con el nombre de la propia carpeta), así como qué usuarios van a poder acceder al mismo. En relación con esto, existe una gran diferencia entre que el directorio resida en una partición FAT y que resida en una NTFS.

Si la carpeta reside en una partición FAT, este filtro de acceso será el único que determine los usuarios que van a poder acceder al contenido de la carpeta, puesto que no es posible determinar permisos sobre la misma o sus archivos. Es decir, el filtro sólo se establece para poder acceder al recurso. Si un usuario tiene permisos suficientes para conectarse a un recurso, tendrá acceso sobre todos los archivos y subcarpetas del recurso. Concretamente, el tipo de acceso sobre todos ellos será el que le permita el permiso sobre el recurso (`Lectura`, `Escritura` o `Control Total`).

Por el contrario, si la carpeta se encuentra en una partición NTFS, ésta tendrá unos permisos establecidos (así como sus subcarpetas y archivos), al margen de estar o no compartida. En este caso también es posible establecer permisos desde la ventana de `Compartir...`, pero entonces sólo los usuarios que puedan pasar *ambos* filtros podrán acceder a la carpeta compartida y a su contenido. En este caso se recomienda dejar `Control Total` sobre `Todos` en los permisos asociados al recurso (opción por defecto), y controlar quién (y cómo)

puede acceder al recurso y a su contenido mediante los permisos asociados a dicha carpeta (y a sus archivos y subcarpetas). Sin embargo, esta no es la opción por defecto en Windows Server 2003 (aunque sí lo era en Windows 2000), que sólo concede inicialmente el permiso de lectura al grupo `Todos` al compartir una carpeta.

Esta recomendación es muy útil, si tenemos en cuenta que de esta forma para cada carpeta (y archivo) del sistema no utilizamos dos grupos de permisos sino uno solo, independientemente de que la carpeta sea o no compartida. Este forma de trabajar obliga al administrador a asociar los permisos correctos a cada objeto del sistema (aunque no esté compartido), pero por otra parte se unifica la visión de la seguridad de los archivos, con lo que a la larga resulta más segura y más sencilla.

Cuando compartimos recursos a otros usuarios en la red (especialmente en un dominio) hay que tener en cuenta no sólo los permisos del recurso y su contenido, sino también los *derechos* del ordenador que comparte el recurso. En concreto, si un usuario ha iniciado una sesión interactiva en un ordenador Windows 2003 denominado A, y desea conectarse a un recurso de red que exporta otro Windows 2003 denominado B, además de poseer suficientes permisos (sobre el recurso, sobre el propio carpeta y sobre su contenido), tiene que tener concedido en B el derecho `Acceder a este equipo desde la red`. De lo contrario, dicho usuario ni siquiera podrá obtener la lista de los recursos que el ordenador B comparte.

1.4.2. Compartición dentro de un dominio

Cuando la compartición de recursos la realizan equipos que forman parte de un dominio Windows 2003, existen consideraciones que el administración debe conocer.

Primero, una vez se ha compartido físicamente una carpeta en la red (según el procedimiento descrito arriba), el administrador del dominio puede además *publicar* este recurso en el directorio. Para ello debe crear un nuevo objeto, en la unidad organizativa adecuada, de tipo `Recurso compartido`. A este objeto se le debe asociar un nombre simbólico y el nombre de recurso de red que representa (de la forma `\\equipo\recurso`). Es importante tener en cuenta que cuando se publica el recurso de esta forma, no se comprueba si realmente existe o no, por lo que es responsabilidad del administrador el haberlo compartido y que su nombre coincida con el de la publicación. Una vez publicado, el recurso puede localizarse mediante búsquedas en el Directorio Activo, como el resto de objetos del mismo. Windows Server 2003 ha introducido otra forma de publicación: entre las opciones de la pestaña "Compartir" del explorador de archivos, puede publicarse dicha compartición en el directorio, simplemente asignándole un nombre simbólico. Si se publica de esta forma, el proceso crea automáticamente el objeto que representa la carpeta compartida en el directorio.

Y segundo, cuando un sistema Windows 2003 se agrega a un dominio, los siguientes recursos se comparten de forma automática y por defecto (estas comparticiones no deben modificarse ni prohibirse):

- `letra_de_unidad$`. Por cada partición existente en el sistema Windows 2003 (`C:`, `D:`, etc.) se crea un recurso compartido denominado `C$`, `D$`, etc. Los administradores del dominio, así como los operadores de copia del domino, pueden conectarse por defecto a estas unidades.

- `ADMIN$`. Es un recurso utilizado por el propio sistema durante la administración remota de un ordenador Windows 2003.
- `IPC$`. Recurso que agrupa los tubos (colas de mensajes) utilizados por los programas para comunicarse entre ellos. Se utiliza durante la administración remota de un ordenador Windows 2003, y cuando se observa los recursos que comparte.
- `NETLOGON`. Recurso que exporta un DC para proporcionar a los ordenadores miembros del dominio el servicio de validación de cuentas globales a través de la red (*Net Logon service*).
- `SYSVOL`. Recurso que exporta cada DC de un dominio. Contiene información del Directorio Activo (por ejemplo, de directivas de grupo) que debe replicarse en todos los DCs del dominio.

En relación con los nombres de estos recursos, es interesante saber que añadir el carácter "`$`" al final de cualquier nombre de recurso tiene un efecto específico: prohibe que dicho recurso se visualice dentro de la lista de recursos que una máquina exporta al resto. Es decir, convierte un recurso en "invisible" para al resto del mundo. En este caso, un usuario remoto sólo podrá conectarse al recurso si conoce su nombre de antemano (y tiene suficientes permisos, obviamente).

1.4.3. Mandatos Windows 2003 para compartir recursos

La compartición de recursos en Windows 2003 puede realizarse en línea de órdenes utilizando los mandatos **net share** y **net use**. La sintaxis de ambos mandatos es la siguiente:

1. Mandato **net share**: Crea, elimina o muestra recursos compartidos.

```
net share
net share recurso_compartido
net share recurso_compartido=unidad:ruta_de_acceso
          [/users:número | /unlimited] [/remark:"texto"]
net share recurso_compartido [/users:número | unlimited]
          [/remark:"texto"]
net share {recurso_compartido | unidad:ruta_de_acceso} /delete
```

2. Mandato **net use**: Conecta o desconecta un equipo de un recurso compartido o muestra información acerca de las conexiones del equipo. También controla las conexiones de red persistentes.

```
net use  [nombre_dispositivo]
         [\\nombre_equipo\recurso_compartido[\volumen]]
         [contraseña | *]] [/user:[nombre_dominio\]nombre_usuario]
         [[/delete] | [/persistent:{yes | no}]]
net use nombre_dispositivo [/home[contraseña | *]]
         [/delete:{yes | no}]
net use  [/persistent:{yes | no}]
```

1.5. Delegación de la administración

Para delegar, total o parcialmente, la administración de una unidad organizativa existe un asistente (*wizard*) que aparece cuando se selecciona la acción `Delegar el control...` en el menú contextual de la unidad organizativa. Este asistente pregunta básicamente los dos aspectos propios de la delegación: *a quién* se delega y *qué* se delega. La primera pregunta se contesta o bien con un usuario o con un grupo (se recomienda un grupo). Para responder a la segunda pregunta, se puede elegir una tarea *predefinida* a delegar (de entre una lista de tareas frecuentes), o bien podemos optar por construir una tarea personalizada. En este último caso, tenemos que especificar la tarea mediante un conjunto de permisos sobre un cierto tipo de objetos del directorio. Esto se explica a continuación.

Internamente, los derechos de administración (o control) sobre un dominio o unidad organizativa funcionan de forma muy similar a los permisos sobre una carpeta o archivo: existe una DACL propia y otra heredada, que contienen como entradas aquellos usuarios/grupos que tienen concedida (o denegada) una cierta acción sobre la unidad organizativa o sobre su contenido. En este caso, las acciones son las propias de la administración de objetos en el directorio (control total, creación de objetos, modificación de objetos, consulta de objetos, etc.), donde los "objetos" son las entidades que pueden ser creados dentro de la unidad: usuarios, grupos, unidades organizativas, recursos, impresoras, etc.

En resumen, la delegación de control sobre una unidad organizativa puede hacerse de forma completa (ofreciendo el *Control Total* sobre la unidad) o de forma parcial (permitiendo la lectura, modificación y/o borrado de los objetos de la misma). Hay que tener en cuenta que en el caso de la delegación parcial, el número de posibilidades es inmenso: por una parte, se incluye la posibilidad de establecer el permiso sobre cada *atributo* de cada tipo de objeto posible; por otra parte, se puede establecer a qué unidades se va a aplicar la regla (sólo en esa unidad organizativa, en todas las que se sitúan por debajo, en parte de ellas, etc.). Por tanto, para una delegación parcial se recomienda el uso del asistente, ya que su lista de delegación de tareas más frecuentes (como por ejemplo "Crear, borrar y administrar cuentas de usuario" o "Restablecer contraseñas en cuentas de usuario") resulta muy útil. Sin embargo, cuando la delegación que buscamos no se encuentra en la lista, tendremos que diseñar una a medida, asignando los permisos oportunos sobre los objetos del directorio que sean necesarios.

2

Administración de dominios en Linux

Indice

2.1. Introducción

En el mundo Linux no existe un concepto de dominio tan elaborado como en el mundo de Windows 2003. Sin embargo, se consigue un efecto similar al activar un servicio en una máquina Linux (que actuaría como "servidor" de cuentas y grupos) y otro servicio que permite la exportación de directorios a máquinas remotas. En concreto, dichos servicios se denominan *LDAP* y *NFS*, respectivamente. Ambos son explicados en este capítulo y en el siguiente, respectivamente.

2.2. Concepto de dominio

Desde el punto de vista de la administración de sistemas, suele denominarse *dominio* a un conjunto de equipos interconectados que comparten información administrativa (usuarios, grupos, contraseñas, etc.) centralizada. Ello requiere fundamentalmente la disponibilidad de (al menos) un ordenador que almacene físicamente dicha información y que la comunique al resto cuando sea necesario, típicamente mediante un esquema cliente-servidor. Por ejemplo, cuando un usuario desea iniciar una conexión interactiva en cualquiera de los ordenadores (clientes) del dominio, dicho ordenador deberá validar las credenciales del usuario en el servidor, y obtener de éste todos los datos necesarios para poder crear el contexto inicial de trabajo para el usuario.

En Windows 2003, la implementación del concepto de dominio se realiza mediante el denominado *Directorio Activo*, un servicio de directorio basado en diferentes estándares como LDAP (Lightweight Directory Access Protocol}) y DNS (*Domain Name System*). En el mundo Unix, los dominios solían implementarse mediante el famoso *Network Information System* (NIS), del que existían múltiples variantes. Sin embargo, la integración de servicios de directorio en Unix ha posibilitado la incorporación de esta tecnología, mucho más potente y escalable que NIS, en la implementación de dominios.

Este capítulo describe cómo una implementación libre del protocolo LDAP para Unix, denominada *OpenLDAP* (`www.openldap.org`), puede utilizarse para implementar dominios en Centos Linux.

2.3. Servicios de directorio y LDAP

En el contexto de las redes de ordenadores, se denomina *directorio* a una base de datos especializada que almacena información sobre los recursos, u "objetos", presentes en la red (tales como usuarios, ordenadores, impresoras, etc.) y que pone dicha información a disposición de los usuarios de la red. Por este motivo, esta base de datos suele estar optimizada para operaciones de búsqueda, filtrado y lectura más que para operaciones de inserción o transacciones complejas. Existen diferentes estándares que especifican servicios de directorio, siendo el demominado *X.500* tal vez el más conocido.

El estándar X.500 define de forma nativa un protocolo de acceso denominado DAP (*Directory Access Protocol*) que resulta muy complejo (y computacionalmente pesado) porque está definido sobre la pila completa de niveles OSI. Como alternativa a DAP para acceder a di-

rectorios de tipo X.500, LDAP (*Lightweight Directory Access Protocol*) ofrece un protocolo "ligero" casi equivalente, pero mucho más sencillo y eficiente, diseñado para operar directamente sobre TCP/IP. Actualmente, la mayoría de servidores de directorio X.500 incorporan LDAP como uno de sus protocolo de acceso.

LDAP permite el acceso a la información del directorio mediante un esquema cliente-servidor, donde uno o varios servidores mantienen la misma información de directorio (actualizada mediante réplicas) y los clientes realizan consultas a cualquiera de ellos. Ante una consulta concreta de un cliente, el servidor contesta con la información solicitada y/o con un "puntero" donde conseguir dicha información o datos adicionales (normalmente, el "puntero" es otro servidor de directorio).

Internamente, el modelo de datos de LDAP (derivado de X.500, pero algo restringido) define una estructura jerárquica de objetos o *entradas* en forma de árbol, donde cada objeto o entrada posee un conjunto de atributos. Cada atributo viene identificado mediante un *nombre* o acrónimo significativo, pertenece a un cierto *tipo* y puede tener uno o varios *valores* asociados. Toda entrada viene identificada unívocamente en la base de datos del directorio mediante un atributo especial denominado *nombre distinguido* o `dn` (*distinguished name*). El resto de atributos de la entrada depende de qué objeto esté describiendo dicha entrada. Por ejemplo, las entradas que describen personas suelen tener, entre otros, atributos como `cn` (*common name*) para describir su nombre común, `sn` (*surname*) para su apellido, `mail` para su dirección de correo electrónico, etc. La definición de los posibles tipos de objetos, así como de sus atributos (incluyendo su nombre, tipo, valor(es) admitido(s) y restricciones), que pueden ser utilizados por el directorio de un servidor de LDAP la realiza el propio servidor mediante el denominado *esquema* del directorio. Es decir, el esquema contiene las definiciones de los objetos que pueden darse de alta en el directorio.

El nombre distinguido de cada entrada del directorio es una cadena de caracteres formada por pares `<tipo_atributo>=<valor>` separados por comas, que representa la ruta invertida que lleva desde la posición lógica de la entrada en el árbol hasta la raíz del mismo. Puesto que se supone que un directorio almacena información sobre los objetos que existen en una cierta organización, cada directorio posee como raíz (o *base*, en terminología LDAP) la ubicación de dicha organización, de forma que la base se convierte de forma natural en el *sufijo* de los nombres distinguidos de todas las entradas que mantiene el directorio. Existen dos formas de nombrar, o estructurar, la raíz de un directorio LDAP:

1. Nombrado "tradicional": formado por el país y estado donde se ubica la organización, seguida por el nombre de dicha organización. Por ejemplo, la raíz o base de la Universidad Politécnica de Valencia podría ser algo así: `"o=UPV, st=Valencia, c=ES"`.

2. Nombrado basado en nombres de dominio de Internet (es decir, en DNS): este nombrado utiliza los dominios DNS para nombrar la raíz de la organización. En este caso, la base de la UPV sería: "dc=upv, dc=es". Este es el nombrado que vamos a utilizar puesto que permite localizar a los servidores de LDAP utilizando búsquedas DNS.

A partir de esa base, el árbol se subdivide en los nodos y subnodos que se estime oportuno para estructurar de forma adecuada los objetos de la organización, objetos que se ubican finalmente como las hojas del árbol. De esta forma, el nombre distinguido de cada entrada

describe su posición en el árbol de la organización (y vice-versa), de forma análoga a un sistema de archivos típico, en el que el nombre absoluto (unívoco) de cada archivo equivale a su posición en la jerarquía de directorios del sistema, y vice-versa. En la Figura 2.1, "Estructura de directorio del dominio admon.com." se muestra un ejemplo de un directorio sencillo (dos usuarios y dos equipos) de la organización `admon.com`.

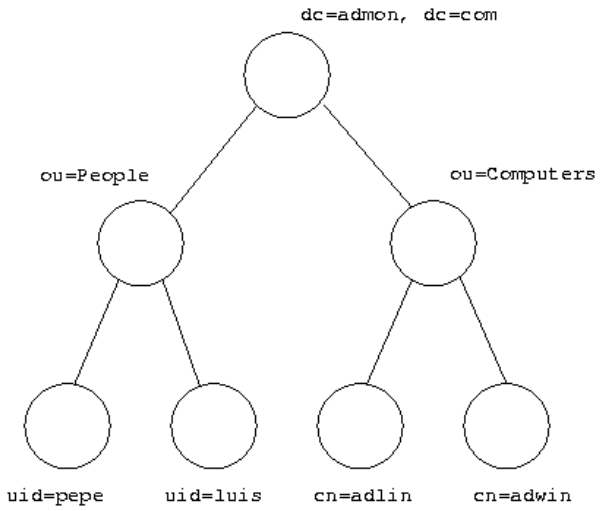

Figura 2.1. Estructura de directorio del dominio `admon.com`.

De acuerdo con dicha figura, la entrada correspondiente al usuario "pepe" tendría como nombre distinguido "`uid=pepe, ou=People, dc=admon, dc=com`". Al margen de ese identificador único, cada entrada u objeto en el directorio puede tener, como hemos dicho, un conjunto de atributos tan descriptivo como se desee. Cada objeto necesita, al margen de su nombre distinguido, su "clase de objeto", que se especifica mediante el atributo `Object-Class` (un mismo objeto puede pertenecer a diferentes clases simultáneamente, por lo que pueden existir múltiples atributos de este tipo para un mismo objeto en el directorio). Este atributo especifica implícitamente el resto de atributos de dicho objeto, de acuerdo con la definición establecida en el esquema. Siguiendo con el ejemplo anterior, a continuación se muestra un subconjunto de los atributos del usuario "pepe":

```
dn: uid=pepe, ou=People, dc=admon, dc=com
objectClass: person
cn: Jose García
sn: García
description: alumno
mail: pepe@admon.com
```

El formato en el que se han mostrado los atributos del objeto se denomina LDIF (*LDAP Data Interchange Format*), y resulta útil conocerlo porque es el formato que los servidores LDAP (y OpenLDAP entre ellos) utilizan por defecto para insertar y extraer información del directorio.

Puesto que nosotros vamos a utilizar el directorio con un uso muy específico (centralizar la información administrativa de nuestro dominio para autentificar usuarios de forma glo-

bal) deberíamos asegurarnos que en el esquema de nuestro directorio existen los tipos de objetos (y atributos) adecuados para ello. Afortunadamente, OpenLDAP posee por defecto un esquema suficientemente rico para cubrir este cometido. Por ejemplo, cada usuario puede definirse mediante un objeto de tipo `posixAccount`, que posee entre otros atributos para almacenar su UID, GID, contraseña, etc. A título de curiosidad, la entrada en el esquema que define el atributo para la contraseña (`userPassword`) se muestra a continuación:

```
attributetype (2.5.5.35 NAME 'userPassword'
     EQUALITY octetStringMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{128})
```

2.4. Visión general de la implementación de un dominio Linux con OpenLDAP

La implementación de un dominio Linux utilizando OpenLDAP es una tarea algo compleja y requiere realizar varios pasos que se resumen a continuación.

- **Instalar el servidor OpenLDAP.**

  El primer paso consiste en elegir uno de los ordenadores de la red para que actue como servidor OpenLDAP, e instalar y configurar en dicho ordenador este servicio de red, tal como se describe en la sección Sección 2.5, “Instalación del servidor OpenLDAP”. Finalizado este paso se dispone de un directorio operativo pero carente de información.

- **Configurar clientes de OpenLDAP.**

  Antes de pasar a introducir información es conveniente verificar que el directorio es accesible desde las herramientas cliente de OpenLDAP, tal como se describe en la sección Sección 2.6, “Instalación de las herramientas cliente de OpenLDAP”. Es importante destacar que este paso sólo configura las herramientas para consultar y modificar la información del directorio. Configurar Linux para que utilice esta información para autenticar usuarios es una acción diferente que se realiza en el último paso.

- **Migrar la información actual de usuarios y grupos.**

  Una vez se ha comprobado el correcto funcionamiento del directorio LDAP, es el momento de introducir la información administrativa relativa a usuarios y grupos. Si esta información ya existe y está almacenada en los ficheros locales del ordenador donde se ejecuta el servicio OpenLDAP, es posible "copiarla" al directorio mediante las acciones que se describen en la sección Sección 2.7, “Migración de la información del servidor”.

- **Configurar la autenticación basada en OpenLDAP.**

  Finalmente, una vez se dispone de un directorio LDAP en el que reside la información necesaria relativa a usuarios y grupos (UIDs, GIDs, contraseñas, etc.), podemos configurar Linux para que utilice esta información para autenticar usuarios. Estas acciones están descritas en la sección Sección 2.8, “Autenticación basada en OpenLDAP”.

Otros aspectos interesantes a considerar en la configuración de OpenLDAP son la asignación de permisos y la utilización de varios servidores, descritos respectivamente en las secciones Sección 2.9, “Permisos de acceso” y Sección 2.10, “Configuración de OpenLDAP con varios servidores”.

Es importante resaltar que buena parte de la configuración que se describe a continuación es *dependiente* del software de LDAP concreto que se va a utilizar (en este caso, OpenLDAP) y de la versión de UNIX utilizada (Centos Linux). La configuración de otros servidores de LDAP, en la misma u otras versiones de Linux, puede ser distinta a la que aquí se va a exponer.

