SERVICIO DE MENSAJERÍA EMPRESARIAL EN TIEMPO REAL UTILIZANDO EL PROTOCOLO JABBER v.2.0

YESID ALBERTO CANO BÁRCENAS

JAVIER ERNESTO LOZANO SALAZAR

UNIVERSIDAD INDUSTRIAL DE SANTANDER

ESPECIALIZACION EN TELECOMUNICACIONES

FACULTAD DE INGENIERIAS FISICOMECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

BUCARAMANGA

2007

SERVICIO DE MENSAJERÍA EMPRESARIAL EN TIEMPO REAL UTILIZANDO EL PROTOCOLO JABBER v.2.0

YESID ALBERTO CANO BÁRCENAS

JAVIER ERNESTO LOZANO SALAZAR

Este proyecto es presentado como requisito para optar al título de Especialista en Telecomunicaciones

Director

MG. FREDY ALFONSO BELTRAN MIRANDA

UNIVERSIDAD INDUSTRIAL DE SANTANDER

ESPECIALIZACION EN TELECOMUNICACIONES

FACULTAD DE INGENIERIAS FISICOMECÁNICAS

ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

BUCARAMANGA

2007

RESUMEN

TITULO: SERVICIO DE MENSAJERÍA EMPRESARIAL EN TIEMPO REAL UTILIZANDO EL PROTOCOLO JABBER v.2.0[*]

AUTORES: CANO BÁRCENAS, Yesid Alberto
LOZANO SALAZAR, Javier Ernesto[**]

PALABRAS CLAVES: Mensajería Instantánea, Empresarial, Jabber, Presencia, Seguridad, XML, XMPP.

DESCRIPCION:

En la actualidad las organizaciones de tipo empresarial requieren de instrumentos ágiles y eficaces para la comunicación entre sus miembros y en muchos casos, con participantes externos. Para satisfacer este requisito, las empresas acuden a las herramientas tradicionales como la telefonía y acuden cada vez más a nuevas tecnologías como la transmisión de voz y video sobre Internet. Pese a que los costos de utilización son cada vez menores conforme evoluciona la tecnología, y que la información en tiempo real cada vez requiere menos ancho de banda, la utilización de los recursos de una red sigue siendo considerable comparada con una transferencia de texto. Esta situación es especialmente crítica en escenarios donde el flujo de información en tiempo real es alto, y se tienen múltiples

[*] Trabajo de grado

[**] Facultad de Ingenierías Fisicomecánicas. Especializacion en Telecomunicaciones. Director: Mg. Fredy Alfonso Beltran Miranda

conexiones concurrentes, lo que produce un deterioro en el rendimiento de las redes.

Por esta razón, muchas empresas han adoptado a la Mensajería Instantánea para que sus empleados establezcan comunicaciones rápidas entre si, sin sacrificar sus recursos de red disponibles. Sin embargo, los servicios de mensajería públicos, pese a que son de uso gratuito, no son adecuados para las organizaciones porque los empleados pueden contactar a personas diferentes a la organización, comprometiendo la seguridad de la información. Sin embargo, debido a la necesidad de reducir los costos de comunicaciones, muchas empresas mantienen el dilema de apoyarse de una Mensajería Instantánea o de no permitirla definitivamente. Nuestro trabajo pretende estimular el uso de la Mensajería Instantánea en el entorno corporativo, describiendo los requisitos que debe cumplir un servicio de Mensajería Instantánea Empresarial y propone la implementación de un servicio de Mensajería Instantánea de bajo costo, ligero y seguro, mediante el uso del servicio de distribución libre, Jabber v.2.0.

ABSTRACT

TITLE: SERVICIO DE MENSAJERÍA EMPRESARIAL EN TIEMPO REAL UTILIZANDO EL PROTOCOLO JABBER v.2.0[*]

AUTHOR: CANO BÁRCENAS, Yesid Alberto
LOZANO SALAZAR, Javier Ernesto[**]

KEYWORDS: Instant Messaging, Corporate, Jabber, Presence, Security, XML, XMPP.

DESCRIPTION:

Today, corporate organizations require agile and efficient instruments for communications between their members and in some cases, with external participants. In order to satisfy this requirement, enterprises rely on traditional media like telephony, and decide to use new technologies as voice and video transmission over the Internet. In spite of costs of utilization are much lower as technology evolves, and real-time information require less bandwidth, the use of network resources is still considerable compared to text transfers. This situation is specially critical in scenarios where the flow of real-time data is high, and there are multiple concurrent connections, which produces a decrease in network performance.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Fisicomecánicas. Especializacion en Telecomunicaciones. Director: Mg. Fredy Alfonso Beltran Miranda

Because of that, most enterprises have adopted Instant Messaging for their employees, to establish fast and reliable communications between them, without sacrifice of available network resources. However, Public Instant Messaging services, in spite of their free cost of utilization, are not adequate because employees could contact people from outside the organization, compromising security of information. Due to the need of reduce costs in communications, most enterprises are on a dilemma of deploy an Instant Messaging service, or completely avoid it. Our work pretends to help stimulate the use of Instant Messaging in a corporate environment, describing the requisites it have to accomplish and proposes the deployment of a light, low cost and safe Instant Messaging service by the use of open source service, Jabber v.2.0.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Pág.

INTRODUCCIÓN

Las tecnologías y anchos de banda pueden ser cada vez más espectaculares, pero debajo de cables y ondas radioeléctricas están los usuarios, que han encontrado en la Mensajería Instantánea (MI[1]) una nueva cultura para hablar a distancia.

El servicio de Mensajería Instantánea ofrece una ventana donde se escribe el mensaje, en texto plano o acompañado de iconos o "emoticons[2]", y se envían a uno o varios destinatarios quienes reciben los mensajes en tiempo real, el receptor lo lee y puede contestar en el acto.

La Mensajería Instantánea ha tenido una aceptación rápida y masiva durante los últimos años. La primera forma de MI apareció en la década del ochenta (80) en los sistemas UNIX con el uso del comando talk, que permitía la conversación entre pares. En la década de los noventas (90) se introdujo ICQ[3], el cual fue el primer protocolo de IM en incluir los conceptos de envío de mensajes, estados y uso de la lista de contactos; desde entonces se han desarrollado paralelamente una variedad de protocolos propietarios como el MSN, AOL, YAHOO y otros.

Inicialmente fue utilizada en los hogares para la comunicación con familiares y amigos, posteriormente adoptada a nivel empresarial. Los propios empleados fueron quienes instalaron las herramientas y comenzaron a utilizarlas para comunicarse con sus compañeros de trabajo, clientes y familia. Posiblemente

---

[1] MI de sus siglas en español Mensajería Instantánea

[2] Figuras que representan estados de animo

[3] I seek you o te busco, es un servicio de mensajería instantánea

haya contribuido el hecho de que las mismas son de fácil acceso, gratuitas y simples de instalar. Esta situación provoco que la instalación se haya realizado, en la mayoría de los casos, sin la supervisión y el consentimiento del personal informático.

Existen varios problemas de seguridad asociados a la mensajería instantánea y una amplia variedad de herramientas que intentan mitigarlos. Algunos de estos problemas están asociados a la posibilidad que brinda MI de transferir archivos, lo que facilita la propagación de virus o gusanos. Uno de los muchos controles que deciden utilizar las organizaciones, es el bloquear la opción de transferencia de archivos y que solo mensajes de texto puedan ser transmitidos. De esta forma se restringe la utilización de MI a la transmisión de "solo texto", lo que para muchos no resulta un riesgo de seguridad.

El concepto general desde el punto de vista Gerencial en las organizaciones sobre el fenómeno de la MI, se basa fundamentalmente en aspectos de productividad de los usuarios sin importar la seguridad de la información corporativa. En las empresas las soluciones de mensajería buscan restringir los contactos a sus pares laborales, con el fin de evitar distracciones, spam[4] y recientemente MI-phishing[5]. De todas formas, solamente una fracción de las organizaciones que utilizan MI lo controla. Los usuarios domésticos básicamente no realizan controles, si bien no son el foco del presente trabajo.

La mensajería instantánea se ha convertido en la killer application (aplicación muy exitosa) de las herramientas informáticas de comunicación mas utilizada en los últimos tiempos, esto se debe posiblemente a que proporciona una manera rápida de comunicarse con clientes, compañeros de trabajo y socios, lo que puede resultar ventajoso en un ambiente de negocios donde segundos pueden hacer la

[4] Mensajes no solicitados, habitualmente de tipo publicitario.

[5] Viene del inglés fishing (pesca), alusión al acto de pescar usuarios

diferencia. Brinda además información presencial de los contactos, indicando si los mismos se encuentran conectados o no y si están disponibles para establecer una conversación.

Con este estudio pretendemos proponer e implementar un servicio para soluciones corporativas de MI que utilice el protocolo Jabber v.2.0 (una versión de software libre), que permita a las organizaciones tener sus propios servidores de MI previniendo de esta forma que los datos sean interceptados y de esta forma mantener el trafico de mensajería instantánea de la organización dentro de la misma.

La Monografía esta organizada de la siguiente manera. El capítulo I introduce conceptos de los Sistemas de Mensajería Instantánea sobre el diseño de los requisitos y las políticas de administración del servicio para las organizaciones. En el capítulo II se presentan aspectos fundamentales del servicio de Mensajería Instantánea Jabber v.2.0 y sus características. El capítulo III se hace una breve descripción de los procedimientos de implantación de la solución de Mensajería Instantánea utilizando el protocolo Jabber v.2.0 para un entorno empresarial. Continuamos con el capítulo IV que abarca los resultados obtenidos, análisis de los resultados y pruebas de funcionamiento en el escenario propuesto. Por último se termina con unas conclusiones y recomendaciones sobre el desarrollo de la propuesta.

1. CONCEPTUALIZACION DE UN SISTEMA DE MENSAJERIA INSTANTANEA

La Mensajería Instantánea es un conjunto de programas que utilizan el protocolo TCP IP[6] que sirven para enviar y recibir mensajes instantáneos con otros usuarios conectados a Internet u otras redes.

La Mensajería Instantánea está llamada a convertirse en el centro del universo de la comunicación a medida que funciones de un nivel más alto se van integrando cada vez más en las diferentes plataformas de MI existentes.

1.1. MARCO CONCEPTUAL PARA LA MENSAJERIA INSTANTANEA

Primero fue el correo electrónico, después llegó el chat[7] y lo último en sumarse son los mensajes instantáneos, los cuales llegan como alternativa más directa, cómoda y veloz para mantener conversaciones con otros clientes por medio de mensajes de texto en tiempo real.

Los Sistemas de Mensajería Instantánea –SMI– resultan un punto intermedio entre los sistemas de chat - se lo conoce también como IRC, siglas de Internet Relay Chat - (instantaneidad y no privacidad) y los mensajes de correo electrónico (no instantaneidad y privacidad). Son programas generalmente gratuitos y versátiles, residen en el escritorio y mientras haya una conexión a Internet están activos.

[6] Conjunto básico de protocolos de comunicación de redes

[7] En inglés significa charla, comunicación escrita a través de Internet

1.1.1. Definición de Mensajería Instantánea - MI. Comencemos con una definición sencilla sobre esta tecnología, con la intención de limitar el alcance de esta monografía.

La MI se puede definir como un servicio, que permite a los usuarios tener un medio de comunicación eficaz que funciona sobre diferentes tipos de dispositivos que dispongan de conectividad con una red IP. Hoy por hoy el servicio más utilizado es el chat de texto entre PCs; aunque la IM se puede utilizar desde terminales de telefonía móvil, o PDAs, y puede incorporar voz y vídeo. Esto significa que una plataforma de IM puede ser la base para el desarrollo de una plataforma integrada de comunicaciones en plena convergencia tecnológica: si además de la voz y el vídeo, le añadimos la telefonía IP, tendremos un medio para comunicarnos con cualquier persona en cualquier momento, independientemente de la forma en que hayamos accedido a Internet (acceso inalámbrico o por cable, terminal móvil o fijo); todo esto sin considerar las capacidades de colaboración que ya hoy se están incorporando.

1.1.2. El concepto de Presencia. Este concepto hace referencia a la función existente en la MI para poder rastrear a los usuarios determinando de esta forma si están en línea o disponibles para el intercambio de mensajes de texto y ficheros. Esta es la idea detrás de la "conciencia de presencia", un concepto basado en la noción de que los dispositivos en una red pueden detectar automáticamente otros dispositivos.

A muchos, la perspectiva de información que puede revelar la disponibilidad de una persona en determinado momento en cualquier parte del mundo, les parece alarmante. Sin embargo, los usuarios se tranquilizan cuando puede ver, a la distancia, los movimientos de quienes conforman sus círculos íntimos, como son los familiares y amigos.

De la mano de la tranquilidad, viene la desagradable sensación de ser observado, una lección que ya experimentaron millones de usuarios de mensajes instantáneos. La información que era privada (o, al menos, difícil de conseguir), de pronto y con poco esfuerzo, puede estar al alcance de empleadores, compañeros de trabajo, amigos, familiares y, a veces, extraños.

La popularidad de los mensajes instantáneos demuestra que la gente esté dispuesta a aceptar las posibilidades que implica la detección de presencia. Millones de personas usan actualmente productos de mensajería instantánea, según estimaciones de la industria: muchos de ellos dicen que lo que más les gusta de la tecnología es la posibilidad de saber si un amigo está en línea (online).

En resumen, los programadores de software deben estar en condiciones de diseñar sistemas con el concepto de presencia, que satisfagan tanto a quienes buscan privacidad como a quienes quieren un contacto constante.

1.1.3. Estandarización en la Mensajería Instantánea. Los proveedores de Mensajería Instantánea están empezando a abrir sus redes de manera que pronto la MI se convertirá en un servicio universal de comunicación que permitirá llegar tanto a usuarios de PC como de redes móviles. Con este tipo de acuerdos las distintas comunidades quedan comunicadas sin necesidad de utilizar distintas cuentas.

Será interesante observar el proceso de consolidación que se producirá como consecuencia de esta apertura. Evidentemente ya no será necesario mantener varias cuentas en distintos proveedores. Con una se podrá llegar a cualquiera. La lucha por el usuario será feroz.

Orange, Microsoft y Yahoo! son algunos de los nombres que han empezado a moverse para abrir sus redes e interconectar sus comunidades. Sin duda un paso importante, en especial por la talla de los protagonistas. Microsoft y Yahoo! son de lejos los claros dominadores de este canal de comunicación que con el tiempo se han hecho importante en las vidas de los usuarios, particularmente entre los más jóvenes.

Ya ha comenzado un periodo de pruebas en el que los usuarios de los clientes de mensajería instantánea de Microsoft y Yahoo podrán verificar si funcionan correctamente entre sí.

Yahoo y Microsoft han iniciado un periodo de pruebas para que los usuarios de sus sistemas de mensajería instantánea Windows Live Messenger y Yahoo Messenger se comuniquen entre sí a través de cualquiera de los dos servicios.

Los usuarios interesados pueden registrarse para probar la beta pública de Yahoo o de Microsoft. Una vez tengan acceso a la beta, podrán intercambiar mensajes instantáneos entre los dos servicios, así como ver si sus contactos están online, visualizar los mensajes de estado de todos, compartir emoticonos, ver mensajes offline y añadir nuevos contactos de uno y otro servicio.

Microsoft ya ha activado la interoperatividad de mensajería instantánea entre su plataforma de mensajería unificada empresarial Live Communications Server (LCS) y los clientes de mensajería de Yahoo y America Online, así como su propio Windows Live Messenger. Los clientes empresariales que hayan adquirido LCS y Office Communicator, la interfaz cliente de LCS, podrán también conectarse a todos esos clientes.

El programa beta estará disponible en Alemania, Argentina, Australia, Brasil, Canadá, China, Corea, España, Francia, Holanda, Hong Kong, India, Italia,

México, Reino Unido, Singapur, Taiwán y Turquía. Una vez terminado el periodo de pruebas, ambas compañías tienen previsto que esta interoperatividad esté disponible para todos sus usuarios a lo largo de los próximos meses.

En cuanto al cliente de mensajería instantánea de AOL, Microsoft no ha notificado si tiene previsto firmar un acuerdo similar con America Online, aunque sí espera que haya acuerdos de este tipo con otras compañías en el futuro.

1.1.4. Características comunes entre los servicios de Mensajería Instantánea. Los sistemas de mensajería tienen unas funciones básicas aparte de mostrar los usuarios que hay conectados y chatear. Unas son comunes a todos o casi todos los clientes o protocolos y otras son menos comunes, a continuación enumeraremos algunas de éstas características:

Contactos

Mostrar varios estados: Disponible, Disponible para hablar, Sin actividad, No disponible, Vuelvo enseguida, Invisible, no conectado. Con el estado invisible se puede ver a los demás pero los demás a uno no.

Mostrar un mensaje de estado: Es una palabra o frase que aparece en las listas de contactos de tus amigos junto a tu nick. Puede indicar la causa de la ausencia, o en el caso del estado disponible para hablar, el tema del que quieres hablar. A veces, es usado por sistema automáticos para mostrar la temperatura, o la canción que se está escuchando, sin molestar con mensajes o peticiones de chat continuos. También se puede dejar un mensaje de estado en el servidor para cuando se esté

desconectado. Registrar y borrar usuarios de la lista de contactos propia.

Inclusión en la lista: Se puede enviar un mensaje explicando los motivos para la admisión. Rechazar un usuario discretamente: cuando no se quiere que un usuario en concreto le vea a uno cuando se conecta, se puede rechazar al usuario si dejar de estar en su lista de contactos. Solo se deja de avisar cuando uno se conecta.

Agrupar los contactos: Familia, Trabajo, Facultad, etc.

También si se desea podemos usar una imagen o foto que identifique al usuario.

Chateo

Existen varios tipos de mensajes como son:

Aviso: Lanza un mensaje solo. No es una invitación a mantener la conversación, solo se quiere enviar una información.

Invitación a chatear: Se invita a mantener una conversación tiempo real.

Mensaje emergente: Es un aviso que se despliega unos segundos y se vuelve a cerrar. No requiere atención si no se desea.

Muchas veces es útil mostrar cuando el otro está escribiendo, así mismo, se puede usar emoticonos

Otra función que se puede encontrar es la creación de salas (grupos de charla) públicas o privadas, que pueden ser permanentes o que desaparezcan al

quedarse sin usuarios. La seguridad usada es la de restringir el acceso mediante invitaciones certificadas solo a los usuarios que se desea.

Otras

Mandar ficheros, que es la posibilidad de usar otros sistemas de comunicación, como una pizarra electrónica, o abrir otros programas como un VNC o una videoconferencia.

