

# **MANUAL DE ADMINISTRACIÓN DEL RADIUS DE TELEFÓNICA DATA PARA LOS SERVICIOS IP**

## **TELEFONICA DATA**

## CONTROL DE LAS MODIFICACIONES

<b>Versión</b>	<b>Descripción de las Modificaciones</b>	<b>Fecha</b>	<b>Visado por</b>
1.0	Creación del documento	13/01/2000	
1.1	Se recogen cambios de instalación y configuración introducidos por la nueva versión del Radius de TData (versión 4.01)	26/01/2000	
1.2	Se recogen cambios de instalación de la herramienta de desconexión introducidos por la nueva versión del Radius de TData (versión 4.01)	28/01/2000	
1.3	Se añaden los atributos Radius empleados en el servicio IV+Básico Delegado sobre tecnología Lucent para indicar la calidad de salida a Internet de un usuario y el número máximo de sesiones simultáneas	15/02/2000	

# ÍNDICE

<b>1. INTRODUCCIÓN .....</b>	<b>4</b>
<b>2. SERVIDOR RADIUS DE TELEFÓNICA DATA .....</b>	<b>5</b>
2.1. FUNCIONALIDADES DEL SERVIDOR RADIUS DE TELEFÓNICA DATA .....	5
2.2. REQUERIMIENTOS DE SISTEMA .....	6
2.3. INSTALACIÓN DEL SERVIDOR RADIUS DE TELEFÓNICA DATA .....	7
2.3.1. <i>Instalación para Solaris</i> .....	7
2.3.2. <i>Instalación para Linux</i> .....	7
2.3.3. <i>Instalación para Windows NT</i> .....	7
2.3.4. <i>Directorios de instalación</i> .....	7
2.3.5. <i>Procedimiento de instalación</i> .....	9
2.3.6. <i>Arranque del servidor Radius</i> .....	9
2.3.7. <i>Ejecutables del servidor Radius</i> .....	10
2.4. CONFIGURACIÓN DEL SERVIDOR RADIUS DE TELEFÓNICA DATA .....	15
2.4.1. <i>Configuración del servidor Radius para el servicio InfoVía Plus Directo</i> .....	20
2.4.2. <i>Configuración del servidor Radius para el servicio InfoVía Plus Básico (Modalidad Delegada)</i> .....	23
2.4.3. <i>Configuración del servidor Radius para la convivencia de ambos servicios</i> .....	25
2.5. INSTALACIÓN, CONFIGURACIÓN Y MANTENIMIENTO DE LA APLICACIÓN DE DESCONEXIÓN .....	26
2.5.1. <i>Instalación para Solaris</i> .....	26
2.5.2. <i>Instalación para Linux</i> .....	26
2.5.3. <i>Instalación para Windows NT</i> .....	27
2.5.4. <i>Directorios de instalación</i> .....	27
2.5.5. <i>Utilización de la herramienta de desconexión</i> .....	27
2.5.6. <i>Fichero de configuración</i> .....	28
2.6. PRESTACIONES DEL SERVIDOR RADIUS DE TELEFÓNICA DATA .....	29
2.6.1. <i>Prestaciones del RADIUS en Solaris</i> .....	29
2.6.2. <i>Prestaciones del RADIUS en Linux</i> .....	33
2.6.3. <i>Prestaciones del RADIUS en Windows NT</i> .....	37
2.6.4. <i>Conclusiones del estudio del servidor Radius de Telefonía Data</i> .....	38
<b>3. USO DE BASES DE DATOS EXTERNAS .....</b>	<b>39</b>
<b>4. SERVIDOR PROXY RADIUS DE TELEFÓNICA DATA .....</b>	<b>40</b>
4.1. DISTRIBUCIONES DEL SERVIDOR PROXY RADIUS.....	40
4.2. ASIGNACIÓN DE DIRECCIONES IP .....	41
4.3. CRITERIOS PARA EL REENVÍO DE PAQUETES .....	41
4.4. MAPEO Y FILTRADO DE ATRIBUTOS RADIUS .....	41
4.5. DETECCIÓN DE SESIONES COLGADAS EN EL SERVIDOR PROXY.....	41
<b>A ANEXO A: FUENTES PARA EL DESARROLLO DE LIBRERIAS DINAMICAS.....</b>	<b>42</b>
A.1 <i>Makefile</i> .....	42
A.2 <i>Fichero UsuExtPwd.c</i> .....	45
A.3 <i>Fichero EscribeFac.c</i> .....	51
<b>5. REFERENCIAS Y GLOSARIO.....</b>	<b>54</b>
5.1. REFERENCIAS.....	54
5.2. GLOSARIO.....	54

## 1. Introducción

El presente documento constituye el manual de instalación, configuración y mantenimiento del Radius de Telefónica Data, para los diferentes modos de operación que dicho Radius ofrece para los clientes, y que son:

- Servidor Radius propiamente dicho
- Acceso a bases de datos de usuarios externas
- Servidor proxy Radius

## 2. Servidor Radius de Telefónica Data

En este apartado se describe el uso del Radius de Telefónica Data como servidor Radius propiamente dicho.

### 2.1. Funcionalidades del servidor Radius de Telefónica Data

La singularidad del servidor Radius de Telefónica Data se basa en dos factores:

- Su interfuncionamiento con la Red IP para la prestación de los servicios InfoVía Plus Directo e InfoVía Plus Básico (Modalidad Delegada) está garantizado.
- Admite una gran variedad de configuraciones.

El servidor Radius de Telefónica Data es compatible con los servicios InfoVía Plus Directo e InfoVía Plus Básico (Modalidad Delegada). Ello sólo requiere unas pequeñas modificaciones en los ficheros de configuración del servidor Radius, como se indica en el apartado 2.4.

A continuación se describen las características principales del servidor Radius de Telefónica Data:

- **Multiplataforma:**

El software Radius se suministra para las plataformas más populares: SUN/Solaris 2.5.1, Linux y Microsoft Windows NT Server 4.0.

- **Bases de Datos de Usuarios:**

La base de datos de usuarios puede consistir en un simple fichero plano que se suministra por defecto, o bien, en una base de datos dbm, a la que se accede mediante una librería dinámica que también se incorpora por defecto en el servidor Radius.

Así mismo, el cliente puede utilizar bases de datos distintas a las ofrecidas por defecto, mediante el desarrollo de librerías dinámicas para el acceso a dichas bases de datos. El servidor Radius incorpora los fuentes necesarios para el desarrollo de dicha librería, tal y como se indica en el apartado 3 del presente documento.

- **Contabilidad:**

La información de contabilidad se guarda en un fichero denominado “detalle” en un directorio configurable.

- **Monitorización del servidor:**

Se genera un fichero “logfile” en el directorio en el que se ejecuta el servidor Radius, que contiene un registro de los mensajes de error producidos por el sistema.

Por otra parte, existe la posibilidad de depuración de la ejecución mediante los ficheros de trazas. Estas pueden tener distinto nivel de detalle, según se especifique en los parámetros de lanzamiento del servidor, o bien posteriormente, mediante la ejecución de un comando desde la línea de comandos. Para una información más detallada, consultar los apartados 2.3.7.1 y 2.3.7.4 del presente documento.

Existen comandos que permiten ver el estatus actual del servidor, gestionar las direcciones IP conectadas, incluso parar la ejecución del servidor. Puede verse información más detallada en el apartado 2.3.7.4 del presente documento.

Así mismo, puede simularse un cliente Radius, como se describe en el apartado 2.3.7.5.

- **Asignación de direcciones/Asignación de Calidades de Acceso a Internet:**

En el caso de que el servidor Radius se utilice para el servicio InfoVía Plus Directo, las direcciones IP se asignarán a partir de los datos contenidos en un fichero de configuración.

En el caso de que el servidor Radius se utilice para el servicio InfoVía Plus Básico (Modalidad Delegada), se podrán asignar distintas calidades de acceso a Internet a distintos usuarios.

- **Configuración de Clientes Radius:**

Para el servicio InfoVía Plus Básico (Modalidad Delegada), existe la posibilidad de configurar hasta 256 clientes Radius.

- **Herramienta de desconexiones:**

Por iniciativa del gestor del servidor Radius, se puede desconectar a cualquier usuario en base a su dirección IP.

Por último, indicar que en el apartado 2.6 del presente documento, pueden consultarse las prestaciones del servidor Radius de Telefónica Data, para cada una de las plataformas para las que se distribuye.

## 2.2. *Requerimientos de sistema*

Las distribuciones que existen actualmente del servidor Radius de Telefónica Data están disponibles para los siguientes sistemas operativos:

- Solaris 2.5.1
- Linux 2.0.36
- Windows NT Server 4.0

El servidor Radius puede instalarse en cualquier plataforma SUN o PC con las siguientes características:

- En plataforma SUN: Cualquier máquina (Sparc o Ultra)
- En plataforma PC: Pentium o superior

En ambos casos se requiere 32 MB de memoria RAM y 10 MB de memoria libre en disco.

Con objeto de obtener un rendimiento adecuado del servidor Radius de Telefónica Data, es recomendable que en la máquina donde se instale el servidor Radius corran únicamente el demonio Radius, las herramientas de accounting y la base de datos de usuarios.

Una posible forma en que el cliente puede cuantificar si el servidor Radius opera en condiciones óptimas, es simulando peticiones Radius al servidor, y midiendo el tiempo que el servidor Radius emplea en atenderlas, así como la tasa de llegadas que el servidor Radius ofrece en tales condiciones, debiéndose encontrar ambos valores dentro de un determinado margen considerado como óptimo. Actualmente está

siendo llevado a cabo el desarrollo de una herramienta que realice las funciones anteriormente descritas para evaluar el rendimiento del servidor Radius.

### 2.3. Instalación del servidor Radius de Telefónica Data

Se muestra a continuación el procedimiento de instalación del servidor Radius de Telefónica Data.

Se indica inicialmente cómo se instala el software Radius para cada uno de los sistemas operativos para los que se distribuye. Posteriormente se indica el procedimiento de instalación, así como los directorios de instalación y los ficheros ejecutables que se crean en dicho proceso, de forma común, puesto que son análogos para los tres sistemas operativos.

#### 2.3.1. Instalación para Solaris

El servidor Radius para Solaris se suministra en un paquete denominado "radiuspsi". Para realizar la instalación deberá ejecutarse como usuario root el comando:

```
pkgadd -d <directorio de instalación del paquete> <nombre del paquete>
```

El paquete instala los ficheros binarios, documentación y fuentes en el directorio /opt/radius.

#### 2.3.2. Instalación para Linux

El servidor Radius para Linux se suministra por medio de un fichero con formato tar denominado "radius\_lin.tar". Para instalar el servidor debe ejecutarse el siguiente comando:

```
tar -xvf radius_lin.tar
```

El paquete instala los ficheros binarios, documentación y fuentes en el directorio /radius para la versión 4.01 del servidor Radius, y en el directorio /opt/radius/ para versiones del servidor Radius anteriores a la indicada. A lo largo del presente documento, se hará referencia al nombre del directorio de instalación utilizado para la última versión del servidor Radius. En caso de que se disponga de alguna versión anterior del servidor Radius, el directorio que se creará en el proceso de instalación será el /opt/radius.

#### 2.3.3. Instalación para Windows NT

El servidor Radius se suministra en un fichero autoejecutable que se instala en el directorio c:\radius.

En el directorio c:\radius se instalan los ficheros binarios, documentación y fuentes del servidor Radius.

#### 2.3.4. Directorios de instalación

El paquete genera los directorios que se indican a continuación en el directorio /opt/radius para Solaris, /radius para Linux, y en el directorio c:\radius para Windows NT:

- **bin/** Contiene los ejecutables del servidor Radius

- **doc/** Contiene los ficheros radius-psi.pdf, atributos-radius.pdf, guia\_cpi.ps y el fichero migracion.pdf.
- **lib/** Contiene librerías y ficheros fuente de diversas utilidades. Se incluye un fichero de documentación denominado LEEME\_msj. Las librerías se emplean para poder usar diversas fuentes de datos para la información sobre los perfiles de usuario. Para el caso de Windows NT, estas librerías y ficheros fuente se encuentran en el directorio c:\radius\lib, para la versión 4.01 del servidor Radius, aunque pueden también encontrarse en algunas versiones anteriores en el directorio c:\radius\src.
- **raddb/** Contiene los ficheros de configuración necesarios para el funcionamiento del servidor.
- **bateria\_infovia/** Este directorio contiene un conjunto de ficheros de configuración de ejemplo, y un script denominado "bateria". Dicho script permite realizar automáticamente un conjunto de pruebas para el servidor Radius. Las pruebas se realizan localmente utilizando las herramientas simula y rad\_tool. Se simulan las peticiones que realizaría un cliente Radius para usuarios del servicio InfoVía Plus Directo. La ejecución deberá realizarse en este directorio, copiando previamente todos los ejecutables suministrados en el directorio bin. Si se ejecutan bateria y radiusd simultáneamente, ha de asegurarse que el puerto UDP en el fichero cfg\_local\_INFOVIA de bateria\_infovia es distinto del de radiusd. Para ejecutar el script de pruebas basta invocarlo desde la línea de comandos con:

```
./bateria
```

- **bateria\_delegado/** Este directorio contiene un conjunto de ficheros de configuración de ejemplo, y un script denominado "bateria". Dicho script permite realizar automáticamente un conjunto de pruebas para el servidor Radius. Las pruebas se realizan localmente utilizando las herramientas simula y rad\_tool. Se simulan las peticiones que realizaría el servidor proxy Radius de la Red IP para usuarios del servicio InfoVía Plus Básico (Modalidad Delegada). La ejecución deberá realizarse en este directorio, copiando previamente todos los ejecutables suministrados en el directorio bin. Si se ejecutan bateria y radiusd simultáneamente, ha de asegurarse que el puerto UDP en el fichero cfg\_local\_INFOVIA de bateria\_delegado es distinto del de radiusd. Para ejecutar el script de pruebas basta invocarlo desde la línea de comandos con:

```
./bateria
```

Además de los directorios anteriores, pueden ser de utilidad los siguientes ficheros de texto que se encuentran en el directorio /opt/radius para Solaris (/radius para Linux y c:\radius para Windows NT):

- **LEEME VERSION.** Notas sobre las distintas versiones del servidor Radius.
- **LEEME DELEGADO.** Documentación general sobre el servidor Radius para el PSI de la Red IP (funcionalidad, ficheros de configuración y procedimiento de instalación, entre otros)
- **INSTALL\_MENS.txt.** Detalle de los procedimientos de instalación y lanzamiento del servidor Radius.