2.5. Instalación del servidor OpenLDAP

El paquete que incorpora el servidor de OpenLDAP se denomina `openldap-servers` y puede instalarse utilizando:

```
yum install openldap-servers
```

Una vez instalado, este paquete instala los ficheros de configuración por defecto bajo el directorio `/etc/openldap`, crea un directorio vacío denominado `/var/lib/ldap` para albergar la base de datos con la información del directorio y sus índices, y finalmente incorpora el servicio o "demonio" de LDAP, denominado **slapd**. Al igual que sucede con la mayoría de servicios, `sladp` puede iniciarse y detenerse utilizando el mandado `service`:

```
service ldap start | stop | restart
```

y puede ser configurado para activarse cuando se inicia el sistema utilizando la orden `chkconfig`:

```
bash# chkconfig --level 35 ldap on
```

La configuración del servicio `slapd` se realiza en `/etc/openldap/slapd.conf` fundamentalmente. Este fichero contiene referencias a los demás ficheros necesarios para el servicio `slapd` (como por ejemplo, las definiciones del esquema), y un conjunto de secciones donde se describen los directorios de los que se mantiene información. Es incluso posible almacenar los directorios en bases de datos de distintos formatos internos y definir opciones específicas diferentes para cada una. En el caso más sencillo, sin embargo, tendremos un único directorio almacenado en una base de datos en el formato por defecto (lbdm), cuyas opciones no modificaremos.

De hecho, de los numerosos parámetros que pueden configurarse en este fichero, sólo vamos a comentar los estrictamente necesarios para configurar un servidor básico de LDAP que luego haga de servidor de un dominio Linux. Para configurar cualquier otro parámetro avanzado, se recomienda la lectura previa del documento *”OpenLDAP Administrator’s Guide”* [http://www.openldap.org/doc/admin22/].

En definitiva, los tres parámetros fundamentales que es necesario configurar son los siguientes (el resto pueden dejarse con sus opciones por defecto):

1. **Sufijo**. Este es el sufijo de las entradas del directorio, es decir, lo que hemos denominado base o raíz del directorio. Por ejemplo, para el dominio "admon.com" deberíamos configurar:

```
suffix  "dc=admon, dc=com"
```

2. **Cuenta del administrador** (del directorio). Esta es la cuenta del usuario administrador del directorio, lógicamente en formato de LDAP. Las credenciales que se sitúan en esta opción (y la siguiente) tienen validez independientemente de que este usuario exista realmente en la base de datos del directorio o no. El nombre por defecto es "manager", pero si queremos lo podemos cambiar por "root" (a la UNIX):

```
rootdn  "cn=root, dc=admon, dc=com"
```

3. **Contraseña del administrador**. Cuando las operaciones a realizar en la base de datos no permitan una conexión anónima (sin acreditarse), y en concreto, cuando necesiten del usuario "administrador del directorio" definido arriba, dichas operaciones deben acompañarse de la contraseña de esta cuenta, que se especifica en la siguiente opción:

```
rootpw  <CONTRASEÑA>
```

   Hay que tener en cuenta que la contraseña que pongamos aquí será visible por cualquier usuario que pueda leer el fichero (aunque por defecto, éste es un permiso sólo de root). Por tanto, es conveniente ponerla cifrada. Para ello, podemos crear una contraseña nueva mediante la orden:

```
slappasswd -h {MD5}
```

   Esta orden nos pide una contraseña y nos la muestra cifrada. Luego simplemente sustitiumos <CONTRASEÑA> por el resultado de dicha orden. Como ejemplo, la sentencia rootpw quedaría como sigue:

```
rootpw  {MD5}EDtdFIXDSXRdagNOPSXPvcTBbA==
```

2.6. Instalación de las herramientas cliente de OpenLDAP

OpenLDAP incluye varias utilidades a nivel de cliente que permiten acceder a la información que almacenan los servidores LDAP. Entre estas utilidades se encuentran, por ejemplo, mandatos tales como `ldapsearch`, `ldapadd` y `ldapmodify`, que consultan, crean y modifican dicha información. El paquete que incorpora las herramientas cliente de OpenLDAP se

denomina `openldap-clients` y puede instalarse utilizando:

```
yum install openldap-clients
```

Para que estos mandatos funcionen correctamente es necesario configurar en el fichero `/etc/openldap/ldap.conf` dos opciones que describen el ordenador donde se encuentra instalado el directorio LDAP así como el sufijo común para las entradas de dicho directorio.

```
BASE dc=admon,dc=com
HOST adlin.admon.com
```

Es importante hacer notar que OpenLDAP aún no soporta la localización del servidor basada en registros de servicio de DNS (como es el caso del Directorio Activo de Windows 2003). Por tanto, en la opción `HOST` es necesario especificar o bien una dirección IP o bien un nombre de ordenador que pueda resolverse correctamente sin utilizar el propio directorio (por ejemplo, que esté dado de alta en el fichero `/etc/hosts` del ordenador cliente, o que pueda resolverse correctamente mediante una consulta DNS).

Una vez realizada la configuración del apartado anterior, ya estamos en condiciones de comprobar que, de momento, todo funciona correctamente.

Si el servicio ha arrancado correctamente, y a pesar de que actualmente el directorio está vacío, deberíamos ser capaces de preguntar al menos por un tipo de atributo denominado "namingContexts". Para ello, utilizamos la orden `ldapsearch`:

```
bash# ldapsearch -x -b '' -s base namingContexts
```

Si todo funciona bien, el resultado debería ser algo así:

```
# filter: (objectclass=*)
# requesting: namingContexts
#
dn:
namingContexts: dc=admon,dc=com

# search result
search: 2
result: 0 Success
```

Es imprescindible que en la salida por pantalla aparezca el sufijo del dominio (`dc=admon,dc=com` en el ejemplo).

También puede resultar útil comprobar si se ha configurado correctamente el "administrador" del directorio y su contraseña. Para ello puede ejecutarse la orden anterior acreditándose como el administrador utilizando las opciones `-D dn` (acreditarse como) y `-W` (solitar contraseña):

```
bash# ldapsearch -D "cn=root,dc=admon,dc=com" -W -x -b '' -s base \
namingContexts
```

2.7. Migración de la información del servidor

Este paso consiste en añadir al directorio, que aún está vacío, toda la información presente en el ordenador que está haciendo de servidor. Esto incluye todos los recursos dados de alta en sus ficheros de configuración, tales como cuentas de usuarios, grupos, ordenadores, etc., así como diferentes "contenedores" o "unidades organizativas" (*OrganizationalUnits*) que distribuyen los objetos de forma racional.

Para ello, OpenLDAP incorpora un conjunto de herramientas que pueden utilizarse para realizar esta migración de forma automática. El paso previo para ello es editar el fichero `/usr/share/openldap/migration/migrate_common.ph`. En concreto, tenemos que editar las dos directivas siguientes:

```
$DEFAULT_MAIL_DOMAIN = "admon.com";
$DEFAULT_BASE = "dc=admon,dc=com";
```

Una vez modificadas ambas, tenemos diferentes opciones para incorporar la información del sistema al directorio. Entre ellas, la más recomendable es realizar la migración por partes, añadiendo primero la "base" (es decir, las entradas correspondientes a la organización y sus unidades organizativas por defecto) y posteriormente migrando los usuarios, grupos, hosts, etc., que se ubicarán dentro de dichas unidades.

Para todo ello existen *scripts* de Perl en el directorio mencionado arriba (donde se ubica `migrate_common.ph`), que podemos utilizar de la siguiente forma (en todo el proceso, el servicio `ldap` debe estar ejecutándose):

- Para la migración de la base, se exporta primero sus objetos a un fichero, en formato LDIF, y luego se insertan en el directorio mediante la orden **ldapadd**:

  ```
  bash# ./migrate_base.pl > /root/base.ldif
  bash# ldapadd -x -c -D "cn=root,dc=admon,dc=com" -W -f /root/base.ldif
  ```

  Nótese que con la opción `-D` estamos acreditándonos, para la operación, como el usuario "administrador del directorio", y con la opción `-W` le decimos a la orden que nos pida la contraseña de dicho usuario de forma interactiva. La opción `-c` consigue que **ldapadd** siga insertando registros a pesar de que se produzcan errores en alguna inserción (cosa que suele ocurrir con el primer registro).

- Para la migración de los usuarios:

  ```
  bash# ./migrate_passwd.pl  /etc/passwd > /root/usus.ldif
  ```

  Una vez copiados todos los usuarios en format LDIF, tenemos que editar el fichero `usuarios.ldif` y eliminar todos los registros que hacen referencia a usuarios especiales de Linux, incluyendo a "root" (no se recomienda en general exportar la cuenta de "root" mediante LDAP, por cuestiones de seguridad). En definitiva, sólo deberían quedar en el fichero los registros asociados a los usuarios comunes que hemos añadido al sistema. Una vez editado el fichero, añadimos los registros al directorio:

```
bash# ldapadd -x -c -D "cn=root,dc=admon,dc=com" -W -f /root/usus.ldif
```

- Para la migración de los grupos:

```
bash# ./migrate_group.pl  /etc/group > /root/grupos.ldif
```

  Como con los usuarios, eliminamos del fichero `grupos.ldif` los grupos especiales, y dejamos sólo los grupos privados de los usuarios comunes que hemos añadido al sistema. Tras la modificación, añadimos esos grupos al directorio:

```
bash# ldapadd -x -c -D "cn=root,dc=admon,dc=com" -W -f /root/grupos.ldif
```

- Y así sucesivament con hosts, servicios, etc. De todas formas, para nuestros propósitos de uso del directorio, con la migración hecha hasta aquí resultaría suficiente.

A partir de este momento, pueden utilizarse las utiliadades **ldapadd** para añadir entradas, **ldapmodify** y **ldapmodrdn** para modificar entradas o nombres relativos de entrada (el nombre distinguido relativo de una entrada es el primer campo del nombre distinguido de dicha entrada), **ldapdelete** para eliminar entradas, **ldappasswd** para modificar la contraseña de una entrada y la ya mencionada **ldapsearch** para buscar entradas, desde cualquiera de los clientes. Se recomienda visitar las páginas de manual correspondientes para más información.

Es importante recordar que las órdenes de añadir y modificar entradas esperan recibir la información en formato LDIF. Por ejemplo, para añadir una entrada de grupo denominada "alumnos" con GID 1000 deberíamos crear primeramente un archivo (`alumnos.ldif`) con el siguiente contenido (y al menos una línea en blanco al final):

```
dn: cn=alumnos,ou=Group,dc=admon,dc=com
objectclass: posixGroup
objectclass: top
cn: alumnos
userPassword: {crypt}x
gidNumber: 1000
```

Y posteriormente añadirlo al directorio mediante la orden:

```
bash# ldapadd -x -D "cn=root,dc=admon,dc=com" -W -f alumnos.ldif
```

Evidentemente, esto resulta muy tedioso y es fácil equivocarse. Por este motivo, se recomienda la utilización de herramientas gráficas que faciliten la labor de crear, modificar y borrar entradas en el directorio. En Sección 2.11, “Herramientas gráficas de administración” se amplía este aspecto.

2.8. Autenticación basada en OpenLDAP

Una vez configurado el servidor de LDAP para almacenar la información del directorio, y los clientes de LDAP para poder preguntar por ella, el siguiente y último paso para organizar un dominio Linux con LDAP es conseguir que el directorio sea utilizado por todos los ordenadores (servidores y clientes) como fuente de información administrativa, añadida a los propios ficheros de configuración locales presentes en cada ordenador.

En principio, la información administrativa que tiene sentido centralizar en un dominio Linux se reduce prácticamente a cuentas de usuario (incluyendo contraseñas) y cuentas de grupo (las cuentas de ordenadores del directorio, es decir, la centralización del fichero `/etc/hosts` del servidor LDAP, pueden obviarse si ya disponemos de resolución de nombres basada en DNS, por ejemplo). En conjunto, la información almacenada en ambos tipos de cuentas permite autentificar a un usuario cuando éste desea iniciar una sesión interactiva en un sistema Linux y, en el caso de que la autentificación sea positiva, crear el contexto de trabajo inicial (es decir, el proceso *shell* inicial) para ese usuario. Manteniendo ambos tipos de cuentas en el directorio permitiría una gestión completamente centralizada de los usuarios del dominio.

Internamente, este proceso de autentificación y creación del contexto inicial que Linux lleva a cabo cuando un usuario desea iniciar una sesión interactiva utiliza dos bibliotecas distintas:

1. **PAM** (*Pluggable Authentication Module*) es una biblioteca de autentificación genérica que cualquier aplicación puede utilizar para validar usuarios, utilizando por debajo múltiples esquemas de autentificación alternativos (ficheros locales, Kerberos, LDAP, etc.). Esta biblioteca es utilizada por el proceso de "login" para averiguar si las credenciales tecleadas por el usuario (nombre y contraseña) son correctas.

2. **NSS** (*Name Service Switch*) presenta una interfaz genérica para averiguar los parámetros de una cuenta (como su UID, GID, *shell* inicial, directorio de conexión, etc.), y es utilizada por el proceso de "login" para crear el proceso de atención inicial del usuario.

La ventaja fundamental de ambas bibliotecas consiste en que pueden reconfigurarse dinámicamente mediante ficheros, sin necesidad de recompilar las aplicaciones que las utilizan. Por tanto, lo único que necesitamos es reconfigurar ambas para que utilicen el servidor LDAP además de los ficheros locales (`/etc/passwd`, etc.) de cada ordenador.

En CentOS Linux, esta configuración es muy sencilla. Primero instalamos (si no lo está ya) un paquete denominado `nss_ldap`, que contiene los módulos de LDAP para PAM y NSS. A partir de ahí, ejecutamos la herramienta de configuración **system-config-authentication**, que configura automáticamente los ficheros de PAM y NSS con los mecanismos de autentificación disponibles. En nuestro caso, basta habilitar el soporte LDAP en los paneles "Información del usuario" y "Autenticación", configurando LDAP indicando la dirección IP del servidor y el sufijo del directorio (en formato LDAP, claro). En principio, no debemos activar la casilla de "Utilizar TLS" (que activaría conexiones seguras) ya que no hemos activado este tipo de conexiones en el servidor.

Es importante recordar que debemos realizar esta configuración en todos los clientes primero, y sólo iniciarla en el servidor cuando hayamos asegurado que todo funciona correctamente. En cualquier caso, no resulta imprescindible configurar el servidor como cliente, si siempre incluimos los usuarios y grupos nuevos tanto en el directorio como en los ficheros locales del mismo (o bien si dichos usuarios nunca van a trabajar en el servidor).

La comprobación desde cualquier cliente que el dominio funciona es muy sencilla. Simplemente ejecutamos:

```
bash# getent passwd
bash# getent group
```

Y comprobamos que nos devuelven, respectivamente, la lista completa de usuarios y grupos del servidor. En realidad, esta forma comprobaría únicamente el funcionamiento correcto de NSS sobre LDAP. Para una comprobación completa (PAM+NSS), la manera más efectiva es intentar iniciar una sesión en un cliente con un usuario que sólo esté definido en el servidor. Hay que recordar que el directorio de conexión del usuario (*home*)debe existir en el ordenador cliente (localmente, o bien montado por NFS).

2.9. Permisos de acceso

En OpenLDAP es posible establecer permisos de acceso, e indicar así quién puede leer, buscar, comparar, modificar, etc., la información almacenada en el directorio. Estos permisos se expresan en el fichero de configuración `/etc/openldap/slapd.conf` mediante una lista de sentencias de acceso. La sintaxis que utilizan estas sentencias es relativamente compleja y está descrita con detalle en la página de manual `slapd.access`. A continuación se describe la sintáxis de las formas más usuales de estas sentencias.

```
access to <what> [by <who> <acess_control>]+
```

donde:

- `<what>` es una expresión que especifica a qué entradas del directorio se aplica la regla. Existen varias opciones, siendo las más comunes las siguientes: (1) puede indicarse todo el directorio mediante un asterisco (`*`), (2) una entrada del directorio (por ejemplo, `dn="cn=jero,ou=people,dc=admon, dc=com"` y (3) un subarbol de entradas del directorio (por ejemplo, `dn.subtree="ou=ventas,dc=admon, dc=com"`). Por defecto, los permisos se aplican a todos los atributos de las entradas especificadas, pero es posible seleccionar qué atributos que se verán afectados utilizando la opción `attrs` (por ejemplo, `dn.subtree="ou=People, dc=admon, dc=com" attrs=userPassword`).
- `<who>` indica a quién (a qué *usuario(s)*) se especifica la regla. Puede tomar diferentes valores, siendo los más comunes los siguientes: `self` (el propietario de la entrada), `dn="..."` (el usuario representado por el nombre distinguido), `users` (cualquier usuario acreditado), `anomymous` (cualquier usuarios no acreditado) y `*` (cualquier usuario).

- `<access_control>` indica qué operación concede la regla: `none` (sin acceso), `auth` (utilizar la entrada para validarse), `compare` (comparar), `search` (búsqueda), `read` (lectura), y `write` (modificación).

Para entender correctamente la semántica de las reglas de control de acceso, es imprescindible conocer el método que sigue `slapd` para evaluarlas, cada vez que recibe una petición por parte de un cliente: primero se busca, secuencialmente, la primera regla cuya expresión `<what>` incluye la entrada, o entradas, afectadas por la petición. Dentro de ella, se busca secuencialmente la primera expresión `<who>` que incluye al *usuario* que ha realizado la petición desde el cliente. Una vez encontrado, si el nivel de acceso expresado por `<access_control>` es mayor o igual al requerido por la petición, entonces ésta se concede, y en caso contrario se deniega. Si no existe ninguna expresión `<who>` que incluya al usuario, o bien no existe ninguna regla cuya expresión `<what>` incluya la información afectada por la petición, se deniega la petición.

Por tanto, el orden en que aparecen las reglas en el fichero, y el orden interno de las cláusulas "by" dentro de cada regla, es relevante: si varias reglas afectan a los mismos objetos, las reglas más específicas deberían ubicarse antes en el fichero; y, dentro de una regla, si incluimos varias cláusulas `by`, también debemos situar las más específicas primero.

Es importante tener en cuenta que el administrador del directorio siempre tiene asignado el permiso de escritura y, por defecto, el resto de usuarios tienen permiso de lectura. Un ejemplo de reglas de acceso que modifica levemente el comportamiento por defecto podría ser el siguiente:

```
access to dn.subtree="ou=People,dc=admon,dc=com" attrs=userPassword
       by self write
       by dn="cn=root,dc=admon,dc=com" write
       by * auth

access to *
       by dn="cn=root,dc=admon,dc=com" write
       by * read
```

En este ejemplo, la primera regla de acceso permite a cada usuario a cambiarse su propia contraseña (la contraseña es el atributo `userPassword` en los objetos de tipo usuario, que se sitúan por defecto dentro de la unidad organizativa "People"), al administrador cambiar la de cualquier usuario y al resto de usuarios sólo pueden utilizar este campo para autentificarse. De esta forma se consigue que un usuario no pueda leer las contraseñas cifradas de otros usuarios. La segunda regla de acceso permite al administrador cambiar cualquier entrada del directorio y al resto de usuarios sólo leerlas, lo que corresponde realmente con el comportamiento por defecto excepto para el atributo `userPassword`.

Finalmente, es importante destacar que las reglas de acceso también permiten *delegar* la administración de parte (o todo) el directorio a ciertos usuarios, de forma análoga a la delegación de control en el Directorio Activo de Windows Server (ver Sección 1.5, “Delegación de la administración”). En este caso, la delegación se traduce habitualmente en conceder permisos de escritura sobre los objetos situados en una cierta unidad organizativa, bien sea sobre todos los atributos de dichos objetos, o bien sobre alguno(s) en concreto.

Por ejemplo, supongamos que queremos modificar el comportamiento de las reglas de acceso anteriores para conseguir que sobre la unidad organizativa `People`, el usuario `javier` tenga capacidad administrativa completa, mientras que el usuario `jose` pueda modificar únicamente las contraseñas de sus usuarios. Para ello debemos modificar las reglas anteriores de forma que queden como se muestra a continuación:

```
access to dn.subtree="ou=People,dc=admon,dc=com" attrs=userPassword
       by dn="cn=javier,ou=People,dc=admon,dc=com" write
       by dn="cn=jose,ou=People,dc=admon,dc=com" write
       by self write
       by dn="cn=root,dc=admon,dc=com" write
       by * auth

access to dn.subtree="ou=People,dc=admon,dc=com"
       by dn="cn=javier,ou=People,dc=admon,dc=com" write
       by dn="cn=root,dc=admon,dc=com" write
       by * read

access to *
       by dn="cn=root,dc=admon,dc=com" write
       by * read
```

Esta forma de construir las reglas viene determinada por la manera particular en que OpenLDAP las evalúa, tal como se ha descrito arriba. En concreto, la primera regla del ejemplo permite que tanto `javier` como `jose` modifiquen las contraseñas de los usuarios de `People`, mientras que la segunda permite que `javier` tenga permisos de modificación sobre *el resto* de los atributos de los usuarios de `People`.

2.10. Configuración de OpenLDAP con varios servidores

Si las necesidades de nuestra red y/o de nuestro directorio hacen aconsejable mantener más de un servidor LDAP (por ejemplo, para poder equilibrar la carga de las consultas, y por aspectos de tolerancia a fallos) podemos configurar varios servidores LDAP que mantengan imágenes sincronizadas de la información del directorio.

Para mantener el directorio replicado en diferentes servidores, OpenLDAP propone un esquema de maestro único y múltiples esclavos. Este esquema funciona de la siguiente forma: cada vez que se produce un cambio en el directorio del servidor maestro, el servicio `slapd` escribe dicho cambio, en formato LDIF, en un fichero local de registro (es decir, *log file* o cuaderno de bitácora). El servidor maestro inicia otro servicio denominado `slurpd` que, cada cierto tiempo se activa, lee dichos cambios e invoca las operaciones de modificación correspondientes en todos los esclavos. En cambio, si un esclavo recibe una operación de modificación directa por parte de un cliente, ésta debe redirigirse automáticamente al servidor maestro.