1.2. USO DE LA MENSAJERÍA INSTANTÁNEA EN EL ENTORNO EMPRESARIAL

Si bien es cierto que la mensajería instantánea se ha impuesto entre los usuarios que quieren comunicarse en forma rápida y directa con sus conocidos y amigos, este sistema pretende ser más que una herramienta para que el navegante pueda divertirse un rato. Hoy, la MI busca su pleno desarrollo a nivel laboral. La carrera ha comenzado y muchas empresas están haciendo de la mensajería instantánea su principal medio de comunicación y colaboración entre compañeros de trabajo, proveedores y/o clientes.

En los últimos años, el uso de la MI dentro de una empresa aumentó exponencialmente. Una inversión en mensajería instantánea corporativa debe tomar en cuenta la necesidad de un crecimiento e innovación futuros en nuevos usos. Una solución bien diseñada escalará conforme crezca el negocio con más empleados, un uso más sofisticado de funciones avanzadas y requerimientos de más ancho de banda. Diseñar alta disponibilidad en una solución otorga a los usuarios la confianza para colaborar de una manera eficiente. El soporte para estándares tales como el Protocolo de inicio de sesión (SIP) proporciona la

capacidad para ampliar los servicios multimedia de valor agregado a la comunidad de usuarios. Los desarrolladores requieren un conjunto robusto de Interfaces de Programación de Aplicación (APIs[8]) para servidor y cliente para ampliar un producto de mensajería corporativa a soluciones personalizadas para sus organizaciones.

Se estima según estudios profesionales, que el uso de servicios de Mensajería Instantánea de forma regulada y bajo políticas de uso puede incrementar la productividad organizacional hasta en un 45% de lo normal. Debido a ello, a continuación sugerimos una forma de establecer criterios básicos para el uso de la MI de servicio público en las empresas u organizaciones.

1.2.1. Determinación de las políticas de uso de la Mensajería Instantánea en las Empresas. En un entorno empresarial lo primero es establecer una coordinación entre la Gerencia General (quien coordinara todo el esfuerzo), el Departamento de Personal o de Recursos Humanos (encargado de informar al personal sobre las políticas de uso y/o sancionar al personal que no las cumpla), y el Departamento de Sistemas o Informática (quien vigilara que las políticas se cumplan).

Tomando en cuenta lo anterior, proponemos un conjunto básico de políticas para el uso de la Mensajería Instantánea de servicio público en un entorno organizacional con los siguientes puntos:

1. Establecer y usar una cuenta de MI exclusiva para cuestiones laborales. Queda prohibido el uso de cuentas MI de índole personal para uso de contactos relacionados con el trabajo o la organización.

[8] Siglas en ingles de Application Programming Interface

2. Los contactos MI en la cuenta del usuario deben ser para propósitos, intereses y funcionamiento de la organización.

3. No existirá ningún contacto no relacionado con lo descrito en el punto anterior.

4. Al empleado se le informa que los mensajes no son de índole privado y que continuamente serán monitoreados para supervisar el contenido.

5. Todo empleado que no cumpla con las políticas de la empresa para el uso de la MI, podrá ser sancionado según los reglamentos empresariales

Los puntos anteriores son los básicos para generar un conjunto de políticas de uso en el funcionamiento de la MI; el adicionar mas criterios depende enteramente del alcance, visión, operaciones y administración propia de la empresa. Sin embargo se considera que con estos 5 puntos mencionados, ya es posible tener una visión total del conjunto de políticas que podamos usar en una organización determinada.

1.2.2. Desventajas del uso de servicios de Mensajería Instantánea Personal en las empresas. Actualmente la Mensajería Instantánea es vista como un simple entretenimiento, capaz de llegar hasta los puestos de trabajo quitando tiempo y atención laboral a los empleados. Además, y para colmo de males, está relacionada con la falta de seguridad en los sistemas, por lo tanto, se la considera una puerta abierta para los hackers.[9] La mensajería gratuita tiene un nivel de privacidad muy bajo: el usuario nunca está seguro de quién puede estar leyendo

[9] Neologismo utilizado para referirse a un experto en computación y telecomunicaciones

los mensajes enviados por él. Esta clase de correspondencia viaja libremente por Internet y es ruteada a través de los servidores de las empresas que brindan el servicio. La inseguridad que provoca el sistema tal vez no pasaría a mayores si el navegante estuviera enviando comentarios triviales o personales a un amigo (aunque tampoco en este caso se justificaría la violación de la privacidad), pero ¿qué pasaría si el usuario estuviera utilizando el servicio para realizar negocios serios? La cuestión sería más complicada. Por eso la búsqueda de una solución que permita el uso de IM, pero que dé garantías también, es inevitable.

Otra de las desventajas en el uso de la MI gratuita puede ser que los usuarios se vuelvan demasiado dependientes de su uso, sin tomar en cuenta otras opciones más tradicionales como el teléfono o el simple "cara a cara".

Por otro lado, los programas de mensajería gratuitos (Yahoo, ICQ, etc) no se relacionan entre sí. Cada proveedor tiene sus propios seguidores. Por lo tanto, un factor para asegurar el éxito futuro de la mensajería instantánea dependerá también de que el usuario pueda interactuar entre los distintos servicios. Según la consultora Gartner la colaboración es un modelo de gestión primordial para las empresas y para aprovechar las soluciones de colaboración nada mejor que asumir las ventajas de IM, evitando los riesgos en la seguridad y en la gestión que caracterizan a los servicios gratuitos.

Siempre se debe tener presente, que si un medio permite la comunicación y manejo de información entre un grupo de individuos será más beneficioso que perjudicial. Lo importante es que todo la normativa exista y sea lo mas clara posible, así mismo, que la plataforma tecnológica detrás de los mensajes, sea lo más robusta posible para asegurar los estándares de calidad y disponibilidad.

No está mal que las compañías comiencen a apostar al desarrollo e implementación de este tipo de mensajería, ya que se han convertido en un

componente vital de las organizaciones y de las infraestructuras de comunicación, de hecho, este es el objetivo general de nuestro trabajo.

1.2.3. Mecanismos de Control. La Mensajería Instantánea Pública es una solución de comunicaciones económica para las empresas. Pero se deben seguir ciertas pautas para que no genere brechas de seguridad.

Debido a que NO todos los usuarios acataran las políticas de uso de la MI, es necesario adquirir, implementar o desarrollar una herramienta que monitoree los mensajes instantáneos, esto con el fin de evitar situaciones tediosas e impracticas en la manipulación de los equipos activos y/o la supervisión personal a los empleados. Después de todo, lo que se busca es incrementar considerablemente la productividad en las organizaciones.

El pensamiento general en muchas empresas y universidades con respecto a los servicios de la MI, es la perdida de tiempo de los empleados al usarlo en funciones diferentes a las asignadas para su cargo; debido a lo anterior, el control comúnmente utilizado por las organizaciones es el negar o bloquear el acceso a dichos servicios; ahora, estas situaciones y los riesgos en seguridad pueden ser mitigados, sin llegar al punto de prohibir o desechar esta herramienta de comunicación.

Existen en el mercado diversas alternativas a los programas de MI, éstas soluciones vienen listas para instalar, son de fácil administración y le proveen capacidades de autenticación (usted administra los usuarios que pueden conectarse), encriptación (la información no viaja como texto plano, es decir, no es posible que monitoreen sus conversaciones), compatibilidad con los sistemas públicos de MI para no eliminar del todo la conexión con el “mundo exterior”. Incluso algunos ofrecen integración con plataformas de colaboración; claro, todo

depende de su interés y de su presupuesto.

Es claro que existen en el mercado soluciones mucho mas avanzadas pero a precios bastante altos para pensar en acceder a ellas. Para las pequeñas y medianas empresas cuyo presupuesto para inversión en tecnología no es grande, una alternativa es el uso de herramientas de licencia abierta, ya que existen opciones que cuentan con las mismas características de seguridad de las más comerciales.

Esta opción requiere de más conocimiento, pues la solución puede no ser tan sencilla para administrar, aunque siempre cuenta con la alternativa de contratar el soporte de quien lo instala.

Algunos casos son programas como Jabber (http://www.jabber.org) del lado del servidor de mensajería, y clientes tan completos como gaim (http://gaim.sourceforge.net) o Pandion (http://www.pandion.be). Si no esta seguro de lo que más le conviene, asesórese de un buen desarrollador de soluciones.

Si definitivamente, la decisión es adquirir una solución de MI pública, asegúrese de mantener sus recursos de antivirus al día, definir políticas de uso de ese tipo de herramientas, generar acuerdos de confidencialidad de la información con sus usuarios, y lo mas importante, crear una cultura del buen uso de la Mensajería Instantánea.

1.2.4. Requisitos funcionales mínimos para un servicio de Mensajería Instantánea de clase empresarial. La aparición de la Mensajería Instantánea Corporativa promete una revolución en las comunicaciones internas de las organizaciones estatales y empresariales. A diferencia de las redes de Mensajería

Instantánea públicas que están orientadas más bien al entretenimiento, la Mensajería Instantánea Corporativa (MIC) debe adecuarse a estándares mucho más altos en varias áreas críticas.

ESTANDARES	DESCRIPCIÓN
Seguridad	Proporcionar los medios para el control, limitar el acceso y proteger comunicaciones vitales y el transporte de datos a través de la red.
Estabilidad	Funcionar perfectamente a pesar de las exigencias de los usuarios o el ambiente de la red en la que se encuentran.
Eficiencia	Utilizar la cantidad mínima de los recursos del sistema y de la red requeridos sin restar recursos importantes de otras aplicaciones de misiones críticas.
Amplitud de Características	Ofrecer un conjunto de características útiles que facilitan el alcance de los objetivos de las metas propuestas
Compatibilidad	No requerir de cambios a los sistemas existentes para poder implementarla.
Escalabilidad	Cumplir los requisitos de un ambiente corporativo siempre cambiante y con demandas crecientes.
Simplicidad	Facilidad de aprendizaje y uso para los usuarios.
Costo	Razonable: para implementar y mantener tanto inicialmente como a largo plazo.

Tabla 1. Estándares para la MIC

Por ello, es que los servicios de M.I. públicos y remotos (ej: AIM, MSN, Yahoo!), no son las mejores opciones como soluciones corporativas; lo anteriormente expuesto surge por la dificultad existente para cumplir adecuadamente con los requisitos de seguridad necesarios de las organizaciones o empresas.

1.2.5. Ventajas de un servicio de Mensajería Instantánea Empresarial en una organización. La adopción generalizada de sistemas seguros de Mensajería Instantánea Empresarial, en las instituciones estatales y organizaciones, es inevitable y está sucediendo en este mismo instante. A continuación brindamos algunos razonamientos con el fin de arrojar alguna luz sobre la importancia de esta herramienta:

- La comunicación en tiempo real y la transferencia de archivos con otras instituciones con las que se desarrollan proyectos conjuntos, con colegas, con proveedores y/o clientes, mejora las relaciones institucionales.
- Los costos asociados con llamadas de larga distancia, utilización del fax, gastos de viajes, envíos por correos privados, y adjuntos de email se reducen dramáticamente.
- Se elimina por completo el uso de la MI insegura (gratis), reduciendo violaciones de seguridad.
- Permite autorizar o prohibir a los empleados utilizar la MI dentro o fuera de la red, e impone el uso profesional de nombres / Nicks o Alias.
- Protege contra los fines de espionaje, robo de información y/o software; así mismo, deniega el ataque a los servicios, que son posibles de vulnerar, cuando los empleados de una organización utiliza los servicios de la MI pública.
- También mejora los servicios, las relaciones de asesoría o soporte que se puedan efectuar en beneficio de administrar la información intercambiada entre las organizaciones y sus sedes.

- Se incrementa el control sobre la comunicación de la información confidencial, dentro y fuera de las organizaciones, protegiendo de ésta manera los derechos Intelectuales de las empresas.
- Desarrolla la responsabilidad en los usuarios, por medio del mapeo de los nombres en pantalla, de los registros minucioso de las conversaciones y las transferencias de archivos, efectuadas con las identificaciones y/o permisos de uso dados por la corporación a sus empleados.

A pesar de su perfil de alto riesgo, la mensajería instantánea ofrece un importante valor empresarial puesto que transforma la forma de comunicación y colaboración de los empleados. Por esta razón, las organizaciones deben mantener seguros y disponibles en todo momento los sistemas de mensajería y los datos.

2. ASPECTOS FUNDAMENTALES DEL SERVICIO DE MENSAJERIA INSTANTANEA JABBER V.2.0

2.1. DESCRIPCIÓN GENERAL

Jabber, si bien es conocido popularmente como un servicio de Mensajería Instantánea, es en realidad una implementación del protocolo XMPP. En general, XMPP proporciona un marco de intercambio de mensajes, incluso para alertar sobre procesos del sistema y por supuesto para su aplicación más popular, la Mensajería Instantánea. No es el propósito de este trabajo entrar en detalles sobre las características de XMPP, pero si dar un entendimiento acerca del funcionamiento de Jabber, y su aplicación para el entorno empresarial.

Jabber como servicio de Mensajería Instantánea es básicamente una Aplicación de Arquitectura Cliente / Servidor que a su vez, utiliza de los servicios proporcionados por la pila de protocolos TCP/IP.

El Servidor principalmente es el responsable de encaminar mensajes entre otras entidades de Jabber, cuyo formato está en lenguaje XML, entre los participantes de una conversación. XML es un lenguaje muy similar al HTML, ya que también hace uso de etiquetas para la presentación de información, luego proporciona flexibilidad en su manejo, a la vez que puede ser fácilmente leída por humanos. Este lenguaje ofrece la posibilidad de dar estructura a los datos, que en este caso son los mensajes. Por su parte, el cliente simplemente actúa como un agente para el envío y recepción de estos mensajes, y a la vez de intérprete de estos.

Desde su concepción, Jabber tiene la propiedad de que su arquitectura es

distribuida, es decir, que el servicio puede implementarse en diferentes ubicaciones y todas ellas tienen la posibilidad de encaminar mensajes entre si. También fue diseñado para intentar dar una solución al problema de interoperabilidad de los servicios existentes de Mensajería Instantánea Pública. Para esto, en su núcleo se han definido Gateways o puertas de enlaces a otros servicios, principalmente MSN, Yahoo y AOL. Estas puertas de enlace se conocen alternativamente como Transportes, y permiten el intercambio de mensajes entre Clientes de Jabber y los de otros servicios.

El siguiente gráfico es una representación simplificada de la arquitectura del servicio Jabber, tal y como se especifica en [RFC3920][10].

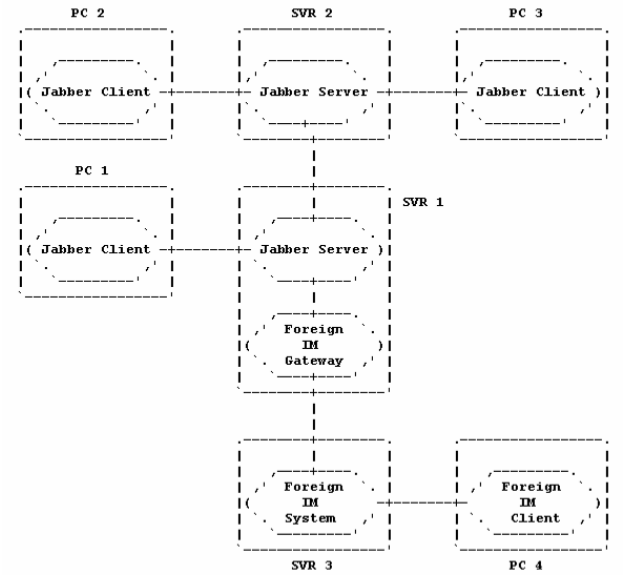

Figura 1. Arquitectura del Servicio Jabber

En resumen, en la arquitectura del servicio de Mensajería Instantánea Jabber pueden distinguirse tres componentes principales a saber:

[10] El RFC3920, Extensible Messaging and Presence Protocol (XMPP): Core, es la propuesta de especificacion para un protocolo de Mensajeria y Presencia

- Clientes de Jabber
- Servidores de Jabber
- Transportes a otros servicios de Mensajería Instantánea Publica.

En secciones posteriores se tratará acerca del funcionamiento y otros detalles de cada uno de los componentes de la arquitectura del servicio Jabber.

2.2. EL CONCEPTO DE IDENTIFICADOR DE JABBER (JID)

Un usuario de Jabber puede representarse como una entidad única a su vez caracterizada por un identificador de Jabber (Jabber ID, o simplemente JID); sin embargo, debe considerarse que no siempre un JID caracteriza a un usuario de Jabber, también puede identificar a una entidad perteneciente a un servicio. En general un JID tiene el formato <entidad>@<nombre-de-dominio-del-servidor>.

Por ejemplo, el JID javier.lozano@lelajels.gotdns.org intuitivamente podemos determinar que corresponde a un usuario que puede autenticar con el servidor lelajels.gotdns.org. Por otro lado, el JID helpdesk@conference.lelajels.gotdns.org representa a la sala con tema helpdesk -un sitio para recibir soporte- del servicio de charla múltiple o conferencia en el servidor lelajels.gotdns.org.

2.3. COMPONENTES DEL SERVIDOR JABBER.

En la versión 2.0 de Jabber, el componente servidor se ha dividido en subcomponentes o procesos, a diferencia del proceso monolítico del servidor, característico de las versiones de la familia 1.4. En otras palabras, en la última

versión de Jabber, no es un único demonio del sistema quien realiza las operaciones del servidor, sino que depende de cinco demonios, cada uno con sus respectivas tareas.

Según [ADMIN], los componentes que hacen parte del servidor Jabber son:

- Enrutador o Router
- Resolución de direcciones o Resolver
- Administrador de Sesión o Session Manager.
- Cliente a Servidor o Client to Server
- Servidor a Servidor o Server to Server.

Adicionalmente, tareas como la autenticación de usuarios y el almacenamiento de datos dependen de la interacción con paquetes externos. Sin embargo, el servicio requiere del uso de una forma de autenticación e interactuar con alguna base de datos para guardar información. La selección de qué paquete utilizar es opcional y recae en el instalador o administrador del servicio.

Cada proceso o componente del servidor, es responsable de diversas tareas que se enunciarán a continuación.

2.3.1. Enrutador o Router. El Enrutador se considera la espina dorsal del servidor Jabber, ya que es este quien acepta conexiones de tipo TCP/IP de las diferentes entidades Jabber y las direcciona a sus respectivos destinos. Sin embargo, estas labores de encaminamiento no dependen de si mismo; para cumplir con su objetivo, el Enrutador debe pasar los paquetes codificados en XML a los otros componentes del servidor Jabber.

Los demás componentes, que descritos a continuación interactúan

constantemente con el componente Enrutador. Es la responsabilidad del Enrutador la administración de estos componentes e implica controlar y supervisar como estos se conectan y el flujo de datos entre ellos.