Todas las actuaciones sobre los ficheros del Radius, ya sean de configuración o herramientas suministradas, se harán, para el caso de Solaris, con el usuario root.

### 2.3.5. Procedimiento de instalación

Los pasos a seguir se detallan a continuación:

1. Crear el directorio de configuración /etc/raddb para Solaris y Linux (c:\raddb para Windows NT). Como se comentó anteriormente, para el caso de Solaris, el directorio se creará como usuario root.
2. Copiar a este directorio los ficheros de configuración: cfg\_local\_INFOVIA, cfg\_OPERADOR\_INFOVIA, redIP\_delegado, usuarios\_INFOVIA, clientes\_INFOVIA, cfg\_clientes\_hw\_INFOVIA, cfg\_pool\_INFOVIA, cfg\_pool\_clie\_INFOVIA y diccionario\_INFOVIA. Los detalles de cada uno de estos ficheros serán explicados en el apartado 2.4.
3. Copiar también a este directorio el fichero radiusd (radiusnt para Windows NT) o radiusdll. radiusdll es la versión del demonio Radius que utiliza librerías dinámicas.
4. Crear el directorio de contabilidad /usr/adm/radacct para Solaris y Linux (c:\radacct para Windows NT). Como se comentó anteriormente, para el caso de Solaris, el directorio se creará como usuario root. Aquí se generarán automáticamente los ficheros de "detalle", los cuales aparecerán en los directorios de cada cliente Radius (cada una de las direcciones IP asignadas al servidor proxy Radius de la Red IP para el caso del servicio InfoVía Plus Básico (Modalidad Delegada), o la dirección IP del Servidor de Túneles del cliente, para el caso del servicio InfoVía Plus Directo).
5. Crear un directorio para almacenar los ejecutables del servidor Radius, accesible desde donde se invoque la ejecución del servidor Radius (puede dejarse por defecto, /opt/radius/bin para el caso de Solaris, /radius/bin para Linux, y c:\radius\bin para Windows NT).
6. En Unix se puede modificar el script de arranque para que se inicie el servidor Radius al arrancar el sistema.

### 2.3.6. Arranque del servidor Radius

Se especifica a continuación el comando que debe ejecutarse para lanzar el servidor Radius:

- Solaris:

```
/opt/radius/bin/radiusd -d <directorio de configuración> -a <directorio de accounting>
```

Si se desean emplear librerías dinámicas, deberá lanzarse el ejecutable radiusdll con el siguiente comando:

```
/opt/radius/bin/radiusdll -d <directorio de configuración> -a <directorio de accounting>
```

- Linux:

```
/radius/bin/radiusd -d <directorio de configuración> -a <directorio de accounting>
```

Si se desean emplear librerías dinámicas, deberá lanzarse el ejecutable radiusdll con el siguiente comando:

```
/radius/bin/radiusdll -d <directorio de configuración> -a <directorio de accounting>
```

- Windows NT:

*start radiusnt -d <directorio de configuración> -a <directorio de accounting>*

### 2.3.7. Ejecutables del servidor Radius

Los ficheros ejecutables del servidor Radius, situados en el directorio /opt/radius/bin para Solaris, /radius/bin para Linux y c:\radius\bin para Windows NT son los siguientes:

#### 2.3.7.1. *radiusd o radiusnt*

*radiusd* es el fichero ejecutable del servidor Radius para Solaris y Linux, siendo *radiusnt* el fichero ejecutable del servidor Radius para Windows NT. Las opciones con las que se ejecuta son las siguientes:

Para Solaris y Linux:

*radiusd [-d <directorio\_configuracion\_radius>] [-a <directorio\_accounting\_radius>] [-u <fichero de usuarios\_radius>][-x][-X][-v][-p]*

Para Windows NT:

*start radiusnt [-d <directorio\_configuracion\_radius>] [-a <directorio\_accounting\_radius>] [-u <fichero de usuarios\_radius>][-x][-X][-v][-p]*

Los valores por defecto para los parámetros son los siguientes:

- **<directorio\_configuracion\_radius>**. Directorio en el que se encuentran los ficheros de configuración del servidor Radius, que se describen en el apartado 2.4. Por defecto, se tomará el directorio /etc/raddb para Solaris y Linux, y c:\raddb para Windows NT.
- **<directorio\_accounting\_radius>**. Directorio donde se deposita la información de accounting. En dicho directorio se creará un subdirectorio cuyo nombre será la dirección IP de cada uno de los clientes Radius. En dicho subdirectorio se creará un fichero denominado “detalle”, que contiene la información de accounting. Por defecto, se tomará el directorio /usr/adm/radacct para Solaris y Linux y c:\radacct para Windows NT.
- **<fichero de usuarios\_radius>**. Fichero donde se definen los usuarios que tienen acceso al cliente, así como sus características. Por defecto se tomará el fichero usuarios\_INFOVIA en el directorio /etc/raddb para Solaris y Linux, y c:\raddb\usuarios\_INFOVIA para Windows NT.
- **-x**. Si se indica esta opción se registrarán las trazas de Nivel 1 sobre el funcionamiento del sistema. Los ficheros de trazas se crearán en el mismo directorio donde se instale el ejecutable *radiusd* o *radiusnt*, y se denominan TRAZAS\_INFOVIA y TRAZAS\_INFOVIA\_ERR.
- **-X**. Con esta acción se permiten activar las trazas de Nivel 2 (trazas de Nivel 1 más los paquetes IP).
- **-v**. Nos permite ver la versión del servidor Radius.
- **-p**. Introduciendo este parámetro se permite actualizar un password utilizando la herramienta *radpass*.

Por último, indicar que en máquinas Unix, el uso de las opciones `-x`, `-X`, impide que el `radiusd` actúe como demonio.

En el caso de Windows NT, el ejecutable del Radius (`radiusnt`) deberá localizar la librería `RadInfovia.dll`.

### 2.3.7.2. *radiusdll*

Es el fichero ejecutable del Radius que usa una librería dinámica. Se puede usar una librería dinámica para poder acceder a una fuente de datos de perfiles de usuarios distinta a la que se ofrece por defecto (en forma de fichero plano, el denominado `usuarios_INFOVIA`). Para Solaris y Linux, se ofrece por defecto una librería dinámica para el acceso a bases de datos `dbm`.

El usuario puede desarrollar la suya propia con los fuentes de las mismas que se proporcionan en el directorio `/opt/radius/lib` de la distribución para Solaris y en el directorio `/radius/lib` para Linux.

En dichos directorios se ofrecen las siguientes librerías dinámicas. La diferencia entre ambas es que la segunda utiliza una base de datos con formato `dbm`.

- `libInfovia.so.1`
- `libInfoviadbm.so.1`

La librería que se use, deberá ser almacenada en el directorio `/usr/lib` con el nombre `libInfovia.so`.

Si se utiliza la base de datos `dbm`, cada vez que se incluya un usuario en el fichero `usuarios_INFOVIA` habrá que utilizar el programa `builddb`.

Este ejecutable no se instala en la versión para Windows NT.

### 2.3.7.3. *radpass*

El proceso `radpass` permite cambiar la `password` de un usuario en el archivo `usuarios_INFOVIA`. Para ello se ejecuta `radpass` en una máquina que cumpla las siguientes características:

- Que en el fichero `/etc/services` se encuentre definido el puerto por el que el cliente atiende las peticiones Radius.
- Que en el fichero `/etc/hosts` esté definida la dirección de la máquina que actúa como servidor Radius del PSI con el alias `radius-server`.

Este ejecutable no se instala en la versión de Windows NT.

### 2.3.7.4. *rad\_tool*

Se ha creado esta aplicación de usuario que permite las siguientes opciones:

```
rad_tool [-d <directorio de configuración>][parar_radius] [ver_status]
[desconectar_usuario<dir_ip>] [reutilizar_direccion<dir_ip>]
[liberar_direccion_delegada<dir_ip>] [usuarios_conectados] [activar_trazas_1]
[desactivar_trazas_1] [activar_trazas_2] [desactivar_trazas_2]
```

- **-d <directorio de configuración>** permite buscar el archivo de configuración `cfg_OPERADOR_INFOVIA` en un directorio diferente del que se lanza esta aplicación. Esta opción sólo está disponible para la versión 4.01 del servidor

Radius, no para versiones anteriores. El resto de opciones de la herramienta rad\_tool están disponibles para todas las versiones del servidor Radius.

- **parar\_radius** fuerza la parada del servidor Radius.
- **ver\_status** permite consultar el estado del servidor Radius en el momento actual, es decir, usuarios conectados y direcciones IP que quedan libres. rad\_tool con esta opción crea los ficheros: Estado\_del\_cliente\_aa.bb.cc.dd (donde aa.bb.cc.dd es la dirección de cada cliente Radius), con los usuarios conectados en ese momento, y Estado\_del\_pool\_X (donde X es el identificativo de cada pool) con las direcciones libres en ese pool.
- **desconectar\_usuario <dir\_ip>** desconecta un usuario del PSI con dirección IP dir\_ip.
- **reutilizar\_direccion <dir\_ip>** libera la dirección IP dir\_ip del pool de direcciones de un PSI.
- **liberar\_direccion\_delegada <dir\_ip>** desactiva la dirección IP dir\_ip de las asignadas por el servidor proxy Radius de la Red IP.
- **usuarios\_conectados** busca en los logs de accounting (archivos que almacenan la información de accounting para una dirección IP dada del servidor proxy Radius de la Red IP) las direcciones IP asignadas por el servidor proxy Radius de la Red IP. Esta opción resulta útil para conocer qué direcciones pueden liberarse con liberar\_direccion\_delegada. Los archivos logs se guardan bajo el directorio ./<IP\_Cliente\_Radius> y con el nombre "detalle" (donde IP\_Cliente\_Radius es uno de los parámetros tal como figura en el fichero cfg\_OPERADOR\_INFOVIA). Si se emplea junto a la opción "-d", los logs de accounting se buscarán en el directorio:  
<directorio\_configuracion>/<IP\_Cliente\_Radius>
- **activar\_trazas\_1, activar\_trazas\_2** activa las trazas de nivel 1 y 2 respectivamente.
- **desactivar\_trazas\_1, desactivar\_trazas\_2** desactiva las trazas de nivel 1 y 2 respectivamente.

Para usar esta utilidad, es preciso que aparezca en el mismo directorio el fichero de configuración cfg\_OPERADOR\_INFOVIA, que se encuentra en el directorio /etc/raddb para Solaris y Linux y c:\raddb en Windows NT. En el caso de querer utilizar la opción de desconectar direcciones delegadas, además se hace necesario instalarla en la misma máquina donde resida el Radius de accounting, puesto que la dirección IP asignada a un usuario sólo se observa en los logs de accounting, y en el directorio que se haya configurado como directorio de accounting (por defecto el directorio /usr/adm/radacct para Solaris y Linux, y c:\radacct en Windows NT).

Adicionalmente, si se desean utilizar las opciones parar\_radius, ver\_status, desconectar\_usuario, reutilizar\_direccion, activar\_trazas\_1, activar\_trazas\_2, desactivar\_trazas\_1, desactivar\_trazas\_2, en la versión 4.01 del servidor Radius, debe incluirse la dirección IP del interfaz Ethernet de la máquina donde se encuentra instalado el servidor Radius, en el fichero clientes\_INFOVIA o bien, en el fichero clientes\_hw\_INFOVIA, con identificador 1000. A continuación se muestra un ejemplo:

```
###clientes_INFOVIA###
```

```

      1              2              3              4
=====
<IP_PSI>          <SECRETO_PSI>          <UDP_PSI>          1000

```

donde:

<IP\_PSI> es la dirección IP del interfaz Ethernet de la máquina donde se encuentra instalado el servidor Radius del PSI.

<SECRETO\_PSI> es el secreto asociado a la herramienta de monitorización del servidor Radius del PSI.

<UDP\_PSI> es el puerto UDP del servidor Radius local.

Estos parámetros son los mismos que han de aparecer en el fichero `cfg_OPERADOR_INFOVIA`.

Para versiones anteriores a la versión 4.01 del servidor Radius, para poder utilizar las opciones de la herramienta `rad_tool` anteriormente indicadas, debe incluirse el cliente "localhost" en el fichero `clientes_INFOVIA` o bien, en el fichero `clientes_hw_INFOVIA`, con identificador 1000. A continuación se muestra un ejemplo:

```
###clientes_INFOVIA###
```

```

      1              2              3              4
=====
<IP_localhost>  <SECRETO_PSI>          <UDP_PSI>          1000

```

donde:

<IP\_PSI> es la dirección IP del cliente "localhost"

<SECRETO\_PSI> es el secreto asociado a la herramienta de monitorización del servidor Radius del PSI.

<UDP\_PSI> es el puerto UDP del servidor Radius local.

Estos parámetros son los mismos que han de aparecer en el fichero `cfg_OPERADOR_INFOVIA`.

### 2.3.7.5. *simula\_delegado*

Esta aplicación simula el funcionamiento de un cliente Radius situado en la misma máquina que el servidor Radius del PSI. Para ello, en los ficheros de configuración, será necesario añadir la dirección IP del interfaz Ethernet de la máquina donde se encuentre instalado el servidor Radius, para la versión 4.01 de dicho servidor.