Evidentemente, este esquema sólo funciona si todos los servidores (maestro y esclavos) parten de un estado del directorio común. Por ese motivo, es necesario copiar manualmente la base de datos del directorio (es decir, el contenido del directorio `/var/lib/ldap` del servidor maestro) a cada esclavo, estando los servicios `slapd` parados en todos ellos, por supuesto.

La configuración del servidor maestro y de cada esclavo sería la siguiente:

1. **Servidor maestro**. En el archivo de configuración `/etc/openldap/slapd.conf` hay que añadir las siguientes líneas por cada servidor esclavo:

```
replica  host=esclavo.admon.com:389
         binddn="cn=Replicator,dc=admon,dc=com"
         bindmethod=simple
         credentials=CONTRASEÑA_PLANA
```

   Y además hay que decirle a `slapd` en qué fichero de registro tiene que escribir los cambios:

```
replogfile   /var/lib/ldap/master-slapd.replog
```

2. **Servidor esclavo**. Por una parte, en el servidor esclavo hay que configurar el archivo `/etc/openldap/slapd.conf` de la misma forma que en el maestro (ver Sección 2.5, "Instalación del servidor OpenLDAP"), exceptuando las líneas expuestas en el apartado anterior, que sólo corresponden al maestro.

   Por otra parte, hay que incluir en dicho fichero las siguientes opciones:

```
rootdn     "cn=Replicator,dc=admon,dc=com"
updatedn   "cn=Replicator,dc=admon,dc=com"
updateref  ldap://maestro.admon.com
```

   La opción `updatedn` indica la cuenta con la que el servicio `slurpd` del servidor maestro va a realizar las modificaciones en la réplica del esclavo. Como puede comprobarse, hemos establecido que esta cuenta sea también el "rootdn" del servidor esclavo. Esa es la forma más sencilla de asegurar que este usuario tendrá permisos para modificar el directorio en este servidor (si no fuera así, deberíamos asegurarnos que esta cuenta tuviera concedido permiso de escritura en el directorio del servidor esclavo, en directiva "access" correspondiente). Por su parte, `updateref` indica al servidor esclavo que cualquier petición directa de modificación que venga de un cliente debe ser redireccionada al servidor maestro del directorio.

2.11. Herramientas gráficas de administración

Entre las diferentes herramientas gráficas con las que se puede manipular un directorio de tipo LDAP en Linux, hemos elegido una denominada *"phpLDAPadmin"* [http://phpldapadmin.sourceforge.net] por su sencillez y comodidad.

Esta herramienta se ejecuta de forma indirecta a través de un servidor Web, Apache normalmente. Una muestra de su interfaz la tenemos en la Figura 2.2, "Vista de la herramienta phpLDAPadmin".

A partir de ese momento, ya estamos en condiciones de añadir, modificar y borrar usuarios y grupos del directorio, mediante una simple interacción con la herramienta. Si esta herramienta se convierte en la herramienta fundamental de gestión de usuarios y grupos en el dominio Linux, necesitamos tener en mente un par de precauciones: en primer lugar, si deseamos mantener la filosofía de grupos privados, debemos crear el grupo privado de un usuario *antes* que el propio usuario, ya que en este último paso sólo podemos seleccionar su grupo primario. Y en segundo lugar, tendremos que gestionar manualmente los directorios de conexión de los usuarios que creemos.

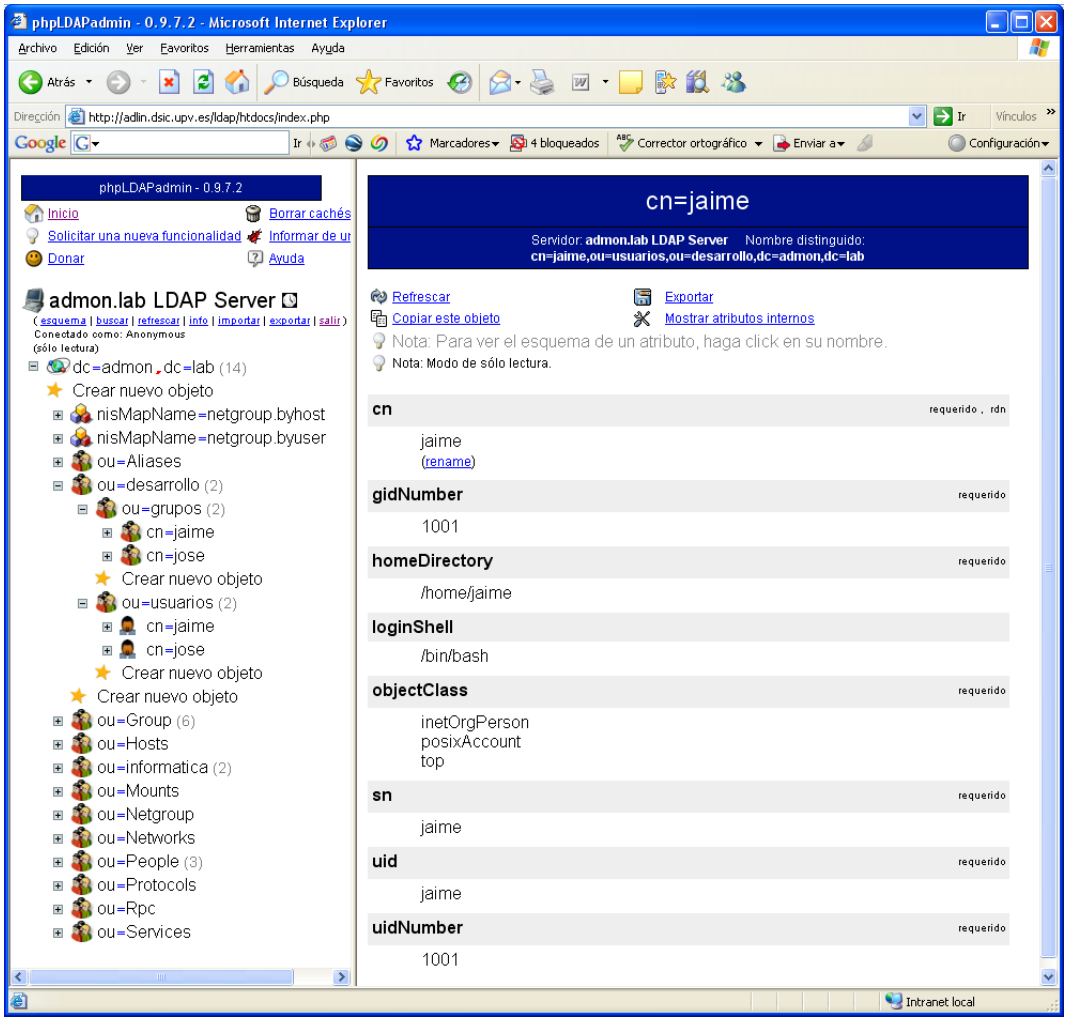

Figura 2.2. Vista de la herramienta phpLDAPadmin

3

Sistema de Archivos en Red (NFS)

Indice

3.1. Introducción

Como es bien sabido por cualquier usuario de Linux, un sistema Linux puede trabajar únicamente con una jerarquía de directorios, de tal forma que si se desea acceder a distintos sistemas de archivos (particiones de discos, cd-roms, disquetes, etc.), todos ellos deben montarse primero en algún punto de dicha jerarquía única.

Siguiendo la misma filosofía, *Network File System* (NFS) es un servicio de red que permite a un ordenador cliente montar y acceder a un sistema de archivos (en concreto, un *directorio*) remoto, exportado por otro ordenador servidor. Este capítulo explica las bases de cómo exportar directorios por NFS desde un sistema Linux y cómo acceder a ellos desde otros sistemas a través de la red.

3.2. Acceso a directorios remotos mediante NFS

Comenzaremos esta sección con un ejemplo. Supóngase que una máquina denominada `faemino` desea acceder al directorio `home/ftp/pub/` de la máquina `cansado`. Para ello, debería invocarse el siguiente mandato (en `faemino`):

```
bash# mount -t nfs  cansado:/home/ftp/pub  /mnt
```

Este mandato indica que el directorio `/home/ftp/pub/`, que debe ser *exportado* por el ordenador `cansado`, va a *montarse* el directorio local `/mnt/` de `faemino`. Es importante tener en cuenta que este directorio local debe existir previamente para que el montaje pueda realizarse. La opción `-t nfs` indica a **mount** el tipo de sistema de archivos que va a montar, aunque en este caso podría omitirse, ya que mount detecta por el carácter que se trata de un montaje remoto (por NFS) gracias al carácter '`:`' en la especificación del "origen" ("`cansado:/home/ftp/pub`").

Una vez el montaje se ha realizado, cualquier acceso a archivos o directorios dentro de `/mnt` (lectura, escritura, cambio de directorio, etc.) se traduce de forma transparente a peticiones al ordenador servidor (`cansado`), que las resolverá y devolverá su respuesta al cliente, todo a través de la red. Es decir, el montaje de directorios mediante NFS permite trabajar con archivos remotos exactamente igual que si fueran locales, aunque lógicamente con una menor velocidad de respuesta.

3.3. Usos típicos de NFS

En general, NFS es muy flexible, y admite las siguientes posibilidades (o escenarios):

- Un servidor NFS puede exportar más de un directorio y atender simultáneamente a varios clientes.
- Un cliente NFS puede montar directorios remotos exportados por diferentes servidores.

- Cualquier sistema UNIX puede ser a la vez cliente y servidor NFS.

Teniendo esto en cuenta, existen varios usos típicos de NFS donde este servicio muestra su utilidad (entre otros):

1. **Directorios de conexión (home) centralizados**. Cuando en una red local de máquinas Linux se desea que los usuarios puedan trabajar indistintamente en cualquiera de ellas, es apropiado ubicar los directorios de conexión de todos ellos en una misma máquina y hacer que las demás monten esos directorios mediante NFS.

2. **Compartición de directorios de uso común**. Si varios usuarios (desde distintas máquinas) trabajan con los mismos archivos (de un proyecto común, por ejemplo) también resulta útil compartir (exportar+montar) los directorios donde se ubican dichos archivos.

3. **Ubicar software en un solo ordenador de la red**. Es posible instalar software en un directorio del servidor NFS y compartir dicho directorio vía NFS. Configurando los clientes NFS para que monten dicho directorio remoto en un directorio local, este software estará disponible para todos los ordenadores de la red.

3.4. Funcionamiento de NFS

En el sistema cliente, el funcionamiento de NFS está basado en la capacidad de traducir los accesos de las aplicaciones a un sistema de archivos en peticiones al servidor correspondiente a través de la red. Esta funcionalidad del cliente se encuentra normalmente programada en el núcleo de Linux, por lo que no necesita ningún tipo de configuración.

Respecto al servidor, NFS se implementa mediante dos *servicios* de red, denominados **mountd** y **nfsd**. Veamos qué acciones controlan cada uno de ellos:

1. El servicio **mountd** se encarga de atender las peticiones remotas de montaje, realizadas por la orden `mount` del cliente. Entre otras cosas, este servicio se encarga de comprobar si la petición de montaje es válida y de controlar bajo qué condiciones se va a acceder al directorio exportado (sólo lectura, lectura/escritura, etc.). Una petición se considera válida cuando el directorio solicitado ha sido explícitamente exportado y el cliente tiene permisos suficientes para montar dicho directorio. Esto se detalla más adelante en el documento.

2. Por su parte, y una vez un directorio remoto ha sido montado con éxito, el servicio **nfsd** se dedica a atender y resolver las peticiones de acceso del cliente a archivos situados en el directorio.

3.5. Instalación y configuración del cliente NFS

Como se ha expresado anteriormente, el cliente NFS no requiere ni instalación ni configura-

ción. Los directorios remotos pueden importarse utilizando el mandato **mount** y el fichero asociado `/etc/fstab`

En este sentido, recuérdese que en cada invocación al mandato **mount** (y/o en cada línea del fichero `/etc/fstab`) se pueden establecer opciones de montaje. En ellas se particulariza el comportamiento que tendrá el sistema de archivos una vez se haya montado en el directorio correspondiente. En el caso de NFS, las opciones más importantes son las que gobiernan el modo de fallo de las peticiones remotas, es decir, cómo se comporta el cliente cuando el servidor no responde:

1. `soft`. Con esta opción, cuando una petición no tiene respuesta del servidor el cliente devuelve un código de error al proceso que realizó la petición. El problema es que muy pocas aplicaciones esperan este comportamiento y ello puede provocar situaciones en las que se pierda información. Por tanto, no es aconsejable.

2. `hard`. Mediante esta opción, cuando el servidor no responde el proceso que realizó la petición en el cliente se queda suspendido indefinidamente. Esta es la opción que se recomienda normalmente, por ser la que esperan las aplicaciones, y por tanto más segura desde el punto de vista de los datos.

   Esta opción se puede combinar con la opción `intr`, que permite matar el proceso mediante el envío de una señal (de la forma tradicional en Linux).

A continuación se muestra un ejemplo, en el que se presenta la línea del archivo `/etc/fstab`(de la máquina `faemino`) relacionada con el ejemplo de la Sección 3.2, “Acceso a directorios remotos mediante NFS”:

```
#device                mountpoint  fs-type  options    dump  fsckorder
...
cansado:/home/ftp/pub  /mnt        nfs      defaults   0     0
```

3.6. Instalación y configuración del servidor NFS

El servidor necesita, además de los dos servicios **mountd** y **nfsd** (ambos se aúnan en el servicio común demominado **nfs**), uno más denominado **portmap** (del inglés **portmapper**), sobre el cual ambos basan su funcionamiento. Por tanto, la configuración de un servidor NFS necesita únicamente tener disponibles dichos servicios e iniciarlos en el nivel de ejecución 3 (o 5, o ambos) de la máquina.

Una vez activos los servicios NFS, el servidor tiene que indicar explícitamente qué directorios desea que se exporten, a qué máquinas y con qué opciones. Para ello existe un fichero de configuración denominado `/etc/exports`. A continuación se muestra uno de ejemplo, sobre el que se explican las características más relevantes:

```
# Directory            Clients and (options)
/tmp                   pc02??.dsic.upv.es(rw) *.disca.upv.es()
/home/ftp/pub          158.42.54.1(rw,root_squash)
```

```
/                       mio.dsic.upv.es(rw,no_root_squash)
/pub                    (rw,all_squash,anonuid=501,anongid=601)
/pub/nopublic           (noaccess)
```

Cada línea especifica un directorio que se va a exportar, junto con una lista de autorizaciones, es decir, qué ordenadores podrán montar dicho directorio y con qué opciones (desde el punto de vista del servidor). Cada elemento de la lista de ordenadores puede especificar un solo ordenador (mediante nombre simbólico o dirección IP) o un grupo de ordenadores (mediante el uso de caracteres comodín como `*' ó `?'). Cuando el ordendor/rango no se especifica (por ejemplo, en las últimas dos líneas), esto significa que el directorio correspondiente se exporta a todos los ordenadores del mundo (conectados a Internet). Por su parte, de entre las posibles opciones de montaje que, entre paréntesis, pueden especificarse para cada ordenador/grupo, las más importantes se resumen a continuación:

`()`	Esta opción establece las opciones que NFS asume por defecto.
`ro`	El directorio se exporta como un sistema de archivos de sólo lectura (opción por defecto).
`rw`	El directorio se exporta como un sistema de archivos de lectura/escritura.
`root_squash`	Los accesos desde el cliente con UID=0 (root) se convierten en el servidor en accesos con UID de un usuario anónimo (opción por defecto)
`no_root_squash`	Se permite el acceso desde un UID = 0 sin conversión. Es decir, los accesos de root en el cliente se convierten en accesos de root en el servidor.
`all_squash`	Todos los accesos desde el cliente (con cualquier UID) se transforman en accesos de usuario anónimo.
`anonuid, anongid`	Cuando se activa la opción `root_squash` ó `all_squash`, los accesos anónimos utilizan normalmente el UID y GID primario del usuario denominado `nobody`, si éste existe en el servidor (opción por defecto). Si se desean utilizar otras credenciales, los parámetros `anonuid` y `anongid` establecen, respectivamente, qué `uid` y `gid` tendrá la cuenta anónima que el servidor utilizará para acceder contenido del directorio.
`noaccess`	Impide el acceso al directorio especificado. Esta opción es útil para impedir que se acceda a un subdirectorio de un directorio exportado.

Es importante destacar que cada vez que se modifica este fichero, para que se activen los cambios, el servidor NFS debe ser actualizado ejecutando el mandato **exportfs -ra**.

Una lista completa de las opciones de montaje y su significado pueden encontrarse en la

página de manual `exports (5)` de Linux. La mayoría de estas opciones establecen como quién se comporta el proceso cliente cuando su petición de acceso llega al servidor. En principio, cada petición lleva asociada el UID y GID del proceso cliente, es decir, se comporta como sí mismo. No obstante, si está activa la opción `all_squash` (o bien el UID es cero y `root_squash` está activado), entonces el UID/GID se convierten en los del usuario anónimo. De todas formas, hay que tener en cuenta que los permisos sobre cada acceso del cliente se evalúan mediante los UID/GID que finalmente son válidos en el servidor (es decir, los originales o los anónimos, según el caso).

Es muy importante resaltar que no existe ningún proceso de acreditación de usuarios en NFS, por lo que el administrador debe decidir con cautela a qué ordenadores exporta un determinado directorio. Un directorio sin restricciones es exportado, en principio, a cualquier otro ordenador conectado con el servidor a través de la red (incluida Internet). Si en un ordenador cliente NFS existe un usuario con un UID igual a "X", este usuario accederá al servidor NFS con los permisos del usuario con el UID igual a "X" del servidor, aunque se trate de usuarios distintos.

4

Configuración Básica de Samba

Indice

4.1. ¿Qué es Samba?

Samba es un producto que se distribuye gratuitamente para varias versiones de UNIX(R), de acuerdo con los términos de la *General Public License* de GNU, y que básicamente permite al sistema UNIX conversar con sistemas Windows a través de la red de forma nativa. De esta forma, el sistema UNIX aparece en el "Entorno de red", y clientes Windows pueden acceder a sus recursos de red e impresoras compartidas como si de otro sistema Windows se tratase. Para ello, Samba implementa los protocolos NetBIOS y SMB. NetBIOS es un protocolo de nivel de sesión que permite establecer sesiones entre dos ordenadores. SMB (*Server Message Block*), implementado sobre NetBIOS, es el protocolo que permite a los sistemas Windows compartir ficheros e impresoras.

Esencialmente, Samba consiste en dos programas, denominados **smbd** y **nmbd**. Ambos programas utilizan el protocolo NetBIOS para acceder a la red, con lo cual pueden conversar con ordenadores Windows. Haciendo uso de estos dos programas, Samba ofrece los siguientes servicios, todos ellos iguales a los ofrecidos por los sistemas Windows:

- Servicios de acceso remoto a ficheros e impresoras.
- Autenticación y autorización.
- Resolución de nombres.
- Anuncio de servicios.

El programa **smbd** se encarga de ofrecer los servicios de acceso remoto a ficheros e impresoras (implementando para ello el protocolo SMB), así como de autenticar y autorizar usuarios. **smbd** ofrece los dos modos de compartición de recursos existentes en Windows, basado en usuarios o basado en recursos. En el modo basado en usuarios (propio de los dominios Windows) la autorización de acceso al recurso se realiza en función de nombres de usuarios registrados en un dominio, mientras que en el modo basado en recursos (propio de Windows 3.11/95) a cada recurso se le asigna una contraseña, estando autorizado el acceso en función del conocimiento de dicha contraseña.

El programa **nmbd** permite que el sistema UNIX participe en los mecanismos de resolución de nombres propios de Windows, lo cual incluye el anuncio en el grupo de trabajo, la gestión de la lista de ordenadores del grupo de trabajo, la contestación a peticiones de resolución de nombres y el anuncio de los recursos compartidos. De esta forma, el sistema UNIX aparece en el "Entorno de Red", como cualquier otro sistema Windows, publicando la lista de recursos que ofrece al resto de la red.

Adicionalmente a los dos programas anteriores, Samba ofrece varias utilidades. Algunas de las más relevantes son las siguientes:

- **smbclient**. Una interfaz similar a la utilidad **ftp**, que permite a un usuario de un sistema UNIX conectarse a recursos SMB y listar, transferir y enviar ficheros.
- **swat** (Samba Web Administration Tool). Esta utilidad permite configurar Samba de forma local o remota utilizando un navegador de web.

- **smbfs** Sistema de ficheros SMB para Linux. Linux puede montar recursos SMB en su jerarquía, al igual que sucede con directorios compartidos vía NFS.

- **winbind**. Permite integrar un servidor Samba en un dominio Windows sin necesidad de crear usuarios UNIX en el servidor Samba que correspondan con los usuarios del dominio Windows, simplificando así la labor de administración.

4.2. El protocolo SMB

Puesto que Samba es, fundamentalmente, una implementación para UNIX del protocolo SMB, quizás la mejor forma de entender Samba es comenzar por describir SMB con un poco más de detalle. Esta sección realiza una pequeña revisión de este protocolo.