2.3.2. Manejador de Sesión (Session Manager). El Manejador de Sesión es el proceso encargado de almacenar y disponer toda la información relacionada con un usuario. El SM[11] debe interactuar con una base de datos externa para guardar tal información. Esta base de datos comúnmente se refiere como el Paquete de Datos de la Aplicación. Entre otras funciones de la SM encontramos:

- Debe gestionar el almacenamiento de la información actual de presencia para un usuario local del servicio Jabber, es decir, debe almacenar el estado de su conexión (En Línea, Ocupado, No Conectado, etc.).
- Interactúa con el paquete de almacenamiento para guardar la información de Nómina o Roster; en otras palabras, permite el almacenamiento de la lista de contactos de un usuario del servicio local de Jabber.
- Es el encargado de negociar suscripciones. Una suscripción es el procedimiento mediante el cual, un usuario del servicio Jabber solicita pertenencia o acceso una entidad (puede ser un usuario o servicio); por ejemplo, cuando se agrega un usuario a la Nómina, el SM envía una solicitud de subscripción, de manera que el usuario agregado autorice al otro a ver su presencia. De igual manera, un usuario puede solicitar suscribirse con un servicio, por ejemplo, conferencia (chat multi-usuario).

El SM debe interactuar continuamente con el componente Cliente a Servidor, que se describe a continuación, para intercambiar mensajes.

[11] Siglas en ingles de Session Manager

2.3.3. Cliente a Servidor - C2S. Este componente es el encargado de manejar las conexiones entre el servidor Jabber y los clientes, es decir, transporta datos entre el servidor y los clientes; para esto, C2S utiliza el puerto 5222 de TCP. Entre sus funciones, también se encuentra el paso de mensajes relacionados con datos de los clientes al SM, para que disponga de ellos y los almacene.

También se encarga del proceso de autenticación de usuarios, es decir, de la administración de las entidades que pueden conectar con el servicio local Jabber. Eventualmente, C2S intercede en el proceso de registro de usuario en las implementaciones que permiten la creación de cuentas y el cambio de contraseñas desde el cliente. C2S debe interactuar con paquetes o módulos externos que proporcionan soporte al proceso de autenticación de usuarios, mecanismos que serán descritos posteriormente en este capitulo.

2.3.4. Servidor a Servidor - S2S. El componente Servidor a Servidor, como su nombre lo indica, es el responsable del mantenimiento de las conexiones entre servidores Jabber. S2S utiliza el puerto 5269 de TCP. Recordemos que la arquitectura del servicio Jabber es distribuida, por lo tanto, este mecanismo es necesario para la comunicación entre diferentes servidores Jabber.

S2S tiene la responsabilidad de verificar la identidad de los servidores externos que hacen peticiones de conexión al servicio por medio de un proceso denominado Marcado Inverso o Dialback, que consiste en el intercambio de una clave entre el servidor remoto y el local; el servidor local “devuelve el llamado” al remoto para confirmar la validez de la clave. De esta forma se reduce el riesgo de suplantación o spoofing. Cuando se establece una conexión entre servidores, S2S es el responsable del paso de paquetes de datos entre ellos.

2.4. SOLUCIONADOR O RESOLVER

El componente Solucionador o Resolver interactúa entre el componente S2S y el servicio DNS. Su labor es la de respaldar al componente S2S en la resolución de nombres de dominio. Para que el servicio Jabber sea accesible desde Internet, el servidor debe ser completamente resoluble. El Resolver es determinante en el proceso de Dialback. El resolver también es importante en un dominio local extenso, con una variedad de máquinas que corren diferentes servicios.

2.5. COMPONENTES AUXILIARES DEL SERVIDOR

2.5.1. Transportes. Según [RFC3920], un Transporte es un servicio de propósito especial asociado al servidor, cuya función principal es la de "traducir" el protocolo XMPP al protocolo utilizado por otro servicio de Mensajería Instantánea no-XMPP. Entre las implementaciones actualmente disponibles se encuentran los transportes a Correo Electrónico, Chat de Internet (IRC), y a los servicios de Mensajería Instantánea propietarios AIM, ICQ, MSN y Yahoo!.

La distribución de Jabber v.2.0 a diferencia de la anterior, si bien puede soportar Transportes, no incluye en el paquete ni librerías para proporcionar esta funcionalidad, ni binarios para ejecutarlos. Para proporcionar Transportes en Jabber v.2.0, es necesario hacer una instalación mínima de Jabber v.1.4, únicamente con los transportes, y vincularla al Enrutador; o crear los procesos individuales de los transportes, mediante un Wrapper.

Los Transportes en un servicio Jabber son especialmente útiles cuando los usuarios desean (o necesitan) interactuar con los de otros servicios de Mensajería Instantánea.

2.5.2. Directorio de Usuarios de Jabber. En implementaciones de Jabber con un número considerable de usuarios o entidades, es necesario garantizar un mecanismo de búsqueda para estos. El Directorio de Usuarios de Jabber (JUD) es un servicio que permite a los usuarios de Jabber, a través de sus clientes, buscar a otros de manera que al encontrarlos, pueda incluirlos a su Nómina y a su vez, solicitar ser incluido en la Nómina del usuario seleccionado. El servicio JUD básicamente hace consultas en lenguaje SQL[12] a una base de datos donde se almacena información relacionada con los usuarios. La pertenencia de un usuario al JUD no es obligatoria, esto quiere decir el ingreso de la información de éstos no recae solo en el administrador del servicio y los usuarios tienen la libertad de pertenecer o no al JUD.

Actualmente JUD para Jabber v.2.0 está soportado por el proyecto Users-Agent, el cual actúa como un proceso que se enlaza con el Enrutador y crea una base de datos en MySQL para almacenar la información.

2.5.3. Conferencia. Por Conferencia, en la implementación XMPP[13], se entiende la característica de intercambiar mensajes entre múltiples usuarios. Por defecto, el servicio Jabber por si solo permite el intercambio mensajes entre dos usuarios en una ventana de conversación. El componente de Conferencia, tiene un comportamiento similar al de Internet Relay Chat, ya que el servicio está basado en el acceso de los usuarios a salas de conversación.

A cada sala de conversación debe definírsele diferentes propiedades, por ejemplo, si la sala es pública o privada, definir el propietario, el moderador, los miembros, si la sala es permanente o por el contrario, temporal; el número de usuarios que pueden acceder a la sala, entre otros.

[12] Siglas en ingles de Structured Query Language (Lenguaje de Consulta Estructurado)

[13] Siglas en ingles de Extensible Messaging and Presence Protocol

El servicio actualmente permite que cualquier usuario pueda crear una sala con un cliente que soporte Conferencia y definir las propiedades para ésta. Para Jabber v.2.0, puede implementarse este servicio por medio del proyecto MU-Conference. El Componente debe construirse mediante un Wrapper e integrarse a Jabber mediante el Enrutador.

2.6. OTROS COMPONENTES

Existen otros componentes que añaden funcionalidad al servicio Jabber, pero en su mayoría, fueron escritos para Jabber v.1.4 y aun no cuentan con soporte nativo para la arquitectura de Jabber v.2.0. Estos servicios por ahora deben implementarse de la misma manera para los Transportes.

Puede observarse un listado completo de los componentes externos disponibles para Jabber, en el sitio web de la comunidad de desarrolladores

2.7. CARACTERÍSTICAS DEL CLIENTE.

Actualmente la mayoría de clientes de Jabber son compatibles con las arquitecturas existentes de la implementación XMPP, es decir que pueden operar tanto con los servidores basados en Jabber v.1.4 como con los de la versión 2.0. Existe una variedad de clientes de Jabber disponibles para instalarse en diferentes sistemas operativos, principalmente de las familias Unix y Windows. Incluso, entre uno y otro se diferencian por el lenguaje de programación sobre el que fueron construidos, por ejemplo C, Java, Perl, Python, Tcl, entre otros. Sin embargo, existen características comunes en todos, las cuales se enuncian a continuación:

- Nómina: El cliente debe mostrar la Nómina de usuarios y la presencia de cada uno de ellos.
- Presencia del usuario: El cliente debe ofrecer un mecanismo para que el usuario ajuste su información de presencia. La información de presencia generalmente muestra estados predefinidos, entre los cuales encontramos típicamente: En línea, Disponible para charlar, Ausente, No disponible y No conectado. Algunos clientes permiten introducir información de presencia personalizada, por ejemplo, añadir una descripción adicional a los estados existentes.
- Autenticación del usuario: El cliente debe ofrecer un espacio para que el usuario introduzca información sobre su identidad. Para autenticar con el servicio, el usuario debe proporcionar su JID, contraseña y un recurso. El recurso es un parámetro opcional, pero permite iniciar varias sesiones simultáneas utilizando diferentes clientes. Por ejemplo, un usuario con JID y recurso glopez@unisinu.edu.co/docencia, puede mantener esta sesión abierta. Cuando se desplaza a su residencia, podría abrir otra sesión desde un cliente diferente con el JID y recurso glopez@unisinu.edu.co/casa.
- Ventanas de conversación: Los clientes permiten la conversación entre dos usuarios por medio de una ventana de conversación diferente de la ventana principal del programa.
- Búsqueda de usuarios: El cliente incluye un mecanismo para la búsqueda de usuarios del servicio Jabber en el JUD, y realizar acciones con estas consultas, por ejemplo, incluir usuarios a la Nomina personal.
- Acceso a conferencia: Casi todos los clientes Jabber incluyen soporte para conversaciones entre múltiples usuarios, permiten la creación de salas en el servicio de conferencia y la edición de sus propiedades.
- Soporte de seguridad: El cliente debe soportar alguna implementación de seguridad para las conexiones con el servidor, por ejemplo mediante Transport Layer Sockets (TLS) o Secure Sockets Layer (SSL).

Entre los clientes más utilizados se encuentra Exodus para la familia de sistemas operativos Windows y Gaim, el cual es multiplataforma. Un listado completo de los clientes disponibles puede observarse en [CLIENTES]

2.8. MECANISMOS DE AUTENTICACIÓN

Para autorizar el acceso a los recursos del servicio Jabber, el proceso Cliente a Servidor necesita consultar un paquete de autenticación externo. Jabber v.2.0 típicamente utiliza bases de datos libres o servicios de directorios LDAP para obtener la información de autenticación de usuario, es decir, su nombre de usuario y contraseña.

Entre los métodos para la autorización del acceso, encontramos los siguientes:

2.8.1. Base de Datos. El servidor puede configurarse para que el proceso de autenticación recaiga sobre una base de datos. Actualmente, Jabber v.2.0 puede integrarse con MySQL, PostgreSQL, BerkeleyDB y Oracle. Cuando el servicio Jabber se compila desde el código fuente, este trae un script para la construcción de la base de datos, la cual lleva por nombre Jabberd2. Esta base de datos además de manejar la autenticación, almacena las variables del servidor.

En la tabla authreg se almacenan básicamente el nombre de usuario, la contraseña y el reino (realm), que corresponde al servidor local de Jabber. De acuerdo a [ADMIN], A continuación se ilustra los tipos de datos que maneja esta tabla:

2.8.2. Pluggable Autentication Modules - PAM. En español [PAM] significa Módulos Ajustables de Autenticación para Linux. Consiste en un conjunto de librerías compartidas, que permiten al administrador del sistema escoger como las aplicaciones autentican a los usuarios.

Para delegar la responsabilidad de la autenticación de usuarios a PAM, debe crearse un archivo de configuración para la aplicación, típicamente en la ruta /etc/pam.d. El mecanismo de autenticación depende del módulo que se especifique en el archivo de configuración. Por ejemplo, para que RADIUS sea la autoridad para la autenticación, en el archivo de configuración de PAM para Jabber debe especificarse entradas con el modulo pam_radius. También podrían usarse las cuentas de usuarios de Unix para la autenticación mediante el uso de PAM, entre otros mecanismos.

2.8.3. Lightweight Directory Access Protocol - LDAP. LDAP como sugiere su nombre en inglés, es un Protocolo Ligero para el Acceso de servicios de Directorios. Al igual que una base de datos, un directorio almacena diferentes tipos de información, sin embargo se diferencian porque este último está optimizado para los accesos de lectura y búsqueda. En un directorio, se especifican diferentes entradas, cada una con un tipo de dato y su valor respectivo. Una entrada típica puede ser el nombre de usuario, otra la cuenta de correo, otra para la contraseña, entre otras más.

Jabber v.2.0 puede obtener información para la autenticación asociándose con el servicio slapd de OpenLDAP. De esta manera, la autenticación de las cuentas de Jabber se realizará mediante consultas al servicio de directorios. LDAP proporciona un mecanismo de autenticación adecuado cuando se tiene una red local geográficamente distribuida y entre diversas plataformas.

2.9. MANEJO DE LOS DATOS POR EL SERVIDOR

Como se ha mencionado anteriormente, Jabber v.2.0 requiere de una base de datos proporcionada por un paquete externo para almacenar información del servicio, especialmente de las sesiones de los usuarios. Según [ADMIN], Jabber v.2.0 maneja los datos mediante agrupaciones que corresponden a un objeto denominado Colección. A cada Colección le corresponde una tabla en la base de datos, y a su vez cada tabla siempre tiene los atributos de Tipo y Propietario. El Tipo especifica que categoría de datos se maneja, por ejemplo ítems de una vCard o una Nomina. El Propietario indica quien posee la colección, por ejemplo para los datos relacionados con los usuarios, el Propietario será el JID. Además cada Colección mantiene uno o más objetos de datos, que corresponden a una tupla de la tabla, con su respectiva llave, valor y tipo. La llave especifica el tipo de dato, el valor es el dato mismo guardado y el tipo su naturaleza, por ejemplo, si es una cadena de caracteres, un numero o un booleano, etc.

A continuación, se mencionan las Colecciones de datos manejadas por Jabber v.2.0:

COLECCIÓN	DESCRIPCIÓN
Active	Almacena la fecha y hora en la que se activó la cuenta por primera vez.
disco-items	Guarda información de descubrimiento persistente de manera que esté disponible cuando se requiere en estado desconectado.
logout	Almacena el JID y la hora de desconexión más reciente de los usuarios.

motd-mesages	Guarda los Mensajes del Día en formato XML.
motd-times	Almacena el JID y el tiempo en que un usuario recibe un MOTD.
privacy-defaults	Guarda el nombre de la lista usada actualmente por un usuario, de manera que esté disponible cuando entre En Línea.
privacy-items	Almacena las listas de privacidad de los usuarios (listas de usuarios denegados y listas de usuarios permitidos).
private	Guarda las preferencias de usuario en formato XML.
Queue	Almacena mensajes XML puestos en cola.
Roster-groups	Guarda los miembros de una Nómina para un JID, únicamente si pertenecen a un grupo.
Roster-items	Guarda los miembros de una Nómina, incluyendo su estado de autorización.
vacation-settings	Almacena un mensaje de ausencia personalizado, incluyendo el tiempo de inicio y el final de la ausencia.
Vcard	Guarda la información relacionada con los datos personales del usuario.
authreg	Almacena la lista de usuarios que autentican en el servicio y sus respectivas contraseñas, cuando la autenticación se delega a la base de datos.

Tabla 2. Colección de Datos de Jabber v2.0.

La siguiente figura es una lista mas detallada de cada uno de los atributos pertenecientes a las Colecciones:

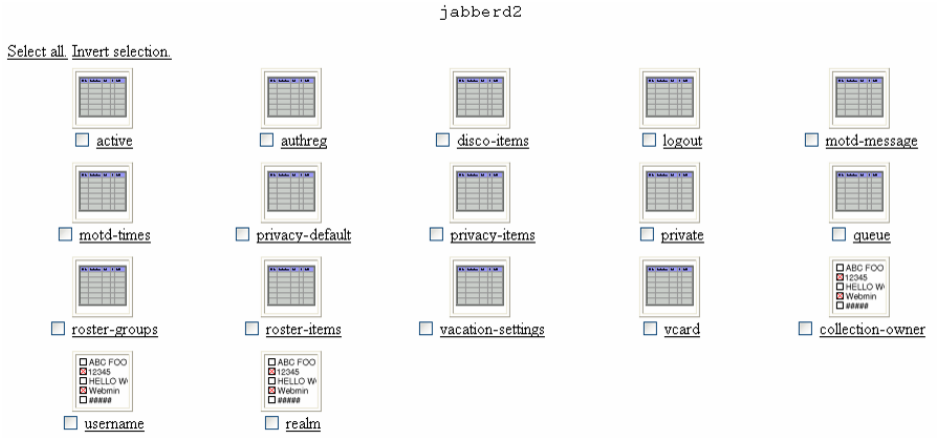

Figura 2. Objetos de la base de datos

2.10. MECANISMOS DE SEGURIDAD

Por defecto, todas las conexiones entre los diferentes componentes de Jabber v.2.0, por ejemplo entre el Enrutador y los procesos, y entre los procesos y los clientes u otros servidores, son paquetes de datos de texto plano en formato XML. Cuando el servicio se utiliza en una red local de pocos usuarios, por ejemplo menos de 20, hacer estas transacciones de datos quizás sea aceptable. Pero cuando el número de usuarios es mucho mayor o las conexiones se originan en cualquier punto de Internet, debe considerarse el uso de un mecanismo de seguridad para proteger los datos. Esta premisa también es válida si mediante el servicio Jabber se intercambian mensajes con información de misión crítica. A continuación se enuncias los mecanismos de seguridad que pueden implementarse en Jabber v.2.0.

2.10.1. Secure Socket Layer - SSL. Debe considerarse que las conexiones entre Cliente y Servidor son transacciones XML, y este lenguaje emplea caracteres ASCII para la construcción de los mensajes. Por lo tanto, al efectuar una captura

de paquetes de datos Jabber, puede observarse su contenido y pueden ser claramente interpretados. C2S también proporciona una forma de codificar los mensajes por medio de conexiones encriptadas en el puerto TCP 5223 (Desde la versión Jabber 1.4) o mediante la encriptación de los mensajes mediante el uso de certificados SSL.

Un certificado [SSL] proporciona un algoritmo de encriptación simétrico como RC4 o IDEA, para la encriptación de los datos que se intercambian entre el cliente y el servidor; además encripta la clave de sesión del algoritmo, utilizando un algoritmo de cifrado de llave publica como RSA. La clave de sesión es distinta para cada transacción, de manera que si un atacante logra revelar una, no le será útil para posteriores sesiones.

Pueden utilizarse certificados SSL para las conexiones entre Cliente y Servidor, Servidor a Servidor e incluso entre el Enrutador y los procesos, modificando algunos parámetros en los archivos de configuración.

2.10.2. Virtual Private Network - VPNs. Otro mecanismo de seguridad que puede utilizarse para proteger las conexiones en Jabber v.2.0, es el uso de Redes Privadas Virtuales o VPNs, del Inglés. Una [VPN] es una conexión entre varios nodos remotos que intercambian datos de manera segura uno a otro sobre una infraestructura de red pública e insegura, por ejemplo Internet. En una VPN, los datos se protegen mediante algún nivel de encriptación; típicamente las VPNs se crean entre dos redes remotas (también llamadas sitio a sitio) o un cliente remoto y una red local (cliente a sitio).