Para versiones anteriores del servidor Radius, habrá de añadirse un cliente "localhost" en los ficheros de configuración del servidor.

Uso:

`simula <Auth | Start | Stop > <Archivo>`

- `Auth Archivo` para simular un paquete de petición de autenticación de un cliente Radius al PSI. El perfil del usuario para el que se solicita la autenticación

está en <Archivo>. <Archivo> también contiene los datos de ese usuario para los paquetes de Accounting: Start y Stop.

- Start Archivo para simular un paquete de comienzo de tarificación para el usuario almacenado en <Archivo>.
- Stop Archivo para simular un paquete de liberación de la conexión y fin de tarificación del usuario almacenado en <Archivo>.

## 2.4. Configuración del servidor Radius de Telefónica Data

Se describen en este apartado los ficheros de configuración del servidor Radius de Telefónica Data, y la correcta configuración de los mismos, en función del servicio para el que se esté utilizando el servidor Radius, puesto que dependiendo del cliente Radius que realiza las peticiones sobre el servidor Radius, variará el comportamiento del mismo.

Los ficheros de configuración del servidor Radius, deberán aparecer en el directorio de configuración /etc/raddb para Solaris y Linux y en el directorio c:\raddb para Windows NT, o bien, en el directorio que se indique como parámetro en el lanzamiento del servidor.

Antes de pasar a analizar los ficheros necesarios y la configuración de los mismos para cada servicio en concreto, se presentan cada uno de los ficheros de configuración, su función y su formato.

- **cfg\_LOCAL\_INFOVIA.** En este fichero se indican diversos parámetros relativos a la configuración del servidor Radius.

FICHERO: cfg\_local\_INFOVIA

=====

Este fichero contiene los datos de configuración de un servidor RADIUS

El primer campo indica el tipo de dato descrito:

UDP==> Indica el puerto UDP usado por ese proceso para escuchar peticiones de autentificación. Si falta este campo se tomará el puerto definido en el /etc/services. Si no está definido en el /etc/services se tomará como puerto el valor 1645

El puerto por el que escuchará peticiones de contabilidad siempre será un número mayor que el anterior (ej: 1646)

ALIAS ==> Nombre del servicio soportado en el servidor local. Se pueden asignar cualquier número de alias al servicio. Por defecto, el nombre del servicio es el nombre de la máquina.

HORA ==> Indica el tiempo en segundos durante el cual, una vez liberada una dirección IP, esta no puede ser reutilizada.

HARDWARE\_INFOVIA\_OMISION. Si esta variable toma el valor 1, se permitirán clientes Hardware, en otro caso, deberá especificarse específicamente en cada perfil de usuario

IDENTIFICADOR	VALOR
UDP	Puerto de escucha del servidor Radius
ALIAS	prueba
HORA	Tiempo en sg
HARDWARE_INFOVIA_OMISION	1

- **clientes\_INFOVIA.** En este fichero se recogen los datos de las direcciones IP de los clientes (máquinas IP) con los que va a comunicarse el servidor Radius.

FICHERO: clientes\_INFOVIA

=====

Este fichero contiene una lista de clientes (máquinas IP) que están autorizadas a lanzar preguntas Radius sobre este servidor y a recibir respuestas Radius desde este servidor.

1. El primer campo es el dirección IP del cliente (máquina IP)
2. El segundo campo es la palabra clave asociada a ese cliente (máquina IP)
3. El tercer campo es el puerto UDP por el que escucha ese cliente (máquina IP)
4. El cuarto campo es el identificador del cliente (máquina IP)

1	2	3	4
Dir_IP	máquina IP	CLAVE	Puerto UDP
			id. cliente

=====

- **clientes\_hw\_INFOVIA.** En este fichero se especifican los datos de las direcciones IP de los clientes hw con los que va a comunicarse el servidor Radius.

FICHERO: clientes\_hw\_INFOVIA

=====

Este fichero contiene una lista de clientes hw que están autorizados a lanzar preguntas Radius sobre este servidor, y a recibir respuestas Radius desde este servidor.

1. El primer campo es la dirección IP del cliente hw.
2. El segundo campo es la palabra clave asociada a ese cliente hw.
3. El tercer campo es el puerto UDP por el que escucha ese cliente hw.
4. El cuarto campo es el identificador del cliente hw. Será distinto a los utilizados en clientes\_INFOVIA.

1	2	3	4
Dir_IP	cliente hw	CLAVE	Puerto UDP
			id. cliente

=====

- **cfg\_pool\_INFOVIA.** Este fichero recoge los datos sobre los diferentes “pools” definidos. Estos datos incluyen la dirección IP de los mismos, su tamaño, identificador y su nivel de privilegio.



FICHERO: cfg\_pool\_INFOVIA

=====

Este es el fichero que va a especificar los pools utilizados así como su tamaño, desde dónde empiezan y el nivel de usuario que está asociado a ese pool.

id\_pool-> Número que identifica el pool. Pueden existir varias líneas con este mismo número.

rango-> Dirección inicial del trozo del pool.

tamano-> Número de direcciones que forman el trozo del pool.

nivel\_usuario-> Indica el nivel que tienen que tener los usuarios para obtener una dirección de ese pool. El valor mínimo es 1.

id_pool	ip_rango	tamano	nivel_usuario
---------	----------	--------	---------------

=====

- **cfg\_pool\_clie\_INFOVIA.** En este fichero se relacionan el identificativo de los “pools” y el identificativo de los clientes a los que se les asigna ese “pool”.

FICHERO: cfg\_pool\_clie\_INFOVIA

=====

Este fichero es el que relaciona un determinado pool con los clientes Radius existentes, es decir, con las máquinas que nos harán las peticiones Radius.

id\_pool-> Número que identifica el pool. Pueden existir varias líneas con este mismo número.

id\_cliente-> Identificador del cliente Radius

id_pool	id_cliente
---------	------------

=====

- **usuarios\_INFOVIA.** En este fichero se recogen los datos de los usuarios autorizados por el PSI.

FICHERO: usuarios\_INFOVIA

=====

Este fichero incluye el formato definido por Ascend para su interfaz por Radius. Contiene información de seguridad y configuración para cada usuario.

El primer campo es el nombre de usuario. Está seguido de una lista con los requerimientos de autorización para cada usuario. Esta puede incluir el password, nombre de usuario, y una fecha de expiración de password del usuario.

Cuando una petición de autenticación es recibida desde el servidor, esos valores son chequeados. Existe la posibilidad de crear un usuario especial llamado DEFAULT el cual siempre deberá estar al final del fichero. Si a este perfil le damos el password

UNIX, se buscará el password del usuario en el fichero /etc/passwd.

Además de los parámetros definidos por ASCEND, los siguientes valores han sido añadidos en el servicio InfoVía Plus Básico (Modalidad Delegada):

-Max-Num-Connections (224): Máximo número de conexiones permitidas para ese usuario.

-Nivel-Internet (225): Calidad de acceso a Internet definida para ese usuario.

Login@menmonico password = "valor password"

Atributo Radius =valor

Es importante destacar que el usuario DEFAULT definido en la descripción de este fichero, sólo podrá utilizarse si la negociación se realiza con PAP. Ello es debido a que, si el perfil del usuario se encuentra definido en el fichero /etc/password y la negociación se realiza con CHAP, la password del usuario llega al servidor Radius encriptada en un solo sentido, con un algoritmo de encriptación diferente al utilizado en Unix. De esta forma, el servidor Radius rechazaría la petición de autenticación del usuario puesto que no coincidiría la contraseña del usuario con la que tiene almacenada en el perfil del mismo.

Por tanto, este usuario no se podrá usar salvo que se fuerce una negociación con PAP.

Por otro lado, tal y como se indica en la descripción de este fichero, se ha definido el atributo Nivel-Internet (225) para el servicio InfoVía Plus Básico Modalidad Delegada, con el que el cliente indica la calidad de salida a Internet para un usuario. Este debería ser el atributo utilizado a tal efecto, independientemente de la tecnología sobre la que se implanta dicho servicio (tecnología TID o tecnología Lucent).

Sin embargo, sobre tecnología Lucent, en principio este no será el atributo empleado por el cliente para definir la calidad de salida a Internet de sus usuarios, sino que se utilizará para ello el atributo Class (25). Para adaptar el uso de este atributo en lugar del ya mencionado Nivel-Internet, los cambios a realizar en los ficheros de configuración son mínimos, y serían los siguientes:

- Puesto que el atributo Class es un atributo Radius standard, se encuentra definido en el diccionario de atributos del servidor Radius, es decir, en el fichero "diccionario\_INFOVIA". En cualquier caso, si no apareciese, debería de incluirse en dicho fichero.
- El atributo Nivel-Internet (225), utiliza el mismo código que el atributo Ascend-Metric, por lo que dicho atributo fue eliminado del diccionario\_INFOVIA. Sin embargo, puesto que el atributo Nivel-Internet utiliza el mismo código que este último, si se diese la posibilidad de recibir el atributo Ascend-Metric en los paquetes Radius, para evitar problemas en el correcto funcionamiento del servidor, sería necesario eliminar del diccionario de atributos Nivel-Internet. Puesto que el atributo Ascend-Metric no tiene ningún uso, no sería necesario eliminar del diccionario\_INFOVIA el atributo Nivel-Internet.
- En el fichero de perfiles de usuarios, usuarios\_INFOVIA, debe incluirse el atributo Class en el perfil del usuario en lugar de Nivel-Internet.

Del mismo modo, para el servicio InfoVía Plus Básico Modalidad Delegada, se ha definido otro atributo propietario, Max-Num-Connections (224), mediante el cual, el cliente indica el número máximo de sesiones simultáneas que permite establecer a un usuario. Este debería ser el atributo utilizado a tal efecto, independientemente de la tecnología sobre la que se implanta dicho servicio (tecnología TID o tecnología Lucent).

Sin embargo, sobre tecnología Lucent, se emplea el atributo Port-Limit (62) para que el cliente indique el número de sesiones simultáneas. Los cambios que este hecho implican en la configuración del servidor Radius de TData son mínimos y análogos a los indicados anteriormente. Son los siguientes:

- Puesto que el atributo Port Limit es un atributo Radius standard, se encuentra definido en el diccionario de atributos del servidor Radius, es decir, en el fichero "diccionario\_INFOVIA". En cualquier caso, si no apareciese, debería de incluirse en dicho fichero.
- El atributo Max-Num-Connections (224), utiliza el mismo código que el atributo Ascend-IPX-Alias, por lo que dicho atributo fue eliminado del diccionario\_INFOVIA. Sin embargo, puesto que el atributo Max-Num-Connections utiliza el mismo código que este último, si se diese la posibilidad de recibir el atributo Ascend-IPX-Alias en los paquetes Radius, para evitar problemas en el correcto funcionamiento del servidor, sería necesario eliminar del diccionario de atributos Max-Num-Connections. Puesto que el atributo Ascend-IPX-Alias no tiene ningún uso, no sería necesario eliminar del diccionario\_INFOVIA el atributo Max-Num-Connections.
- En el fichero de perfiles de usuarios, usuarios\_INFOVIA, debe incluirse el atributo Por-Limit en el perfil del usuario en lugar de Max-Num-Connections.

Por último, puntualizar que el atributo User-Name (1) que ha de recibir el servidor Radius de TData para proceder a la autenticación del usuario en el paquete Access-Request, debe contener el string login@mnemonico. De lo contrario, no se podrían definir diferentes mnemónicos en el servidor Radius de TData.

- **redIP\_delegado.** En este fichero se recogen los datos de los servidores proxy Radius de la Red IP con los que va a conectarse el servidor Radius del PSI.

FICHERO: redIP\_delegado

=====

Este fichero contiene una lista de las direcciones IP de los servidores proxy Radius de la Red IP que están autorizados a lanzar preguntas Radius sobre este servidor, y a recibir respuestas Radius desde este servidor.

1. El primer campo es la dirección IP del servidor proxy Radius de la Red IP.
2. El segundo campo es el secreto compartido entre el servidor proxy Radius de la Red IP y el servidor Radius del PSI.
3. El tercer campo es el puerto UDP por el que escucha el servidor proxy Radius de la Red IP.

4. El cuarto campo es el identificador<sup>1</sup> del servidor proxy Radius de la Red IP.

1	2	3	4
Dir_IP proxy Radius	CLAVE	Puerto UDP	id. cliente

- **cfg\_OPERADOR\_INFOVIA.** Este fichero deberá aparecer, si se está usando la herramienta rad\_tool, en el mismo directorio que ésta. Los parámetros que han de estar presentes en este fichero, dependen del cliente Radius con el que se comunica el servidor Radius del PSI, y se muestran en los apartados siguientes.

```
FICHERO:  cfg_OPERADOR_INFOVIA
=====
IDENTIFICADOR->Nombre del parámetro
VALOR-> Valor que adquiere ese parámetro.

IDENTIFICADOR      VALOR
=====
```

### 2.4.1. Configuración del servidor Radius para el servicio InfoVía Plus Directo

Una vez se han analizado los ficheros de configuración del servidor Radius de forma genérica, el presente apartado se centra en los ficheros de configuración necesarios de entre los presentados, para el correcto funcionamiento del servidor Radius del PSI en la prestación del servicio InfoVía Plus Directo a los usuarios finales.

Como bien es sabido, las funciones del servidor Radius del PSI en el servicio InfoVía Plus Directo, residen tanto en la realización de la AAA del usuario, como en la asignación de una dirección IP al mismo.

El cliente Radius que realizará peticiones sobre el servidor Radius del PSI, será el Servidor de Túneles.