SMB es un protocolo de comunicación de alto nivel que puede implementarse sobre diversos protocolos como TCP/IP, NetBEUI y IPX/SPX, tal como muestra la Figura 4.1, "Protocolos sobre los que puede implementarse SMB.", junto con la ubicación de dichos protocolos en los niveles OSI y en la pila TCP/IP. Entre todas esas alternativas, tanto en el caso de Samba como de Windows 2000/XP, SMB se implementa habitualmente encima de NetBIOS sobre TCP/IP (esta alternativa se ha convertido en el estándar de facto para compartir recursos entre sistemas Windows). Sin embargo, no incidiremos más en los protocolos que soportan SMB o en su implementación, puesto que todo ello queda fuera del contexto de este tema.

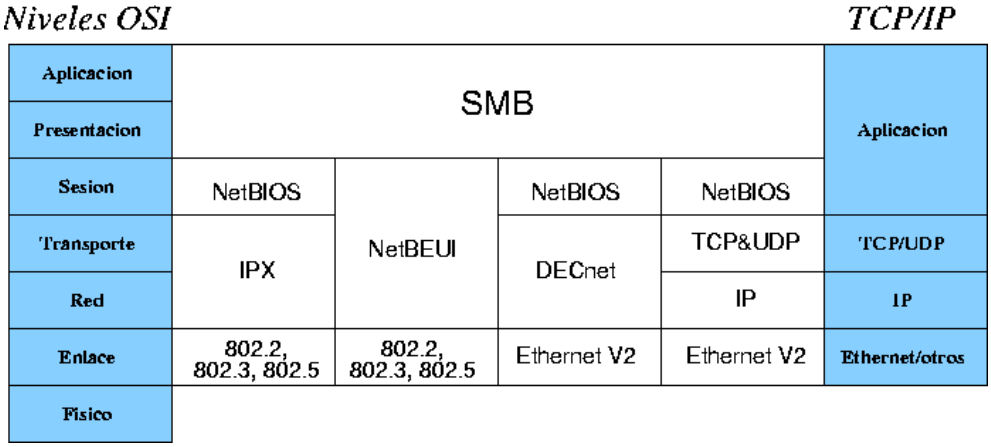

Figura 4.1. Protocolos sobre los que puede implementarse SMB.

Históricamente, este protocolo fue desarrollado inicialmente por IBM como el *IBM PC Network SMB Protocol* o *Core Protocol* a principios de los años 80. Desde entonces, diversos fabricantes (especialmente Microsoft) han ido ampliando su funcionalidad progresivamente, creando diferentes *variantes* (versiones) de SMB. Desafortunadamente, en ocasiones el cambio de versión ha conllevado el rebautizar el propio protocolo. En este sentido, SMB ha recibido, entre otros, los siguientes nombres: Core Protocol, DOS Lan Manager, LAN Manager, NTLM (NT Lan Manager), y en los últimos años, CIFS (*Common Internet File System*). Todos

ellos, por tanto, hacen referencia a SMB, aunque se diferencien en algunos detalles de su funcionalidad y/o implementación.

Si nos fijamos en su interfaz, SMB es un protocolo de tipo *cliente/servidor*, donde el ordenador "servidor" ofrece recursos (archivos, impresoras, etc.) que pueden ser utilizados remotamente por los ordenadores "cliente" a través de la red. Asimismo, es un protocolo de los denominados *petición/respuesta*, indicando que las comunicaciones se inician siempre desde el cliente como una petición de servicio al servidor (dicha petición se denomina precisamente SMB), que la procesa y retorna una respuesta a dicho cliente. (En realidad, existe un caso en que el servidor envía un mensaje al cliente sin haber recibido una petición de éste, pero la discusión del protocolo a ese nivel queda fuera del ámbito de este texto). La respuesta del servidor puede ser positiva (con el resultado de procesar la petición del cliente) o negativa (mensaje de error), en función del tipo de petición, la disponibilidad del recurso, el nivel de acceso (permisos) del cliente, etc.

El siguiente aspecto relevante de SMB es saber qué mecanismos de autentificación soporta este protocolo para controlar el acceso del cliente a los recursos compartidos. En concreto, SMB soporta dos modos de autentificación alternativos, denominados *share* y *user*:

- Cuando compartimos un recurso en modo **share**, la protección de dicho recurso recae en una contraseña que asociamos al mismo, de forma que cualquier usuario de un sistema cliente remoto que conozca dicha palabra de paso podrá acceder sin mayores restricciones al recurso (este es el mecanismo de autentificación por defecto en las implememementaciones de SMB para Windows 9X, por ejemplo).

- Sin embargo, en modo **user**, el servidor recibe inicialmente del sistema cliente unas credenciales de usuario (nombre, dominio y contraseña), que debe autentificar para autorizar el acceso al recurso. Concretamente, si el dominio de las credenciales es conocido, la autentificación se delega a algún controlador de dicho dominio; y en caso contrario, el usuario y la contraseña se autentifican contra la base de datos *local* del equipo servidor. En cualquier caso, en modo *user*, el control de acceso sobre el recurso se realiza en función de qué permisos posee sobre dicho recurso el usuario cuyas credenciales se han enviado desde el cliente. En otras palabras, una vez el sistema servidor ha identificado y autentificado al usuario que desea conectarse al recurso, este sistema dispone ya de un SID válido con el que puede contrastar los permisos que dicho SID posee sobre el recurso. Es conveniente recordar en este punto que si el recurso en cuestión es una carpeta compartida, se tienen en cuenta tanto los permisos *del recurso*, como los permisos *NTFS* de la carpeta y sus archivos. El modo *user* es el mecanismo de autentificación por defecto en las versiones de SMB de sistemas Windows NT y posteriores.

Finalmente, revisaremos brevemente el funcionamiento interno del protocolo SMB, utilizando para ello un ejemplo concreto. Supongamos que un sistema cliente desea acceder a una carpeta compartida que exporta el servidor (en modo *user*). En este escenario, se produciría el siguiente intercambio de mensajes entre ellos:

1. **Petición: Sesión NetBIOS**. El objetivo de este mensaje es establecer una sesión fiable para subsiguientes mensajes entre los ordenadores cliente y servidor. Es imprescindible

que el cliente conozca el nombre NetBIOS del servidor para poder alcanzarlo; el nombre NetBIOS del cliente es parte del mensaje, por lo que ambos saben quién es el otro.

2. **Respuesta: Sesión NetBIOS**. Si no hay error en el mensaje anterior, el servidor envía un mensaje de reconocimiento (ACK), aceptando la conexión.

3. **Petición: Dialecto SMB**. El cliente envía en este mensaje una lista con los dialectos o variantes de SMB que soporta, puesto que es habitual que un sistema Windows soporte varias versiones de SMB simultáneamente.

4. **Respuesta: Dialecto SMB**. El servidor contesta con el dialecto que prefiere para la comunicación subsiguiente, o un código de error si no soporta ninguna de las alternativas ofrecidas por el cliente.

5. **Petición: Inicio de sesión**. El cliente envía las credenciales de usuario (usuario, dominio, contraseña) con las que éste desea conectarse al servidor. Recuérdese que por defecto, se emplean las credenciales con las que el usuario se conectó interactivamente al sistema cliente, pero se pueden especificar otras explícitamente.

6. **Respuesta: Inicio de sesión**. El servidor autentifica las credenciales de usuario (ver modo *user* descrito arriba). Si las credenciales son buenas, el servidor posee ya un SID válido que le permite, antes que nada, comprobar si el usuario posee el derecho de conectarse al servidor (directiva "tener acceso a este equipo desde la red"). En caso afirmativo, se acepta la conexión y el servidor construye un identificador numérico particular para esa conexión (denominado *User ID* o *UID*) que devuelve al cliente. Los UIDs pueden ser reutilizados durante la vida del sistema, pero son únicos para todas las conexiones simultáneas que mantiene el servidor en un momento dado, por lo que identifican unívocamente una conexión (aceptada). Todos los mensajes posteriores del cliente deben contener este identificador para ser aceptados por el servidor.

   Por otro lado, si las credenciales estaban mal (o si los derechos eran insuficientes), el servidor envía un código de error en lugar del UID.

7. **Petición: Conexión a un recurso concreto**. El cliente envía entonces un mensaje que contiene una cadena que identifica el recurso al que desea acceder (por ejemplo, `\\pc01\impresora` o `\\pc01\carpeta`).

8. **Respuesta: Conexión a un recurso concreto**. Si el recurso solicitado por el cliente existe (y el SID asociado a la conexión posee suficientes permisos), el servidor construye un identificador denominado *Tree ID* o *TID*, que será utilizado por el cliente para hacer referencia a dicho recurso en posteriores mensajes de esa conexión.

Tras esta secuencia típica de conexión al recurso (carpeta compartida), y si todo ha funcionado correctamente, el sistema cliente ya está en condiciones de acceder a la carpeta. Mediante el envío de los SMBs correspondientes, el cliente ya puede abrir archivos, leerlos, modificarlos, etc., utilizando siempre los identificadores (UID y TID) que el servidor ha construido durante el intercambio de mensajes inicial.

4.3. Configuración de Samba

La configuración de Samba se realiza en el fichero `/etc/samba/smb.conf`. En este fichero se establecen las características del servidor Samba, así como los recursos que serán compartidos en la red. La utilización de este fichero es bastante sencilla, ya que aunque existe un gran número de opciones, muchas de ellas pueden obviarse dado que siempre existe una valor por defecto para cada opción, que suele ser apropiado. A título de ejemplo, a continuación se muestra un fichero de configuración simple, que exporta el directorio de conexión de cada usuario como un recurso de red distinto, y el directorio `/espacio/pub` como el recurso de red `pub`:

```
[global]
[homes]
   comment = Home Directories
[pub]
   path = /espacio/pub
```

Como se ve en el ejemplo, el fichero `/etc/samba/smb.conf` se encuentra divido en secciones, encabezados por una palabra entre corchetes. Dentro de cada sección figuran opciones de configuración, de la forma `etiqueta = valor`, que determinan las características del recurso exportado por la sección. Al final del capítulo (Sección 4.8, “Opciones del servidor Samba” y Sección 4.9, “Opciones del recurso”) del presente documento se citan las opciones más importantes que se pueden establecer en cada sección.

Existen tres secciones predefinidas, denominadas `global`, `homes` y `printers`, y tantas secciones adicionales como recursos extra se quieran compartir. El cometido de dichas secciones predefinidas se describe brevemente a continuación:

`[global]` Define los parámetros de Samba a nivel global del servidor, así como los parámetros que se establecerán por defecto en el resto de las secciones.

`[homes]` Define automáticamente un recurso de red por cada usuario conocido por Samba. Este recurso, por defecto, está asociado al directorio de conexión de cada usuario en el ordenador en el que Samba está instalado.

`[printers]` Define un recurso compartido por cada nombre de impresora conocida por Samba.

Para cualquier otro recurso (directorio o impresora) que se quiera compartir hay que definir una sección adicional en el fichero de configuración. El encabezamiento de dicha sección (`pub` en el ejemplo anterior) corresponderá al nombre que el recurso tendrá en la red.

Por otra parte, Samba ofrece una interfaz de edición de este fichero basada en web denominada **swat**. Esta herramienta permite configurar Samba utilizando un navegador de red, tanto de forma local como remota. Para ello, basta con acceder a la direción `http://nombre_de_ordenador_samba:901/` mediante cualquier navegador.

4.4. Niveles de seguridad

Una de las consideraciones más importantes a la hora de configurar Samba es la selección del nivel de seguridad.

Desde la perspectiva de un cliente, Samba ofrece dos modos de seguridad, denominados *share* y *user*, emulando exactamente las dos opciones de SMB que veíamos en la Sección 4.2, "El protocolo SMB":

1. **Modo Share**. En modo *share*, cada vez que un cliente quiere utilizar un recurso ofrecido por Samba, debe suministrar una contraseña de acceso asociada a dicho recurso.

2. **Modo User**. En modo *user*, el cliente debe establecer en primer lugar una sesión con el servidor Samba, para lo cual le suministra un nombre de usuario y una contraseña. Una vez Samba valida al usuario, el cliente obtiene permiso para acceder a los recursos ofrecidos por Samba.

En cualquiera de ambos, Samba tiene que asociar un usuario del sistema UNIX en el que se ejecuta Samba con la conexión realizada por el cliente. Este usuario es el utilizado a la hora de comprobar los permisos de acceso a los ficheros y directorios que el sistema UNIX/Samba comparte en la red.

La selección del nivel de seguridad se realiza con la opción `security`, la cual pertenece a la sección `[global]`. Sus alternativas son las siguientes:

```
security = share | user | server | domain
```

Desde la perspectiva del cliente, el nivel `share` corresponde al modo de seguridad *share* y los niveles `user`, `server` y `domain` corresponden todos ellos al modo de seguridad *user*. A continuación se describen someramente los cuatro niveles.

El nivel `share` es utilizado normalmente en entornos en los cuales no existe un dominio Windows. En este caso, se asocia una contraseña por cada recurso, que debe proporcionarse correctamente desde el cliente cuando se pide la conexión.

En el nivel `user`, el encargado de validar al usuario es el sistema UNIX donde Samba se ejecuta. La validación es idéntica a la que se realizaría si el usuario iniciase una sesión local en el ordenador UNIX. Para que este método sea aplicable, es necesario que existan los mismos usuarios y con idénticas contraseñas en los sistemas Windows y en el sistema UNIX donde Samba se ejecuta.

Desde la aparición de sistemas Windows como Windows 98, Windows NT 4.0 (a partir del *Service Pack 3*), Windows 2000 y posteriores, la utilización de este nivel se ha vuelto complicada, ya que dichos sistemas Windows transmiten las contraseñas cifradas por la red. Puesto que Samba no posee acceso a las contraseñas cifradas por Windows, el sistema UNIX ya no puede realizar la validación. Existen dos métodos para resolver este problema. El primero consiste en modificar el registro del sistema Windows para permitir la transferencia de contraseñas sin cifrar por la red. El segundo método obliga a utilizar una tabla de contrase-

ñas adicional en el sistema UNIX, en la cual se almacenan las contraseñas cifradas de los usuarios Windows.

En el nivel `server`, Samba delega la validación del usuario en otro ordenador, normalmente un sistema Windows. Cuando un cliente intenta iniciar una sesión con Samba, éste último intenta iniciar una sesión en el ordenador en el cual ha delegado la validación con la misma acreditación (usuario+contraseña) recibidos del cliente. Si la sesión realizada por Samba es satisfactoria, entonces la solicitud del cliente es aceptada. Este método aporta la ventaja de no necesitar que las contraseñas se mantengan sincronizadas entre los sistemas Windows y UNIX, ya que la contraseña UNIX no es utilizada en el proceso de validación. Adicionalmente, no hay inconveniente en utilizar contraseñas cifradas, ya que la validación la realiza un sistema Windows.

Por último, existe la posibilidad de utilizar el nivel `domain`. Este nivel es similar al nivel `server`, aunque en este caso el ordenador en el que se delega la validación debe ser un DC, o una lista de DCs. La ventaja de este método estriba en que el ordenador Samba pasa a ser un verdadero miembro del dominio Windows, lo que implica, por ejemplo, que puedan utilizarse las relaciones de confianza en las que participa el dominio Windows. Esto significa, en pocas palabras, que usuarios pertenecientes a otros dominios en los que los DCs confían son conocidos por Samba.

Dadas las ventajas del nivel `domain`, este documento se centra fundamentalmente en este método. Para detalles específicos de los otros niveles, se recomienda la consulta de la documentación original de Samba.

4.5. Configuración de Samba en el nivel `domain`

Los pasos a seguir para configurar Samba con el nivel de seguridad `domain` son los siguientes:

1. Desde alguno de los DCs del dominio, dar de alta el sistema UNIX donde se ejecuta Samba en el dominio (si en dicho dominio ya existía una cuenta de máquina con el mismo nombre, hay que borrar dicha cuenta y volver a crearla).

2. Detener el servidor Samba. Para ello:

```
bash# service smb stop
```

3. Utilizando `swat`, o editando manualmente el fichero de configuración `/etc/samba/smb.conf`, configurar Samba en modo `domain`. En concreto, en la sección `[global]` hay que establecer las siguientes opciones:

```
security = domain
workgroup = DOMINIO
encrypt passwords = yes
password server = DC1_DEL_DOMINIO, DC2_DEL_DOMINIO, ...
```

En esta configuración, la última línea hace referencia a los ordenadores que realiza-

rán la autenticación de los usuarios. Esta línea se puede simplificar, y sustituir por: `password server = *`

4. Agregar el sistema UNIX al dominio:

```
bash# net rpc join member -U administrador -W "DOMINIO" -S "DC"
```

5. Iniciar el servidor Samba:

```
bash# service smb start
```

4.6. Tratamiento de los accesos como invitado

Cuando se utiliza el nivel de seguridad `domain`, el tratamiento de los accesos como usuario invitado requiere algunas consideraciones. En particular, hay que tener en cuenta tres factores:

1. Si el usuario que accede ha sido acreditado por el dominio Windows al cual pertenece el servidor Samba.
2. Si el usuario que accede existe, como usuario UNIX, en el servidor Samba.
3. El valor que tenga actualmente asignado la opción global de Samba `map to guest`.

La Tabla 4.1, “Resumen de los accesos como invitado en modo "domain"” a continuación indica, en función de los tres factores anteriores, si Samba permite o no el acceso y, en caso de permitirlo, si al usuario que accede se le considera como él mismo o como invitado.

Tabla 4.1. Resumen de los accesos como invitado en modo "domain"

MAP TO GUEST = NEVER		
	En Windows...	
En Samba...	**Acreditado**	**No acreditado**
Existe	Mismo usuario	No permitido
No Existe	No permitido	No permitido
MAP TO GUEST = BAD USER		
	En Windows...	
En Samba...	**Acreditado**	**No acreditado**
Existe	Mismo usuario	No permitido
No Existe	Invitado	Invitado

MAP TO GUEST = BAD PASSWORD		
	En Windows...	
En Samba...	**Acreditado**	**No acreditado**
Existe	Mismo usuario	Invitado
No Existe	Invitado	Invitado

En cualquier caso, para que el usuario "invitado" pueda acceder a un recurso compartido, este acceso tiene que permitirse expresamente para el recurso con la opción `guest ok` (cuyo valor por defecto es `no`).

4.7. El sistema de ficheros CIFS para Linux

Linux dispone de soporte para montar recursos SMB. En concreto, los sistemas Linux actuales son capaces de montar recursos compatibles con la nueva especificación del protocolo SMB, denominada CIFS (*Common Internet File System*). De esta forma, Linux, al igual que puede montar en un directorio local un directorio exportado vía NFS, puede montar un recurso SMB/CIFS ofrecido por un servidor SMB (un sistema Windows o un servidor Samba, por ejemplo).

No obstante, existe una diferencia significativa entre NFS y CIFS. En NFS no se requiere autenticar al usuario que realiza la conexión; el servidor NFS utiliza el UID del usuario del ordenador cliente para acceder a los ficheros y directorios exportados. Un servidor SMB, por contra, requiere autenticar al usuario, para lo que necesita un nombre de usuario y una contraseña. Por ello, para montar un recurso SMB/CIFS se utiliza el mandato **mount** indicándole un tipo de sistema de archivos específico, denominado `cifs`:

```
bash# mount -t cifs -o user=USUARIO,password=CONTRASEÑA,domain=DOMINIO
//ORDENADOR/RECURSO /PUNTO/DE/MONTAJE
```

Si se omite la opción `password`, el sistema solicita al usuario que introduzca una contraseña. Si el servidor SMB valida al usuario, a partir del directorio `/PUNTO/DE/MONTAJE` se consigue el acceso al recurso `//ORDENADOR/RECURSO`.

Como de costumbre en los montajes de sistemas de archivos, podemos optar por registrar el montaje en el fichero `/etc/fstab`. Sin embargo, dicho registro presenta un problema en el caso del sistema de archivos `cifs`, puesto que el montaje siempre supone la petición de una contraseña, bien escrita en las opciones de montaje, bien solicitada por teclado en el momento de realizar dicho montaje. Obviamente, esto dificulta el montaje automático en tiempo de inicio, a menos que escribamos la contraseña en el fichero `fstab`, lo cual no resulta muy buena idea por motivos de seguridad (dicho archivo puede ser leido por cualquier usuario). La alternativa consiste en utilizar un fichero de credenciales (opción de montaje `credentials=FICHERO`), donde escribimos el nombre del usuario y su contraseña. A pesar de que la contraseña en dicho fichero también se escribe en texto plano, resulta suficiente que dicho fichero pueda ser leído por el usuario que realiza el montaje (por ejemplo, `root`, si es un montaje automático durante el inicio), lo cual permite un nivel de seguridad un poco mayor. Se recomienda consultar la página de manual `mount.cifs(8)` para obtener detalles

sobre esta opción.

A continuación se muestra una tabla con las principales opciones de montaje del sistema de archivos CIFS:

Tabla 4.2. Opciones de montaje del sistema de archivos CIFS

Opción	Descripción
`user`	Usuario con el que se realiza la conexión al ordenador remoto
`password`	Contraseña del usuario
`domain`	Dominio al que pertenece el usuario
`uid`	Nombre del usuario linux que será el propietario del directorio donde se ha montado el recurso de red
`gid`	Nombre del grupo linux que será el grupo propietario del directorio donde se ha montado el recurso de red
`dir_mode`	Bits de premiso de los directorios
`file_mode`	Bits de permiso de los ficheros

4.8. Opciones del servidor Samba

En la Tabla 4.3, "Principales opciones de la sección [global] de Samba" se describen algunas opciones del servidor Samba. Estas opciones tan sólo son aplicables a la sección `[global]`.