Las VPNs no solamente son útiles para proteger el tráfico de Jabber, sino que permiten crear toda una infraestructura de red segura y de bajo costo para organizaciones geográficamente distribuidas, utilizando solamente accesos a

Internet. También permite la conexión segura de Teletrabajadores (usuarios remotos que necesitan acceder a recursos de la red local) desde sus dispositivos cliente.

2.11. VENTAJAS DEL USO DE JABBER V.2.0

Jabber v.2.0 posee diferentes características que lo hacen muy atractivo a la hora de implementar un servicio de Mensajería Instantánea, sin importar si es para el uso público o para un servicio a nivel interno de una organización. Algunas de sus ventajas representativas respecto a otros servicios de Mensajería Instantánea existentes son:

- Es software con licencia GNU GPL. Significa que su uso es libre y de igual forma su distribución. Cualquiera puede copiar el producto e incluso modificarlo y darle valor agregado con fines de comercialización. No hay números de licencia, ni alguna otra restricción para su implementación. Sin embargo, se recomienda de alguna forma contribuir a su desarrollo, por ejemplo donando dinero al equipo de programadores, haciendo modificaciones propias y divulgarlas a la comunidad en Internet, o simplemente promover su uso.
- Flexibilidad en la Implementación. Jabber v.2.0 posee una arquitectura modular. El administrador del sistema puede determinar que componentes instalar y cuales no. Por ejemplo, si se desea establecer un servicio público, probablemente se quiera instalar Transportes a otros servicios de Mensajería Instantánea. Si el servicio es privado, muchas características no son necesarias.
- Fácil configuración del servicio. El servicio Jabber es configurable mediante un conjunto de archivos en lenguaje XML, un archivo por cada proceso. Estos residen típicamente en la ruta /etc/jabber en los sistemas Unix. Para

modificar los parámetros, basta con descomentar (o comentar) algunas líneas de texto. Los archivos contienen comentarios que explican la acción que ejecuta un parámetro.

- Amplio soporte. Existen diversos recursos en Internet destinados a la solución de problemas relacionados con el servicio, por ejemplo foros y listas de correo. En estos recursos muchos administradores del servicio comparten su experiencia en la configuración del servicio y en la resolución de problemas.
- Es un protocolo ligero. Jabber fue concebido en esencia para el intercambio de mensajes de texto. Entonces, el tamaño de los mensajes es muy pequeño y el consumo de ancho de banda total para un gran número de usuarios simultáneos es relativamente bajo, lo que lo hace ideal en redes donde el manejo del ancho de banda es crítico.
- Es un protocolo seguro. Gracias al uso de SSL, puede garantizarse confidencialidad de datos en las conversaciones.

3. IMPLEMENTACION SUGERIDA DE JABBER V.2.0 EN UN ENTORNO EMPRESARIAL

En páginas anteriores se trataron las principales características del servicio Jabber v.2.0, para dar un mejor entendimiento de su funcionalidad. Estas no solamente hacen de él un servicio público de mensajería alternativo a los más populares; también puede adaptarse para proporcionar un servicio privado, por ejemplo un servicio de Mensajería Instantánea para una organización. Este capitulo pretende ilustrar el procedimiento para la implementación de un servicio de Mensajería Instantánea para un escenario propuesto típico de una empresa.

3.1. DESCRIPCIÓN DEL ESCENARIO PROPUESTO

Consideremos una compañía con un número de empleados no superior a 500 y cuenta con sedes en diferentes ubicaciones geográficas y “teletrabajadores”, es decir, personal que permanece por fuera de las instalaciones de la organización. Básicamente, la organización desea establecer una herramienta de comunicación que permita determinar la presencia de sus empleados y el contacto en tiempo real entre el personal. La plataforma que soporta al servicio estaría ubicada en la sede principal, y la conectividad con las otras sedes se proporciona mediante redes privadas virtuales. Ninguna persona ajena a la compañía debe tener acceso al servicio, aunque se debe permitir que los empleados puedan acceder a él desde una ubicación remota. Se prevé que por medio del servicio se intercambie información estrictamente de carácter laboral y eventualmente información confidencial, por ejemplo contraseñas o información bancaria. El administrador del servicio desea asegurarse que los empleados den un uso adecuado y racional al

servicio. El servicio debe permitir la realización de conversaciones múltiples, por ejemplo para realizar juntas con personal en otras ciudades o directivos que se encuentran de viaje. También es deseable que los usuarios que necesiten comunicarse con otros diferentes a los habituales, tengan una manera de buscarlos para posteriormente contactarlos. Se ha determinado previamente que es una prioridad mantener un nivel de baja utilización del ancho de banda, luego no es necesario el uso de conversaciones de voz o video.

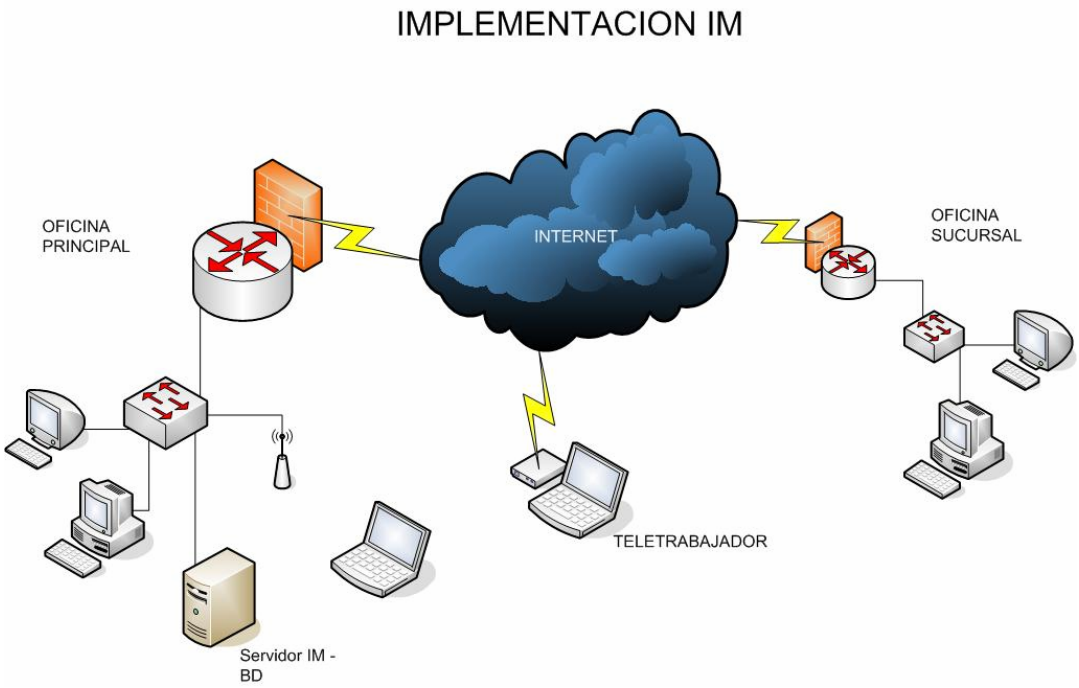

Figura 3. Escenario Empresarial Propuesto.

3.2. RECURSOS

La compañía tiene varios servidores de red en plataformas Unix, entre ellos dispone de un servidor de correo electrónico, con un número de cuentas igual al de empleados. Este servidor tiene sistema operativo SUSE Linux, el cual posee

como intercambiador de correo a Postfix, luego las cuentas de correo corresponden a cuentas de usuarios regulares del sistema. Adicionalmente este sistema operativo incluye entre sus paquetes de instalación la base de datos libre MySQL. En cuanto a las estaciones de trabajo y Pcs de escritorio, todos tienen sistemas operativos de la familia Windows.

3.3. CONSIDERACIONES EN JABBER V.2.0 PARA LA IMPLEMENTACIÓN COMO SERVICIO DE CLASE EMPRESARIAL

Teniendo en cuenta el escenario anteriormente propuesto y los recursos disponibles, debemos hacer algunas consideraciones para proporcionar una solución a estos requerimientos utilizando el servicio de Mensajería Instantánea mediante Jabber v.2.0., como son:

- Configurar el Cortafuegos o Enrutador que separa a la red de área local de Internet para que permita conexiones de clientes desde el exterior; para esto deben permitirse conexiones por el puerto TCP 5222. Por otro lado, el servidor Jabber no debe comunicarse con otros servidores en el exterior, por tratarse de un servicio corporativo, de manera que se deben denegar las conexiones Servidor a Servidor bloqueando las conexiones por el puerto TCP 5269.
- Se recomienda ubicar el servidor al interior de la red de área local. Para no afectar los servicios de correo y mensajería y puedan ser vistos desde el exterior, en lo posible debe configurarse un redireccionamiento de puertos a la dirección de red interna donde se encuentre el servidor.
- Configurar el servicio de resolución de nombres de dominio para que los clientes puedan determinar mediante el nombre de máquina la dirección del servicio de mensajería. Si múltiples servicios residen en una maquina, es útil definir alias de nombres para diferenciar los servicios.

- Utilizar las cuentas de usuarios previamente creadas para el servicio de correo para autenticar con el servicio Jabber. Esto elimina la necesidad de tener que introducir manualmente los usuarios a una base de datos o que los usuarios registren cuentas con la base de datos la primera vez que accedan al servicio. Además se debe deshabilitar el cambio de contraseñas por parte de los usuarios.
- Debemos asegurarnos que las conexiones entre Clientes y el Servidor se realicen de forma segura, por lo que será necesario darle al servicio soporte de encriptación mediante certificados SSL.
- No implementar Transportes a otros servicios de Mensajería Instantánea, ya que las comunicaciones a otros servidores serán completamente deshabilitadas para mantener el servicio privado. Sin embargo es útil dar soporte a charlas multiusuario (Conferencia) y al directorio local de usuarios de Jabber (JUD).
- Llevar un registro o Log de las conversaciones que sostienen los usuarios, como medida de control, para tener pruebas que permitan tomar correctivos en caso de que un usuario utilice inadecuadamente el servicio.
- Utilizar un cliente básico o ligero, que se ajuste a las máquinas menos potentes y de manera que no tenga elementos que puede convertirse en un distractor, como por ejemplo, emoticons. Exodus es una buena elección para los clientes basados en Windows.

3.4. AJUSTES PRELIMINARES

Para realizar una instalación de Jabber desde el código fuente, es necesario hacer algunos preparativos al sistema, para evitar posibles problemas post-instalación y asegurar que su funcionamiento sea correcto.

3.4.1. Creación de usuario y grupo del sistema para Jabber. Por razones de seguridad, se debe hacer un usuario y grupo para correr el servidor Jabber 2.0. No se recomienda correr el servicio con privilegios de superusuario. Para esto, usando la línea de comandos y con una sesión del usuario root, se creará un usuario Jabber vinculado a un grupo Jabber:

```
linux:~# groupadd jabber
linux:~# useradd -g jabber jabber
```

3.4.2. Creación de rutas para almacenar los archivos PID y Logs del servicio. El servicio necesita de los archivos PID para guardar en ellos el identificador del proceso en memoria del sistema, el cual es típicamente un número, el cual es único para un proceso determinado. También es útil guardar los Logs del sistema, ya que en ellos se registran diferentes acciones que ejecuta el servicio, así como alertas útiles a la hora de resolver un problema. Para guardar estos archivos se recomienda crear directorios, una en el directorio de PIDs, típicamente /var/run; y otra en el de Logs en /var/log. A los directorios creados se sugiere definirles un propietario, en nuestro caso el usuario Jabber previamente creado.

Para el directorio de PIDs usaríamos la siguiente secuencia de comandos:

```
linux:/# cd var
linux:/var# cd run
linux:/var/run# mkdir jabber
linux:/var/run# chown -R jabber:jabber jabber/
```

De manera similar procedemos con la creación del directorio para los Logs.

```
linux:/# cd var
linux:/var# cd log
linux:/var/log# mkdir jabber
linux:/var/log# chown -R jabber:jabber jabber/
```

3.5. PROCEDIMIENTO DE INSTALACIÓN DEL SERVIDOR

Una vez realizados los ajustes preliminares sugeridos, podemos proceder con la instalación del servidor en la ubicación ya determinada.

3.5.1. Configuración de variables de tiempo de compilación. Siempre que se realice una instalación desde el código fuente es necesario ajustar algunas variables de entorno para integrar el servidor con diferentes parámetros del sistema.

Se recomienda colocar el código fuente en un directorio temporal. En nuestro caso, vamos a utilizar el directorio /opt/instaladores. En esta ubicación, descomprimimos el código fuente y procedemos a configurar variables:

```
linux:/var/log# cd /opt/instaladores
linux:/opt/instaladores# ls
jabberd-2.0s11.tar.gz

linux:/opt/instaladores# tar -zxvf jabberd-2.0s11.tar.gz

linux:/opt/instaladores# ls
jabberd-2.0s11  jabberd-2.0s11.tar.gz

linux:/opt/instaladores# cd jabberd-2.0s11
linux:/opt/instaladores/jabberd-2.0s11# ls
acinclude.m4  config.rpath  Doxyfile.in  Makefile.in  README.win32  sx
aclocal.m4    config.sub    etc         man          resolver      TODO
AUTHORS       configure     expat       mio          router        tools
c2s           configure.in  INSTALL     missing      s2s           util
ChangeLog     contrib       install-sh  NEWS         scod
config.guess  COPYING       ltmain.sh   PROTOCOL     sm
config.h.in   depcomp       Makefile.am README       subst
```

Una vez hecho esto, observamos la salida de ayuda del comando configure para asegurarnos de las opciones de compilación que debemos ajustar. La salida del

comando es algo extensa, por lo tanto se incluyen solo los parámetros más importantes:

```
linux:/opt/instaladores/jabberd-2.0s11# ./configure --help
'configure' configures jabberd 2.0s11 to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...
....

Fine tuning of the installation directories:
  --bindir=DIR           user executables [EPREFIX/bin]
  --sbindir=DIR          system admin executables [EPREFIX/sbin]
  --libexecdir=DIR       program executables [EPREFIX/libexec]
  --datadir=DIR          read-only architecture-independent data [PREFIX/share]
  --sysconfdir=DIR       read-only single-machine data [PREFIX/etc]
  --sharedstatedir=DIR   modifiable architecture-independent data [PREFIX/com]
  --localstatedir=DIR    modifiable single-machine data [PREFIX/var]
  --libdir=DIR           object code libraries [EPREFIX/lib]
  --includedir=DIR       C header files [PREFIX/include]
  --oldincludedir=DIR    C header files for non-gcc [/usr/include]
  --infodir=DIR          info documentation [PREFIX/info]
  --mandir=DIR           man documentation [PREFIX/man]
...
  --enable-idn           enable IDN support (yes)
  --enable-ssl           enable SSL/TLS support (yes)
  --enable-mysql         enable MySQL auth/reg/storage support (yes)
  --enable-pgsql         enable PostgreSQL auth/reg/storage support (no)
  --enable-sqlite        enable SQLite 3 storage support (no)
  --enable-db            enable Berkeley DB auth/reg/storage support (no)
  --enable-ldap          enable OpenLDAP auth/reg support (no)
```

Es importante ajustar los directorios de instalación a las rutas de la distribución de Linux usada, también las opciones del sistema que se utilizarán; por ejemplo, se utilizará como paquete de base de datos MySQL, luego debe habilitarse su correspondiente opción. La seguridad de las comunicaciones entre los clientes y el servidor se hará mediante SSL, luego es necesario especificar soporte para esta característica. La entrada del comando de configuración de las opciones sería la siguiente:

```
linux:/opt/instaladores/jabberd-2.0s11# ./configure --enable-idn |\
> --enable-ssl --enable-mysql --bindir=/usr/bin --sbindir=/usr/sbin  |\
> --libexecdir=/usr/libexec --datadir=/usr/share --sysconfdir=/etc |\
> --localstatedir=/var --libdir=/usr/lib --includedir=/usr/include |\
> --oldincludedir=/usr/include --infodir=/usr/share/info
```

Si se cumple con todas las dependencias, el proceso de ajuste de las opciones de tiempo de compilación no debería retornar errores y podemos proceder a compilar. Si por el contrario se presentan errores, deben resolverse estas dependencias. Verifique que los paquetes especificados (idn, MySQL, openSSL) se encuentren instalados y sus rutas sean las típicas, de lo contrario también será necesario especificar las rutas de los paquetes. El como resolver estas dependencias queda por fuera de del alcance de este procedimiento sugerido.

3.5.2. Compilación e Instalación. Una vez configurada la instalación procedemos a compilar. Estando ubicados en el directorio del código fuente, entramos el siguiente comando:

```
linux:/opt/instaladores/jabberd-2.0s11# make
```

Si la compilación no ha retornado errores, esta se ha realizado con éxito. En caso contrario, es posible que falte alguna librería de C en el sistema y deba instalarse. Nuevamente, resolver estas dependencias queda por fuera del propósito de este documento.

Una vez compilado el código, procedemos a instalar mediante la siguiente secuencia de comandos:

```
linux:/opt/instaladores/jabberd-2.0s11# make install
```

Si la compilación del sistema fue exitosa, es poco probable recibir errores en el

tiempo de instalación. Para verificar la ruta de los ejecutables del servicio, comprobamos buscando el ejecutable de Jabber 2.0:

```
linux:/opt/instaladores/jabberd-2.0s11# which jabberd
/usr/bin/jabberd
```

El servidor se ha instalado con éxito. Sin embargo, es necesario crear una base de datos para almacenar la información del servidor, procedimiento que se describe a continuación

3.6. CREACIÓN DE LA BASE DE DATOS PARA ALMACENAMIENTO

Afortunadamente, en el código fuente de Jabber v.2.0, existe un script de SQL que permite la creación de la base de datos y todas las Colecciones que necesita el servicio. De lo contrario, tendríamos que construir la base de datos y los objetos de forma manual.

Para crear la base de datos en MySQL, se debe acceder a la línea de comandos bajo una sesión de superusuario. Entramos la siguiente secuencia de comandos en la consola de administración de MySQL:

```
linux:/ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 4.1.13

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> \. db-setup.mysql
```

Puede observarse como salida de esta secuencia de comandos los objetos

creados de esta base de datos. Es recomendable crear un usuario con privilegios para administrar esta base de datos. En nuestro caso, hemos creado previamente un usuario de la base de datos llamado Jabberd2, de manera que utilizaremos el usuario Jabber para administrar la base de datos. Debemos otorgar permisos al usuario Jabberd2:

```
mysql> GRANT select,insert,delete,update ON jabberd2.* to jabberd2@localhost IDENTIFIED by 'password-aqui';
```

Para comprobar si la base de datos se ha creado, podemos usar el comando mysqlshow; con este comando también podemos observar las Colecciones que pertenecen a la base.

```
linux:/ # mysqlshow
Databases
jabberd2
mysql
test
tmp

linux:/ # mysqlshow jabberd2
Database: jabberd2
Tables
active
authreg
disco-items
logout
...
```

En este punto, la base de datos para Jabber v.2.0 está lista para guardar datos.