Teniendo en cuenta estas premisas, los ficheros de configuración que habrán de estar presentes en el servidor Radius se indican a continuación. Así mismo, se muestra un ejemplo de configuración de cada uno de ellos<sup>2</sup>:

- **cfg\_LOCAL\_INFOVIA.**

Ejemplo de configuración:

<sup>1</sup> El identificador asociado al servidor proxy Radius de la Red IP puede ser cualquier número, siempre y cuando no se utilice el mismo indentificador para diferentes servidores proxy Radius de la Red IP.

<sup>2</sup> Los valores para cada uno de los parámetros de los ficheros de configuración empleados en los diferentes ejemplos son ficticios. Se muestran de manera orientativa, y habrán de ser sustituidos por los valores reales de cada parámetro para cada cliente en concreto.

IDENTIFICADOR	VALOR
UDP	1645
HORA	30
HARDWARE_INFOVIA_OMISION	1

- **clientes\_INFOVIA**

Ejemplo de configuración:

1	2	3	4
172.16.192.1	CLAVE	1645	1
172.16.193.1	CLAVE	1645	2
<IP_PSI>	Operador	1645	1000

donde <IP\_PSI> corresponderá a la dirección IP del interfaz Ethernet de la máquina donde se encuentra instalado el servidor Radius del PSI para la versión 4.01 del servidor Radius. Para versiones anteriores del mismo, dicho parámetro corresponderá al cliente "localhost".

- **clientes\_hw\_INFOVIA**

Ejemplo de configuración:

1	2	3	4
194.179.25.48	CLAVE	1645	3
194.179.25.50	CLAVE	1645	4
<IP_PSI>	Operador	1645	1000

donde <IP\_PSI> corresponderá a la dirección IP del interfaz Ethernet de la máquina donde se encuentra instalado el servidor Radius del PSI para la versión 4.01 del servidor Radius. Para versiones anteriores del mismo, dicho parámetro corresponderá al cliente "localhost".

- **cfg\_pool\_INFOVIA**

Ejemplo de configuración:

id_pool	ip_rango	tamano	nivel_usuario
1	192.168.2.10	5	1
2	192.168.2.20	1	2
2	192.168.2.60	1	2
3	192.168.2.30	1	3
3	192.168.2.70	2	3
4	192.168.2.40	2	4

- **cgf\_pool\_clie\_INFOVIA**

Ejemplo de configuración:

id_pool	id_cliente
1	1
2	1
1	2
2	2
3	3
4	4
4	5
1	1000
2	1000
3	1000
4	1000

- **usuarios\_INFOVIA**

Ejemplo de configuración de un perfil de usuario:

```
Usul@mnemonico password = "madrid"
    User-Service = Framed-User,
    Framed-Protocol = PPP,
    Framed-Address = 192.168.2.33,
    Framed-Netmask = 255.255.255.254
```

- **cfg\_OPERADOR\_INFOVIA.** Los parámetros necesarios en este fichero cuando el servidor Radius del cliente atiende peticiones de usuarios del servicio InfoVía Plus Directo son los siguientes:

UDP\_PSI-> Indica el puerto UDP del servidor a controlar.

IP\_PSI->Indica la dirección IP del interfaz Ethernet de la máquina donde se encuentra instalado el servidor Radius del PSI, para la versión 4.01 del servidor Radius. Para versiones anteriores del servidor Radius, este atributo debe indicar la dirección IP del cliente "localhost".

SECRETO\_PSI->Indica el secreto asociado a la herramienta de monitorización del servidor Radius del PSI.

Ejemplo de configuración:

IDENTIFICADOR	VALOR
UDP_PSI	1645
IP_PSI	<IP_PSI>
SECRETO_PSI	Operador

## 2.4.2. Configuración del servidor Radius para el servicio InfoVía Plus Básico (Modalidad Delegada)

Se indican en este apartado los ficheros de configuración necesarios de entre los presentados anteriormente, para el correcto funcionamiento del servidor Radius del PSI en la prestación del servicio InfoVía Plus Básico (Modalidad Delegada) a los usuarios finales.

Para el caso del servicio InfoVía Plus Básico (Modalidad Delegada), el servidor Radius del cliente es el encargado de realizar la AAA del usuario, mientras que la asignación de una dirección IP al mismo es llevada a cabo por la Red IP.

El cliente Radius que realizará peticiones sobre el servidor Radius del PSI, será, en el caso de este servicio, el servidor proxy Radius de la Red IP.

Teniendo en cuenta estas premisas, los ficheros de configuración que habrán de estar presentes en el servidor Radius se indican a continuación. Así mismo, se muestra un ejemplo de configuración de cada uno de ellos<sup>3</sup>:

- **cfg\_LOCAL\_INFOVIA.**

Ejemplo de configuración:

IDENTIFICADOR	VALOR
=====	
UDP	1645
HARDWARE_INFOVIA_OMISION	1

- **clientes\_INFOVIA.** Teniendo en cuenta la funcionalidad de este fichero, indicada en apartados anteriores, se puede prescindir del mismo, puesto que no se emplea para el servicio InfoVía Plus Básico (Modalidad Delegada). En cualquier caso, debe aparecer en el directorio de configuración.
- **clientes\_hw\_INFOVIA.** Teniendo en cuenta la funcionalidad de este fichero, indicada en apartados anteriores, se puede prescindir del mismo, puesto que no se emplea para el servicio InfoVía Plus Básico (Modalidad Delegada). En cualquier caso, debe aparecer en el directorio de configuración.

Tal y como ya se ha indicado, para poder utilizar las opciones de la herramienta rad\_tool, es necesario incluir en el fichero clientes\_INFOVIA o bien, en el fichero clientes\_hw\_INFOVIA, la dirección IP del interfaz Ethernet de la máquina en la que se encuentre instalado el servidor Radius, para la versión 4.01 del mismo. Para versiones anteriores a la indicada, sería necesario definir el cliente "localhost".

- **cfg\_pool\_INFOVIA.** Teniendo en cuenta la funcionalidad de este fichero, indicada en apartados anteriores, se puede prescindir del mismo, puesto que no se emplea para el servicio InfoVía Plus Básico (Modalidad Delegada). En cualquier caso, debe aparecer en el directorio de configuración aunque se encuentre vacío.

<sup>3</sup> Los valores para cada uno de los parámetros de los ficheros de configuración empleados en los diferentes ejemplos son ficticios. Se muestran de manera orientativa, y habrán de ser sustituidos por los valores reales de cada parámetro para cada cliente en concreto.

- **cgf\_pool\_clie\_INFOVIA.** Teniendo en cuenta la funcionalidad de este fichero, indicada en apartados anteriores, se puede prescindir de este fichero, puesto que no se emplea para el servicio InfoVía Plus Básico (Modalidad Delegada). En cualquier caso, debe aparecer en el directorio de configuración aunque se encuentre vacío.
- **usuarios\_INFOVIA.**

Ejemplo de configuración de un perfil de usuario:

```
Usul@mnemonico password = "madrid"
    Nivel-Internet = 1,
    Session-Timeout = 0,
    Idle-Timeout = 10,
    Max-Num-Connections = 5
```

Como ya se ha indicado anteriormente, sobre tecnología Lucent no se emplea el atributo Nivel-Internet para indicar la calidad de salida a Internet de un usuario, sino que el atributo utilizado con este propósito es el atributo Class. Del mismo modo, el atributo utilizado para indicar el número máximo de sesiones simultáneas permitidas a un usuario sería el Port-Limit, en lugar de Max-Num-Connections. Por tanto, en el perfil de un usuario se definiría de la siguiente forma:

```
Usul@mnemonico password = "brasil"
    Class = "string",
    Port-Limit = 2
```

- **redIP\_delegado**

Ejemplo de configuración:

1	2	3	4
1.0.11.23	CLAVE	1645	22
1.23.40.23	CLAVE	1645	23

- **cfg\_OPERADOR\_INFOVIA.** Los parámetros necesarios en este fichero cuando el servidor Radius del cliente atiende peticiones de usuarios del servicio InfoVía Plus Básico (Modalidad Delegada) son los siguientes:

UDP\_PSI-> Indica el puerto UDP del servidor Radius local.

UDP\_Cliente\_Radius-> Indica el puerto UDP asignado al (los) servidor (servidores) proxy Radius de la Red IP.

IP\_PSI->Indica la dirección IP del interfaz Ethernet de la máquina donde se encuentra instalado el servidor Radius del PSI para la versión 4.01 de dicho servidor. Para versiones anteriores del mismo, dicha dirección IP correspondería al cliente "localhost".



IP\_Cliente\_Radius-> Indica la dirección IP del servidor (servidores) proxy Radius de la Red IP.

SECRETO\_PSI-> Indica el secreto asociado a la herramienta de monitorización del servidor Radius del PSI.

SECRETO\_Cliente\_Radius-> Indica el secreto compartido entre el servidor proxy Radius de la Red IP y el servidor Radius del PSI.

Ejemplo de configuración:

IDENTIFICADOR	VALOR
=====	=====
UDP_PSI	1645
UDP_Cliente_Radius	1645
IP_PSI	<IP_PSI>
IP_Cliente_Radius	1.0.11.23
IP_Cliente_Radius	1.23.40.23
SECRETO_PSI	Operador
SECRETO_Cliente_Radius	CLAVE

### 2.4.3. Configuración del servidor Radius para la convivencia de ambos servicios

Por último, se muestran los ficheros de configuración del servidor Radius del cliente en un escenario de convivencia de los servicios InfoVía Plus Directo e InfoVía Plus Básico (Modalidad Delegada), en el que el servidor Radius del cliente recibirá peticiones de autenticación de usuarios de cada uno de los servicios.

Dependiendo del usuario de quien el servidor Radius reciba una petición de autenticación, variará el comportamiento del mismo, de la forma en que se ha indicado con anterioridad.

Del mismo modo, el servidor Radius del cliente recibirá peticiones Radius tanto del Servidor de Túneles del cliente, para usuarios del servicio InfoVía Plus Directo, como del servidor proxy Radius de la Red IP, para usuarios del servicio InfoVía Plus Básico (Modalidad Delegada). Cada cliente Radius deberá ser definido en los ficheros de configuración correspondientes, para que el servidor Radius responda de la forma adecuada ante las peticiones realizadas por cada uno de ellos.

Teniendo en cuenta las conclusiones de los apartados 2.4.1 y 2.4.2, los ficheros de configuración necesarios en un escenario de convivencia de ambos servicios, son los que a continuación se indican.

- **cfg\_LOCAL\_INFOVIA**
- **clientes\_INFOVIA**
- **clientes\_hw\_INFOVIA**
- **cfg\_pool\_INFOVIA**
- **cgf\_pool\_clie\_INFOVIA**

- **usuarios\_INFOVIA**
- **redIP\_delegado**
- **cfg\_OPERADOR\_INFOVIA**

La configuración de los ficheros `cfg_LOCAL_INFOVIA`, `clientes_INFOVIA`, `clientes_hw_INFOVIA`, `cfg_pool_INFOVIA` y `cfg_pool_clie_INFOVIA`, es la misma que la indicada en el apartado 2.4.1 del presente documento.

La configuración del fichero `redIP_delegado` es la misma que la indicada en el apartado 2.4.2 del documento.

El fichero de perfiles de usuarios, `usuarios_INFOVIA`, contendrá ahora perfiles de los dos tipos de usuarios, los correspondientes al servicio InfoVía Plus Directo y los correspondientes al servicio InfoVía Plus Básico (Modalidad Delegada).

La configuración del fichero `cfg_OPERADOR_INFOVIA`, es la misma que la indicada en el apartado 2.4.2 del documento.

## 2.5. **Instalación, configuración y mantenimiento de la aplicación de desconexión**

Se presenta en este apartado la herramienta que ofrece la funcionalidad en el servidor Radius de Telefónica Data necesaria para enviar al servidor proxy Radius de la Red IP peticiones de desconexión de usuarios que han accedido a través del cliente.

La herramienta de desconexión se distribuye para los mismos sistemas operativos que el software Radius.

Se indica inicialmente cómo se instala la herramienta de desconexión para cada uno de los sistemas operativos para los que se distribuye. Posteriormente se indican los directorios de instalación y la forma en que se utiliza dicha herramienta de forma común, puesto que son análogos para los tres sistemas operativos.

### 2.5.1. **Instalación para Solaris**

La herramienta de desconexión para SUN/Solaris 2.5.1 se suministra en un paquete denominado "minitool".

Para realizar la instalación, deberá ejecutarse como usuario root el siguiente comando:

```
pkgadd -d <directorio de instalación del paquete>
```

El paquete instala el fichero binario, la documentación, los fuentes, y un fichero de configuración de ejemplo en el directorio `/opt/rad_tool`

### 2.5.2. **Instalación para Linux**

Para el sistema operativo Linux, se suministra un fichero en formato tar.

Para la versión 4.01 del servidor Radius, dicho fichero se denomina `mini_tool_linux.tar`. Para realizar la instalación deberá ejecutarse el comando:

```
tar -xvf mini_tool_linux.tar
```

El paquete instala el fichero binario, la documentación, los fuentes y un fichero de configuración de ejemplo en el directorio `/mini_tool`.

Para versiones anteriores del servidor Radius, se suministra el fichero denominado `radtool_lin.tar`. Para realizar la instalación deberá ejecutarse el comando:

```
tar -xvf radtool_lin.tar
```

El paquete instala el fichero binario, la documentación, los fuentes y un fichero de configuración de ejemplo en el directorio `/opt/rad_tool`.

### 2.5.3. Instalación para Windows NT

La herramienta de desconexión de usuarios para Windows NT se suministra como un fichero autoejecutable que se instala en el directorio `c:\mini_tool` para la versión 4.01 del servidor Radius. Para versiones anteriores del servidor, se instala en el directorio `c:\rad_tool`.

### 2.5.4. Directorios de instalación

Para la versión 4.01 del servidor Radius, el paquete genera los directorios que se indican a continuación en el directorio `/opt/rad_tool` para Solaris, `/mini_tool` para Linux, y en `c:\mini_tool` para Windows NT. Como ya se ha indicado, para versiones anteriores del servidor, los directorios se generan en el directorio `/opt/rad_tool` para Solaris y Linux, y en `c:\rad_tool` para Windows NT.