Tabla 4.3. Principales opciones de la sección `[global]` de Samba

Opción	Significado	Valor por defecto
`netbios name`	Nombre (NetBIOS) del ordenador Samba.	Primer componente del nombre DNS del ordenador.
`workgroup`	Nombre del dominio (o grupo de trabajo) al que pertenece Samba.	nulo
`security`	Nivel de seguridad (`share, user, server, domain`).	`user`
`encrypt passwords`	Utilizar contraseñas cifradas de Windows (en modo `domain`, sí deben utilizarse).	no
`password server`	Ordenador Windows utilizado para la autentificación. En modo `domain`, debe ser una lista de los DCs del dominio.	nulo
`map to guest`	Establece en qué condiciones un acceso a Samba debe considerarse en modo invitado (en el nivel `domain`, este parámetro afecta sólo cuando el acceso no ha sido acreditado por el DC del dominio).	`never`
`log level`	Nivel de detalle en la auditoría de Samba. Es un número que indica la cantidad de información a auditar. A mayor valor, más cantidad de información.	Se establece en el script que inicia el servicio Samba.

Opción	Significado	Valor por defecto
`log file`	Nombre del fichero donde se almacenan mensajes de auditoría de Samba.	Se establece en el script que inicia el servicio Samba.

4.9. Opciones del recurso

En la Tabla 4.4, "Principales opciones de los recursos en Samba" se describen algunas opciones aplicables a cada recurso compartido. Pueden establecerse también en la sección global, siendo en este caso utilizadas como valores por defecto para cada recurso compartido.

Tabla 4.4. Principales opciones de los recursos en Samba

Opción	Significado	Valor por defecto
`read only ({yes/no})`	Recurso exportado como sólo lectura.	`yes`
`browseable ({yes/no})`	El servicio aparece en la lista de recursos compartidos al explorar el ordenador Samba desde el Entorno de Red Windows.	`yes`
`path`	Ruta absoluta al directorio compartido por el recurso.	nulo
`comment`	Descripción del servicio (cadena de caracteres).	nulo
`guest ok ({yes/no})`	Permitir accesos como invitado al recurso.	`no`
`guest account`	Si un acceso se realiza como invitado, se utiliza el usuario indicado para representar la conexión.	`nobody`
`guest only ({yes/no})`	Todos los accesos se aceptan en modo invitado.	`no`
`copy`	Duplica otro recurso existente.	nulo
`force user`	Los accesos al recurso se realizan como si el usuario que accede es el usuario indicado.	nulo (se utiliza el mismo usuario que ha realizado la conexión).
`force group`	Los accesos al recurso se realizan como si el usuario que accede pertenece al grupo indicado.	nulo (se utiliza el grupo primario del usuario que ha realizado la conexión).
`hosts allow`	Lista ordenadores desde los que se permite acceder al recurso	lista vacía (i.e., todos los ordenadores).
`hosts deny`	Lista ordenadores desde los que no se permite acceder al recurso. En caso de conflicto, prevalece lo indicado en `hosts allow`.	lista vacía (ningún ordenador).
`valid users`	Lista de usuarios que pueden acceder al recurso.	lista vacía (i.e., todos los usuarios).
`follow symlinks ({yes/no})`	Permitir el seguimiento de los enlaces simbólicos que contenga el recurso.	`yes`.
`inherit permissions ({yes/no})`	Al crear ficheros y subdirectorios nuevos, estos heredan los permisos UNIX de la carpeta donde se crean.	`no`.

5

Integración de dominios mediante servidores Windows Server 2003

Indice

5.1. Introducción

Cuando una organización posee múltiples ordenadores en los que hay instalado el mismo sistema operativo, la opción de administración más adecuada suele ser la creación de un *dominio* o conjunto lógico de sistemas que admite gestionarse de forma centralizada, típicamente mediante un esquema cliente/servidor. Para ello, es imprescindible que en dicho sistema operativo exista alguna tecnología que permita esa centralización de aspectos como cuentas de usuarios/grupos, autentificación, políticas de seguridad, etc. En la actualidad, tanto en el caso de Windows Server 2003 como el de Linux, esta tecnología existe y está basada en un servicio de directorio compatible con el estándar LDAP.

Sin embargo, es cada vez más habitual que las organizaciones posean simultáenamente sistemas de diferentes tipos. En este tipo de escenarios, la opción más sencilla (y a menudo la única posible) consiste en gestionar los sistemas de cada tipo de forma independiente, utilizando las herramientas administrativas que cada fabricante proporciona. En el caso más favorable, podemos crear un dominio que integre los sistemas de cada tipo (un dominio de sistemas Linux, un dominio de sistemas Windows (NT, 2000, 2004, XP), etc.), de forma que la gestión está centralizada en cada dominio. Pero obviamente, eso aún supone la repetición de acciones de administración en cada dominio: creación de los mismos usuarios y grupos, configuración de permisos y políticas de seguridad, administración de recursos compartidos entre sistemas, etc.

Es evidente que la opción más coherente en este tipo de entornos mixtos sería la creación de un *dominio heterogéneo* (o *multi-plataforma*), que agrupara todos los sistemas de la organización en una única administración centralizada. Lamentablemente, esta alternativa no es sencilla, ya que los diferentes fabricantes de sistemas no suelen diseñarlos para que sean compatibles entre sí (especialmente cuando el fabricante es una empresa que defiende un producto comercial). Incluso en el caso de sistemas que basan sus tecnologías en estándares (como hemos dicho, Windows Server 2003 y Linux implementan sus dominios mediante LDAP), esta integración no resulta trivial. Ello es debido fundamentalmente a diferencias de diseño entre los sistemas y a que normalmente ningún sistema implementa los estándares de forma completa y correcta hasta el último detalle.

Este capítulo se centra en cómo conseguir un dominio heterogéneo de sistemas Windows y Linux mediante una gestión centralizada en servidores Windows Server 2003 (o más correctamente, *controladores de dominio*). En concreto, se pretende abordar la labor fundamental en la integración de sistemas: la centralización de usuarios (y grupos) de ambos sistemas en una única base de datos, en este caso el Directorio Activo de Windows Server 2003.

5.2. Aspectos básicos a considerar

Globalmente, lo que se pretende conseguir es centralizar la administración de cuentas de usuario y grupo mediante un dominio Windows 2003 de tal forma que sistemas Linux puedan ser clientes de dicho dominio igual que si fueran sistemas Windows *miembros* del dominio. El resultado final de la configuración será una única base de datos de usuarios/grupos almacenada en el Directorio Activo de Windows y la posibilidad de que dichos usuarios puedan iniciar una sesión en cualquiera de los sistemas cliente (Windows o Linux), sin nece-

sitar cuentas de usuario locales en ese sistema.

Para el caso de clientes Linux, este objetivo global supone la necesidad de ofrecerles tanto un mecanismo de autentificación que puedan utilizar como el conjunto de atributos de usuario (o "atributos UNIX") necesario para que los usuarios puedan iniciar sesión satisfactoriamente (incluyendo el UID, GID, directorio de conexión o *home*, *shell* inicial, etc.). El sistema de autentificación por defecto ofrecido por los dominios Windows (basado en Kerberos) es directamente utilizable por los sistemas Linux. Sin embargo, en el caso de los atributos UNIX es necesario realizar configuraciones adicionales. A pesar de que es factible configurar a los clientes Linux para que consulten un directorio LDAP como el Directorio Activo de Windows, éste último no incluye por defecto ningún atributo de tipo UNIX.

Para que estos atributos UNIX estén disponibles en los clientes Linux, tenemos dos alternativas posibles de partida. En la primera alternativa, los atributos UNIX se almacenan en el Directorio Activo junto con el resto de información de usuario y de grupo, y son proporcionadas a los clientes Linux cuando es necesario. En este caso, se requiere realizar las siguientes configuraciones tanto en el servicio de directorio Windows como en los sistemas Linux cliente:

A. **En el Directorio Activo**. Puesto que en esta alternativa los atributos UNIX se almacenan en el Directorio Activo, es necesario modificar dicho directorio para que se almacene la información que Linux necesita para cada usuario y cada grupo. Como veremos, será necesario modificar el *esquema* del Directorio Activo para que las cuentas de usuario y grupo incorporen estos nuevos atributos.

B. **En los clientes Linux**. Las modificaciones a realizar permitirán, por un lado, que los sistemas Linux puedan acceder al Directorio Activo de los controladores Windows para consultar las listas de usuarios y grupos del dominio; y por otro lado, que los sistemas Linux sean capaces de autentificar los inicios de sesión de dichos usuarios contra los DCs Windows.

En la segunda alternativa, tanto la autentificación como la obtención de atributos UNIX para los usuarios se realiza desde el propio cliente Linux, sin involucrar al Directorio Activo. Para ello se instala un servicio en el cliente Linux, denominado winbind, que integra a dicho cliente en el dominio Windows como cualquier otro miembro. Una vez integrado, el servicio winbind autentifica a los usuarios contra el Directorio Activo y obtiene de él aquellos atributos de usuarios y grupos que son comunes a ambos tipos de sistemas (como el nombre de usuario o grupo, o la lista de miembros de un grupo, por ejemplo). Para los atributos UNIX, que no existen en el Directorio Activo, el servicio winbind incorpora un mecanismo que los genera dinámicamente cuando el sistema cliente los requiere.

Las dos siguientes secciones se centran en discutir las configuraciones exactas que hay que realizar para permitir la integración bajo cada una de ambas alternativas.

5.3. Integración de clientes Linux mediante Windows Services for UNIX

5.3.1. Modificaciones del Directorio Activo

Para conseguir que un sistema Linux vea como *cuentas de usuario* los usuarios globales que definidos en el Directorio Activo, es imprescindible conseguir que la definición de un usuario global Windows incluya los atributos que definen a un usuario en Linux. Y exactamente igual ocurre con las *cuentas de grupo*. En concreto, los sistemas Linux necesitan, al menos, los siguientes atributos:

1. **Para cada usuario**:

   - Un nombre de usuario (*login name*).
   - Una contraseña (cifrada).
   - Un identificador numérico de usuario (UID).
   - Un identificador numérico de grupo primario (GID).
   - Un directorio de conexión inicial (*home*).
   - Un programa de atención al usuario (*shell*).

2. **Para cada grupo**:

   - Un nombre de grupo.
   - Un identificador numérico de grupo (GID).
   - Una lista de los nombres de de usuario que pertenecen a ese grupo.

Como es lógico, algunos de los atributos que el Directorio Activo almacena por defecto para un usuario y grupo "Windows" pueden servir también para el caso de clientes Linux (es el caso del nombre del usuario o del grupo y de la contraseña). Sin embargo, hay atributos como el UID, GID, directorio *home*, etc., que no existen en el Directorio Activo simplemente porque Windows no los utiliza. Por ello, si queremos ofrecérselos a los clientes Linux, necesitaremos incorporarlos primero al Directorio Activo. Esto puede conseguirse mediante la modificación del *esquema* del directorio. Una vez añadidos los nuevos atributos a los objetos de tipo usuario y grupo, ya pueden crearse cuentas que contengan todos los datos necesarios para clientes tanto Windows como Linux.

Hasta la primera versión de Windows Sever 2003, existían dos formas de realizar la mo-

dificación del esquema del Directorio Activo para añadir los "atributos UNIX" a los objetos de tipo usuario y grupo: un parche de libre distribución denominado MKS AD4Unix y una suite de utilidades de Microsoft denominadas genéricamente Windows Services for UNIX (o SFU). En el primer caso, el parche realizaba la modificación del esquema y de la herramienta Usuarios y Equipos de Active Directory para incluir dichos atributos UNIX. En el caso de las SFU, además, se incluía un conjunto de servicios tales como cliente y servidor de NIS, NFS y Telnet, varios *shells*, utilidades comunes en UNIX, etc.

A partir de Windows Server 2003 R2, las SFU vienen de serie en el sistema en lugar de en un paquete aparte. Por tanto, no resulta necesario modificar el esquema del Directorio Activo para disponer de los atributos UNIX. Sin embargo, para poder acceder a esta funcionalidad, sí es necesario instalar el "servidor NIS", que incluye la pestaña "Atributos UNIX" para usuarios y grupos en la herramienta Usuarios y Equipos de Active Directory. Esto se realiza instalando un componente de Windows (desde el Panel de Control) denominado "Identity Management for UNIX" dentro de los "Servicios de Active Directory". En la figura Figura 5.1, “Pestaña de atributos UNIX en Usuarios y Equipos de Active Directory.” se muestran los elementos de esta pestaña. Para poder acceder a los atributos es necesario seleccionar el "dominio NIS" (que coincide por defecto con el nombre NetBIOS del dominio), y ya se está en condiciones de rellenar el resto de los atributos del usuario: UID, *shell*, directorio *home* y grupo primario (GID).

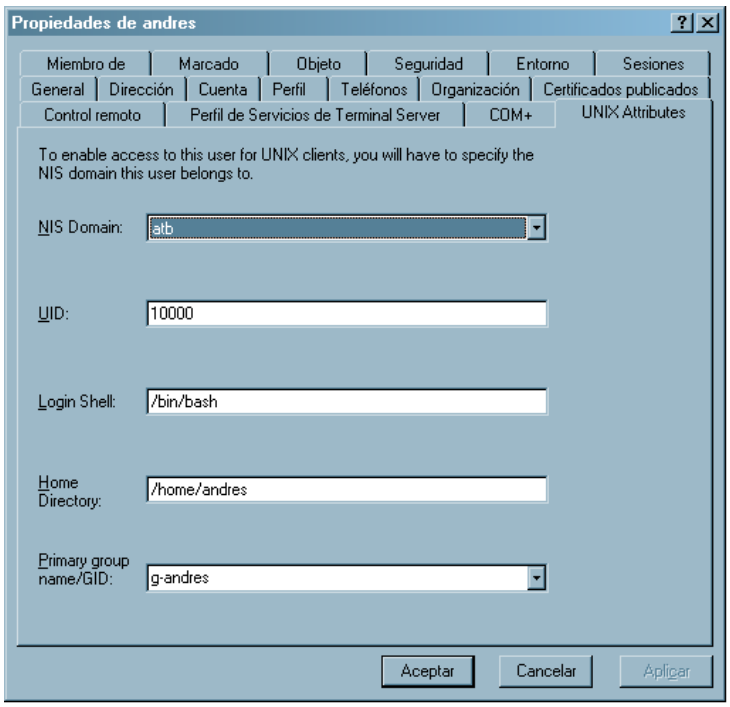

Figura 5.1. Pestaña de atributos UNIX en Usuarios y Equipos de Active Directory.

Como puede verse en la figura, la elección del grupo no es un número o identificador que pueda teclearse, sino un desplegable de los grupos para los que previamente hemos activado

sus atributos UNIX. Esto obliga a activar primero los atributos UNIX de los grupos (que queramos sean visibles desde UNIX) y luego hacer lo propio con los usuarios. A este respecto, es importante destacar que en el Directorio Activo existe una limitación que impide a un grupo llamarse igual que un usuario en el mismo dominio (aunque residan en unidades organizativas distintas). Esto dificulta la implementación de la estrategia de grupos privados que mantienen los sistemas Linux basados en RedHat (en el ejemplo, el grupo primario del usuario "andres" se ha denominado "g-andres" por este motivo).

5.3.2. Configuración de los clientes Linux

En la Sección 2.8, "Autenticación basada en OpenLDAP" se explica cómo es posible configurar un sistema Linux como cliente LDAP. Esta sección recuerda los princpales conceptos relacionados e introduce las peculiaridades que aparecen cuando el sistema Linux debe ser cliente de un servidor LDAP Windows (es decir, de un servicio de Directorio Activo).

5.3.2.1. Linux como cliente de OpenLDAP

En general, Linux separa el acceso interno a las cuentas de usuario/grupo y la autentificación de usuarios en dos bibliotecas diferentes, denominadas respectivamente *NSS* y *PAM*. En ambos casos, estas bibliotecas pueden configurarse dinámicamente para obtener sus datos de múltiples fuentes alternativas (ficheros locales, servidores NIS, servidores LDAP, etc.). Si deseamos configurar el sistema Linux como cliente LDAP, debemos establecer que, para ambas bibliotecas, la fuente de información debe ser un servidor LDAP.

Si el cliente es un sistema Linux basado en RedHat, y el servidor es otro sistema Linux ejecutando OpenLDAP, la configuración es realmente sencilla: primero hay que asegurarse que está instalado el paquete RPM `nss_ldap` (que contiene los módulos LDAP para PAM y NSS) y posteriormente debemos ejecutar la herramienta **authconfig** (también puede invocarse su versión en modo gráfico, **system-config-authentication**). En dicha herramienta hay que activar la recepción de información de cuentas y la autentificación de tipo LDAP, indicando en ambos casos la dirección del servidor y la *base* o sufijo LDAP. Internamente, esta herramienta modifica principalmente los siguientes ficheros:

- `/etc/nsswitch.conf`. Contiene la configuración del cliente NSS del sistema Linux. En nuestro caso, hay que comprobar que las líneas que configuran de dónde hay que leer la información de usuarios, contraseñas *shadow* y grupos incluyan la opción `ldap`:

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
```

- `/etc/pam.d/system-auth`. Contiene la configuración básica de autentificación del sistema Linux mediante PAM. La herramienta **authconfig** configura automáticamente los cuatro tipos de módulos (`auth`, `account`, `password`, y `session`) para que utilicen LDAP como alternativa:

```
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     [default=bad success=ok user_unknown=ignore service_err=ignore
system_err=ignore] /lib/security/$ISA/pam_ldap.so

password    required      /lib/security/$ISA/pam_cracklib.so retry=3 type=
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok
md5 shadow
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     optional      /lib/security/$ISA/pam_ldap.so
```

- `/etc/ldap.conf`. Contiene la configuración de las bibliotecas NSS y PAM cuando se utilizan contra un servidor LDAP. Si el servidor en cuestión es OpenLDAP, el fichero ya está preparado para interpretar adecuadamente los atributos LDAP de los objetos "usuario" y "grupo" del directorio como los atributos UNIX necesarios (UID, GID, contraseña, etc.), con lo que no es necesaria ninguna modificación de posterior.

5.3.2.2. Linux como cliente del Directorio Activo

Si el servicio LDAP contra el que el sistema Linux debe actuar es un Directorio Activo de Windows Server 2003, la configuración es similar a la explicada en la sección anterior, aunque existen un par de salvedades importantes que hay que considerar: en primer lugar, el fichero `/etc/ldap.conf` no es el adecuado y resulta necesario modificar su configuración, y en segundo el Directorio Activo no permite por defecto las consultas anónimas (que hacen los clientes LDAP Linux normalmente). A continuación se expone cómo solucionar ambos problemas. Por último, al final de la sección, se incluye un apartado donde se comenta las dos posibilidades de configurar el cliente de autenticación en el sistema Linux (biblioteca PAM): LDAP y Kerberos.

Tras la ejecución de la herramienta **authconfig** (igual que en la sección anterior), es necesario modificar *manualmente* el fichero `/etc/ldap.conf`, ya que no está preparado para interpretar correctamente los atributos de los objetos del directorio como los atributos UNIX que necesitan NSS y PAM. En particular, debemos asegurarnos que el fichero incluye las siguientes líneas (suponiendo que el dominio Windows en cuestión se denomina "admon.lab"):

```
# Your LDAP server.
host 158.42.170.14

# The distinguished name of the search base.
base dc=admon,dc=lab

# The distinguished name to bind to the server with.
# Optional: default is to bind anonymously.
binddn cn=linux,ou=special-users,dc=admon,dc=lab
```

```
# The credentials to bind with.
# Optional: default is no credential.
bindpw -Admon-

# Base search
nss_base_passwd dc=admon,dc=lab?sub
nss_base_shadow dc=admon,dc=lab?sub
nss_base_group dc=admon,dc=lab?sub

# Services for UNIX 3.5 mappings
nss_map_objectclass posixAccount User
nss_map_objectclass shadowAccount User
nss_map_objectclass posixGroup Group
nss_map_attribute uid sAMAccountName
nss_map_attribute uidNumber uidNumber
nss_map_attribute gidNumber gidNumber
nss_map_attribute loginShell loginShell
nss_map_attribute uniqueMember member
nss_map_attribute homeDirectory unixHomeDirectory

# pam configuration
pam_login_attribute msSFU30Name
pam_filter objectclass=User
ssl no
pam_password md5
```

En el fichero anterior, las líneas `nss_base_*` especifican en qué contenedores se ubican las cuentas de usuario y grupo, mientras que las líneas `nss_map_*` asocian adecuadamente los atributos de las cuentas de usuario y grupo a sus correspondientes del Directorio Activo. Finalmente, las líneas que comienzan por `pam_*` configuran PAM para que sea compatible con dicho servicio de directorio.

Para comprobar que efectivamente la asociación entre los atributos de los objetos del Directorio Activo y los atributos UNIX correspondientes es la adecuada, a continuación se muestran los atributos del usuario del Directorio Activo "andres", correspondientes al usuario de la figura Figura 5.1, “Pestaña de atributos UNIX en Usuarios y Equipos de Active Directory.”.