3.7. INSTALACIÓN DE COMPONENTES EXTERNOS AL SERVIDOR

Para proporcionar la funcionalidad de charlas multiusuario, procederemos a

integrar el servicio MU-Conference con nuestra instalación de Jabber v.2.0. De igual manera, para permitir búsquedas de usuarios en el servicio de Mensajería Instantánea local, mediante la provisión del servicio de directorio de usuarios Jabber Users – Agent. En las siguientes secciones se sugiere un procedimiento de instalación para estos servicios.

3.7.1. Aprovisionamiento del servicio de Conferencia con MU-Conference. Debido a la arquitectura modular de Jabber v.2.0, MU-Conference debe construirse desde el código fuente, pero de manera que este se ejecute como un proceso independiente en el sistema. MU-Conference por defecto, debe construirse contra el código fuente de Jabber v.1.4, luego para integrarlo con Jabber v.2.0 es necesario compilarlo contra un Wrapper, llamado Jabber Component Runtime o JCR. Este Wrapper permite generar un binario para MU-Conference, de manera que se ejecute como un proceso independiente del de Jabber.

De manera similar al procedimiento de instalación de Jabber v.2.0, colocamos los paquetes de código fuente en un directorio temporal, en nuestro caso usamos la ruta /opt/instaladores. En este directorio, ubicamos los paquetes jcr-0.2.4.tar.gz y mu-conference-0.6.0.tar.gz

```
linux:/# cd /opt/instaladores/
linux:/opt/instaladores# ls
mu-conference-0.6.0.tar.gz      jcr-0.2.4.tar.gz
...
```

Procedemos a descomprimir los paquetes. En el directorio de instaladores deben observarse los directorios con el código fuente, tanto del JCR como del MU-Conference:

```
linux:/opt/instaladores# tar -zxvf jcr-0.2.4.tar.gz
linux:/opt/instaladores# tar -zxvf mu-conference-0.6.0.tar.gz
```

En primer lugar, debemos entrar al directorio con el código fuente de JCR. No es necesario configurar opciones de instalación. Procedemos directamente a compilar las librerías de JCR mediante el comando make.

```
linux:/opt/instaladores# cd jcr-0.2.4
linux:/opt/instaladores/jcr-0.2.4# make
```

Ahora debemos mover el código fuente de MU-Conference adentro del directorio con las librerías creadas de JCR, para posteriormente compilar.

```
linux:/opt/instaladores# mv  mu-conference-0.6.0/ jcr-0.2.4/
linux:/opt/instaladores# cd jcr-0.2.4/
linux:/opt/instaladores/jcr-0.2.4# ls
CONFIGURATION  INSTALL  jcomp  LICENSE   mu-conference-0.6.0  src
FUNCTIONS      jabberd  lib    Makefile  README
```

Procedemos a copiar los archivos jcr-0.2.4/src/jcomp.mk y jcr-0.2.4/src/main.c en el directorio jcr-0.2.4/mu-conference-0.6.0/src/

```
linux:/opt/instaladores/jcr-0.2.4/src# cp jcomp.mk ../mu-conference-0.6.0/src/
linux:/opt/instaladores/jcr-0.2.4/src# cp main.c ../mu-conference-0.6.0/src/
```

Una vez copiados estos archivos, nos desplazamos al directorio jcr-0.2.4/mu-conference-0.6.0/src/ y compilamos con la secuencia de comandos make -f jcomp.mk

```
linux:/opt/instaladores/jcr-0.2.4# cd mu-conference-0.6.0/src
linux:/opt/instaladores/jcr-0.2.4/mu-conference-0.6.0/src# make -f jcomp.mk
```

Como resultado de la compilación, debe aparecer el archivo binario (entiéndase, ejecutable) de MU-Conference:

```
linux:/opt/instaladores/jcr-0.2.4/mu-conference-0.6.0/src# ls
admin.c       conference_room.c  hash.c    main.o        roles.o  xdata.o
admin.o       conference_room.o  hash.o    Makefile      utils.c  xdb.c
conference.c  conference_user.c  jcomp.mk  mu-conference  utils.o  xdb.o
conference.o  conference_user.o  main.c    roles.c       xdata.c
```

Este binario finalmente podemos copiarlo en el directorio donde se encuentran los ejecutables de Jabber v.2.0, en nuestro caso la ruta /usr/bin. Solo resta configurar el servicio para que interactúe con Jabber, procedimiento que se introducirá posteriormente en este capitulo.

3.7.2. Aprovisionamiento del servicio JUD mediante Users - Agent. Users-Agent es el componente de JUD mas utilizado en la implementación Jabber v.2.0, debido a que es un proceso independiente que se vincula al servicio por medio del componente Enrutador. A continuación se describe su procedimiento de instalación.

Antes de realizar la instalación, debemos comprobar que los siguientes módulos de PERL[14] en sus últimas versiones, se encuentren instalados en el sistema:

```
Net::XMPP      Net::Jabber      DBI      XML::Stream
```

Si estos módulos no se encuentran instalados, pueden típicamente pueden instalarse desde los CDs del sistema operativo o mediante el servicio CPAN[15].

Es conveniente descomprimir el paquete de instalación de users-agent en una

[14] PERL es el acrónimo para Practical Extraction and Report Language, un popular lenguaje de programación para plataformas Unix.

[15] CPAN es el acrónimo para Comprehensive Perl Archive Network, un servicio para la distribución de modulos PERL.

ubicación temporal, en nuestro caso /opt/instaladores:

```
linux:~ # cd /opt/instaladores/
linux:/opt/instaladores # tar -zxvf Users-Agent-1.2.tar.gz
```

Una vez ejecutada esta acción, accedemos a la carpeta creada al descomprimir el paquete. Es recomendable copiar el ejecutable del componente en el directorio /usr/bin, así como el archivo de configuración config.xml en la ubicación de los archivos de configuración de Jabber v.2.0, en /etc/jabberd.

```
linux:/opt/instaladores/ #cd  users-agent-1.2
linux:/opt/instaladores/users-agent-1.2 # ls
.  ..  CHANGES  INSTALL  README  boot.sh  config.xml createDB  users-agent

linux:/opt/instaladores/users-agent-1.2 # cp users-agent /usr/bin
linux:/opt/instaladores/users-agent-1.2 # cp config.xml /etc/jabberd
```

Finalmente, procedemos a crear la base de datos en MySQL, que requiere Users-Agent para almacenar la información de los usuarios. Para esto, el paquete de instalación dispone de un script de shell de linux, createDB. Sin embargo, antes de ejecutar el script, se debe editar algunas líneas para introducir la contraseña del usuario root de la base de datos.

```
my $DBHandle = DBI->connect("$dbs[0]","root","password-aqui ");
...
my $DBHandle = DBI->connect("DBI:mysql:database=JUD","root","password-aqui");
```

Adicionalmente, algunas versiones de MySQL no soportan el atributo INDEX, por lo que se recomienda suprimirlos de la siguiente función:

```
my $sth = $DBHandle->prepare("CREATE TABLE jud (jid VARCHAR(100), name VARCHAR(100) INDEX, first VARCHAR(50) INDEX, last VARCHAR(50) INDEX, nick VARCHAR(50) INDEX, email VARCHAR(50) INDEX);");
```

Una vez realizados estos cambios al script, lo podemos ejecutar para crear la base

de datos, de la siguiente manera:

```
linux:/opt/instaladores/users-agent-1.2 # ./createDB
```

El servicio JUD en este punto está completamente instalado, sin embargo debe configurarse para que se integre con el Enrutador de Jabber v.2.0.

3.8. CONFIGURACIÓN DE LOS SERVICIOS DE JABBERD V.2.0

Para que el servicio Jabber v.2.0 funcione de acuerdo con nuestras necesidades, deben modificarse las opciones que ofrecen los archivos de configuración. Cada proceso relacionado con Jabber v.2.0 (Enrutador, Resolver, SM, C2S y S2S) tiene su propio archivo de configuración, el cual es un archivo de texto plano en lenguaje XML. En nuestro caso, hemos especificado que los archivos de configuración estén ubicados en la ruta /etc/jabberd. A continuación se observa un listado de los archivos de configuración:

```
linux:/etc/jabberd # ls
.             jabberd.cfg.dist   router-users.xml.dist  server.pem
..            muc-jcr.xml        router.xml             sm.xml
c2s.xml       resolver.xml       router.xml.dist        sm.xml.dist
c2s.xml.dist  resolver.xml.dist  s2s.xml                templates
jabberd.cfg   router-users.xml   s2s.xml.dist
```

La instalación de Jabber v.2.0 proporciona copias de seguridad de los archivos de configuración (aquellos denotados por la extensión .dist). Cuando se hacen modificaciones a los archivos de configuración se sugiere copiar la configuración actual a otro archivo (por ejemplo, finalizando con extensión .bk) de manera que pueda recuperarse la configuración en caso de cometer un error. Los archivos de extensión .dist son útiles cuando no hay otra alternativa más que recuperar la configuración inicial.

Para realizar la configuración del servicio es conveniente definir tareas de administración, de tal forma que se facilite la comprensión de los efectos sobre la funcionalidad del servicio al modificar un parámetro; y también para hacer este procedimiento de manera organizada.

3.8.1. Ajustar nombre de máquina. De esta manera se identificará el servicio en la red. El nombre de máquina debe ser completo, es decir, acompañado del dominio. Este debe ser resuelto por el servicio DNS[16], especialmente cuando se espera recibir conexiones desde Internet.

En sm. xml, debemos editar la sección <id>, por ejemplo:

```
<sm>
  <id>lelajels.gotdns.org</id>
...
```

En c2s.xml, también debemos editar la opción <id>, pero en la sección <local>:

```
 <!-- Local network configuration -->
  <local>
    <id>lelajels.gotdns.org</id>
...
```

3.8.2. Especificar la ruta de los archivos PID. Anteriormente, en los preparativos para la instalación, se crearon los directorios para guardar los archivos PID. Jabber necesita conocer donde guardar estos archivos, por lo que necesitamos especificar en los archivos de configuración la ruta a estos directorios.
Cada proceso tiene su archivo PID.

[16] Siglas en ingles de Domain Name System, en español Sistema de Nombres de Dominio

Para sm. xml:

```
<pidfile>/var/run/jabber/sm.pid</pidfile>
...
```

Para c2s.xml:

```
<pidfile>/var/run/jabber/c2s.pid</pidfile>
...
```

Para router, xml:

```
<pidfile>/var/run/jabber/router.pid</pidfile>
...
```

De igual manera pueden configurarse para s2s.xml y resolver.xml

3.8.3. Autenticación mediante PAM. Puesto que en el sistema ya existen cuentas de usuarios creadas, utilizadas por el servicio de correo electrónico, no es necesario introducir los usuarios a la base de datos de Jabber v.2.0 para la autenticación. En su lugar, le especificamos al servicio que utilice PAM para la comprobación de los usuarios. En el archivo c2s.xml introducimos los siguientes cambios:

```
<!-- Authentication/registration database configuration -->
  <authreg>
    <!-- Backend module to use -->
    <module>pam</module>
...
```

Ahora, debemos configurar PAM. Se debe crear un archivo de configuración para Jabber v.2.0. En SUSE Linux, la ruta de configuración de PAM es /etc/pam.d. Con un editor de texto (en nuestro caso, usamos pico) creamos en este directorio un

archivo jabber:

```
linux:/# cd /etc/pam.d
linux:/etc/pam.d# pico jabberd
```

En este archivo introducimos las siguientes líneas y se guardan los cambios:

```
#%PAM-1.0
auth     required       pam_unix2.so    nullok
password required       pam_unix2.so
account  required       pam_unix2.so
session  required       pam_unix2.so
```

Estas líneas le especifican a PAM que realice la comprobación de usuarios mediante los usuarios del sistema y verifique las contraseñas de la forma tradicional, es decir, mediante el archivo /etc/passwd.

3.8.4. Almacenamiento de datos mediante MySQL. SUSE Linux incluye como su paquete por defecto para bases de datos a MySQL. Por esta razón, en ella anteriormente creamos la base de datos para el almacenamiento de la información del servicio. Ahora debemos indicarle al servidor que las Colecciones que el necesita las encuentra en los directorios de datos asociados a MySQL.

En sm.xml, debemos editar las siguientes líneas para que el SM reconozca a MySQL como su contenedor de datos:

```
 <!-- Storage database configuration -->
  <storage>
    <!-- By default, we use the MySQL driver for all storage -->
    <driver>mysql</driver>
...
```

También se deben especificar algunos parámetros para MySQL, como el puerto

TCP para las conexiones a la base de datos y el usuario autorizado para manipularla:

```
<!-- MySQL driver configuration -->
  <mysql>
    <!-- Database server host and port -->
    <host>localhost</host>
    <port>3306</port>

    <!-- Database name -->
    <dbname>jabberd2</dbname>

    <!-- Database username and password -->
    <user>jabberd2</user>
    <pass>password</pass>
    <transactions/>
  </mysql>
```

3.8.5. Manejo de los Logs del servidor. Jabber v.2.0, a diferencia de su antecesor, por defecto registra sus avisos con los Logs del sistema (syslog), luego no es necesario cambiar esta configuración si se es familiar con el manejo de Logs del sistema. Sin embargo, si prefiere, puede almacenarlos en la ruta creada en los preparativos para la instalación. Cada proceso tiene su respectivo archivo Log, luego debemos ajustar todos los archivos de configuración:

Para sm.xml:

```
<log type='file'>
<file>/var/log/jabber/sm.log</file>
</log>
```

Para c2s.xml:

```
<log type='file'>
<file>/var/log/jabber/c2s.log</file>
```

```
</log>
```

Este procedimiento se repite para los procesos restantes (s2s, resolver, router).

3.8.6. Proveer conexiones seguras mediante SSL. Para cumplir con el requisito de confidencialidad con en las conexiones Cliente a Servidor, es necesario especificarle al proceso c2s que todas las transacciones deben ser codificadas utilizando un certificado SSL que se encuentra en la ubicación /etc/jabber. Originalmente Jabber v.2.0 no proporciona ningún certificado de seguridad, luego es necesario crear nuestro propio certificado por medio de la utilidad proporcionada por OpenSSL.

Para generar la llave pública y la privada, entramos la siguiente secuencia de comandos desde un directorio temporal:

```
linux:/tmp/ssl # openssl req -new -x509 -newkey rsa:1024 -days 365 -keyout |\
> private.pem -out certificate.pem
```

A continuación se genera una clave aleatoria mediante RSA. Inicia un asistente para la creación del certificado, el cual pregunta por el passphrase para la clave privada. Posteriormente el asistente pide alguna información de identificación, la cual es opcional.

```
Generating a 1024 bit RSA private key
...++++++
.....................++++++
writing new private key to 'private.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CO
State or Province Name (full name) [Some-State]:Santander
Locality Name (eg, city) []:Bucaramanga
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Familiar Ltd
Organizational Unit Name (eg, section) []:IT
Common Name (eg, YOUR name) []:Javier
Email Address []:lelajels@excite.com
```

En el directorio temporal puede observarse la clave privada y el certificado creados. Por seguridad, debemos remover el passphrase de la clave privada:

```
linux:/tmp/ssl # ls
.  ..  certificate.pem  private.pem
linux:/tmp/ssl # openssl rsa -in private.pem -out private.pem
Enter pass phrase for private.pem:
writing RSA key
```

Posteriormente debemos combinar la llave privada con la llave pública, para construir el certificado y borramos la clave privada:

```
linux:/tmp/ssl # cat private.pem >> certificate.pem
linux:/tmp/ssl # rm private.pem
```

El certificado está listo para asociarse con el servicio. Debemos moverlo al directorio de los archivos de configuración de Jabberd.

```
linux:/tmp/ssl # mv certificate.pem /etc/jabberd/
```

En el archivo de configuración c2s.xml, modificamos las siguientes líneas para que las comunicaciones Cliente a Servidor soporten seguridad mediante SSL.

```
<local>
...
```

```
 <pemfile>/etc/jabberd/certificate.pem</pemfile>
...
   <require-starttls/>
...
   <ssl-port>5223</ssl-port>
</local>
```

Al servicio se le ha especificado que el certificado de seguridad está en el directorio de configuración de Jabberd. Adicionalmente, se obliga a los clientes a que soliciten una conexión segura con el servidor. Opcionalmente, se habilita el puerto TCP 5223 para establecer transacciones encriptadas, para asegurar que los clientes antiguos también puedan tener conexiones seguras.

3.8.7. Deshabilitar creación de usuarios y cambios de password. Jabberd por defecto permite el registro de nuevas cuentas de usuario desde los clientes. Sin embargo, esto no es deseable en el ámbito empresarial porque los empleados podrían crear múltiples cuentas, lo que acarrearía un uso inadecuado del servicio. Es recomendable deshabilitar el registro de nuevas cuentas de usuario, incluso si se usa PAM para la autenticación, donde el registro público no funciona.

```
<!-- Registration configuration -->
   <register>
...

   <!-- <enable/> -->
     ...
```

Opcionalmente, podemos permitir que el usuario cambie su contraseña del servicio. Sin embargo, esta función no funciona cuando la autenticación la realiza PAM, luego optamos por deshabilitarla.

```
<!-- Password change only. When registration is disabled, it may
 still be useful to allow clients to change their password. If
 you want this, uncomment this when you disable registration. -->
```

```
    <!--
    <password/>
     -->
  </register>
```

3.8.8. Configuración de MU-Conference. Previamente hemos instalado a MU-Conference como componente de charlas multiusuario. Ahora debemos crear un archivo de configuración para este proceso, cuyo nombre será muc-jcr.xml, y se ubicará en la carpeta de los archivos de configuración /etc/jabberd.