- `doc/` Contiene el fichero `mini_tool.pdf`
- `src/` Fuentes de la aplicación de desconexión de usuario (`mini_tool`) y un Makefile para el sistema operativo correspondiente
- `bin/` Fichero ejecutable que constituye la herramienta de desconexión.

### 2.5.5. Utilización de la herramienta de desconexión

La aplicación “`mini_tool`” se utiliza con la siguiente sintaxis:

```
mini_tool liberar_direccion_delegada <dir_ip>
```

Donde:

- `liberar_direccion_delegada <dir_ip>` libera la dirección IP `dir_ip` de las asignadas al usuario del cliente por el servidor proxy Radius de la Red IP.

Para usar esta utilidad, es preciso que aparezca en el mismo directorio el fichero “`cfg_desconexion`”.

Se muestra a continuación un ejemplo de la salida que se produce ante una petición de desconexión:

```
$/mini_tool liberar_direccion_delegada 1.1.1.1
```

Código de respuesta: 41

Identificador\_recibido: 35 , identificador\_enviado: 35

La sesión ya finalizó o la dirección IP no esta asignada a este proveedor.

En el caso de que se haya producido un error o existan problemas de comunicación con el servidor proxy Radius de la Red IP, la salida es la siguiente:

```
$/mini_tool liberar_direccion_delegada 1.1.1.1
```

Temporiza...id 181

Código de respuesta: 0

Identificador\_recibido: 0, identificador\_enviado: 181

Problema de conexión con el servidor

### 2.5.6. Fichero de configuración

El fichero de configuración de la herramienta de desconexión es el denominado "cfg\_desconexion". Este fichero contiene diversos parámetros relativos al servidor proxy Radius de la Red IP (cliente Radius) al que se le enviarán las peticiones de desconexión de usuarios desde el servidor Radius del PSI.

Su contenido será de la forma que se indica a continuación:

FICHERO: cfg\_desconexion

=====

IDENTIFICADOR->Nombre del parámetro

VALOR-> Valor que adquiere ese parámetro.

IDENTIFICADOR	VALOR
UDP	Puerto UDP de escucha del cliente Radius
IP_Cliente_Radius	Dirección IP del cliente Radius
SECRETO	Secreto a intercambiar con el cliente Radius

## 2.6. Prestaciones del servidor Radius de Telefónica Data

Se muestran en este apartado los resultados del estudio de prestaciones del servidor Radius de Telefónica Data, para las distintas plataformas para las que se distribuye, y que como ya se ha indicado en otros puntos del documento, son Solaris, Linux y Windows NT.

Debe tenerse en cuenta que las pruebas se han realizado sobre máquinas en las que no corría conjuntamente ningún otro proceso.

### 2.6.1. Prestaciones del RADIUS en Solaris

#### 2.6.1.1. Escenario

El estudio de las prestaciones se realizó utilizando una máquina ULTRA 1 con una CPU a 167 MHz, 192 MB de RAM y Solaris 2.6<sup>4</sup>. Durante las pruebas se utilizó un simulador, ejecutándose desde otra máquina, que realizaba solicitudes de autenticación correspondientes a usuarios elegidos aleatoriamente.

La base de datos de usuarios puede ser de dos tipos:

- Fichero plano de texto
- Fichero en formato dbm

El fichero con formato dbm lleva asociado un índice del identificador de usuario lo que reduce muy notablemente el tiempo de búsqueda cuando el número de usuarios es considerable.

Se han realizado por tanto dos conjuntos de pruebas, utilizando fichero plano y utilizando dbm.

#### 2.6.1.2. Resultados utilizando fichero plano

##### 2.6.1.2.1. Resultados en Máquina de Pruebas

En la siguiente tabla se presentan los tiempos de servicio obtenidos para distintos tamaños del fichero de usuarios.

#Usuarios en fichero	Tiempo CPU (ms)
10	8
100	75
500	256
1000	536
5000	2610

El tiempo de CPU muestra una tendencia lineal con el número de usuarios almacenados en el fichero plano.

<sup>4</sup> Como se observa, la versión de Solaris empleada en las pruebas de prestaciones del servidor Radius es la 2.6. SUN garantiza que Solaris 2.6 es compatible binario con Solaris 2.5.1, y no debería haber problemas al ejecutar el software Radius en Solaris 2.6. Sin embargo, el servidor Radius de Telefónica Data se ha desarrollado para Solaris 2.5.1

Se observan también accesos de escritura a disco (tarificación), aunque son escasos y no bloqueantes, dado que se escribe al final de un fichero y por tanto, la caché de ficheros del sistema operativo reduce apreciablemente el volumen de los mismos.

La tasa máxima depende por tanto únicamente de la CPU y del tamaño del fichero de usuarios. En la siguiente tabla se muestra la capacidad máxima según el número de usuarios para la máquina empleada en las pruebas:

#Usuarios en fichero	Llamadas por segundo
10	100
100	11
500	3.3
1000	1.5
5000	0.3

Dado que el Radius es monoproceso y monothread y, por tanto, no se puede conseguir paralelismo, la única manera de aumentar las prestaciones es con una CPU más rápida.

Por consiguiente, independientemente de la máquina empleada, a partir de 1000 usuarios la alternativa de fichero plano es poco recomendable.

#### 2.6.1.2.2. Resultados en Máquina con distinta CPU

Dado que la capacidad del sistema en este caso se debe exclusivamente a CPU, la capacidad para una máquina con CPU de velocidad distinta pero con la misma arquitectura que la utilizada en las pruebas (CPU UltraSparc I) es lineal con la velocidad de reloj (así se indica en los benchmarks de Sun). Así, la capacidad máxima con una plataforma Solaris con CPU Sun Ultra-I a 336 MHz y con un fichero con 100 usuarios sería de 22 llamadas/segundo. Sin embargo, si la configuración HW permitiera más de 100 llamadas/segundo, los accesos a escritura en disco limitarían la capacidad a ese valor.

#### 2.6.1.3. Resultados utilizando fichero dbm

##### 2.6.1.3.1. Resultados en Máquina de Pruebas

Opcionalmente, se puede utilizar un Radius que emplea las librerías dbm de Solaris para autenticar a los usuarios almacenados en un fichero con formato dbm. Este fichero se genera utilizando un programa incluido en la distribución (builddbm) a partir de un fichero plano de texto con los usuarios y sus campos asociados.

El tiempo de CPU es ahora independiente del número de usuarios (al menos hasta 1 millón de usuarios). Los resultados se resumen en la siguiente tabla (prueba con fichero de 1 millón de usuarios):

Tiempo CPU (ms)	Disco (escritura) (ms)	Disco (lectura) (ms)
13	2	13

Los accesos de lectura a disco son los debidos a la búsqueda del perfil del usuario. Debido al tamaño del fichero y al elevado tiempo entre accesos de un mismo usuario, la eficiencia de la caché es reducida y en general, cada autenticación conlleva un acceso de lectura a disco. Los accesos a escritura no son bloqueantes, al menos, mientras el disco donde se produzcan los accesos a escritura no llegue a saturación, por lo que los 2 ms del tiempo de escritura no afectan a la tasa máxima de llegadas. Además, hay que tener en cuenta que el sistema debe estar por debajo de saturación para que los tiempos de respuesta no se disparen (se desbordan colas, vencen temporizadores, etc...). Se puede fijar un máximo del 85%.

Teniendo en cuenta estos factores se ofrecen los siguientes resultados:

#Usuarios en fichero	Llamadas por segundo
10	32
100	32
1000	32
100000	32
1000000	32

En realidad, los valores que se muestran en la tabla son pesimistas para los tamaños del fichero dbm entre 10 y 10.000 usuarios. La eficiencia de la caché del sistema de ficheros tiene como consecuencia que, al cabo de un tiempo, la mayor parte de las peticiones no tienen que acceder a disco y, por tanto, la capacidad máxima del sistema será superior al valor mostrado. Sin embargo, y dado que el valor exacto depende de bastantes factores, como memoria de la máquina, procesos corriendo en la misma máquina que leen y escriben de disco (ficheros) y por tanto pueden desalojar páginas en la caché, etc., se opta por una cota pesimista.

En el caso de un fichero dbm con más de 100.000 usuarios, sí que se puede confirmar que, por el tamaño de los ficheros y los accesos a posiciones aleatorias del mismo, la eficiencia de la caché es reducida y en media se accede a lectura a disco por cada petición.

#### 2.6.1.3.2. Resultados en Máquina con Distinto Hardware

En este caso, como se ha visto, las prestaciones vienen determinadas tanto por el tiempo empleado en CPU como los accesos de lectura a disco. En general, salvo que se utilice una configuración en "stripe" de varios discos, los tiempos de acceso a disco serán muy parecidos a los obtenidos. En cuanto al tiempo de servicio en CPU, será lineal con la velocidad de reloj de la CPU, pero con la misma arquitectura que la CPU utilizada en las pruebas (UltraSparc- I). Además, hay que tener en cuenta que el sistema debe estar por debajo de saturación, para que los tiempos de respuesta no se disparen (se desbordan colas, vencen temporizadores, etc). Se puede fijar un máximo del 85%.

Así, en un sistema con una CPU a 336 MHz, el tiempo de CPU se reduce aproximadamente a la mitad, los accesos a disco serán muy parecidos y, teniendo en cuenta el factor anterior de utilización máxima, se tendría una capacidad máxima de 43 llamadas/ segundo.

#### 2.6.1.4. Actualización del fichero dbm de Perfiles de Usuario

Un aspecto a tener en cuenta cuando se utiliza la base de datos de usuarios con ficheros en formato dbm es el tiempo de generación de los mismos.

La utilidad de generación del fichero en formato dbm sólo permite generar éste fichero a partir del fichero plano de texto con todos los usuarios del PSI. Es decir, si se da de alta o baja un usuario hay que regenerar de nuevo el fichero con todos los usuarios. Si el número de usuarios es elevado, el tiempo de generación puede ser un factor importante a tener en cuenta.

##### 2.6.1.4.1. Tiempo necesario para generar el fichero dbm según el número de perfiles

Se han realizado una serie de pruebas para medir el tiempo necesario en generar la base de datos dbm según el número de usuarios, midiendo además la utilización de recursos de la máquina para así determinar el recurso limitante en el proceso.

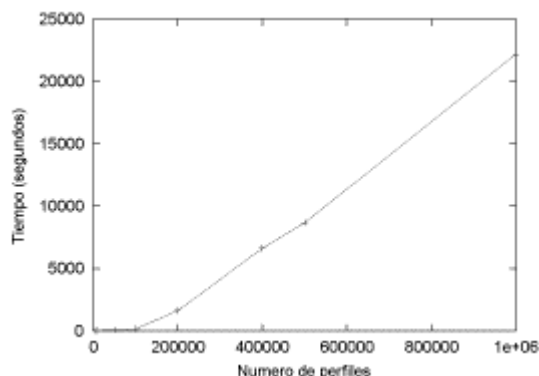
En la siguiente tabla se muestran los tiempos de creación del fichero dbm según el número de perfiles:

Número de Perfiles	Tiempo de creación
5000	14sg
10000	29sg
50000	66sg
100000	2min 42sg
200000	27min 3sg
400000	1h 50min 29sg
1000000	6h 9min 50sg

Gráficamente:



### Tiempo de creación frente al número de perfiles



Se observa un aumento **no lineal** del tiempo de creación del fichero dbm con el aumento del número de perfiles. En concreto, se observa una ruptura de la tendencia lineal inicial entre 100.000 y 200.000 usuarios. Al doblarse el número de usuarios de 100.000 a 200.000 el tiempo necesario se ha multiplicado por 10. A partir de este punto, 200.000 perfiles, se vuelve a observar una tendencia aproximadamente lineal. Se ha visto que este punto de ruptura puede depender de la “variabilidad” de los nombres de usuarios del fichero. Así, se encontró que, con cadenas de usuarios muy aleatorias, este punto de ruptura era menor (entre 50.000 y 100.000).

El “cuello de botella” en la generación del fichero dbm son los accesos a disco. La utilización del mismo es próxima al 100% mientras se está generando el fichero dbm. Es decir, el rendimiento del mecanismo de generación del fichero dbm no depende del tipo de CPU ni del número de CPUs de que disponga la máquina, únicamente de la “velocidad” del disco.

## 2.6.2. Prestaciones del RADIUS en Linux

### 2.6.2.1. Escenario

El estudio de las prestaciones se realizó utilizando un PC con Linux de la distribución Debian con la versión de kernel 2.0.36, con una CPU Pentium II a 400 MHz, y 128 MB de RAM. Durante las pruebas se utilizó un simulador, ejecutándose desde otra máquina, que realizaba solicitudes de autenticación correspondientes a usuarios elegidos aleatoriamente.

Como en el caso de la plataforma Solaris, la base de datos de usuarios puede ser de dos tipos:

- Fichero plano de texto
- Fichero en formato dbm

El fichero con formato dbm lleva asociado un índice del identificador de usuario, lo que reduce muy notablemente el tiempo de búsqueda cuando el número de usuarios es considerable.

Se han realizado por tanto dos conjuntos de pruebas, utilizando fichero plano y utilizando dbm.

## 2.6.2.2. Resultados utilizando fichero plano

### 2.6.2.2.1. Resultados en Máquina de Pruebas

En la siguiente tabla se presentan los tiempos de servicio obtenidos para distintos tamaños del fichero de usuarios.

#Usuarios en fichero	Tiempo CPU (ms)
10	15.2
500	65
1000	118
5000	689

El tiempo de CPU muestra una tendencia lineal con el número de usuarios almacenados en el fichero plano.

Se observan también accesos de escritura a disco (tarificación), aunque son escasos y no bloqueantes, dado que se escribe al final de un fichero y, por tanto, la caché de ficheros del sistema operativo reduce apreciablemente el volumen de los mismos.