```
# andres, Desarrollo, admon.lab
dn: CN=andres,OU=Desarrollo,DC=admon,DC=lab
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: andres
givenName: andres
distinguishedName: CN=andres,OU=Desarrollo,DC=admon,DC=lab
instanceType: 4
whenCreated: 20070531144519.0Z
whenChanged: 20070601104631.0Z
displayName: andres
uSNCreated: 16398
memberOf: CN=proy1,OU=Desarrollo,DC=admon,DC=lab
memberOf: CN=g-andres,OU=Desarrollo,DC=admon,DC=lab
uSNChanged: 28711
name: andres
objectGUID:: GwlsqoY+Ykm5wZZFVtIKew==
userAccountControl: 512
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 128251773380156250
```

```
lastLogoff: 0
lastLogon: 128251789138750000
pwdLastSet: 128250963195312500
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAHCoq2isuNW66PFYOUgQAAA==
accountExpires: 9223372036854775807
logonCount: 8
sAMAccountName: andres
sAMAccountType: 805306368
userPrincipalName: andres@admon.lab
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=admon,DC=lab
dSCorePropagationData: 20070531144632.0Z
dSCorePropagationData: 20070531144632.0Z
dSCorePropagationData: 20070531144632.0Z
dSCorePropagationData: 16010108151056.0Z
uid: andres
msSFU30Name: andres
msSFU30NisDomain: admon
uidNumber: 10000
gidNumber: 10000
unixHomeDirectory: /home/andres
loginShell: /bin/bash
```

Por otro lado, existe un ligero problema relativo a la exploración anónima del directorio que debemos resolver: por defecto, el Directorio Activo no permite realizar búsquedas anónimas (de usuarios no autentificados) en la base de datos del directorio. Como la biblioteca NSS de Linux realiza ese tipo de búsquedas, esto puede ocasionar algunos problemas a la hora de visualizar los nombres de usuarios y grupos (por ejemplo, en el *prompt* del intérprete de órdenes o al realizar un **ls -l**). Para solucionarlo tenemos dos alternativas: configurar el Directorio Activo para que permita las consultas anónimas o configurar el cliente LDAP de Linux para que haga todas las consultas como un usuario conocido por el dominio Windows (ya que por defecto, el grupo `Usuarios Autentificados` tiene permisos de lectura en todo el directorio).

Ambas alternativas tienen sus problemas de seguridad. En el primer caso, admitir consultas anónimas en el Directorio Activo no es buena idea si el DC es accesible por red desde el exterior de la organización, puesto que el contenido del directorio quedaría público a todo el mundo. En el segundo caso, el usuario como el que se realizarían las consultas desde el cliente Linux debe tener sus credenciales (usuario y contraseña) en texto claro (sin cifrar) en el fichero `/etc/ldap.conf`, que posee permisos de lectura para cualquier usuario conectado localmente al sistema Linux.

Quizás la opción menos comprometida sería la segunda, pero utilizando además un usuario especial que sólo sirviera para ese fin, y cuya funcionalidad en los sistemas Linux/Windows del dominio estuviera restringida. Para ello es necesario crear un usuario en el Directorio Activo (no hace falta activar sus atributos UNIX) y especificar dicho usuario y su contraseña en el fichero `/etc/ldap.conf`, en las líneas `binddn` y `bindpw`, respectivamente. En el ejemplo de arriba, se ha utilizado un usuario denomindo "linux" (dentro de la unidad organizativa "special-users") cuya contraseña es "-Admon-".

Posteriormente, podemos restringir las acciones que dicho usuario pueda realizar en los sistemas Windows del dominio (mediante directivas de seguridad o GPOs) y en los sistemas Linux, especificando el fichero `/bin/false` como *shell* en sus atributos UNIX.

5.3.2.2.1. Autenticación LDAP vs. Kerberos

En la configuración de Linux como cliente de un dominio Windows descrita arriba, el sistema Linux se configura como cliente de cuentas (bibilioteca NSS) y de autenticación (biblioteca PAM) de los controladores del dominio a través del protocolo LDAP. Sin embargo, la forma nativa en que los sistemas Windows pertenecientes a un dominio realizan la *autenticación* es mediante el protocolo Kerberos. En el dominio, los DCs son servidores de Kerberos y todos los miembros de dicho dominio autentican los inicios de sesión utilizando dicho protocolo.

En los sistemas Linux, es posible configurar la biblioteca PAM para que utilice Kerberos y realice la autenticación de usuarios mediante este protocolo en los inicios de sesión. Para ello, se puede utilizar la misma herramienta de configuración descrita arriba (**system-config-authentication**), pero en la pestaña de autenticación debe seleccionarse la habilitación del protocolo Kerberos en lugar de LDAP. La configuración de Kerberos tiene básicamente tres apartados: el entorno o *realm*, el servidor KDC (*Kerberos Distribution Center*) y el servidor de administración. En el primer caso, debe teclearse el nombre completo (DNS) del dominio Windows pero **en mayúsculas**, y en los otros dos casos, el nombre completo o la dirección IP de uno de los servidores (DCs) del dominio. La Figura 5.2, “Configuración de la autenticación Kerberos en Linux.” muestra cómo realizar esta configuración para el dominio Windows "admon.lab".

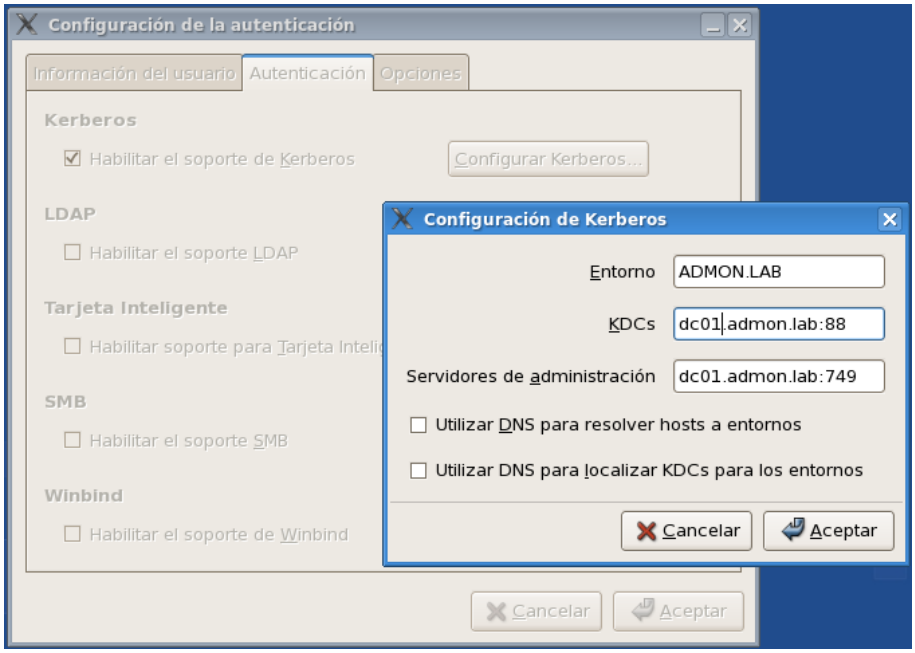

Figura 5.2. Configuración de la autenticación Kerberos en Linux.

La configuración de la biblioteca PAM a través de Kerberos respecto a LDAP tiene dos ventajas fundamentales: en primer lugar, el cliente Linux se comporta exactamente igual que un sistema Windows miembro del dominio respecto a la autenticación. Y en segundo lugar,

no resulta necesario configurar ningún mecanismo especial en el sistema Linux para encriptar las comunicaciones (mediante certificados), ya que el propio protocolo Kerberos garantiza la confidencialidad en la manipulación de las contraseñas utilizadas en la autenticación.

5.4. Integración de clientes Linux mediante Winbind

Hasta ahora hemos visto las posibilidades y características que ofrece un software como SFU para conseguir que diferentes clientes linux se integren en un dominio Windows. El método empleado no es en sí complicado pero requiere de una serie de conocimientos por parte del administrador. Este ha sido capaz de alterar el esquema del Directorio Activo para que soporte todas las características necesarias para que los clientes Linux se autentifiquen en un dominio Windows.

En esta sección se plantea una alternativa de integración que no requiere modificar ningún aspecto del Directorio Activo. Se trata de un servicio incluido en la suite Samba, denominado **winbind**, y que permite solventar el problema de la autenticación unificada.

Winbind utiliza una implementación UNIX de las llamadas RPC de Microsoft, PAM (Pluggable Authentication Modules) y NSS (Name Service Switch) para permitir que los usuarios de un dominio Windows se muestren y actúen como usuarios Linux en sistemas Linux.

Winbind proporciona tres funciones independientes:

- Autenticación de las credenciales de usuario vía PAM. Esta funcionalidad permite autentificar usuarios Windows, almacenados en el Directorio Activo, desde el sistema Linux cliente (de forma idéntica a cómo se describía en la Sección 5.3.2.2, "Linux como cliente del Directorio Activo").

- Resolución de nombres de usuario y grupo via NSS. Esta funcionalidad permite al sistema Linux utilizar de forma natural su biblioteca NSS para resolver nombres de usuario y de grupo contra el Directorio Activo

- Asociación dinámica entre SIDs de Windows y UID/GIDs de Linux. Por defecto, Winbind mantiene una base de datos local denominada *winbind_idmap.tdb*, en la que se almacenan las asociaciones entre los UIDs y GIDs de Linux y los SIDs de Windows. En particular, cuando un usuario no tiene un UID/GID local (es decir, cuando es un usuario almacenado en el Directorio Activo), Winbind genera automáticamente un nuevo UID/GID para este usuario, y almacena la asociación entre esos atributos UNIX y el SID del usuario en esta base de datos. Así, siempre que el usuario inicie sesión en el futuro en el sistema Linux, Winbind le asignará el mismo UID/GID (ya que el SID siempre es único), manteniendo la consistencia en el acceso al sistema de archivos. Como alternativa a este almacenamiento local, winbind puede también recuperar esta información de un repositorio de base de datos mediante la opción **idmap backend**. De esta manera, las asociaciones se pueden guardar de múltiples formas alternativas (en un servidor LDAP, en un fichero XML, etc.)

El funcionamiento de Winbind le permite unificar la administración de cuentas UNIX y Windows al permitir a un sistema Linux ser un miembro *completo* de un dominio Windows. Una vez dado este paso, el sistema Linux reconocerá a los usuarios Windows como si fueran usuarios Linux nativos, de la misma manera que se utiliza NIS o LDAP en un entorno puramente Linux, gracias a que Winbind se enlaza con la biblioteca de resolución de nombres (**NSS**) del sistema Linux. El resultado final es que cualquier programa de Linux que pregunte al sistema operativo por un nombre de usuario o grupo, este le preguntará al controlador de dominio Windows. La única característica especial a considerar cuando se utiliza Winbind como método de resolución de usuarios y grupos es que los nombres de usuarios y grupos vienen precedidos por el nombre del dominio Windows al que pertenecen: *DOMAIN\user*. Esto es necesario, ya que permite a Winbind determinar qué controlador de dominio tiene que utilizar para resolver el nombre de usuario o grupo en un entrono de red donde existen relaciones de confianza entre diferentes dominios. Además, Winbind proporciona un servicio de autenticación via *PAM* que proporciona la autenticación contra un dominio Windows de cualquier aplicación del sistema compatible con *PAM*.

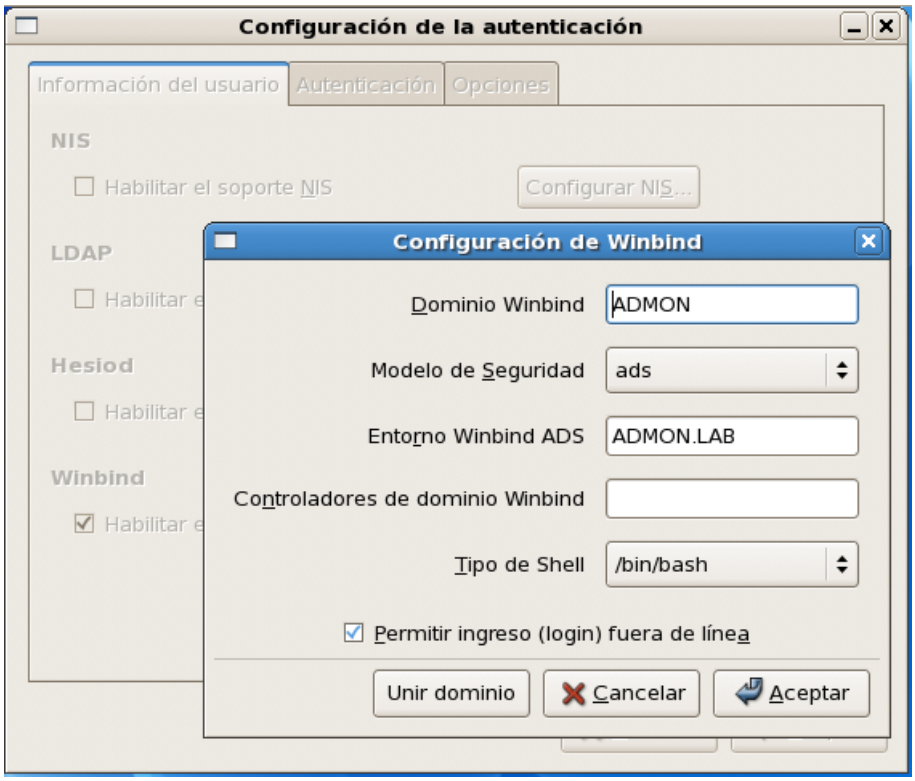

Figura 5.3. Configuración de la autenticación Kerberos en Linux.

En distribuciones basadas en RedHat (como CentOS), la configuración del sistema Linux como cliente del Directorio Activo mediante Winbind puede realizarse casi por completo mediante la herramienta gráfica de configuración de la autenticación (authconfig), tal como muestra la Figura 5.3, "Configuración de la autenticación Kerberos en Linux.". Alternativamente, se puede configurar siguiendo los pasos siguientes:

1. **Configuración del fichero /etc/samba/smb.conf**.

   Se necesita añadir varios parámetros al fichero *smb.conf* para controlar el comportamiento del servicio que implementa winbind (**winbindd**). El fichero de configuración que se muestra a continuación incluye las entradas necesarias en la sección [global].

```
[global]
# separate domain and username with '\', like DOMAIN\username
winbind separator = \
# use uids from 10000 to 20000 for domain users
idmap uid = 10000-20000
# use gids from 10000 to 20000 for domain groups
idmap gid = 10000-20000
# allow enumeration of winbind users and groups
winbind enum users = yes
winbind enum groups = yes
# give winbind users a real shell
template homedir = /home/winnt/%D/%U
template shell = /bin/bash
```

2. **Unir el cliente Linux al dominio Windows**.

   Todas las máquinas que desean participar en un *dominio de seguridad* necesitan ser miembros del dominio, incluidos los controladores de dominio (DCs). Para lograr este objetivo en un cliente Linux, tenemos que utilizar el comando *net*.

```
root# net rpc join -S PDC -U Administrador
```

   Este comando permite comunicar con un controlador de dominio via llamadas RPC, para ello es necesario que el servicio smbd este corriendo. La respuesta sera "*Joined the domain DOMAIN*"

3. **Ejecutar el servicio winbindd**.

   Winbind es un *demonio* (servicio) independiente de la suite SAMBA (smbd, nmbd). El comando que lo implementa se denomina *winbindd*. Para lanzarlo, tecleáríamos en la shell lo siguiente:

```
root# /usr/sbin/winbindd
```

   Para comprobar que dicho servicio resuelve los usuarios y grupos del dominio, podemos emplear la utilidad *wbinfo*.

```
root# wbinfo -u
CEO\Administrator
CEO\burdell
CEO\Guest
CEO\jt-ad
CEO\krbtgt
CEO\TsInternetUser

root# wbinfo -g
CEO\Domain Admins
CEO\Domain Users
CEO\Domain Guests
CEO\Domain Computers
CEO\Domain Controllers
```

```
CEO\Cert Publishers
CEO\Schema Admins
CEO\Enterprise Admins
CEO\Group Policy Creator Owners
```

4. **Configuración de la resolución de nombres (NSS)**.

Una vez estamos seguros que el servicio winbind resuelve los nombres de usuarios y grupos, hay que conseguir que cualquier aplicación del sistema sea capaz de resolverlos tambien. En un sistema Linux se utiliza la libreria NSS para resolver el nombre de los diferentes objetos del sistema. El fichero de configuración que nos permite definir las fuentes de datos de NSS es */etc/nsswitch.conf*.

```
passwd:  files winbind
shadow:  files winbind
group:   files winbind
```

Una vez salvado el fichero comprobaremos que el sitema resuelve los mismos usuarios y grupos que resuelve winbind.

```
root# getent passwd
CEO\burdell:15000:15003:Mary Orville:/home/winnt/CEO/burdell:/bin/bash
```

5. **Configuración de la autenticación (PAM)**.

La libreria PAM utiliza una interfaz de modulos apilables que permiten cambiar su comportamiento y definir las fuentes de información de autenticación. En un sistema CentOS Linux, el funcionamiento de la libreria PAM se centraliza en un único fichero denominado */etc/pam.d/system-auth*.

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_winbind.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so broken_shadow
account     sufficient    /lib/security/$ISA/pam_succeed_if.so uid < 100
quiet
account     [default=bad success=ok user_unknown=ignore] /lib/security/
$ISA/pam_winbind.so
account     required      /lib/security/$ISA/pam_permit.so

password    requisite     /lib/security/$ISA/pam_cracklib.so retry=3
password    sufficient    /lib/security/$ISA/pam_unix.so nullok
use_authtok md5 shadow
password    sufficient    /lib/security/$ISA/pam_winbind.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
```

Tras esta configuración básica de Winbind, debemos prestar especial atención a la forma particular en la que se realiza asociacion de identificadores de usuario Windows (SIDs) a

identificadores de usuarios Linux (UIDs/GIDs), que Winbind denomina *identity mapping*. Como se ha explicado anteriormente, Winbind dispone de varios mecanismos alternativos para realizar el almacenamiento de esas asociaciones. En este sentido, si existen varios sistemas Linux que utilicen Winbind, un mismo usuario podría recibir UIDs/GIDs diferente en cada sistema, lo cual puede ocasionar problemas si entre dichos sistemas Linux utilizan aplicaciones que confían en la unicidad de UIDs/GIDs, como el protocolo NFS, por ejemplo. Este problema puede resolverse mediante dos configuraciones de *identity mapping*: almacenar las asociaciones en una base de datos única compartida por todos los sistemas Linux (normalmente, un servidor de tipo LDAP), o bien utilizar una asociación algorítmica, que dado un SID de un usuario o grupo Windows, calcule siempre el mismo UID o GID.

Esta última alternativa se denomina IDMAP_RID, y en realidad lo que permite es que los sistemas Linux cliente generen el mismo UID/GID para un mismo **RID**, es decir, para la parte del SID que es única dentro de cada dominio. Por este motivo, IDMAP_RID no funciona correctamente cuando hay varios dominios relacionados (por ejemplo, en un bosque), ya que el algoritmo de asignación asignaría el mismo UID/GID a dos usuarios de dominios diferentes que tuvieran el mismo RID (aunque obviamente su SID sería distinto). Un ejemplo concreto de configuración con esta opción se muestra a continuación.

```
[global]
workgroup = ADMON
realm = ADMON.LAB
security = ADS
allow trusted domains = No
idmap backend = rid:ADMON=500-100000000
idmap uid = 500-100000000
idmap gid = 500-100000000
template shell = /bin/bash
winbind use default domain = Yes
winbind enum users = No
winbind enum groups = No
winbind nested groups = Yes
```

5.5. Directorios *home* centralizados

En un dominio Linux puro, es habitual centralizar los directorios de conexión (*home*) de los usuarios en uno de los sistemas del dominio, y compartirlo al resto de sistemas mediante el sistema de archivos en red NFS (ver Capítulo 3, *Sistema de Archivos en Red (NFS)*). De esta forma, los ficheros de cada usuario se mantienen centralizados y el usuario trabaja siempre con los mismos datos, independientemente del ordenador en que esté trabajando.

En el caso de un domino heterogéneo de sistemas Linux y Windows, existen numerosas alternativas para conseguir que los usuarios tengan disponibles sus datos de forma centralizada: exportar los directorios personales a través de SMB o de NFS, y en ambos casos, exportarlos desde un sistema Windows o Linux. En esta sección vamos a comentar estas alternativas (aunque algunas con mayor nivel de detalle que otras, por ser su configuración menos automática).

La primera alternativa sería que los directorios *home* residieran físicamente en un sistema Linux, que puede ser cliente de autentificación del dominio Windows, tal como se ha conta-

do en este capítulo. En este caso, es necesario crear manualmente los directorios y asignarles los atributos de protección adecuados (propietario, grupo y permisos). Si lo hacemos así, la configuración más fácil es exportar dichos directorios mediante NFS para los demás sistemas Linux, y mediante Samba para que se conviertan en recursos directamente accesibles desde sistemas Windows. En ambos casos (NFS para Linux y SMB para Windows) las configuraciones pueden automatizarse completamente (en el caso de Windows, mediante *scripts* de inicio de sesión) para que no necesiten ninguna intervención del usuario. Ambas configuraciones han sido comentadas respectivamente en el Capítulo 1, *Administración de dominios Windows 2003* y Capítulo 2, *Administración de dominios en Linux*, respectivamente.

La segunda alternativa sería que los directorios *home* residieran físicamente en un sistema Windows miembro del dominio (que puede ser alguno de los DCs, o bien cualquier otro miembro). En este caso, obviamente los directorios se exportarían como recursos SMB para el resto de sistemas Windows del dominio, que podrían acceder al mismo mediante una conexión a la unidad de red correspondiente. Respecto a los sistemas Linux que deben ser clientes de estos recursos, puede optarse por hacerlo mediante NFS o mediante el propio protocolo SMB. En el primer caso, es necesario utilizar una *suite* existente en Windows Server denominada "Windows Services for UNIX", que incluye entre otros servicios un servidor de NFS. En el segundo caso, es necesario que los clientes Linux sean configurados para admitir como directorios de trabajo de los usuarios los correspondientes recursos de red exportados directamente por SMB. El resto de la sección trata de explicar cómo puede hacerse esta configuración. En concreto, se comenta primero las configuraciones a realizar en el sistema Windows que alojará los directorios de los usuarios, y posteriormente se introducen dos alternativas para realizar el montaje desde los sistemas Linux: la utilización conjunta de **pam_mkhomedir** y **pam_mount** y la utilización de **pam_script**.