El componente debe integrarse al Enrutador de Jabberd v.2.0 para que el servicio entre en operación.

```
<jcr>
   <name>muclinker</name>
  <host>conference.lelajels.gotdns.org</host>
  <ip>127.0.0.1</ip>
  <port>5347</port>
  <secret>secret</secret>
```

Configuramos otras opciones para el servicio. Debemos definir un directorio para las salas de conferencia o spool, una ubicación para los logs de este proceso y otra para el archivo PID:

```
  <spool>/var/spool/mu-conference/lelajels.gotdns.org</spool>
  <logdir>/var/log/mu-conference</logdir>
  <pidfile>/var/lib/mu-conference/mu-conference.pid</pidfile>
```

Finalmente, se definen otras opciones como la vCard[17] del servicio, formato de las notificaciones, y un usuario de administración:

```
 <!--
```

[17] Son tarjetas de visita electrónicas, en ellas van los datos del remitente.

```
    <logstderr/>
  -->
  <loglevel>124</loglevel>

    <conference xmlns="jabber:config:conference">
      <public/>
      <vCard>
        <FN>Salas de Charla Multi-usuario</FN>
        <DESC>Este es el servicio de conferencia empresarial</DESC>
        <URL>http://lelajels.gotdns.org/</URL>
      </vCard>
      <history>40</history>
      <logdir>./logs/</logdir>
      <notice>
        <join>se ha unido a la conversacion</join>
        <leave>se ha ido</leave>
        <rename>se conoce ahora como</rename>
      </notice>
      <sadmin>
        <user>presence@localhost</user>
      </sadmin>
    </conference>
</jcr>
```

3.8.9. Configuración de Users - Agent. Es necesario realizar dos ajustes para que el servicio JUD funcione correctamente. Primero, debemos garantizar el acceso del usuario de MySQL jabberd2 a la base de datos.

Desde el cliente MySQL en una sesión de terminal, entramos la siguiente secuencia de comandos:

```
linux:/ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 86 to server version: 4.1.13

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> use JUD;
Database changed
mysql> GRANT ALL PRIVILEGES ON *.* TO 'jabberd2'@'localhost'
−> IDENTIFIED BY 'secret' WITH GRANT OPTION;
```

De esta forma, hemos dado al usuario jabberd2 de MySQL todos los privilegios sobre la base de datos JUD. Ahora, debemos integrar el servicio con el Enrutador de Jabberd v.2.0, modificando el archivo de configuración /etc/config.xml. En nuestro caso, el archivo de configuración quedaría de la siguiente forma:

En la sección <server>, creamos el vínculo al Enrutador, de manera que debe especificarse el host, puerto y clave de acceso al recurso:

```
<config>
  <server>
    <connectiontype>accept</connectiontype>
    <hostname>127.0.0.1</hostname>
    <port>5347</port>
    <secret>secret</secret>
  </server>
```

En la sección componente especificamos el nombre de dominio para el servicio, de manera similar a lo hecho con el componente de conferencia:

```
<component>
    <name>users.lelajels.homelinux.org</name>
  </component>
```

Finalmente, el servicio necesita conocer el nombre de usuario y la contraseña para acceder a la base de datos de directorio de usuarios en MySQL

```
<mysql>
    <dbname>JUD</dbname>
    <username>jabberd2</username>
    <password>secret</password>
    <limit>250</limit>
```

```
  </mysql>
</config>
```

El servicio se encuentra listo para su operación con Jabberd v.2.0.

4. RESULTADOS OBTENIDOS

4.1. ARRANQUE DE LOS SERVICIOS

Antes de realizar pruebas para verificar que los servicios funcionan correctamente, es necesario saber como inicializar los servicios. En la mayoría de distribuciones de Linux, incluyendo a SUSE, basta con ejecutar el binario del proceso correspondiente.

Para el arranque de los procesos de Jabberd v.2.0, nos ubicamos en el directorio de los ejecutables e introducimos la siguiente secuencia de comandos:

```
linux:/usr/bin # ./jabberd &
```

El uso de un ampersand al final del comando permite que el proceso se ejecute en memoria (background) y no ocupe el terminal en el que actualmente se trabaja. En el caso de los componentes externos, tales como MU-Conference y Users-Agent, el arranque de estos servicios es muy similar, con la diferencia que debemos especificar la ruta a los archivos de configuración.

```
linux:/usr/bin # ./mu-conference -c /etc/jabberd/muc-jcr.xml &
```

```
linux:/usr/bin # ./users-agent -c /etc/jabberd/config.xml &
```

En estos casos el uso de la opción -c permite especificar la ruta a los archivos de configuración en el caso de que éstos no se encuentren en el mismo directorio de los ejecutables.

El arranque de estos servicios podría optimizarse mediante la creación de shell scripts que automaticen este proceso, de manera que pueda hacerse desde el tiempo de arranque del sistema.

4.2. PRUEBAS DEL SISTEMA

4.2.1. Procesos en Ejecución. Si cada uno de los componentes del servicio jabber han sido instalados y configurados correctamente, debemos observar en memoria del sistema los procesos para cada uno de ellos. Recordemos que cada proceso tiene un PID que se almacena en las rutas previamente especificadas en los archivos de configuración. Para comprobar los PIDs asignados por el sistema, accedemos al directorio para estos archivos y observamos el contenido de estos:

Para los componentes internos del servicio:

```
linux:~ # cd /var/lib/jabberd/pid/
linux:/var/lib/jabberd/pid # ls
.  ..  c2s.pid  resolver.pid  router.pid  s2s.pid  sm.pid
linux:/var/lib/jabberd/pid # cat sm.pid
13368
linux:/var/lib/jabberd/pid # cat router.pid
13356
linux:/var/lib/jabberd/pid # cat resolver.pid
13362
linux:/var/lib/jabberd/pid # cat sm.pid
13368
linux:/var/lib/jabberd/pid # cat c2s.pid
13385
linux:/var/lib/jabberd/pid # cat s2s.pid
```

Para los componentes externos del servicio:

MU-Conference:

```
linux:/var/lib/mu-conference # cat mu-conference.pid
13399
linux:/var/lib/mu-conference #
```

Users-Agent: (Sin ubicación para el PID).

Para observar los procesos del sistema, podemos usar el comando ps -ef. Debemos comprobar que los servicios de Jabber se encuentren activos, incluyendo el paquete externo de autenticación y almacenamiento:

```
linux:/ # ps -ax

  PID TTY      STAT   TIME COMMAND
...
4622 ?        S      0:00 /bin/sh /usr/bin/mysqld_safe --user=mysql
4727 ?        Sl     0:02 /usr/sbin/mysqld --basedir=/usr
...
13356 pts/0    S      0:01 /usr/bin/router -c /etc/jabberd/router.xml
13362 pts/0    S      0:00 /usr/bin/resolver -c /etc/jabberd/resolver.xml
13368 pts/0    S      0:01 /usr/bin/sm -c /etc/jabberd/sm.xml
13385 pts/0    S      0:01 /usr/bin/c2s -c /etc/jabberd/c2s.xml
13396 pts/0    S      0:00 /usr/bin/s2s -c /etc/jabberd/s2s.xml
13399 pts/0    Sl     0:00 mu-conference -c /etc/jabberd/muc-jcr.xml
13409 pts/0    S      0:06 perl -x ./users-agent -c /etc/jabberd/config.xml
```

4.2.2. Registro de Logs. El servidor Jabberd v.2.0 constantemente registra diferentes mensajes relacionados con el funcionamiento del servicio, por ejemplo, el inicio de los procesos, advertencias de errores, conexiones de los clientes, entre otros. Como de habló en paginas anteriores, puede permitirse al servicio que registre los sucesos con los del sistema (syslog) o en un archivo de log (.log).

Para el caso de que los logs se registren con el sistema, la visualización de estos depende de la distribución de Linux que se use. Para SUSE Linux, pueden visualizarse por medio de la herramienta de administración YaST o simplemente

observando el archivo /var/log/messages. A continuación se ilustra el registro para la conexión de un usuario del servicio.

```
linux:/var/log # cat messages
...
Jan 25 11:47:31 linux jabberd/c2s[13385]: [6] [200.122.233.218, port=1580] connect

Jan 25 11:47:32 linux jabberd/c2s[13385]: [6] SASL authentication succeeded: mechanism=PLAIN; authzid=yacano@lelajels.homelinux.org

Jan 25 11:47:32 linux jabberd/c2s[13385]: [6] bound: jid=yacano@lelajels.homelinux.org/Exodus

Jan 25 11:47:32 linux jabberd/c2s[13385]: [6] requesting session: jid=yacano@lelajels.homelinux.org/Exodus

Jan 25 11:47:32 linux jabberd/sm[13368]: session started: jid=yacano@lelajels.homelinux.org/Exodus
```

4.3. PRUEBAS DE CONECTIVIDAD

Hasta el momento hemos verificado que los servicios asociados a Jabber se encuentren en ejecución, sin embargo, esto no es garantía de que funcionen correctamente. Por tal razón, es conveniente realizar algunas pruebas de conectividad para asegurarnos que sea las diferentes opciones se han configurado correctamente. Algunas pruebas que se pueden ejecutar se describen a continuación.

4.3.1. Resolución de Nombre de la maquina. Para que los clientes sean capaces de descubrir el host donde se ejecuta el servicio mediante el JID, el nombre del host debe estar registrado con al menos una entrada en un servicio DNS. De lo contrario, para que los clientes conecten con el servicio deberá especificarse una dirección IP y el puerto, lo cual no es eficiente.

Para realizar la comprobación basta con ejecutar un comando de resolución de nombres, por ejemplo nslookup.

```
linux:~ # nslookup lelajels.gotdns.org
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   lelajels.gotdns.org
Address: 200.114.27.70
```

El host está asociado con una IP pública, lo que permitirá que acepte conexiones por fuera de la LAN, por ejemplo para los teletrabajadores. Opcionalmente pueden definirse Alias de nombres, cuando varios servicios se ejecutan en el mismo host, por ejemplo con los servicios de conferencia.

4.3.2. Mapeo de Puertos. Para que el servicio Jabber pueda aceptar conexiones de los diferentes componentes, es necesario que el host que los ejecuta, si bien no se puede deshabilitar el cortafuegos, por lo menos debe tener abiertos los puertos TCP requeridos. En el caso del Enrutador o Cortafuegos que separan la LAN del exterior, sólo deben abrirse los puertos necesarios, por ejemplo se abren los puertos de c2s para que conecten los empleados externos, pero se cierran los del s2s para evitar que usuarios de otros servicios Jabber puedan acceder al nuestro. La siguiente es la salida entregada por el escaner de puertos nmap:

```
[root@www ~]# nmap -P0 -sV -p 1-10000 lelajels.homelinux.org

Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2007-01-25 14:18 COT
Interesting ports on 200.114.27.60:

(The 9997 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 4.1 (protocol 1.99)
5222/tcp  open  unknown
10000/tcp open  http    Webmin httpd
```

Al escáner de puertos se le especificó que probara en el host el rango de puertos TCP de 1 a 10000 (-p 1-10000) y que no usara el método de ping (-P0, previendo que el nombre de dominio apunte a una IP pública asignada a un Firewall). En caso de encontrar puertos abiertos, debe intentar determinar el servicio de red asociado a ese puerto (-sV).

En el ejemplo, el escáner de puertos claramente identificó algunos servicios disponibles al exterior, con excepción de uno. Recordemos que el puerto 5222 corresponde al componente Cliente a Servidor de Jabberd v.2.0. Esto permite suponer que las conexiones de clientes que se encuentran al exterior de la LAN son permitidas, pero no las conexiones s2s, de manera que no se puede contactar con usuarios de otros servicios Jabber diferentes al nuestro.

4.3.3. Conexión a los servicios mediante Telnet. Una prueba util para determinar si el servicio se encuentra en operación es la de establecer una conexión con el servidor mediante telnet, en nuestro caso a través del puerto TCP que utiliza el componente c2s, 5222. Sin embargo, para probar el funcionamiento del servicio, se debe establecer una conversación con el protocolo del servicio, es decir en XML.

Para efectos de la prueba, lo haremos desde un host remoto (al exterior de la LAN) y enviaremos un pequeño fragmento de XML.

```
[root@www ~]# uname -a
Linux www.sts.com.co 2.6.9-5.ELsmp #1 SMP Wed Jan 5 19:30:39 EST 2005 i686 i686 i386 GNU/Linux


[root@www ~]# telnet lelajels.homelinux.org 5222
Trying 200.114.27.60...
Connected to lelajels.homelinux.org (200.114.27.60).
Escape character is '^]'.
<stream:stream
to='lelajels.homelinux.org'    # Cambiar por el dominio del servicio.
xmlns='jabber:client'
xmlns:stream='http://etherx.jabber.org/streams'>
```

El servidor nos envía este fragmento en respuesta a la anterior petición:

```
<?xml version='1.0'?><stream:stream xmlns:stream='http://etherx.jabber.org/streams' xmlns='jabber:client' from='lelajels.homelinux.org' id='7aphgbhqd6k03wzjn82nylnbq3r3tss03hijs0a5'></stream:stream>
```

Al obtener respuesta, cerramos la conexión con la siguiente etiqueta:

```
</stream:stream>
Connection closed by foreign host.
```

Al obtener respuesta por parte del servidor, significa que el servicio esta activo y en espera de conexiones por parte de los clientes.

4.4. PRUEBAS DE CONEXIÓN MEDIANTE CLIENTE EXODUS

Una vez verificado que el servicio se encuentre en ejecución y listo para aceptar conexiones por parte de los clientes, podemos acceder al servicio desde cualquier host de la LAN o del exterior, siempre y cuando el usuario se encuentre registrado

con el servidor. Para verificar la conexión de los clientes se utilizará Exodus, ya que este cliente proporciona la mayoría de características del servicio Jabber.

Para empezar el software de Exodus, damos doble clik en cualquiera de los iconos que aparecen en el escritorio del computador, como lo muestra la siguiente figura.

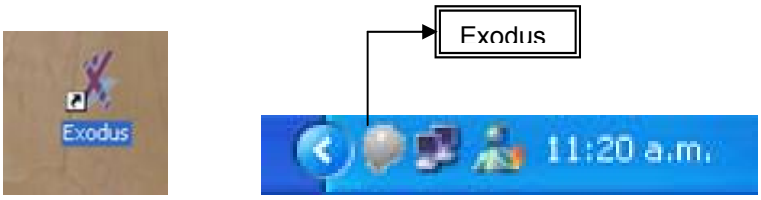

Figura 4. Iconos para iniciar el Servicio de Exodus

4.4.1. Autenticación. En nuestro caso particular que la autenticación se hará con las cuentas de usuario del sistema, podemos visualizar las cuentas creadas para saber que usuarios pueden autenticar con el servidor. En SUSE Linux, la utilidad de administración YaST puede mostrarnos esta información:

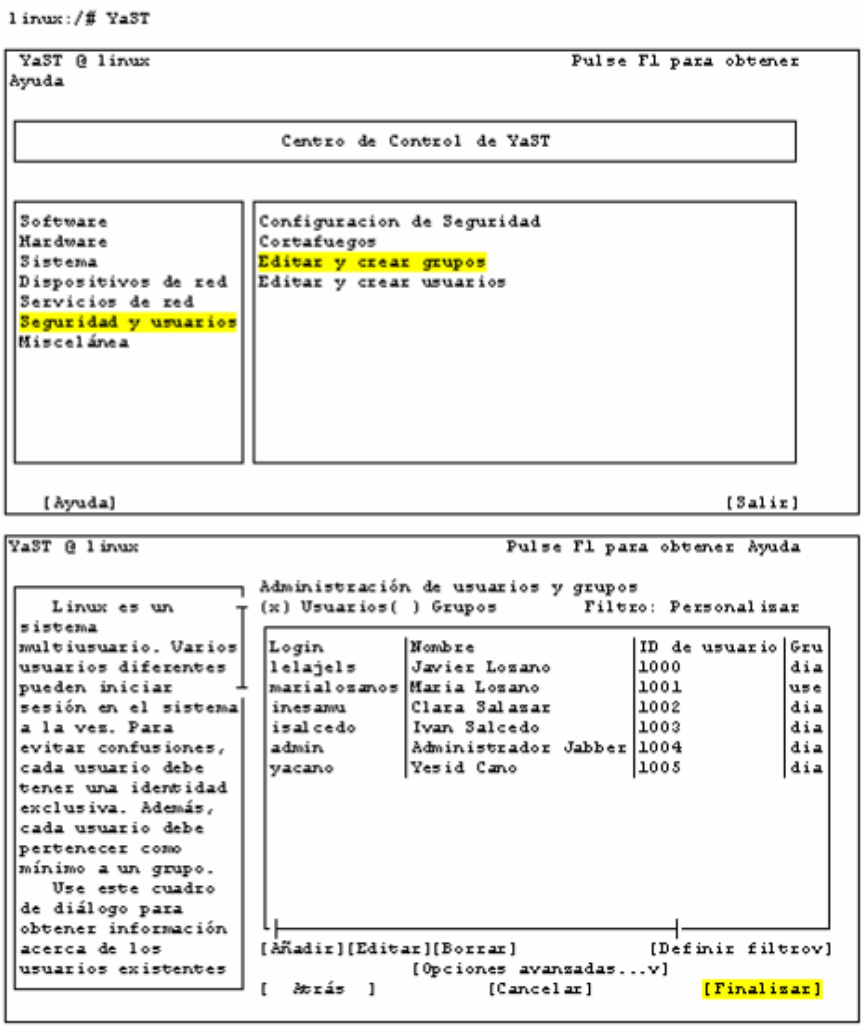

Figura 5. Administración por YaST

A continuación en un cliente, se ilustrará la conexión al servicio con el usuario yacano, el cual se encuentra registrado en el sistema.

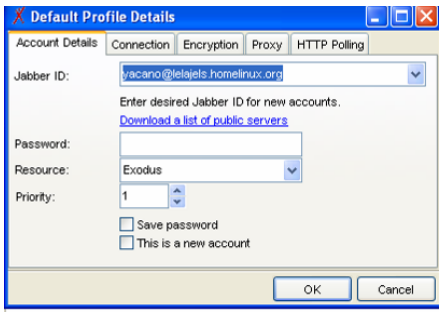

Figura 6. Digitar cuenta del usuario

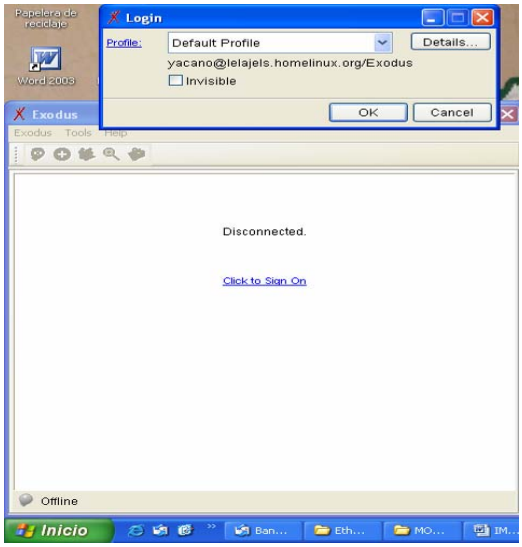

Figura 7. Inicio de Sesión del usuario

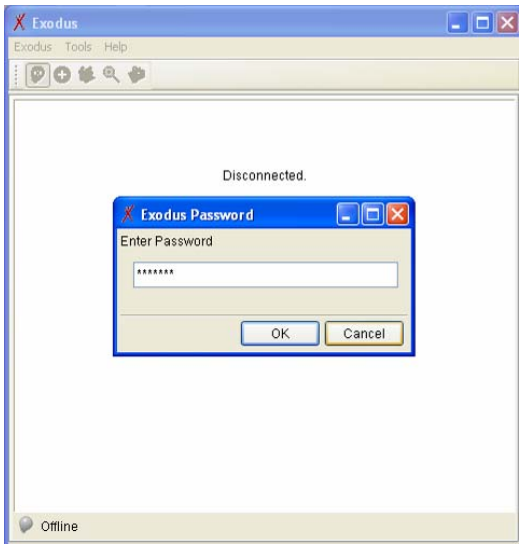

Figura 8. Caja para digitar la clave o password

A continuación, observamos el inicio de la sesión para la cuenta del usuario yacano.