La tasa máxima depende, por tanto, únicamente de la CPU y del tamaño del fichero de usuarios. En la siguiente tabla se muestra la capacidad máxima según el número de usuarios para la máquina empleada en las pruebas:

#Usuarios en fichero	Llamadas por segundo
100	55
500	13
1000	7.2
5000	1.2

Dado que el Radius es monoproceso y multithread y por tanto no se puede conseguir paralelismo, la única manera de aumentar las prestaciones es con una CPU más rápida.

Por consiguiente, independientemente de la máquina empleada, a partir de 5000 usuarios la alternativa de fichero plano es poco recomendable.

### 2.6.2.2.2. Resultados en Máquina con distinta CPU

Dado que la capacidad del sistema en este caso se debe exclusivamente a CPU, la capacidad para una máquina con CPU de velocidad distinta pero con la misma arquitectura que la utilizada en las pruebas (PENTIUM II), es aproximadamente lineal con la velocidad de reloj (así se indica en los benchmarks de INTEL). Así, la capacidad máxima con un PC PENTIUM II a 300 MHz y con un fichero con 100 usuarios sería de 41 llamadas/segundo.

### 2.6.2.3. Resultados utilizando fichero dbm

#### 2.6.2.3.1. Resultados en Máquina de Pruebas

Opcionalmente, se puede utilizar un Radius que emplea las librerías dbm para autenticar a los usuarios almacenados en un fichero con formato dbm. Este fichero se genera utilizando un programa incluido en la distribución (builddb) a partir de un fichero plano de texto con los usuarios y sus campos asociados.

Los resultados se resumen en la siguiente tabla:

#usuarios	Tiempo CPU (ms)	Tiempo disco-lectura (ms)
1000	2.7	0
10000	4.7	0
100000	31.7	~13
500000	365	~13

A partir de 100000 usuarios en el fichero dbm en media se accede a disco en lectura por cada usuario que se conecta. Debido al tamaño del fichero y al elevado tiempo entre accesos de un mismo usuario, la eficiencia de la caché es reducida y en general, cada autenticación conlleva un acceso de lectura a disco. Los tiempos ofrecidos en la tabla son pesimistas debido a la dificultad de la medida de disco en esta plataforma.

Existen también accesos de escritura a disco (tarificación de usuarios, logs), aunque el volumen de los mismos es muy reducido y no son bloqueantes (debido al buffer de I/O del sistema operativo), al menos mientras el disco donde se produzcan los accesos a escritura no llegue a saturación debido a otra tarea o proceso que haga uso exhaustivo del mismo disco, por lo que éstos no afectan a la tasa máxima de llegadas.

Además hay que tener en cuenta que el sistema debe estar por debajo de saturación para que los tiempos de respuesta no se disparen (se desbordan colas, vencen temporizadores, etc). Se puede fijar un máximo del 85%.

Teniendo en cuenta estos factores se ofrecen los siguientes resultados:

#Usuarios en fichero	Llamadas por segundo
100	>100
10000	>100
100000	19
500000	2.2

Se observa, a diferencia de la plataforma Sun Solaris, que la capacidad depende del número de usuarios en el fichero dbm y, de manera muy notable, a partir de 100.000 usuarios.

### 2.6.2.3.2. Resultados en máquina con distinta CPU

En este caso, como se ha visto, las prestaciones vienen determinadas tanto por el tiempo empleado en CPU como los accesos de lectura a disco. En cuanto al tiempo de servicio en CPU, será lineal con la velocidad de reloj de la CPU, pero con la misma arquitectura que la CPU utilizada en las pruebas (Pentium-II). Además hay que tener en cuenta que el sistema debe estar por debajo de saturación para que los tiempos de respuesta no se disparen (se desbordan colas, vencen temporizadores, etc). Se puede fijar un máximo del 85%.

Así, un sistema con una CPU a 300 MHz y 100000 usuarios, el tiempo de CPU aumenta en un 33% aproximadamente, los accesos a disco serán muy parecidos y teniendo en cuenta el factor anterior de utilización máxima, se tendría una capacidad de 15 llamadas/segundo.

### 2.6.2.4. Actualización del fichero dbm de perfiles de usuarios

Un aspecto a tener en cuenta cuando se utiliza la base de datos de usuarios con ficheros en formato dbm, es el tiempo de generación de los mismos.

La utilidad de generación del fichero en formato dbm sólo permite generar este fichero a partir del fichero plano de texto con todos los usuarios del PSI. Es decir, si se da de alta o baja un usuario hay que regenerar de nuevo el fichero con todos los usuarios. Si el número de usuarios es elevado, el tiempo de generación puede ser un factor importante a tener en cuenta.

#### 2.6.2.4.1. Tiempo necesario para generar el fichero dbm según el número de perfiles

Se han realizado una serie de pruebas para medir el tiempo necesario en generar la base de datos dbm según el número de usuarios, midiendo además la utilización de recursos de la máquina, para así determinar el recurso limitante en el proceso.

En la siguiente tabla se muestran los tiempos de creación del fichero dbm según el número de perfiles:

Número de Perfiles	Tiempo de creación
1000	2.66
10000	42.8
100000	27min 12sg
500000	2h 16min 30sg

Se observa un incremento no lineal del tiempo necesario para generar el fichero dbm según el número de usuarios.

El “cuello de botella” en la generación del fichero dbm son los accesos a disco. La utilización del mismo es próxima al 100% mientras se está generando el fichero dbm. Es decir, el rendimiento del mecanismo de generación del fichero dbm no depende del tipo de CPU ni del número de CPUs de que disponga la máquina, únicamente de la “velocidad” del disco. Además, según se ve en la tabla de tiempos, cada vez supone

más trabajo insertar un nuevo usuario a medida que va aumentando el número de usuarios insertados en el fichero dbm.

### 2.6.3. Prestaciones del RADIUS en Windows NT

#### 2.6.3.1. Escenario

El estudio de las prestaciones se realizó utilizando un PC con Windows NT 4.0 con una CPU Pentium II a 400 MHz y 128 MB de RAM. Durante las pruebas se utilizó un simulador, ejecutándose desde otra máquina, que realizaba solicitudes de autenticación correspondientes a usuarios elegidos aleatoriamente.

Para esta plataforma sólo existe la opción de utilizar fichero plano de texto como base de datos de usuarios.

#### 2.6.3.2. Resultados utilizando fichero plano

##### 2.6.3.2.1. Resultados en Máquina de Pruebas

En la siguiente tabla se presentan los tiempos de servicio obtenidos para distintos tamaños del fichero de usuarios.

#Usuarios en fichero	Tiempo CPU (ms)
100	367
500	855
1000	1860

La tasa máxima depende del tamaño del fichero de usuarios. En la siguiente tabla se muestra la capacidad máxima según el número de usuarios para la máquina empleada en las pruebas:

#Usuarios en fichero	Llamadas por segundo
100	2.31
500	1
1000	0.5

##### 2.6.3.2.2. Resultados en Máquina con distinto Hardware

Dado que la capacidad del sistema en este caso se debe exclusivamente a CPU, la capacidad para una máquina con CPU de velocidad distinta pero con la misma arquitectura que la utilizada en las pruebas (PENTIUM II), es aproximadamente lineal con la velocidad de reloj (así se indica en los benchmarks de INTEL). Así, la capacidad máxima con un PC PENTIUM II a 300 MHz y con un fichero con 100 usuarios sería de 1.7 llamadas/segundo.

#### 2.6.4. Conclusiones del estudio del servidor Radius de Telefónica Data

A continuación se detallan los aspectos más importantes del estudio:

- La plataforma Windows NT es, con diferencia, la que menos rendimiento ofrece (el PC empleado en las pruebas es el mismo que en el caso de la plataforma Linux).
- Para clientes con pocos usuarios (menos de 1.000) la opción más adecuada (por sencillez, coste y prestaciones) es utilizar Radius con fichero plano de texto en plataforma Linux
- La utilización de fichero plano como base de datos de usuarios cuando el número de estos es significativo penaliza las prestaciones. El proceso Radius tiene que hacer una búsqueda secuencial del perfil del usuario en el fichero. Cuanto más grande sea el número de usuarios en el fichero, en media, más tiempo tardará en procesar cada petición de autenticación. Cuando el número de usuarios supera cierto valor (por ejemplo a partir de 5.000 usuarios), independientemente de la plataforma y el Hardware empleado, la capacidad máxima del sistema no será suficiente.
- La opción más adecuada para clientes con un número de usuarios superior a 5.000, es emplear la opción de base de datos de usuarios en formato dbm. Esta opción sólo está disponible en las plataformas Solaris y Linux.
- En este caso, un aspecto importante a destacar es el tiempo de generación del fichero en formato dbm a partir del fichero plano. Esto es así ya que no sólo hay que generarlo al principio, si no que cada vez que se da de alta/baja un usuario o se modifican los perfiles de un usuario, hay que regenerar el fichero dbm por completo.
- El tiempo de generación aumenta de forma no lineal con el número de usuarios. Éste puede llegar a 6 horas con una base de datos de 1 millón de usuarios (para la plataforma Solaris). En el caso de Linux sobre PC los tiempos de generación son apreciablemente mayores.
- Además, mientras se está generando el fichero en formato dbm, se accede continuamente a disco en accesos de escritura. Si esto se produce mientras el sistema está activo (admitiendo usuarios) y el disco que se utiliza para leer los datos de usuarios (fichero dbm antiguo) es el mismo que el que utiliza la herramienta de generación del nuevo fichero dbm, ambos procesos se interfieren y las prestaciones máximas de ambos se ven reducidas (puede llegar a suponer una reducción del 50% de la capacidad máxima del Radius).
- Continuando el último punto, se debería regenerar el nuevo fichero dbm en un disco distinto al empleado por el fichero dbm que se quiere reemplazar y que estará siendo utilizado por el Radius. Hay que tener en cuenta que comandos de sistema operativo como “cp”, “tar” o “ar” para copiar ficheros normales no funcionan con ficheros en formato dbm. Deben usarse links simbólicos “ln” o el comando “mv”. Además, habrá que rearrancar el Radius para que utilice el nuevo fichero.
- Cabe mencionar que en la actualidad, un cliente con 100.000 usuarios en su base de datos, no recibe más de 3 llamadas/segundo en la hora cargada.

### 3. Uso de bases de datos externas

El servidor Radius de Telefónica Data, ofrece por defecto una base de datos de usuario en forma de fichero plano, cuya configuración se ha tratado en apartados anteriores, el fichero denominado usuarios\_INFOVIA.

Sin embargo, el cliente puede utilizar una librería dinámica para poder acceder a una base de datos de usuario diferente a la ofrecida por defecto.

Para los sistemas operativos Solaris y Linux, el servidor Radius de Telefónica Data ofrece una librería dinámica por defecto, para el acceso a bases de datos dbm, cuya utilización se ha tratado anteriormente, en el apartado 2.3.7.2 del presente documento.

Esta base de datos en formato dbm puede construirse a partir del fichero usuarios\_INFOVIA mediante la utilidad "builddb". Como también se indicó anteriormente, al utilizar esta base de datos, cada vez que se incluya o se excluya un usuario en el fichero usuarios\_INFOVIA, o se modifican los perfiles de un usuario, hay que regenerar el fichero dbm por completo, utilizando el programa builddb. Así mismo, será necesario reiniciar el servidor Radius para que utilice el nuevo fichero dbm generado.

Por otro lado, el cliente puede desarrollar su propia librería dinámica para el acceso a bases de datos externas, empleando para ello los fuentes que se proporcionan en el directorio /opt/radius/lib para el caso de Solaris, /radius/lib para Linux, y en el directorio c:\radius\lib para el caso de Windows NT, aunque para algunas versiones del servidor Radius para este sistema operativo, pueden aparecer en el directorio c:\radius\src.

Dichos fuentes son los siguientes:

- UsuExtPwd.c
- EscribeFac.c

El objetivo de UsuExtPwd.c es buscar el nombre de usuario en la base de datos de usuario. Devuelve un string con los datos del usuario. Opcionalmente, se puede comprobar también si la password del usuario es correcta.

El objetivo de EscribeFac.c consiste en la entrega de los datos de facturación para su almacenamiento o procesado.

El código de estos fuentes puede verse en el Anexo A del presente documento.

Para desarrollar las librerías, se suministra a modo de ejemplo un fichero "Makefile" que las genera, y que igualmente puede localizarse en el directorio /opt/radius/lib para Solaris, y en el directorio /radius/lib para Linux. El código de este fichero puede verse en el Anexo A del presente documento.

Para el caso de Windows NT, el fichero ejecutable del servidor (radiusnt), utiliza siempre una librería dinámica, la denominada RadInfovia.dll, que deberá localizarse en el mismo directorio que dicho fichero ejecutable.

## 4. Servidor Proxy Radius de Telefónica Data

**La funcionalidad ofrecida por el Radius de Telefónica Data de actuar como un servidor proxy Radius está actualmente en desarrollo, como se indica en mayor detalle a lo largo de este apartado.**

Hasta el momento, se ha tratado el servidor Radius de Telefónica Data como un servidor Radius único para el cliente, encargado de realizar la AAA del usuario y de asignarle una dirección IP al mismo en el caso del servicio InfoVía Plus Directo, y encargado de realizar la AAA del usuario en el caso del servicio InfoVía Plus Básico (Modalidad Delegada).

Así mismo, se ha tratado la forma en que el servidor Radius puede conseguir la información que precisa para desempeñar las funciones anteriormente indicadas, bien mediante el empleo de un fichero plano o de un fichero dbm, que incorpora por defecto, como ya se ha indicado, o bien accediendo a una base de datos externa propia del cliente, realizando para ello los desarrollos correspondientes.