Lo primero que hay que configurar es el propio recuro compartido SMB, que puede exportar cualquier sistema Windows del dominio o cualquier sistema Linux con Samba correctamente configurado. Lo más sencillo resulta exportar un directorio o carpeta por cada usuario como un recurso de red que debe denominarse como el propio usuario. Si decidimos utilizar para ello un sistema Linux con Samba, sólo debemos activar el recurso denominado `homes`. Si lo realizamos desde un sistema Windows, debemos crear y compartir una carpeta personal por usuario (esta acción puede automatizarse mediante el uso de la pestaña `Perfil del usuario` en la herramienta **Usuarios y Equipos de Active Directory**). Si lo hacemos así, la conexión de dicho recurso como unidad de red del usuario en cualquier sistema Windows del dominio ya está resuelto.

5.5.1. Uso de pam_mkhomedir y pam_mount

En el caso de los clientes Linux, puesto que el directorio personal de cada usuario va a residir en un recurso SMB, necesitamos *montar* dicho recurso mediante el sistema de archivos `cifs` (ver Sección 4.7, “El sistema de ficheros CIFS para Linux”). En principio, el montaje debería hacerse en la ruta de su directorio *home* según los datos de su cuenta de usuario (esta ruta es, habitualmente, `/home/nombre_de_usuario`). Sin embargo, en el caso de un montaje de tipo SMB, esta opción plantea algunos inconvenientes relacionados con los ficheros/directorios de configuración del entorno gráfico que se guardan en el directorio de conexión de cada usuario. Por tanto, se ha optado por una alternativa un poco distinta: crear el *home* del usuario como un directorio local en cada sistema Linux y montar automáticamente su

unidad de red personal Windows en un subdirectorio particular (denominado, por ejemplo, `/home/nombre_de_usuario/Personal`).

Entre otros motivos, hemos elegido esta alternativa porque puede automatizarse completamente, sin requerir por parte del administrador la creación de los directorios de conexión locales en cada sistemas Linux ni el montaje de los recursos SMB de forma manual. Para ello pueden utilizarse dos extensiones de la biblioteca PAM, que se explican a continuación:

A. **pam_mkhomedir**. Mediante esta extensión (que no es necesario instalar, sino sólo activar), si el directorio de conexión de un usuario no existe en el sistema, éste se crea automáticamente cuando el usuario inicia sesión por primera vez.

   La configuración de esta biblioteca es muy sencilla. Basta añadir la siguiente línea en la sección `session` del fichero `/etc/pam.d/system-auth`:

```
session  required  /lib/security/$ISA/pam_mkhomedir.so  \
skel=/etc/skel/ umask=0022
```

B. **pam_mount**. Esta extensión viene como un paquete instalable (en formato RPM) con el mismo nombre. Una vez instalado, la configuración se lleva a cabo en dos ficheros: uno para establecer los parámetros del montaje (servidor, recurso, directorio local, etc.), y otro para instruir a PAM sobre cuándo realizar el montaje (cuándo ejecutar **pam_mount**). A continuación se explican los cambios concretos en ambos ficheros:

   En primer lugar, en el fichero `/etc/security/pam_mount.conf` basta añadir una línea al final, tomando como plantilla una existente que especifica cómo realizar montajes de recursos exportados por sistemas Windows Server 2003 pertenecientes a un dominio. Suponiendo que el dominio Windows se denomina `ADMON` y que el ordendor `SERVIDOR` exporta la carpeta personal de cada usuario como un recurso con el nombre de ese usuario, la línea quedaría así:

```
volume * smb SERVIDOR & /home/&/Personal uid=&,gid=&,dmask=0750, \
workgroup=ADMON - -
```

   Y en segundo lugar, el fichero `/etc/pam.d/system-auth` necesita incluir dos líneas, una de tipo `auth` y otra de tipo `session` incluyendo las llamadas al módulo **pam_mount**. La ubicación de ambas líneas entre las de su tipo es significativa para el correcto funcionamiento del módulo. Con todas las modificaciones realizadas, la configuración final del fichero `system-auth` sería:

```
#%PAM-1.0

auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        required      /lib/security/$ISA/pam_mount.so
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     [default=bad success=ok user_unknown=ignore \
service_err=ignore system_err=ignore] /lib/security/$ISA/pam_ldap.so
```

```
password    required      /lib/security/$ISA/pam_cracklib.so retry=3 \
type=
password    sufficient    /lib/security/$ISA/pam_unix.so nullok \
use_authtok md5 shadow
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     required      /lib/security/$ISA/pam_mkhomedir.so \
skel=/etc/skel/ umask=0022
session     optional      /lib/security/$ISA/pam_mount.so
session     optional      /lib/security/$ISA/pam_ldap.so
```

De esta forma, la primera vez que un usuario se conecta al sistema Linux, primero se crea su directorio de conexión y posteriormente se monta en un subdirectorio denominado `Personal` el recurso SMB que contiene su carpeta Windows de uso personal. Cuando el usuario se desconecta del sistema, el directorio permanece, aunque el recurso se desmonta auotmáticamente.

Por último, debemos tener en cuenta que el uso de estas extensiones de PAM (en especial, **pam_mkhomedir**) tiene algunos efectos secundarios que debemos considerar:

- Si pensamos permitir el acceso al sistema Linux de forma remota mediante **ssh**, tenemos que modificar ligeramente la configuración del *demonio* **sshd**, debido a una inconsistencia entre la implementación de este servicio y **pam_mkhomedir**. Para resolverlo, debemos descomentar una línea del fichero `/etc/ssh/sshd_config` (se encuentra casi al final) y configurarla como sigue:

  ```
  UsePrivilegeSeparation no
  ```

- Debido a otra inconsistencia entre **pam_mkhomedir** y la orden **su**, esta orden no funcionará bien cuando intentemos cambiar de usuario a uno que no tenga aún creado su directorio de conexión (es decir, cuando interviene **pam_mkhomedir**). La causa está en que el UID efectivo del proceso que crea el directorio del usuario no es el de `root`, y por tanto no tiene permisos para crear el directorio en `/home/`. Este problema puede solucionarse únicamente mediante permisos de escritura a todos los usuarios en `/home/`, cosa muy poco sensata en cualquier caso. Por tanto, si ese es el efecto que observa en su sistema Linux, no podrá utilizar **su - usuario** para usuarios que no tengan directorio *home* ya creado.

5.5.2. Uso de pam_script

Como alternativa al uso combinado de las bibliotecas de PAM **pam_mkhomedir** y **pam_mount** vistas arriba, esta sección explica otra de estas bibliotecas denominada **pam_script**, que permite la ejecución de *scripts* cada vez que un usuario inicia o abandona una sesión en el sistema Linux. Puesto que los scripts pueden ser libremente escritos por el

administrador, esta opción permite un máximo nivel de flexibilidad en la configuración del entorno de trabajo inicial del usuario, así como de las acciones automáticas que deben ejecutarse cuando el usuario finaliza su sesión.

La biblioteca **pam_script** permite la ejecución automática de un script en tres momentos diferentes durante el proceso de inicio/fin de sesión de un usuario en el servicio de PAM que se desee (login, ssh, gdm, kdm, etc.):

1. **Autenticación.** Si dentro del fichero de configuración del servicio PAM establecemos este módulo en la sección `auth`, cada vez que un usuario se autentifique utilizando dicho módulo se ejecutará un script, cuya ubicación por defecto es `/etc/security/onauth`.

2. **Inicio de sesión.** Si dentro del fichero de configuración del servicio PAM establecemos este módulo en la sección `session`, cada vez que un usuario inicie una sesión utilizando dicho módulo se ejecutará un script, cuya ubicación por defecto es `/etc/security/onsessionopen`.

3. **Fin de sesión.** Si dentro del fichero de configuración del servicio PAM establecemos este módulo en la sección `session`, cada vez que un usuario cierre una sesión utilizando dicho módulo se ejecutará un script, cuya ubicación por defecto es `/etc/security/onsessionclose`.

A modo de ejemplo, a continuación se muestra cómo debería quedar el fichero de configuración PAM del servicio **gdm** (pantalla de inicio en modo gráfico del gestor de ventanas Gnome) para poder utilizar **pam_script** en los tres casos descritos arriba. Se ha elegido un servicio específico como **gdm** en lugar de **system-auth** porque éste último es invocado desde todos los servicios que permiten conectarse al sistema (login, gdm, ssh, ftp, etc.) y de esta manera ilustramos que es posible asociar script de inicio con sólo alguna(s) forma(s) de iniciar sesión en el sistema. El fichero de configuración de este servicio se denomina `/etc/pam.d/gdm`, y quedaría así:

```
#%PAM-1.0
auth       required pam_env.so
auth       required pam_stack.so service=system-auth
auth       required     pam_script.so expose=1 runas=root
auth       required pam_nologin.so

account    required pam_stack.so service=system-auth
password   required pam_stack.so service=system-auth
session    required pam_stack.so service=system-auth
session    required     pam_script.so expose=1 runas=root
session    optional     pam_console.so
```

En principio, los scripts se ejecutan con los atributos de protección del usuario implicado, aunque se puede estableer que se ejecuten en el contexto de protección de otro usuario, mediante la opción `runas=usuario`. En el caso de **gdm** de arriba, los tres scripts se ejecutarán como `root`.

Por otro lado, los scripts siempre reciben dos parámetros como argumentos: el primer argumento es el nombre del usuario implicado en la autenticación/sesión, y el segundo argumento es el nombre del servicio para el que se está ejecutando el módulo PAM. Además,

existe una variable de entorno denominada PAM_AUTHTOK que contiene la contraseña de dicho usuario. Esto puede ser útil, por ejemplo, para realizar montajes de recursos SMB, que requieren de dicha contraseña. El valor de retorno del script es significativo: un valor igual a cero se entiende como éxito mientras que un valor diferente se supone fallido. Si el módulo PAM es requerido, un módulo que falle supone que el proceso de autenticación o inicio de sesión falla para el usuario. Por tanto, también se puede utilizar el script para realizar comprobaciones de cualquier tipo que permitan o impidan al usuario entrar al sistema.

A modo de ejemplo, a continuación se muestra un script simple, vinculado al módulo de autenticación (/etc/security/onauth), que crearía el directorio del usuario en el caso de que no existiera, y posteriormente montaría su directorio personal Windows dentro de dicho directorio. Este script, por tanto, realizaría las mismas acciones que los módulos **pam_mkhomedir** y **pam_mount** comentados anteriormente:

```
USER=$1
SERVICE=$2

if [ ! -d /home/$USER ]
then
    mkdir /home/$USER 2>/dev/null
    find /etc/skel -name ".*" -exec cp -r \{\} /home/$USER/ \;
    chown -R $USER.g-$USER /home/$USER 2>/dev/null
fi

if [ ! -d /home/$USER/Personal ]
then
    mkdir /home/$USER/Personal 2>/dev/null
    chown $USER.g-$USER /home/$USER/Personal 2>/dev/null
fi

mount -t cifs //dc01/$USER /home/$USER/Personal -o user=$USER, \
password=$PAM_AUTHTOK,workgroup=ADMON,uid=$USER,gid=g-$USER 2>/dev/null

exit 0
```

Para completar la configuración, debería implementarse también el script de cierre de sesión /etc/security/onsessionclose para desmontar el recurso SMB del directorio del usuario.

6

Integración de dominios mediante servidores Linux

Indice

6.1. Introducción

La integración de sistemas es la pieza angular que debe afrontar un administrador de sistemas. Cuando en una organización deben convivir diferentes sistemas operativos, el administrador debe facilitar a los usuarios la forma de acceder a los recursos independientemente de la plataforma que estos decidan utilizar.

Unas credenciales únicas facilitarán al usuario el acceso a la red e implicará una mayor seguridad y control del funcionamiento de la misma. Esto conlleva la creación de un repositorio único donde se almacena la información de cada elemento de la red. Máquinas, usuarios y servicios podrán ser fácilmente creados, modificados y eliminados si disponemos de un único punto de administración.

La implantación de un dominio de red que nos permita implementar los objetivos anteriores, obliga a que el administrador tenga que escoger las herramientas software que sean capaces de llevar dicho reto a cabo. La elección del software condicionara el modelo de dominio que se pueda ofrecer.

Desde el punto de vista de la administración de sistemas, suele denominarse dominio a un conjunto de equipos interconectados que comparten información administrativa (usuarios, grupos, contraseñas, etc.) centralizada. Ello requiere fundamentalmente la disponibilidad de (al menos) un ordenador que almacene físicamente dicha información y que la comunique al resto cuando sea necesario, típicamente mediante un esquema cliente-servidor. Por ejemplo, cuando un usuario desea iniciar una conexión interactiva en cualquiera de los ordenadores (clientes) del dominio, dicho ordenador deberá validar las credenciales del usuario en el servidor, y obtener de éste todos los datos necesarios para poder crear el contexto inicial de trabajo para el usuario. En Windows 2000, la implementación del concepto de dominio se realiza mediante el denominado Directorio Activo, un servicio de directorio basado en diferentes estándares como LDAP (Lightweight Directory Access Protocol}) y DNS (Domain Name System). En el mundo Linux, los dominios solían implementarse mediante el famoso Network Information System (NIS), del que existían múltiples variantes. Sin embargo, la integración de servicios de directorio en Linux ha posibilitado la incorporación de esta tecnología, mucho más potente y escalable que NIS, en la implementación de dominios.

La elección está hecha y en este capítulo veremos la implantación de un dominio que de cabida tanto a plataformas Linux como Microsoft Windows. Para implementar todos los servicios que deben ofrecerse (DNS, DHCP, LDAP, SMB, HTTP, SMTP) se ha elegido el Sistema Operativo Linux, corriendo sobre diferentes paquetes de software libre (Bind, OpenLDAP, Samba, NFS, Apache).

La pareja OpenLDAP-Samba nos permitirá crear un dominio de red donde tanto los clientes Linux como Microsoft Windows podrán acceder y compartir información facilitando al usuario la tarea de tener que buscar dicha información.

OpenLDAP es un software robusto, rápido y escalable, que no tiene nada que envidiar a otros servicios de directorio. Además su código es "libre" y altamente configurable a diferencia del Directorio Activo de Microsoft que es una implementación de un servidor LDAP preconfigurada para realizar una tarea determinada. El Directorio Activo viene con un conjunto

de herramientas administrativas personalizadas para ejecutar aplicaciones Windows. En cambio, la complejidad de OpenLDAP será apreciada por aquellos administradores que deseen construir un servicio de directorio personalizado.

Esa característica nos permitirá integrarlo con otra pieza de software de red llamada Samba y que implementa a la perfección las funciones necesarias para que las máquinas con sistema operativo Microsoft Windows se sientan como en su casa, en su dominio.

La principal característica de Samba es ofrecer servicios de impresión y de ficheros a clientes Windows. Conocidos son los "benchmarks" realizados entre Servidores Windows 2003 y servidores Samba. Pero Samba puede funcionar también como controlador de dominio para los clientes Windows, implementando todas las características de un dominio Windows NT.

Por tanto tenemos a nuestra disposición dos piezas de software libre que nos van a permitir implementar un dominio de seguridad que dé servicio tanto a clientes Linux (OpenLDAP, nss_ldap) como a clientes Windows (OpenLDAP, Samba).

6.2. Estructura del Servicio de Directorio

Como se puede leer en el capítulo dedicado a dominios Linux "*En el contexto de las redes de ordenadores, se denomina directorio a una base de datos especializada que almacena información sobre los recursos, u "objetos", presentes en la red (tales como usuarios, ordenadores, impresoras, etc.) y que pone dicha información a disposición de los usuarios de la red*".

En esta sección vamos a definir la estructura (Unidades Organizativas y objetos) del directorio que albergará la información de usuarios, máquinas y servicios que pretendemos ofrecer.

La raíz del directorio será un objeto de clase **dc** que albergará a las diferentes Unidades Organizativas:

```
# dsic2, upv, es
dn: dc=dsic2,dc=upv,dc=es
objectClass: dcObject
objectClass: organization
o: dsic2,dc=upv
dc: dsic2
```

La unidad organizativa encargada de almacenar la información de usuario se denominará Users y su definición en formato LDIF es la siguiente:

```
# Users, dsic2, upv, es
dn: ou=Users,dc=dsic2,dc=upv,dc=es
objectClass: organizationalUnit
ou: Users
```

La unidad organizativa encargada de almacenar la información de grupos se denominará Groups y su definición en formato LDIF es la siguiente:

```
# Groups, dsic2, upv, es
```

```
dn: ou=Groups,dc=dsic2,dc=upv,dc=es
objectClass: organizationalUnit
ou: Groups
```

La unidad organizativa encargada de almacenar la información de máquinas se denominará Computers y su definición en formato LDIF es la siguiente:

```
# Computers, dsic2, upv, es
dn: ou=Computers,dc=dsic2,dc=upv,dc=es
objectClass: organizationalUnit
ou: Computers
```

Un objeto de tipo usuario tendrá la siguiente definición en el Servicio de Directorio:

```
# usu1, Users, dsic2, upv, es
dn: uid=usu1,ou=Users,dc=dsic2,dc=upv,dc=es
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: sambaSAMAccount
cn: usu1
sn: usu1
uid: usu1
uidNumber: 1000
gidNumber: 513
homeDirectory: /home/usu1
loginShell: /bin/bash
gecos: System User
description: System User
sambaLogonTime: 0
sambaLogoffTime: 2147483647
sambaKickoffTime: 2147483647
sambaPwdCanChange: 0
displayName: System User
sambaSID: S-1-5-21-2758841116-2178560935-410125937-3000
sambaPrimaryGroupSID: S-1-5-21-2758841116-2178560935-410125937-513
sambaHomeDrive: H:
sambaAcctFlags: [U]
sambaPwdLastSet: 1099149278
sambaPwdMustChange: 1103901278
```

Como se puede apreciar este tipo de objeto hereda propiedades de varias clases. La clase **posixAccount** que define los atributos correspondientes a una cuenta de usuario Linux, la clase **sambaSAMAccount** que define los atributos correspondientes a una cuenta de usuario Windows.

En cambio para definir un objeto tipo grupo se utilizará la siguiente entrada de directorio:

```
# RRHH, Groups, dsic2, upv, es
dn: cn=RRHH,ou=Groups,dc=dsic2,dc=upv,dc=es
objectClass: posixGroup
objectClass: sambaGroupMapping
cn: RRHH
gidNumber: 1000
sambaSID: S-1-5-21-2758841116-2178560935-410125937-3001
sambaGroupType: 2
displayName: RRHH
```

La definición de un grupo hereda las propiedades de dos clases: **posixGroup**, **sambaGroupMapping**. Estas clases proporcionaran la información necesaria a los equipos Linux y

Windows.

6.3. Instalación del software

El software de servidor OpenLDAP puede ser bajado libremente de la web oficial de OpenLDAP [http://www.openldap.org]; en el momento de escribir estas líneas la versión oficial estable es la 2.4.7, pero en este curso seguiremos utilizando los paquetes que vienen por defecto con la distribución CentOS.

Para instalar el software utilizaremos la herramienta yum que es el administrador de paquetes por defecto utilizado en CentOS.

```
# yum install openldap-servers
```

Esto instalará el servidor LDAP y todas sus dependencias.

El software servidor de Samba también es libre y su última versión estable (3.0.28) puede ser encontrada en la web de Samba [www.samba.org]. En este documento se utilizará el paquete precompilado que distribuye CentOS Linux. De nuevo, para instalar el software utilizaremos yum:

```
# yum install samba
```

Para poder administrar o interactuar con el servicio de directorio y con el servidor Samba necesitamos alguna herramienta de fácil utilización que nos permita crear y modificar objetos en el directorio. En este curso se han elegido las utilidades de IDEALX denominadas smbldap-tools. Estas utilidades se pueden bajar de la siguiente dirección: https://gna.org/projects/smbldap-tools/

Estas utilidades son scripts en perl que dependen a su vez de algunos módulos perl: perl(Net::Ldap), perl(Crypt::SmbHash).

Al final del capítulo se puede encontrar una referencia a todos estos comandos.

6.4. Configuración del servidor OpenLDAP

Necesitamos configurar el servidor OpenLDAP para que actue como una base de datos SAM. Por tanto, para conseguir dicho objetivo, debemos de hacer posible que:

- OpenLDAP acepte la modificación al esquema de Samba.
- el servidor LDAP ejecute como base dc=admon.com
- se añadan las entradas mínimas para empezar a usarlo.