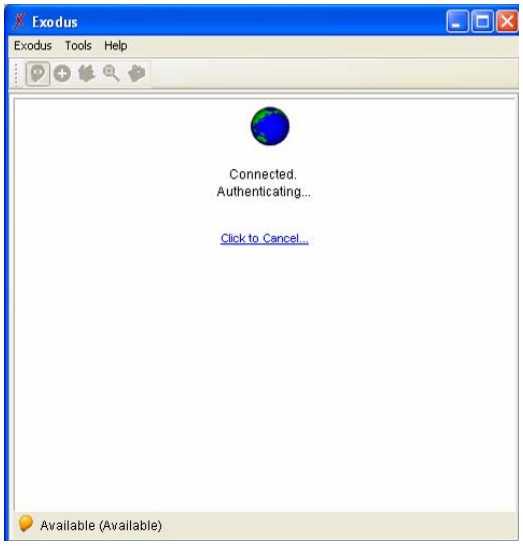

Figura 9. Autorización de Conexión

Figura 10. Icono de conexión establecida

De esta forma acaba el “inicio de sesión”. A partir de aquí el cliente notificara los cambios que realicen en los otros clientes que están en nuestra lista de contactos.

4.4.2. Intercambio de mensajes. Al seleccionar cualquier contacto que forma parte de nuestra Nómina, inmediatamente se despliega una ventana que permite mantener una conversación únicamente con esa persona. En ella podemos visualizar los mensajes que se intercambian mediante las conexiones TCP/IP que los clientes establecen con el servidor.

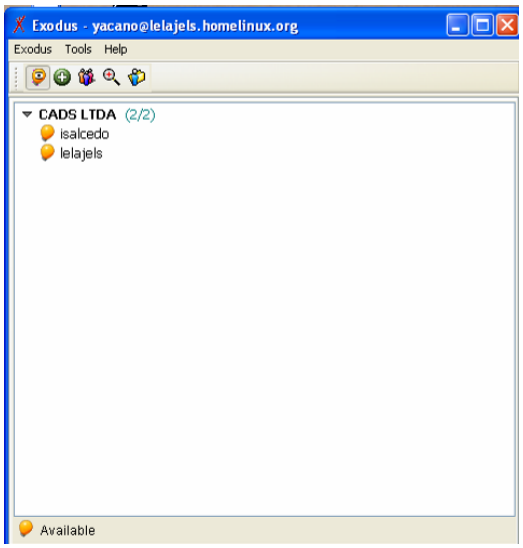

Figura 11. Pantalla de contactos

Figura 12. Creación de conversaciones

4.4.3. Descubriendo servicios. Exodus proporciona un mecanismo para determinar los servicios adicionales al de Jabber presentes en el servidor. El Jabber Browser, establece una conexión con el servidor y permite explorar que servicios se encuentran disponibles. Para nuestro caso, el Jabber Browser de Exodus muestra dos servicios disponibles, el de Conferencia y el Directorio de Usuarios.

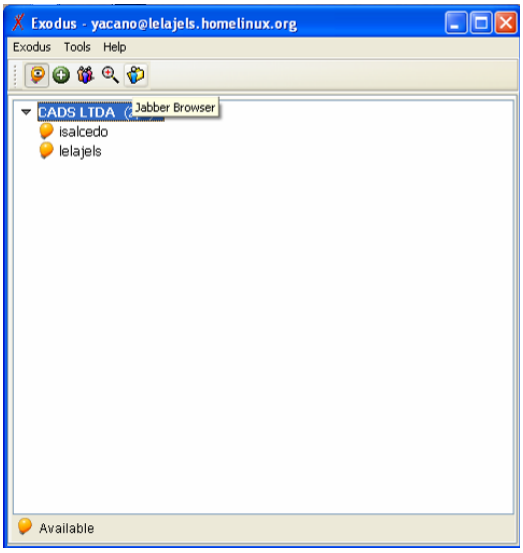

Figura 13. Uso de la opción Browser

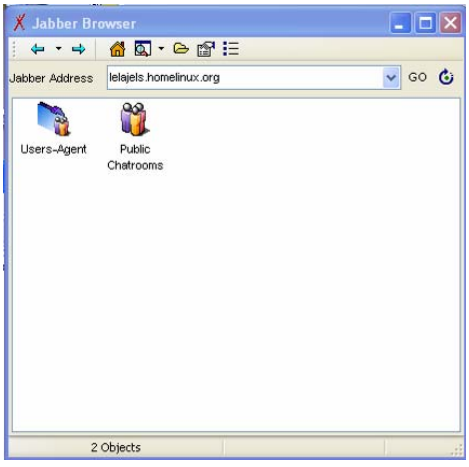

Figura 14. Directorios de usuarios y salones

4.4.4. Conferencia. Exodus incluye soporte para realizar conversaciones entre múltiples usuarios. Esto permite aprovechar las características de Conferencia implementadas en nuestro servicio Jabber. El comportamiento del servicio de conferencia es muy similar al del Internet Relay Chat, IRC.

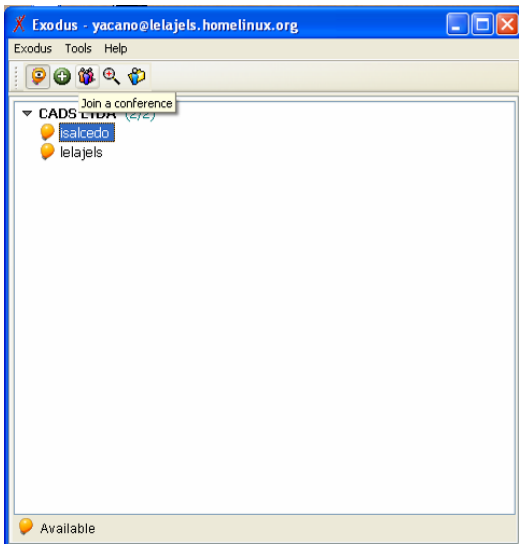

Figura 15. Opción de conferencia

Si conocemos el JID que identifica a una sala en el servicio, podemos acceder a ella por medio de este asistente.

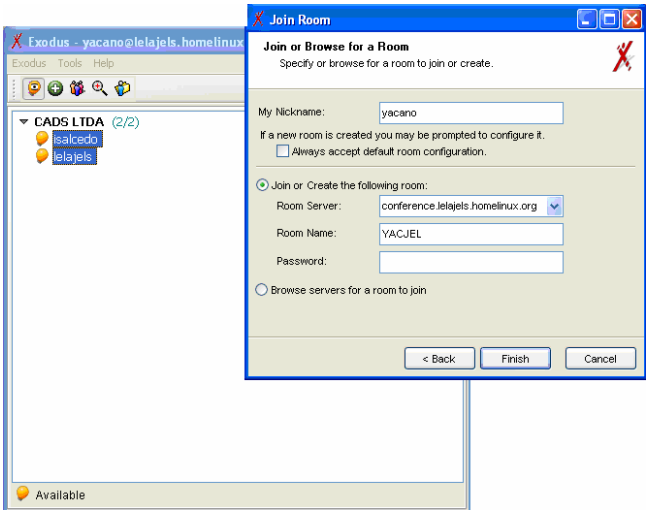

Figura 16. Inicio de una conferencia

De igual forma, podemos utilizar el Jabber Browser para encontrar una sala, y participar en una conversación múltiple.

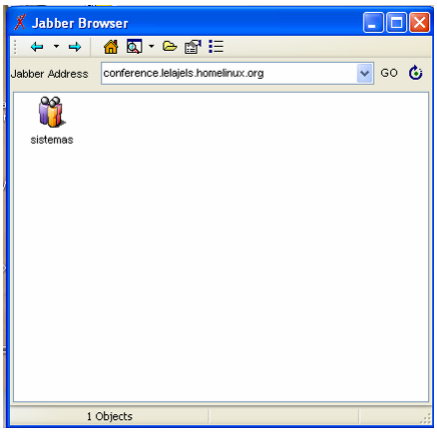

Figura 17. Listado de salas

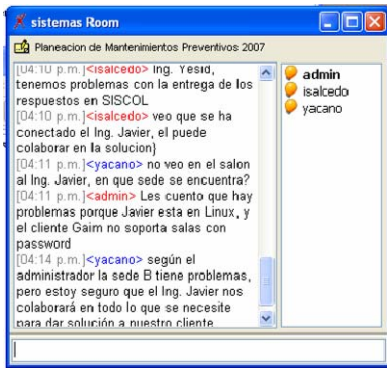

Figura 18. Conversación en una sala

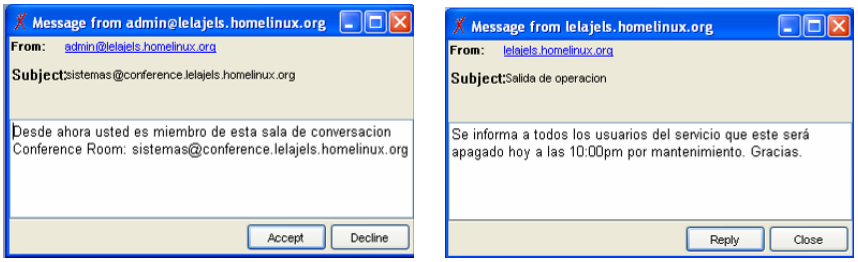

Figura 19. Mensajes tipo Broadcast

4.4.5. Directorio de usuarios. Por medio de Exodus los usuarios pueden registrar sus datos personales en el servicio de Directorio de Usuarios (JUD). Esto puede conseguirse por medio de la opción “Registrándose con un Servicio”.

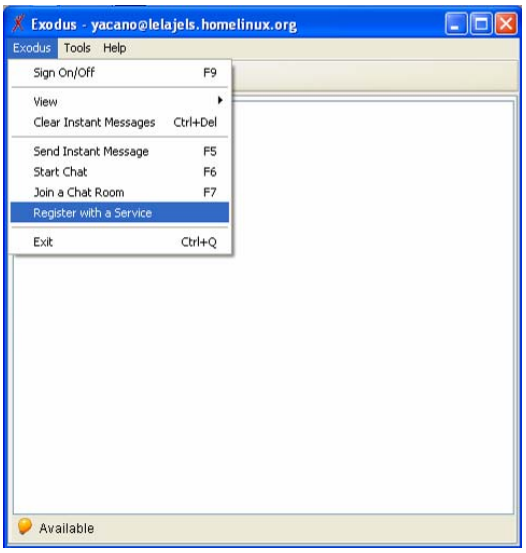

Figura 20. Registro de Servicio

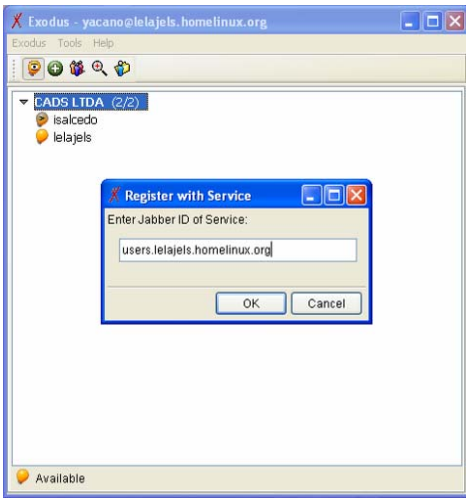

Figura 21. Nombre del servicio JUD

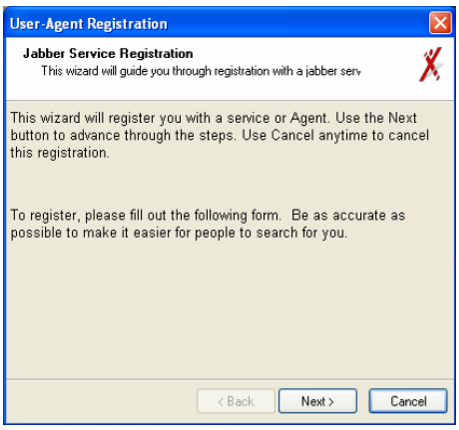

Figura 22. User Agent Registration

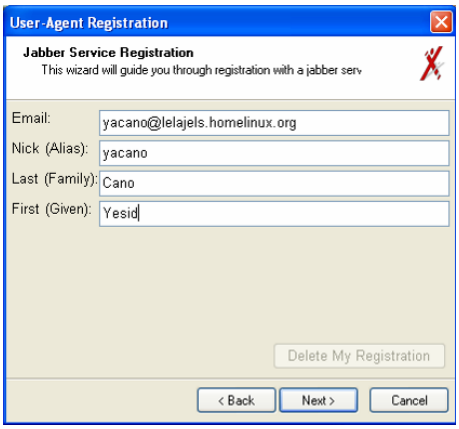

Figura 23. Datos de registro de usuario en el JUD

Figura 24. Final de registro en el JUD

Puede usarse el Jabber Browser para buscar un usuario de manera amigable en

el servicio de Directorio de Usuarios (JUD).

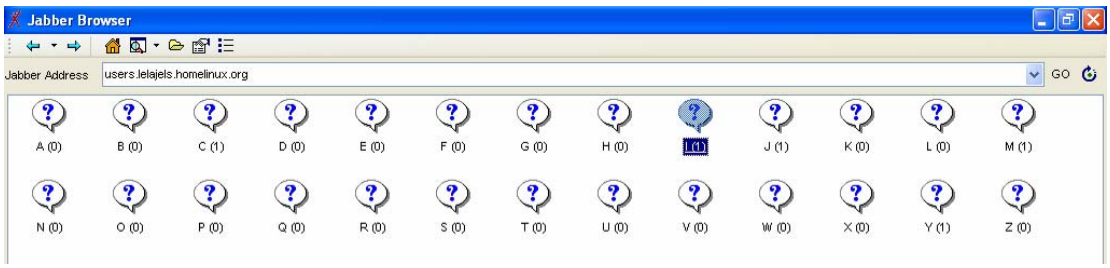

Figura 25. Listado de usuarios por primer nombre

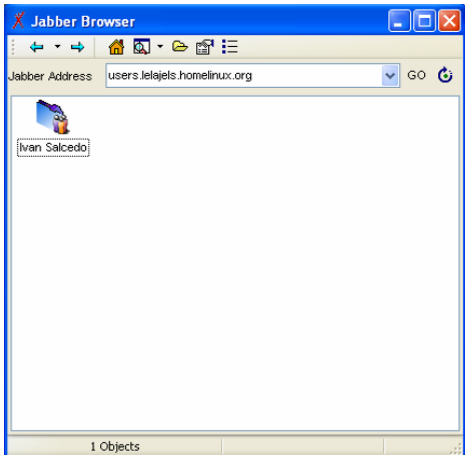

Figura 26. Usuarios cuyo primer nombre empieza por (I)

4.5. CONEXIÓN MEDIANTE CLIENTE SOAPBOX

Otra aplicación para aceptar las conexiones de parte de los clientes desde cualquier host de la LAN o del exterior, es utilizando SoapBox, ya que proporciona características del servicio XMPP/Jabber. Ofrece servicios similares como son: listas de contacto y capacidad de crear o ensamblar charlas de la comunicación, también maneja la presencia con los modos de disponibilidad y alarmas que se activan cuando entra uno de los contactos o al enviar los mensajes.

A continuación mostraremos una secuencia de pasos a seguir para el uso de esta opción de comunicación; para iniciar este software, damos doble clik en el icono que aparece en el escritorio del computador como lo muestra la siguiente figura.

Figura 27. Otra opción de software SoapBox

Figura 28. Lista de contactos en SoapBox

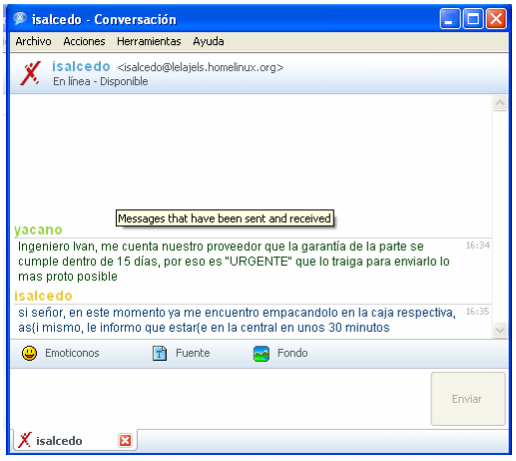

Figura 29. Conversación usando SoapBox

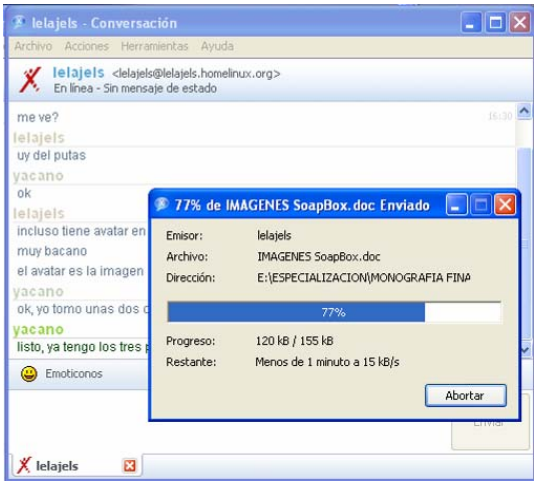

Figura 30. Transferencia de archivos con SoapBox

4.6. PRUEBAS DE ALMACENAMIENTO EN BASE DE DATOS

Esta prueba simplemente tiene por objeto verificar que el paquete externo de almacenamiento de información, en nuestro caso MySQL, se encuentre guardando los datos relacionados con el servicio.