Sin embargo, también será posible la utilización del servidor Radius de Telefónica Data como un servidor proxy Radius. Actualmente, se está llevando a cabo el desarrollo de esta nueva funcionalidad del servidor Radius.

La razón fundamental por la que se ha planteado el desarrollo de un servidor proxy Radius, nace de la necesidad de clientes que desean mantener su entorno íntegro, sin llevar a cabo modificaciones que se hacen necesarias para la prestación de los servicios de la Red IP. Esto tiene implicaciones principalmente en aquellos clientes que decidan prestar el servicio InfoVía Plus Directo, puesto que como es sabido, para este servicio es el servidor Radius del cliente quien debe asignar una dirección IP al usuario.

El servidor proxy intermediará entre el PSI y la red tanto para los paquetes de autenticación como para los paquetes de contabilidad, enviando los atributos Radius necesarios en cada paquete.

Actualmente está siendo llevado a cabo el desarrollo del servidor proxy Radius, desarrollo que comprende dos fases. Las funciones que realizará el servidor proxy Radius para solventar la situación descrita anteriormente se muestran en los siguientes apartados. En ellos se especifica en qué fase del desarrollo se encontrará disponible cada una de las funciones.

La primera fase del desarrollo estará disponible en Abril del 2000, aunque se dispondrá de una versión beta del servidor proxy para poder comenzar a realizar pruebas con los clientes en Marzo del 2000. La disponibilidad de la segunda fase del mismo se traslada a Junio del 2000, disponiendo igualmente de una versión beta en Abril del 2000.

### 4.1. Distribuciones del Servidor Proxy Radius

Inicialmente, el servidor proxy Radius se distribuirá para la plataforma Solaris 2.5.1.

Sólo será utilizado en el servicio InfoVía Plus Directo.

Estas características son las mismas para la fase I y para la fase II del desarrollo.



## 4.2. **Asignación de direcciones IP**

El servidor proxy será quien se encargue de la asignación de dirección IP a usuarios finales. Dicha dirección IP será asignada cuando se reciba la respuesta del proveedor.

Será el PSI quien indique a qué pool pertenece la dirección IP que se asignará a un usuario, empleando para ello un atributo Radius que el propio PSI seleccionará.

En la fase I del desarrollo del servidor proxy, se considerará un atributo fijo a tal efecto. Se utilizará para ello el atributo mayormente utilizado, que corresponde al atributo Radius "Ascend-Assign-IP-Pool".

En la fase II del desarrollo del servidor proxy, el cliente podrá seleccionar el atributo Radius que emplea para indicar de qué pool se asignará una dirección IP al usuario. Puesto que el atributo Radius elegido a tal efecto puede variar de un PSI a otro, esta información será procesada por el servidor proxy mediante una librería dinámica. Se desarrollará por tanto una API personalizada para el tratamiento de dicho atributo.

## 4.3. **Criterios para el reenvío de paquetes**

El servidor proxy Radius seleccionará el servidor Radius del PSI al que reenviará los paquetes Radius basándose en los siguientes parámetros:

- Número de teléfono llamado, obtenido mediante el atributo Called-Station-Id.
- Mnemónico, obtenido mediante el atributo User-Name
- Clientes Radius, conocido mediante la dirección IP origen de la que proviene la petición.

En la fase I del desarrollo, no se aplicará ninguno de los criterios indicados anteriormente para reenviar los paquetes Radius, sino que el servidor proxy Radius reenviará al servidor Radius especificado todos los paquetes que reciba.

En la fase II del desarrollo, se aplicarán los criterios anteriormente indicados.

## 4.4. **Mapeo y filtrado de atributos Radius**

Será necesario que el servidor proxy realice funciones de sustitución y filtrado de atributos Radius para controlar el paso de atributos Radius al Servidor de Túneles.

La conversión de atributos se llevará a cabo en la fase II del desarrollo, pero no en la fase I del mismo.

## 4.5. **Detección de sesiones colgadas en el Servidor Proxy**

Será necesario modificar la herramienta rad\_tool para poder detectar sesiones colgadas en el servidor proxy.

Las modificaciones necesarias en esta herramienta serán llevadas a cabo tanto en la fase I como en la fase II del desarrollo.

Las actuaciones hechas sobre el proxy a través del programa rad\_tool se notificarán adecuadamente a los servidores Radius del PSI (generalmente a través de mensajes de stop).

## A ANEXO A: FUENTES PARA EL DESARROLLO DE LIBRERIAS DINAMICAS

En el caso de que el cliente quiera emplear una base de datos de usuarios externa, en lugar de las suministradas por defecto en el servidor Radius, ha de desarrollar su propia librería dinámica para el acceso a dicha base de datos, empleando para ello los fuentes que se proporcionan en el servidor Radius.

Se muestra a continuación el código de los fuentes que se necesitan para llevar a cabo tal desarrollo, y que son `UsuExtPwd.c` y `EscribeFac.c`. Dichos fuentes se encuentran, en el directorio `/opt/radius/lib` para Solaris y en el directorio `/radius/lib` para Linux. Para Windows NT, se encuentran en el directorio `c:\radius\lib` en la última versión del servidor Radius (versión 4.01), si bien para versiones anteriores a la indicada, pueden localizarse en el directorio `c:\radius\src`.

También se muestra el código del fichero Makefile que genera, a modo de ejemplo, dichas librerías, para los sistemas operativos Solaris y Linux.

### A.1 Makefile

Se indica el código del fichero Makefile:

FICHERO: Makefile

```
# Makefile for RADIUS -
# Remote Authentication Dial In User Service
#
# ASCEND: @(#)Makefile 1.3 (95/07/25 00:55:22)
#
#Host OS:
# aix:
# add -Daix to CFLAGS
#
# HP-UX:
# CFLAGS=
# CC= gcc -traditional
#
# SCO:
# add LIBS= -lsocket
#
# SunOs:
# CFLAGS= -DOSUN
# CC= gcc -traditional
#
# Solaris with gcc:
# CFLAGS= -O -Dsys5 -DSOLARIS
# CC= gcc -ansi
# add LIBS= -lnsl -lsocket
#
# Solaris with cc:
# CFLAGS= -O -Dsys5 -DSOLARIS
# CC= cc
# add LIBS= -lnsl -lsocket
#
# unixware:
# add -Dunixware to CFLAGS
# add LIBS= -lucb -lnsl -lsocket
#
# BSDI:
# CC= gcc # Note: cannot use -ansi on some versions
#
```

```

# Linux:
#   add -Dsys5 to CFLAGS
#
#
#External Authentication Servers:
# Security Dynamics (ACE):
#   add -DACE to AUTH_SERVERS
#   add SD_INCLUDES to AUTH_INCLUDES
#   add -lstdiclient to AUTH_LIBS
# Enigma Logic (SAFWORD):
#   add -DSAFWORD to AUTH_SERVERS
#   add SW_INCLUDES to AUTH_INCLUDES
#   add -lidp to AUTH_LIBS
#

SRCDIR=.

# Pick the combination of external authentication servers to use.  Requires
# the authentication source and libraries provided by the server vendors.
#
AUTH_SERVERS=

# Specify the authentication server include list to use
#
AUTH_INCLUDES=

AUTH_LIBS=

# Pick your compiler
#
#CC= gcc -ansi -Wall      # set the GCC compiler to ANSI mode and picky
CC= gcc                  # set the GCC compiler to ANSI mode (Linux)
                        # set the GCC compiler to ANSI mode (Linux)
#CC= gcc                 # set the GCC compiler for BSDI and friends
#CC= cc -Xc -v -fd#      # set the SUN compiler to full-whine ANSI mode
#CC= cc -Xc -v          # set the SUN compiler to partial-whine ANSI mode
#CC= cc -Xc             # set the SUN compiler to ANSI mode
#CC= cc                 # Solaris SUN compiler

# Pick your compiler flags
# -Dsys5                System V
# -DSOLARIS             Solaris (use sys5 also)
# -DOSUN                SunOS 4.1.x
# $(AUTH_SERVERS)      See above
#

#   use this for BSDI
#CFLAGS= -O -DBSDI $(AUTH_SERVERS)
#   use this for SunOS
#CFLAGS= -O -DOSUN $(AUTH_SERVERS)
#   use this for Solaris 2.x
CFLAGS= -O -Dsys5 -DSOLARIS $(AUTH_SERVERS)
#CFLAGS= -g -Dsys5 -DSOLARIS $(AUTH_SERVERS)
#   use this for Linux
#CFLAGS= -O -Dsys5 $(AUTH_SERVERS)
#CFLAGS= -g -Dsys5 $(AUTH_SERVERS)

#   use this for BSDI
#DLLFLAGS=
#LDLLFLAGS=
#   use this for SunOS
#DLLFLAGS=
#LDLLFLAGS=
#   use this for Solaris 2.x
DLLFLAGS= -KPIC
LDLLFLAGS= -G

```

```
# use this for Linux
#DLLFLAGS= -fpic
#LDLLFLAGS= -shared

# Pick any load options and the load libraries. The AUTH_LIBS
# are selected above
#
LDFLAGS=

# See above for the things you may need to add here
#
#LIBS= $(AUTH_LIBS) # use this for Linux, BSDI and friends
#LIBS= -lnsl $(AUTH_LIBS) # use this for SunOS
LIBS= -lnsl -lsocket $(AUTH_LIBS) # use this for Solaris 2.x

# use this for SunOs. Linux, BSDI
#BMLIBS= -ldb
# use this for Solaris
DBMLIBS=

CMN_OBJS=
SERVER_OBJS=$(CMN_OBJS)
SERVERDBM_OBJS=$(CMN_OBJS)

SERVER_SRCS=$(SERVER_OBJS:%.o:%.c)
SRVR_SOURCES=$(SERVER_OBJS:.o=.c)
SOURCES=$(SRVR_SOURCES) $(SRCDIR)/radpass.c $(SRCDIR)/md5.c

ALL_INCLUDES=$(INCLUDES) $(SRCDIR)/md5.h

all: sim radpass lib rad_tool dbm libmsj

dbm: builddbm libdbm libmsjdbm

sim: simula

SIMULA_OBJS= $(SRCDIR)/simula.o $(SRCDIR)/sim_dict.o $(SRCDIR)/md5.o

SIMULA_INCLUDES=$(SRCDIR)/sim_rad.h

simula: simula.o sim_dict.o md5.o
$(CC) $(CFLAGS) -o simula $(SIMULA_OBJS) $(LIBS)

simula.o: $(SRCDIR)/simula.c $(SIMULA_INCLUDES)
$(CC) $(CFLAGS) -c $(SRCDIR)/simula.c

sim_dict.o: $(SRCDIR)/sim_dict.c $(SIMULA_INCLUDES)
$(CC) $(CFLAGS) -c $(SRCDIR)/sim_dict.c

rad_tool: rad_tool.o md5.o
$(CC) $(CFLAGS) -o rad_tool rad_tool.o md5.o $(LIBS)

rad_tool.o: $(SRCDIR)/rad_tool.c
$(CC) $(CFLAGS) -c $(SRCDIR)/rad_tool.c

radpass: radpass.o md5.o util.o

radpass.o: $(SRCDIR)/radpass.c $(INCLUDES)
$(CC) $(CFLAGS) -c $(SRCDIR)/radpass.c

md5.o: $(SRCDIR)/md5.c $(SRCDIR)/md5.h
$(CC) $(CFLAGS) -c $(SRCDIR)/md5.c

usr_read.o: $(SRCDIR)/usr_read.c $(INCLUDES)
$(CC) $(CFLAGS) -c $(SRCDIR)/usr_read.c
```

```

builddb: builddb.o usr_read.o
        $(CC) $(CFLAGS) -o builddb builddb.o usr_read.o $(DBMLIBS) $(LIBS)

builddb.o: $(SRCDIR)/builddb.c
        $(CC) $(CFLAGS) -c $(SRCDIR)/builddb.c

libdb:      libInfoviadbm.so.1

libmsjdb:   libInfoviamsjdbm.so.1

lib:        libInfovia.so.1

libmsj:     libInfoviamsj.so.1

libInfovia.so.1: EscribeFac.o UsuExt.o
        $(CC) $(LDLLFLAGS) $(CFLAGS) -o libInfovia.so.1 EscribeFac.o  UsuExt.o

libInfoviamsj.so.1: EscribeFacMsj.o UsuExt.o
        $(CC) $(LDLLFLAGS) $(CFLAGS) -o libInfoviamsj.so.1 EscribeFacMsj.o
        UsuExt.o ../bin/libfifo.a

libInfoviadbm.so.1: EscribeFac.o UsuExtDbm.o
        $(CC) $(LDLLFLAGS) $(CFLAGS) -o libInfoviadbm.so.1 EscribeFac.o
        UsuExtDbm.o $(DBMLIBS)

libInfoviamsjdbm.so.1: EscribeFacMsj.o UsuExtDbm.o
        $(CC) $(LDLLFLAGS) $(CFLAGS) -o libInfoviamsjdbm.so.1 EscribeFacMsj.o
        UsuExtDbm.o $(DBMLIBS) ../bin/libfifo.a

EscribeFac.o: $(SRCDIR)/EscribeFac.c
        $(CC) $(DLLFLAGS) $(CFLAGS) -c $(SRCDIR)/EscribeFac.c

EscribeFacMsj.o: $(SRCDIR)/EscribeFac.c
        $(CC) $(DLLFLAGS) $(CFLAGS) -DMSJ_MODE -c $(SRCDIR)/EscribeFac.c -o
        EscribeFacMsj.o

UsuExt.o: $(SRCDIR)/UsuExt.c
        $(CC) $(DLLFLAGS) $(CFLAGS) -c $(SRCDIR)/UsuExt.c

UsuExtDbm.o: $(SRCDIR)/UsuExt.c
        $(CC) $(DLLFLAGS) $(CFLAGS) -DDBM_MODE -c $(SRCDIR)/UsuExt.c -o
        UsuExtDbm.o

lint:
        -lint -hbacvx -DLINT $(SOURCES)

clean:
        rm -f *.o radpass builddb simula rad_tool libInfovia*.so*