Para conseguir los objetivos anteriores, definiremos la estructura del directorio según el

siguiente DIT (LDAP Directory Information Tree).

```
dc=admon, dc=com
|
---- ou = Users         : cuentas de usuario tanto para linux como para Windows
|
---- ou = Computers     : cuentas de máquina para los sistemas Windows
|
---- ou = Groups        : cuentas de grupos tanto para linux como para Windows
|
---- ou = DSA           : cuentas espciales de sistema
```

Esta estructura es conforme con las recomendaciones del RFC 2307bis. Por tanto, la conjuncion de Samba y OpenLDAP nos permitirá almacenar la siguiente información:

- Cuentas de usuario Microsoft Windows utilizando la clase de objeto *sambaSAMAccount* (**samba.schema**)
- Cuentas de máquina Microsoft Windows utilizando la clase de objeto *sambaSAMAccount*.
- Cuentas de usuario UNIX utilizando las clases de objeto *posixAccount* y *shadowAccount* (**nis.schema**)
- Grupos de usuarios utilizando las clases de objetos *posixGroup* y *sambaGroupMapping*.
- Cuentas de seguridad utilizadas por software de clientes (Samba y Linux) que utilizan la clase de objeto *simpleSecurityObject* (**core.schema**)

Los ficheros de configuración del servidor OpenLDAP se encuentran bajo el directorio `/etc/openldap`. El fichero de configuración principal del servidor se denomina `/etc/openldap/slapd.conf`. El fichero de configuración de las herramientas clientes se llama /etc/openldap/ldap.conf, y los diferentes ficheros que configuran el esquema que ofrece el servicio de directorio se encuentran bajo el subdirectorio schema.

El fichero base de configuración sobre el que vamos a trabajar a lo largo del libro es el siguiente:

```
# Fichero /etc/openldap/slapd.conf
#

# Diferentes ficheros que definen partes del esquema
#
include  /etc/openldap/schema/core.schema
include  /etc/openldap/schema/cosine.schema
include  /etc/openldap/schema/inetorgperson.schema
include  /etc/openldap/schema/nis.schema
include  /etc/openldap/schema/samba.schema

# Ficheros que mantienen el pid del proceso servidor
# y los argumentos pasados
#
pidfile  /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

```
# Formato de la base de datos que almacenará el directorio
#
database ldbm

# Raíz(sufijo) del servicio de directorio
# dn y password del administrador del directorio
#
suffix  "dc=admon,dc=com"
rootdn  "cn=admin,dc=admon,dc=com"
rootpw  {SSHA}86kTavd9Dw3FAz6qzWTrCOKX/c0Qe+UV

# Ubicación de los datos físicos del servicio de directorio
#
directory /var/lib/ldap

# Indices
#
index objectClass           eq
index cn                    pres,sub,eq
index sn                    pres,sub,eq
index uid                   pres,sub,eq
index displayName           pres,sub,eq
index uidNumber             eq
index gidNumber             eq
index memberUID             eq
index sambaSID              eq
index sambaPrimaryGroupSID  eq
index sambaDomainName       eq
index default               sub
```

En el fichero de configuración del servidor OpenLDAP se aprecia la inclusión de un fichero de configuración del esquema que nos va a permitir definir los atributos necesarios para poder dar soporte a los objetos que requiere un cliente Windows. El fichero /etc/openldap/schema/samba.schema viene en el paquete Samba. Por tanto habrá que copiarlo de la ubicación `/usr/share/doc/samba-3.X.X/LDAP/samba.schema` al directorio `/etc/openldap/schema`.

Una vez definido el fichero de configuración de OpenLDAP y el esquema que va a soportar, el siguiente paso sería rellenar el directorio con la estructura básica que queremos implementar. Para ello sería necesario obtener el fichero en formato LDIF. Si utilizamos las herramientas comentadas anteriormente de IDEALX, existe un comando (smbldap-populate) que nos permite crear toda esa infraestructura en el servidor LDAP. Lo único necesario sería configurar el fichero /etc/smbldap-tools/smbldap.conf que define algunos parámetros necesarios para dicha operación

```
# smbldap-populate
```

6.5. Configuración del cliente Linux

La configuración de un cliente Linux para que se integre en un dominio LDAP, implica la modificación de varios ficheros de configuración. Básicamente, el cliente Linux debe de saber de donde obtener los usuarios y grupos del dominio y como autenticarlos.

Lo primero se consigue utilizando la librería NSS (Name Service Switch) que permite resolver nombres en el sistema y que para ello depende de un fichero llamado `/etc/nsswitch.conf` donde se definen las fuentes de datos para los usuarios y grupos que será capaz de ver el sistema Linux. Al editar dicho fichero, las líneas que controlan de donde se obtendrá la resolución de usuarios y grupos, son las siguientes:

```
passwd: files ldap
shadow: files ldap
group:  files ldap
hosts:  files dns wins
```

Estas líneas indican al sistema que para obtener la información de un usuario (UID,GID,HOME ...) primero consultará los ficheros del sistema (`/etc/passwd`,`/etc/shadow`) además de una fuente de datos LDAP.

La tarea de autenticar usuarios en un sistema Linux se relega a un módulo denominado PAM (Pluggable Authentication Module) que consiste en insertar una capa en medio de cualquier aplicación que necesite autenticación, independizando de este modo a la aplicación de la necesidad de saber donde se encuentran los datos de autenticación.

En un sistema CentOS Linux existe un fichero de configuración general de PAM que luego incluyen las aplicaciones que necesitan autentificarse. Este fichero se llama /etc/pam.d/system-auth. La modificación necesaria para que la fuente de datos sea un servidor LDAP se realiza de la siguiente forma:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     required      /lib/security/$ISA/pam_unix.so
account     [default=bad success=ok user_unknown=ignore service_err=ignore \
system_err=ignore] /lib/security/$ISA/pam_ldap.so

password    required      /lib/security/$ISA/pam_cracklib.so retry=3 type=
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok
md5 shadow
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so

session     required      /lib/security/$ISA/pam_limits.so
session     required      /lib/security/$ISA/pam_unix.so
session     optional      /lib/security/$ISA/pam_ldap.so
```

El módulo PAM que permite la autenticación se denomina **pam_ldap** y viene en el paquete **nss_ldap**, el cual implenta tanto el módulo pam_ldap como la librería necesaria (/

`lib/libnss_ldap-2.3.2.so`) para que el sistema de resolución de nombres (NSS) pueda leer la información de un servidor LDAP.

Estas dos piezas de software (pam_ldap, libnss_ldap) se apoyan en un fichero de configuración denominado /etc/openldap/ldap.conf donde se define a que servidor LDAP atacar, cual es la raíz del dominio etc ...

```
SIZELIMIT 200
TIMELIMIT 15
DEREF never

host  ldap.admon.com
base dc=admon,dc=com
binddn cn=admin,dc=admon,dc=com
bindpw not24get

pam_password md5

nss_base_passwd ou=Users,dc=admon,dc=com?one
nss_base_shadow ou=Users,dc=admon,dc=com?one
nss_base_group  ou=Groups,dc=admon,dc=com?one
```

En un sistema CentOS Linux, las tareas anteriores se podían haber realizado utilizando la herramienta de configuración de la autenticación denominada **system-config-authentication**, que permite al administrador definir la información necesaria para la librería NSS, así como para pam_ldap.

Hay que señalar que todo este proceso que se ha realizado en un cliente Linux, deberíamos hacerlo también en el servidor, para que este fuera capaz de ver los usuarios y grupos necesarios. Esto se debe a que cuando pasemos a configurar el servidor Samba, los usuarios Samba tienen que tener su correspondiente usuario Linux y si por tanto el servidor Linux no es capaz de ver los usuarios como tal Linux que es, difícilmente podrá hacer que Samba los sirva.

6.6. Configuración del servidor Samba

Todos los ficheros de configuración que tienen que ver con el servidor Samba se encuentran bajo el directorio `/etc/samba`. El fichero principal de configuración se denomina `/etc/samba/smb.conf`.

El objetivo es implantar un controlador de dominio Windows que obtenga la información de los usuarios de un servidor LDAP, para ofrecer la posibilidad a los clientes Windows que inicien sesión en el dominio. Por tanto habrá que instruir al servidor Samba que lea los usuarios de la base de datos LDAP.

Un fichero de configuración apropiado para conseguir dicho objetivo sería el siguiente:

```
[global]
        unix charset = LOCALE
        workgroup = DSIC2
        netbios name = groucho
        passdb backend = ldapsam:ldap://127.0.0.1
        username map = /etc/samba/smbusers
        log level = 2
        syslog = 0
```

```
        log file = /var/log/samba.log
        max log size = 50
        smb ports = 139 445
        name resolve order = hosts wins bcast
        wins server = 158.42.250.200
        time server = Yes
        show add printer wizard = No
        add user script = /usr/sbin/smbldap-useradd -a -m '%u'
        delete user script = /usr/sbin/smbldap-userdel -r %u
        add group script = /usr/sbin/smbldap-groupadd -p '%g'
        delete group script = /usr/sbin/smbldap-groupdel '%g'
        add user to group script = /usr/sbin/smbldap-groupmod -m '%u' '%g'
        delete user from group script = /usr/sbin/smbldap-groupmod -x '%u' '%g'
        set primary group script = /usr/sbin/smbldap-usermod -g '%g' '%u'
        add machine script = /usr/sbin/smbldap-useradd -w '%u
        passwd program = /usr/sbin/smbldap-passwd '%u'
        logon script = scripts\logon.bat
        logon path = \\%L\profiles\%U
        logon drive = X:
        domain logons = Yes
        preferred master = Yes
        ldap admin dn = cn=root,dc=dsic2,dc=upv,dc=es
        ldap group suffix = ou=Groups
        ldap idmap suffix = ou=Idmap
        ldap machine suffix = ou=Computers
        ldap passwd sync = Yes
        ldap suffix = dc=dsic2,dc=upv,dc=es
        ldap user suffix = ou=Users
        idmap backend = ldap:ldap://127.0.0.1
        idmap uid = 10000-20000
        idmap gid = 10000-20000
        printer admin = Administrator
        map acl inherit = Yes
        printing = cups
        printcap name = CUPS

[netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        guest ok = Yes
        locking = No

[profiles]
        comment = Profile Share
        path = /var/lib/samba/profiles
        read only = No
        profile acls = Yes

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        browseable = yes
        guest ok = no
        read only = yes
        write list = Administrator
```

Se puede verificar la corrección de este fichero ejecutando en el siguiente comando:

```
# testparm -s
```

Si el test ha ido bien, sería el momento de arrancar el servidor Samba y comprobar que se comporta como un verdadero controlador de dominio. Pero como hemos elegido como base de datos donde guardar la información del dominio el servidor LDAP, Samba tiene que tener un mecanismo de comunicación con el servidor LDAP.

En el fichero de configuración aparecen una serie de directivas que hacen mención a la configuración LDAP.

- **passdb backend = ldapsam:ldap://127.0.0.1**

  Esta directiva indica cual es la base de datos donde almacenar la información de cuentas (usuarios, máquinas y grupos). Los valores que puede tomar son: *tdbsam, ldapsam, mysql, XML*.

  En nuestro caso el soporte elegido es ldapsam, lo cual implica que toda la información estará en un servidor LDAP. Otro parámetro es la dirección del servidor: ldap://127.0.0.1.

- **ldap admin dn = cn=root,dc=dsic2,dc=upv,dc=es**

  Aquí indicamos cual es el administrador del servicio de directorio.

- **ldap user suffix = ou=Users**

  Define la Unidad Organizativa donde almacenar cuentas de usuario. La ubicación de esta OU será relativa a la raíz del directorio.

- **ldap group suffix = ou=Groups**

  Define la Unidad Organizativa donde almacenar cuentas de grupo. La ubicación de esta OU será relativa a la raíz del directorio.

- **ldap machine suffix = ou=Computers**

  Define la Unidad Organizativa donde almacenar cuentas de máquina. La ubicación de esta OU será relativa a la raíz del directorio.

El único parámetro que no aparece en el fichero de configuración y que es necesario para que el administrador pueda llevar a cabo las operaciones necesarias sobre el servidor, es la contraseña que le permita autentificarse.

En vez de definirlo en el fichero `/etc/samba/smb.conf` se almacena en un fichero especial con un formato particular y que se denomina `/etc/samba/secrets.tdb`. El siguiente comando permite introducir esa contraseña en el fichero:

```
# smbpasswd -w contraseña
```

Dicha contraseña debe ser la misma que aparece en el fichero de configuración del servidor LDAP (`/etc/openldap/slapd.conf`).

Ahora nos encontramos en disposición de arrancar el servidor Samba y comprobar que se ha creado el dominio.

```
#  smbclient -L localhost -U%
#  net getlocalsid
```

Si las cosas han funcionado bien, deberíamos obtener información sobre el dominio y el SID del mismo.

```
SID for domain DSIC2 is: S-1-5-21-3504140859-1010554828-2431957765
```

Si obtenemos una respuesta como la de arriba, procederemos a crear objetos en el dominio. Para ello nos basaremos en las utilidades que se instalaron a la hora de crear el servicio de directorio (smbldap-tools). Existen diferentes scripts que nos permitirán añadir, modificar y borrar usuarios, grupos y maquinas en el dominio par que tanto clientes Windows como Linux puedan autentificarse. Estos mismos scripts se utilizan para que automáticamente una máquina se dé de alta en el dominio.

```
add user script = /usr/sbin/smbldap-useradd -a -m '%u'
delete user script = /usr/sbin/smbldap-userdel -r %u
add group script = /usr/sbin/smbldap-groupadd -p '%g'
delete group script = /usr/sbin/smbldap-groupdel '%g'
add user to group script = /usr/sbin/smbldap-groupmod -m '%u' '%g'
delete user from group script = /usr/sbin/smbldap-groupmod -x '%u' '%g'
set primary group script = /usr/sbin/smbldap-usermod -g '%g' '%u'
add machine script = /usr/sbin/smbldap-useradd -w '%u
passwd program = /usr/sbin/smbldap-passwd '%u'
```

6.7. Configuración del cliente Windows

La configuración de un cliente Microsoft Windows para que se integre en un dominio creado por un servidor Samba, sigue los mismos pasos que si el controlador de dominio fuera un servidor Windows NT/2000/2003.

Lo único que habrá que hacer es ir a MiPC->Propiedades->Identificación de Red->Propiedades que nos permitirá cambiarle el nombre al PC o unirse a un dominio. Elegimos unirse a un dominio, el cual nos solicitará una cuenta de usuarios con privilegios administrativos y para ello utilizaremos la cuenta Administrator.

Una vez que se una al nuevo dominio, reiniciaremos la máquina y comprobaremos que todo ha ido bien. También sería necesario comprobar que ha sido creada la cuenta de máquina necesaria en el servidor Samba y que define la relación de confianza implícita que existirá entre la máquina y el dominio.

```
dn: uid=sem0101$,ou=Computers,dc=dsic2,dc=upv,dc=es
objectClass: top
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: sambaSamAccount
cn: sem0101$
sn: sem0101$
uid: sem0101$
uidNumber: 1004
gidNumber: 553
homeDirectory: /dev/null
loginShell: /bin/false
description: Computer
sambaSID: S-1-5-21-2758841116-2178560935-410125937-3008
sambaPrimaryGroupSID: S-1-5-21-2758841116-2178560935-410125937-2107
displayName: sem0101$
sambaPwdCanChange: 1099655465
```

```
sambaPwdMustChange: 2147483647
sambaPwdLastSet: 1099655465
sambaAcctFlags: [W          ]
```

Una vez que disponemos de un cliente Windows en el dominio sería factible poder administrar el dominio desde el, con las herramientas que Microsoft Windows nos proporciona: User Manager y Server Manager. Esto es posible, aunque con algunas salvedades. No todos los atributos no pueden ser manejados.

6.8. Conclusiones

Como se ha podido apreciar a lo largo de este capítulo, es posible implantar un dominio (conjunto de usuarios y máquinas) solo con software libre y que integre tanto a clientes Windows como Linux.

Si nuestras necesidades no requieren las capacidades extras que supone tener instalado un dominio Active Directory y nos es suficiente con las características que ofrece un dominio Windows NT, esta es una solución escalable, altamente configurable y barata.

En cuanto a la administración del dominio, recomendaría la utilización de las herramientas desarrolladas por IDEALX (smbldap-tools) que aunque en línea de comandos, nos permiten realizar todas la tareas de creación, modificación y borrado de cuentas en el dominio.

También existe la posibilidad de crear relaciones de confianza entre dominios, para poder otorgar de una forma fácil y eficiente privilegios a usuarios de otros dominios.

En definitiva hacer hincapié en que Samba+OpenLDAP es un binomio que promete seguir creciendo para poder ofrecer soluciones fiables al administrador de sistemas.

6.9. Comandos de referencia

```
[root@groucho root]# smbldap-useradd -?
(c) IDEALX 2004 (http://www.idealx.com)- Licenced under GPL
Usage: /usr/local/sbin/smbldap-useradd [-awmugdsckABCDEFGHMNPST?] username
  -o    add the user in the organazional unit (relative to the user suffix)
  -a    is a Windows User (otherwise, Posix stuff only)
  -w    is a Windows Workstation (otherwise, Posix stuff only)
  -i    is a trust account (Windows Workstation)
  -u    uid
  -g    gid
  -G    supplementary comma-separated groups
  -n    do not create a group
  -d    home
  -s    shell
  -c    gecos
  -m    creates home directory and copies /etc/skel
  -k    skeleton dir (with -m)
  -P    ends by invoking smbldap-passwd
  -A    can change password ? 0 if no, 1 if yes
  -B    must change password ? 0 if no, 1 if yes
  -C    sambaHomePath (SMB home share, like '\\PDC-SRV\homes')
  -D    sambaHomeDrive (letter associated with home share, like 'H:')
  -E    sambaLogonScript (DOS script to execute on login)
  -F    sambaProfilePath (profile directory, like '\\PDC-SRV\profiles\foo')
```

```
  -H    sambaAcctFlags (samba account control bits like '[NDHTUMWSLKI]')
  -N    canonical name
  -S    surname
  -M    local mailAddress (comma seperated)
  -T    mailToAddress (forward address) (comma seperated)
  -?    show this help message
```

```
[root@groucho root]# smbldap-userdel -?
(c) IDEALX 2004 (http://www.idealx.com)- Licenced under GPL
Usage: /usr/local/sbin/smbldap-userdel [-r?] username
  -r    remove home directory
  -R    remove home directory interactively
```

```
[root@groucho root]# smbldap-groupadd -?
(c) IDEALX 2004 (http://www.idealx.com)- Licenced under GPL
Usage: /usr/local/sbin/smbldap-groupadd [-agorst?] groupname
  -a   add automatic group mapping entry
  -g   gid
  -o   gid is not unique
  -r   group-rid
  -s   group-sid
  -t   group-type
  -p   print the gidNumber to stdout
  -?   show this help message
```

```
[root@groucho root]# smbldap-populate -?
(c) IDEALX 2004 (http://www.idealx.com)- Licenced under GPL
Usage: /usr/local/sbin/smbldap-populate [-abeiug?] [ldif]
  -u uidNumber first uidNumber to allocate (default: 1000)
  -g gidNumber first uidNumber to allocate (default: 1000)
  -a user       administrator login name (default: Administrator)
  -g user       guest login name (default: nobody)
  -e file       export ldif file
  -i file       import ldif file
  -?            show this help message
```

```
Este comando se encarga de manejar la base de datos SAM:

[root@groucho root]# pdbedit -?
Usage: [OPTION...]
  -L, --list                        list all users
  -v, --verbose                     be verbose
  -w, --smbpasswd-style             give output in smbpasswd style
  -u, --user=USER                   use username
  -f, --fullname=STRING             set full name
  -h, --homedir=STRING              set home directory
  -D, --drive=STRING                set home drive
  -S, --script=STRING               set logon script
  -p, --profile=STRING              set profile path
  -U, --user SID=STRING             set user SID or RID
  -G, --group SID=STRING            set group SID or RID
  -a, --create                      create user
  -r, --modify                      modify user
  -m, --machine                     account is a machine account
  -x, --delete                      delete user
  -b, --backend=STRING              use different passdb backend as default
                                    backend
  -i, --import=STRING               import user accounts from this backend
  -e, --export=STRING               export user accounts to this backend
  -g, --group                       use -i and -e for groups
  -P, --account-policy=STRING       value of an account policy (like maximum
                                    password age)
  -C, --value=LONG                  set the account policy to this value
  -c, --account-control=STRING      Values of account control
  --force-initialized-passwords     Force initialization of corrupt password
```

```
                                    strings in a passdb backend
  -z, --bad-password-count-reset    reset bad password count
  -Z, --logon-hours-reset           reset logon hours
```

```
El comando net es una herramienta de administración de Samba y servidores
remotos CIFS.

[root@groucho root]# net -h
No command: net
  net time              to view or set time information
  net lookup            to lookup host name or ip address
  net user              to manage users
  net group             to manage groups
  net groupmap          to manage group mappings
  net join              to join a domain
  net cache             to operate on cache tdb file
  net getlocalsid [NAME]        to get the SID for local name
  net setlocalsid SID   to set the local domain SID
  net changesecretpw    to change the machine password in the local secrets
database only
                        this requires the -f flag as a safety barrier
  net status            Show server status

  net ads <command>     to run ADS commands
  net rap <command>     to run RAP (pre-RPC) commands
  net rpc <command>     to run RPC commands

Type "net help <option>" to get more information on that option
Valid targets: choose one (none defaults to localhost)
        -S or --server=<server>         server name
        -I or --ipaddress=<ipaddr>      address of target server
        -w or --workgroup=<wg>          target workgroup or domain

Valid miscellaneous options are:
        -p or --port=<port>             connection port on target
        -W or --myworkgroup=<wg>        client workgroup
        -d or --debuglevel=<level>      debug level (0-10)
        -n or --myname=<name>           client name
        -U or --user=<name>             user name
        -s or --configfile=<path>       pathname of smb.conf file
        -l or --long                    Display full information
        -V or --version                 Print samba version information
        -P or --machine-pass            Authenticate as machine account
```

A

Nota Legal

Integración de Sistemas Linux/Windows

por Fernando Ferrer García y Andrés Terrasa Barrena

Versión 4.0, marzo 2009