```

## A.2 Fichero UsuExtPwd.c

Se indica el código del fichero UsuExtPwd.c:

FICHERO: UsuExtPwd.c

```

static char sccsid[] =

#ifdef _WIN32
#include <winsock.h>
#include <stdio.h>
#include <time.h>
#define DllExport __declspec( dllexport )
#else

#include <sys/types.h>
#include <sys/socket.h>

```

```
#include <sys/time.h>
#include <netinet/in.h>

#include <stdio.h>
#include <netdb.h>
#include <pwd.h>
#include <time.h>
#include <ctype.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#endif

#if defined(DBM_MODE)
#   if defined(SOLARIS) || defined(NCR) || defined(MAXION)
#       include </usr/ucbinclude/dbm.h>
#   else
#if defined(unixware) || defined(SINIX)
#       include <rpcsvc/dbm.h>
#else
#       include <dbm.h>
#endif /* unixware */
#   endif /* SOLARIS */
#endif /* DBM_MODE */

/*Codigos devueltos por UsuExt*/

/*Codigos de Error*/
#define MISSING_NEWLINE (-93)
#define NO_USER_OR_DEFAULT_NAME (-92)
#define ZERO_LENGTH_NAME (-91)
#define NO_USER_FILE (-90)
#define NO_WHITESPACE (-89)
#define USERFILE_RENAME_FAILED (-88)
#define USERFILE_READ_ERR (-87)
#define USERFILE_WRITE_ERR (-86)
#define END_OF_USERS_LIST (-85)
#define MEMORY_ERR (-84)

/*Codigos de Exito*/
#define AUT_LOCAL 0 /* Se devuelve el perfil del usuario pero la password
se comprueba en el ejecutable de RADIUS*/

/* Si la password se comprueba aqui, en la dll:*/
#define AUT_EXT_OK 3 /* Usuario autenticado*/
#define AUT_EXT_NO_OK 4 /* Usuario NO autenticado*/

#define AUTH_PASS_LEN 254
#define AUTH_VECTOR_LEN 16
#define PW_PASSWORD 2
#define PW_CHAP_PASSWORD 3

typedef struct {
    int type;
    int size;
    char strvalue[AUTH_PASS_LEN];
    char challenge[AUTH_VECTOR_LEN];
} STRUCT_PASS;

#ifdef _WIN32
DllExport
#else
extern
#endif
```

```

int
#ifdef _WIN32
WINAPI
#endif
UsuExt(char * nombre_archivo,char * nombre_usuario, char * line);

#ifndef _WIN32
/*****
*
*   Function: userinfo_open
*
*   Purpose: Opens the users information file. Do different
*   things different ways depending on whether we are using
*   a flat file or a dbm database.
*
*   Returns: zero upon success, non-zero upon failure
*
*****/

static int
userinfo_open(userfd, infoname)
FILE **userfd;
char *infoname;
{
#ifdef DBM_MODE
    (void)userfd;
    return dbmopen(infoname);
#endif
}

/*****
*
*   Function: userinfo_close
*
*   Purpose: Close the users information file. Do different
*   things different ways depending on whether we are using
*   a flat file or a dbm database.
*
*****/

static void
userinfo_close(userfd)
FILE *userfd;
{
#ifdef DBM_MODE
    (void)userfd;
    dbmclose();
#endif /* SCO50 */
}
#endif /* _WIN32 */

/*****
*
*   Funcion: UsuExt
*
*   Prototipo: int UsuExt(char * nombre_archivo,
*                       char * nombre_usuario,
*                       char * line);
*
*           nombre_archivo: Puntero al nombre del archivo
*                           de la base de datos de usuarios.
*                           Necesario para DBM.
*
*           nombre_usuario: Puntero al nombre del usuario a

```

```
*
*                                     consultar.
*
*           line:           Puntero al array de 4096
*                           caracteres ubicado por radiusd
*                           donde se debera copiar los datos
*                           del perfil del usuario localizado.
*                           Ademias en el se recibe la password
*
* La Password se obtiene de la cadena line y se
* almacena en las variables:
*
*           type_pass: Tipo de Password:
*                       PW_PASSWORD ( PAP Password recibida
*                                   ya descriptada).
*
*                       PW_CHAP_PASSWORD ( CHAP Password).
*
*           size_pass: Longitud de la Password incluido
*                       un caracter nulo de fin de cadena.
*
*           str_pass: Cadena con la password descriptada
*                       acabada con un caracter nulo.
*
*           challenge: Challenge que se utilizo para en-
*                       criptar la Password en el caso de
*                       CHAP.
*
*
* Objetivo: Busca el nombre de usuario en la base de datos.
*           Devuelve un string con los datos del usuario.
*           Opcionalmente puede comprobar tambien si es
*           correcta la Password del usuario.
*
* Devuelve: AUT_LOCAL:   Devuelve string con los dados del
*                       usuario pero la password se comprueba en
*                       radiusdll.
*
*           AUT_EXT_OK:   Devuelve string con los dados del
*                       usuario, la password ya ha sido comprobada en la
*                       dll y es correcta.
*
*           AUT_EXT_NO_OK: La password ya ha sido
*                       comprobada en la dll y NO es correcta
*                       Usuario rechazado.
*
*           1: No se usa esta funcion.
*               radiusdll buscara el
*               usuario internamente.
*
*           <0: Errores codificados por radiusd
*
* Errores:
* MISSING_NEWLINE           (-93)
* NO_USER_OR_DEFAULT_NAME  (-92)
* ZERO_LENGTH_NAME         (-91)
* NO_USER_FILE             (-90)
* NO_WHITESPACE            (-89)
* USERFILE_RENAME_FAILED  (-88)
* USERFILE_READ_ERR       (-87)
* USERFILE_WRITE_ERR      (-86)
* END_OF_USERS_LIST       (-85)
* MEMORY_ERR               (-84)
*
*
* Estructura de los datos de usuario:
*
*
```



```

*           Los datos seran un string con dos lineas
*           basados en la estructura de un usuario del
*           archivo usuarios_INFOVIA.
*
*           Ejemplo:
*           Para el usuario usul,
*
*           usul Password = "madrid"
*           SESIONES-INFOVIA = 0,
*           NIVEL-INFOVIA = 1,
*           Ascend-Idle-Limit = 0,
*           Ascend-Maximum-Time = 10
*
*           Se debera de rellenar en line
*
*           strcpy( line, "Password=madrid\nSESIONES-INFOVIA=0,NIVEL-INFOVIA=1,Asc...");
*
*
*****/

#ifdef _WIN32
DllExport
#else
extern
#endif
int
#ifdef _WIN32
WINAPI
#endif
UsuExt(char *nombre_archivo,char* nombre_usuario,char *line)
{

#ifdef DBM_MODE
FILE      *userfd;
int       mode;
datum    named;
datum    contentd;
int      stat;
#endif /* DBM_MODE */

#ifdef AUT_EXT
int type_pass, size_pass;
char str_pass[AUTH_PASS_LEN];
char challenge[AUTH_VECTOR_LEN];
int hay_password=0;
char *ptr;
int i;
/* Obtenemos password*/

ptr=line;
while(*ptr !=0)
    ++ptr;

++ptr;

if(*ptr==' ')
{
    hay_password=1;
    ++ptr;
    memcpy(&type_pass, ptr, sizeof(int));
    ptr=ptr+sizeof(int)+1;
    memcpy(&size_pass, ptr, sizeof(int));
    ptr=ptr+sizeof(int)+1;
    memcpy(str_pass, ptr, size_pass);
    str_pass[size_pass] = 0;

```

```

    ptr=ptr+size_pass+1;
    memcpy(challenge, ptr, AUTH_VECTOR_LEN);
    if (type_pass == 2)
        printf("Password=%s\n",str_pass);
    else if (type_pass == 3) {
        printf("Chap_Password = ");
        for (i=0; i<size_pass; i++) printf("%02x ",
(u_int)(u_char)str_pass[i]);
        printf("\n");
        printf("ChapChallenge = ");
        for (i=0; i<16; i++) printf("%02x ", (u_int)(u_char)challenge[i]);
        printf("\n");
    }
}
#endif /* AUT_EXT */

#ifdef DBM_MODE

/*
 * Check for valid input, zero length names not permitted
 */

if( userinfo_open(&userfd, nombre_archivo) != 0 ) {
    return(NO_USER_FILE);
}

named.dptr = nombre_usuario;
named.dsize = strlen(nombre_usuario)+1;
contentd = fetch(named);

if(contentd.dsize == 0) { /* name not found */
    named.dptr = "DEFAULT";
    named.dsize = strlen("DEFAULT")+1;
    contentd = fetch(named);
    if(contentd.dsize == 0) { /* default not found */
#ifdef !defined(SCO50)
        dbmclose();
#endif /* SCO50 */
        return(NO_USER_OR_DEFAULT_NAME);
    }
}
strcpy(line, contentd.dptr);
userinfo_close(userfd);

#ifdef AUT_EXT_AU
/*Ejemplo de autenticacion:

    longitud password de 1 a 5->NO_OK
    longitud password de 6 a 8 ->OK
    longitud password > 8 ->Autenticacion en el ejecutable radiusddl1
*/
if(hay_password)
{
    if(size_pass>8)
        return(AUT_LOCAL);
    else if(size_pass<=5)
        return(AUT_EXT_NO_OK);
    else
        return(AUT_EXT_OK);
}
#endif /* AUT_EXT_AU */

return(AUT_LOCAL);

```

```
#else /* DBM_MODE */
    return(1);
#endif /* DBM_MODE */
}
```

### A.3 Fichero EscribeFac.c

Se indica el código del fichero EscribeFac.c:

FICHERO: EscribeFac.c

```
#ifdef _WIN32
#include <winsock.h>
#include <direct.h>
#include <stdio.h>
#include <time.h>
#define DllExport __declspec( dllexport )
#else

#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/file.h>
#include <netinet/in.h>

#include <stdio.h>
#include <netdb.h>
#include <pwd.h>
#include <time.h>
#include <limits.h>
#include <ctype.h>
#include <unistd.h>
#include <signal.h>
#include <errno.h>
#include <sys/wait.h>
#include <sys/stat.h>
extern int log_err (char *msg);
#endif
/*****
*
*   Funcion: EscribeFac
*
*   Prototipo: int EscribeFac( char * detail, char * buffer)
*
*           detail: Puntero al nombre del archivo de
*                   la base de datos de tarificacion.
*
*           buffer: Puntero a los datos de
*                   tarificacion.
*
*
*   Objetivo: Entrega los datos de tarificacion para su
*             almacenamiento o procesado
*
*   Devuelve: 0 exito.
*             1 fallo.
*
*   Estructura de los datos de tarificacion:
*
*           Los datos seran un string de pares <parametro>=<valor>
*           separados por comas.
*
*
*****/
```

```
#ifdef _WIN32
DllExport
#endif
int
#ifdef _WIN32
WINAPI
#endif
EscribeFac( char * detail, char * buffer)
{
    long          curtime;
    FILE          *outfd;
    char          *ptr;

    struct tm *pres;

    if((outfd = fopen(detail, "a")) == (FILE *)NULL) {
        return(1);
    }
    curtime = time(0);
    pres=localtime(&curtime);
    fprintf( outfd,"%02d/%02d/%4d %02d:%02d:%02d\n\t",
        pres->tm_mday, pres->tm_mon+1, pres->tm_year+1900,
        pres->tm_hour, pres->tm_min, pres->tm_sec);

    ptr = buffer;
    while (*ptr != 0)
    {
        if ( *ptr == ',')
        {
            fputs("\n\t", outfd);
        }
        else
        {
            fputc(*ptr, outfd);
        }
        ptr++;
    }
    fputs("\n", outfd);
    fputs("\n", outfd);

    fclose(outfd);
#ifdef MSJ_MODE
    /* Empieza modificacion V3.0 msj */
    detail = detail + strlen(detail) + 1;
    if (memcmp(detail,"V3.0",4) != 0) {
        /* radiusd no es V3.0 */
        return(0);
    }
    ptr++;
    if (*ptr == 0) {
        return(0);
    }
    {
        void DefinirDirectorioConfig(char szDirectorio[]);
        int EnviarMensajeUsuario(char szUsuario[],
            char szPassword[],
            char szDirIp[]);

        char passwd[256];
        char name[256];
        char dirip[16];
        detail = detail + strlen(detail) + 1;
        DefinirDirectorioConfig(detail);
        strcpy(dirip, ptr);
        ptr = ptr + strlen(ptr) + 1;
        strcpy(name, ptr);
        ptr = ptr + strlen(ptr) + 1;
        strcpy(passwd, ptr);
    }
}

```

```
/** printf("Path=%s, Name=%s,  
Pwd=%s,Dirip=%s\n",detail,name,passwd,dirip);***/  
    if (!EnviarMensajeUsuario(name, passwd, dirip))  
#if defined(aix) || defined(SC050) || defined(_WIN32) || defined(__osf__)  
    return(0);  
#else  
    log_err("No se ha podido enviar msj\n");  
#endif  
}  
#endif  
    return(0);  
}
```

## 5. Referencias y glosario

### 5.1. Referencias

- “*Remote Authentication Dial In User Service (RADIUS)*”. RFC 2138.
- “*RADIUS Accounting*”. RFC 2139.

### 5.2. Glosario

AAA	Authentication, Autorization and Accounting
CHAP	Challenge Handshake Authentication Protocol
IP	Internet Protocol
PSI	Proveedor de Servicios de Internet
NA	Nodo de Acceso
PAP	Password Authentication Protocol
PPP	Point to Point Protocol
RADIUS	Remote Authentication and Dial In User Service
RDSI	Red Digital de Servicios Integrados
RTC	Red Telefónica Conmutada
SC	Servidor de Comunicaciones