4.6.1. Acceso a la base de datos jabberd2. En primer lugar, debemos acceder a la base de datos creada en MySQL para almacenar la información, cuyo nombre es jabberd2:

```
linux:~ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 159 to server version: 4.1.13

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use jabberd2
```

```
Database changed
```

Como se observó en el capitulo anterior, verificamos las Colecciones en la base de datos. Obtenemos la lista de tablas:

```
mysql> show tables;
+--------------------+
| Tables_in_jabberd2 |
+--------------------+
| active             |
| authreg            |
| disco-items        |
| logout             |
| motd-message       |
| motd-times         |
| privacy-default    |
| privacy-items      |
| private            |
| queue              |
| roster-groups      |
| roster-items       |
| vacation-settings  |
| vcard              |
+--------------------+
14 rows in set (0.05 sec)
```

Una vez conocidas las tablas, procedemos a realizar consultas para visualizar la información contenida en dichas Colecciones.

4.6.1.1. Usuarios activos. En la siguiente figura podemos observar la tabla que contiene los JID de los usuarios que han autenticado con nuestro servicio de Jabber y el timestamp en el que fueron creados, en otras palabras, entraron en actividad.

Table active in database jabberd2

Select all Invert selection

	collection-owner	object-sequence	time
☐	jlozano@lelajels.homelinux.org	1	1164673322
☐	suse@lelajels.homelinux.org	2	1164674020
☐	lelajels@lelajels.homelinux.org	3	1168306162
☐	marialozanos@lelajels.homelinux.org	4	1168307284
☐	inesamu@lelajels.homelinux.org	5	1168309378
☐	isalcedo@lelajels.homelinux.org	6	1168736630
☐	admin@lelajels.homelinux.org	7	1168828749
☐	yacano@lelajels.homelinux.org	8	1169736921

Select all Invert selection

Edit selected rows | Add row | Delete selected rows

Figura 31. JID de usuarios autenticados

4.6.1.2. Disco Ítems. Esta tabla no contiene datos, pues están deshabilitadas las transacciones s2s.

4.6.1.3. Logout. En la figura que sigue observamos el timestamp correspondiente a la última desconexión de los usuarios activos en el sistema.

Table logout in database jabberd2

Select all Invert selection

	collection-owner	object-sequence	time
☐	jlozano@lelajels.homelinux.org	7	1168305638
☐	suse@lelajels.homelinux.org	6	1168287395
☐	lelajels@lelajels.homelinux.org	97	1169825914
☐	marialozanos@lelajels.homelinux.org	62	1169703345
☐	inesamu@lelajels.homelinux.org	74	1169746525
☐	isalcedo@lelajels.homelinux.org	96	1169778074
☐	admin@lelajels.homelinux.org	89	1169774225
☐	yacano@lelajels.homelinux.org	94	1169776512

Select all Invert selection

Edit selected rows | Add row | Delete selected rows

Figura 32. Listado de timestamp de desconexión de usuarios

4.6.1.4. Motd-message. En la siguiente figura mostramos el código XML para un mensaje del día o broadcast creado por el administrador.

Table motd-message in database jabberd2

Select all. Invert selection.

	collection-owner	object-sequence	xml
☐	lelajels.homelinux.org	3	NAD<route xmlns='http://jabberd.jabberstudio.org/ns/component/1.0' from='lelajels.homelinux.org' to='lelajels.homelinux.org'><message xmlns='jabber:client' from='admin@lelajels.homelinux.org/Exodus' to='lelajels.homelinux.org/announce' id=''><x xmlns='jabber:x:delay' from='lelajels.homelinux.org' stamp='20070125T21:07:22'/><subject>Salida de operacion</subject><body>Se informa a todos los usuarios del servicio que este serÃ¡ apagado hoy a las 10:00pm por mantenimiento. Gracias.</body></message></route>

Select all. Invert selection.

Edit selected rows | Add row | Delete selected rows

Figura 33. Mensaje del día o Broadcast

4.6.1.5. Motd-times. Esta figura nos muestra el timestamp en que los usuarios activos del servicio han recibido los mensajes del día o broadcasts enviados por el administrador.

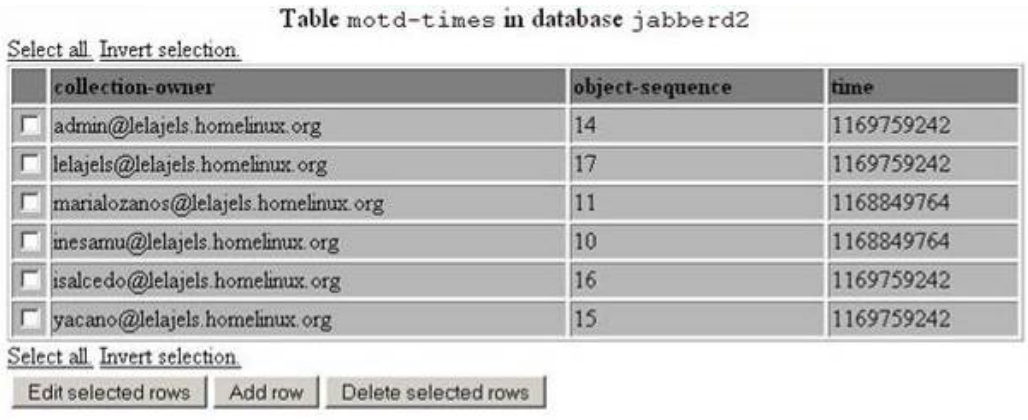

Table motd-times in database jabberd2

Select all. Invert selection.

	collection-owner	object-sequence	time
☐	admin@lelajels.homelinux.org	14	1169759242
☐	lelajels@lelajels.homelinux.org	17	1169759242
☐	marialozanos@lelajels.homelinux.org	11	1168849764
☐	inesamu@lelajels.homelinux.org	10	1168849764
☐	isalcedo@lelajels.homelinux.org	16	1169759242
☐	yacano@lelajels.homelinux.org	15	1169759242

Select all. Invert selection.

Edit selected rows | Add row | Delete selected rows

Figura 34. Listado de usuarios que recibieron mensajes del día o Broadcast

4.6.1.6. Privacy-default. Esta tabla no contiene datos.

4.6.1.7. Privacy-items

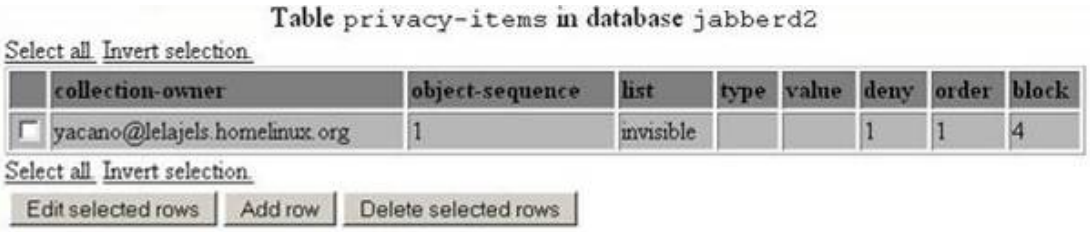

Figura 35. Listado de los usuarios de la tabla privacy-items

4.6.1.8. Private. Esta tabla no contiene datos.

4.6.1.9. Queue. Esta tabla no contiene datos.

4.6.1.10. Roster-groups. Esta tabla se muestra el JID de cada uno de los contactos de los usuarios del servicio con sus respectivos grupos en la Nómina.

Table roster-groups in database jabberd2

Select all Invert selection

	collection-owner	object-sequence	jid	group
☐	suse@lelajels.homelinux.org	4	jlozano@lelajels.homelinux.org	Amigos
☐	marialozanos@lelajels.homelinux.org	8	lelajels@lelajels.homelinux.org	Friends
☐	lelajels@lelajels.homelinux.org	20	marialozanos@lelajels.homelinux.org	Friends
☐	inesamu@lelajels.homelinux.org	24	marialozanos@lelajels.homelinux.org	Friends
☐	lelajels@lelajels.homelinux.org	18	inesamu@lelajels.homelinux.org	Friends
☐	inesamu@lelajels.homelinux.org	19	lelajels@lelajels.homelinux.org	Friends
☐	marialozanos@lelajels.homelinux.org	25	inesamu@lelajels.homelinux.org	Friends
☐	isalcedo@lelajels.homelinux.org	31	lelajels@lelajels.homelinux.org	Friends
☐	lelajels@lelajels.homelinux.org	32	isalcedo@lelajels.homelinux.org	Friends
☐	admin@lelajels.homelinux.org	35	lelajels.homelinux.org/announce@lelajels.homelinux.org	Friends
☐	lelajels@lelajels.homelinux.org	39	yacano@lelajels.homelinux.org	Friends
☐	isalcedo@lelajels.homelinux.org	43	yacano@lelajels.homelinux.org	Friends
☐	yacano@lelajels.homelinux.org	44	isalcedo@lelajels.homelinux.org	CADS LTDA
☐	yacano@lelajels.homelinux.org	45	lelajels@lelajels.homelinux.org	CADS LTDA

Select all Invert selection

Edit selected rows | Add row | Delete selected rows

Figura 36. El JID de cada uno de los contactos de los usuarios

4.6.1.11. Roster-items. Similar a la anterior pero nos muestra además el nick o alias para cada usuario.

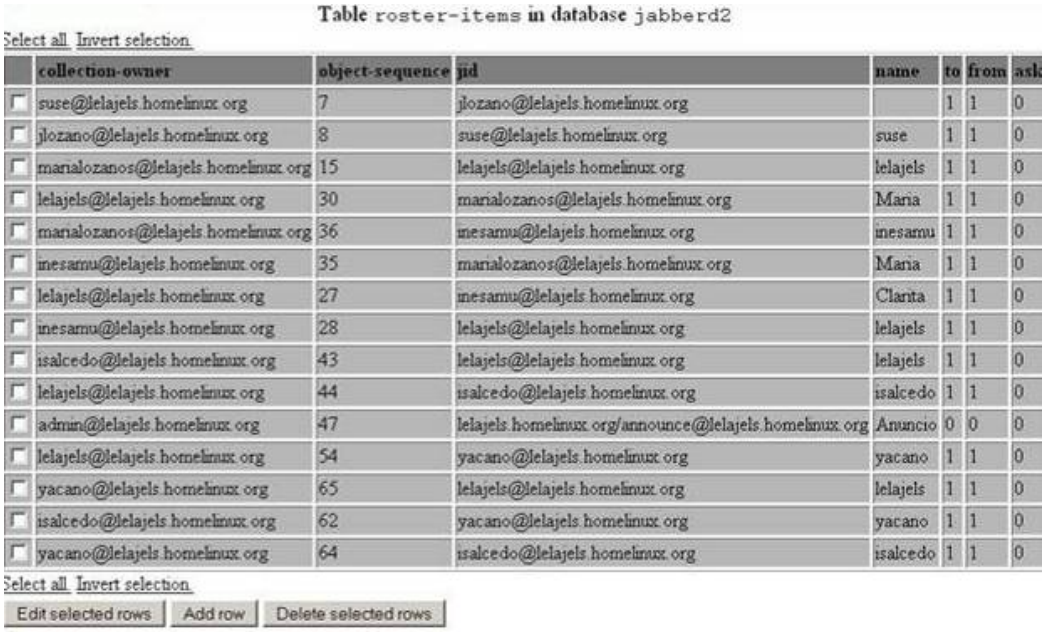

Table roster-items in database jabberd2

Select all Invert selection

	collection-owner	object-sequence	jid	name	to	from	ask
☐	suse@lelajels.homelinux.org	7	jlozano@lelajels.homelinux.org		1	1	0
☐	jlozano@lelajels.homelinux.org	8	suse@lelajels.homelinux.org	suse	1	1	0
☐	marialozanos@lelajels.homelinux.org	15	lelajels@lelajels.homelinux.org	lelajels	1	1	0
☐	lelajels@lelajels.homelinux.org	30	marialozanos@lelajels.homelinux.org	Maria	1	1	0
☐	marialozanos@lelajels.homelinux.org	36	inesamu@lelajels.homelinux.org	inesamu	1	1	0
☐	inesamu@lelajels.homelinux.org	35	marialozanos@lelajels.homelinux.org	Maria	1	1	0
☐	lelajels@lelajels.homelinux.org	27	inesamu@lelajels.homelinux.org	Clarita	1	1	0
☐	inesamu@lelajels.homelinux.org	28	lelajels@lelajels.homelinux.org	lelajels	1	1	0
☐	isalcedo@lelajels.homelinux.org	43	lelajels@lelajels.homelinux.org	lelajels	1	1	0
☐	lelajels@lelajels.homelinux.org	44	isalcedo@lelajels.homelinux.org	isalcedo	1	1	0
☐	admin@lelajels.homelinux.org	47	lelajels.homelinux.org/announce@lelajels.homelinux.org	Anuncio	0	0	0
☐	lelajels@lelajels.homelinux.org	54	yacano@lelajels.homelinux.org	yacano	1	1	0
☐	yacano@lelajels.homelinux.org	65	lelajels@lelajels.homelinux.org	lelajels	1	1	0
☐	isalcedo@lelajels.homelinux.org	62	yacano@lelajels.homelinux.org	yacano	1	1	0
☐	yacano@lelajels.homelinux.org	64	isalcedo@lelajels.homelinux.org	isalcedo	1	1	0

Select all Invert selection

Edit selected rows | Add row | Delete selected rows

Figura 37. El JID de cada contacto de los usuarios mostrando el nick o alias

4.6.1.12. Vacation-settings. Esta tabla no contiene datos.

4.6.1.13. Vcard. Esta tabla nos muestra los datos de los perfiles de usuario creados. No debe confundirse con el JUD.

```
mysql> select * from vcard;
+----------------------------------+-----------------+------+----------+-------------------------------+------+-------+-------+-------------+------------+----
--+---------+----------+------------+------------+--------------+------------+-----------+-------------+-------------+
| collection-owner                 | object-sequence | fn   | nickname | url                           | tel  | email | title | role        | bday       |
desc | n-given | n-family | adr-street | adr-extadd | adr-locality | adr-region | adr-pcode | adr-country | org-orgname | org-
orgunit |
+----------------------------------+-----------------+------+----------+-------------------------------+------+-------+-------+-------------+------------+----
--+---------+----------+------------+------------+--------------+------------+-----------+-------------+-------------+
| lelajels@lelajels.homelinux.org |               4 | NULL | El Brujo | http://lelajels.homelinux.org | NULL | NULL  | NULL  |
Engineering | 1981-11-19 | NULL | Javier  | Lozano   | NULL       | NULL       | Bucaramanga  | Santander | NULL        | NULL
| NULL        | NULL       |
+----------------------------------+-----------------+------+----------+-------------------------------+------+-------+-------+-------------+------------+----
```

```
--+---------+----------+------------+------------+--------------+------------+-----------+--------------+--------------+-------------+
1 row in set (0.00 sec)
```

4.6.2. Información personal de los usuarios. De igual forma a como se hizo con la base de datos jabberd2, accedemos a la base del servicio JUD y realizamos algunas consultas básicas para comprobar su funcionamiento.

En primer lugar, accedemos a la base de JUD a través del cliente mysql:

```
mysql> use JUD;
Database changed
```

Observamos las tablas creadas para esta base de datos:

```
mysql> show tables;
+---------------+
| Tables_in_JUD |
+---------------+
| jud           |
+---------------+
1 row in set (0.00 sec)
```

En la única tabla creada, se almacenan el primer nombre y apellido de cada usuario, así como su nickname y el correo electrónico de contacto. Haciendo una consulta a esta tabla, podemos observar los usuarios registrados con el servicio de JUD.

```
mysql> SELECT * FROM jud;
+----------------------------------------+------+--------+---------+----------+---------------------------------+
| jid                                    | name | first  | last    | nick     | email                           |
+----------------------------------------+------+--------+---------+----------+---------------------------------+
| lelajels@lelajels.homelinux.org     |      | Javier | Lozano  | El Brujo | lelajels@intercable.net.co      |
| inesamu@lelajels.homelinux.org      |      | Clara  | Salazar | Clarita  | inesamu@hotmail.com             |
| marialozanos@lelajels.homelinux.org |      | Maria  | Lozano  | MariaMe  | marialozanos@msn.com            |
| yacano@lelajels.homelinux.org       |      | Yesid  | Cano    | yacano   | yacano@lelajels.homelinux.org   |
```

| isalcedo@lelajels.homelinux.org | | Ivan | Salcedo | isalcedo | isalcedo@lelajels.homelinux.org |
+---------------------------------------+------+--------+---------+----------+---------------------------------+
5 rows in set (0.00 sec)

Como se observó anteriormente, las bases de datos fueron adecuadamente integradas con el servicio y almacenan la información de acuerdo a los eventos que se van presentando.

4.7. PRUEBAS DE SEGURIDAD DE DATOS

4.7.1. Sesión en texto plano. Los siguientes son ejemplos de transacciones que se establecen entre el servidor y los clientes, tomando muestras mediante el analizador de protocolos Ethereal. Los mensajes en XML con color rojo corresponden a peticiones hechas por los clientes al servidor, mientras que aquellas sin color corresponden a las respuestas por parte del servidor.

En esta transacción un cliente se autentica con el servicio por medio del mecanismo de texto plano. Sin embargo, puede observarse que la contraseña que envía el cliente está preencriptada por algún algoritmo, por ejemplo MD5.

<stream:stream to="lelajels.homelinux.org" xmlns="jabber:client" xmlns:stream="http://etherx.jabber.org/streams" xml:lang="es" version="1.0" >
<?xml version='1.0'?><stream:stream xmlns:stream='http://etherx.jabber.org/streams' xmlns='jabber:client' from='lelajels.homelinux.org' version='1.0' id='kef4k5f09f7w1ce5iekmfqkhqvb9q0wbecgzhvzr'>
<stream:features xmlns:stream='http://etherx.jabber.org/streams'><mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'><mechanism>PLAIN</mechanism></mechanisms></stream:features>
<auth mechanism="PLAIN" xmlns="urn:ietf:params:xml:ns:xmpp-sasl">bGVsYWplbHNAbGVsYWplbHMuaG9tZWxpbnV4Lm9yZwBsZWxhamVscwBzdHJva2VvZmx1Y2s=</auth>
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>

<stream:stream to="lelajels.homelinux.org" xmlns="jabber:client" xmlns:stream="http://etherx.jabber.org/streams" version="1.0" >
<?xml version='1.0'?><stream:stream xmlns:stream='http://etherx.jabber.org/streams' xmlns='jabber:client'

from='lelajels.homelinux.org' version='1.0' id='b7ncj2h2h2oxt9ej4ptvpq742n3kj3bubz6v1oxi'>
<stream:features xmlns:stream='http://etherx.jabber.org/streams'><bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'/><session xmlns='urn:ietf:params:xml:ns:xmpp-session'/></stream:features>
<iq id="jcl_18" type="set"><bind xmlns="urn:ietf:params:xml:ns:xmpp-bind"><resource>Exodus</resource></bind></iq>
<iq xmlns='jabber:client' id='jcl_18' type='result'><bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'><jid>lelajels@lelajels.homelinux.org/Exodus</jid></bind></iq>
<iq id="jcl_19" type="set"><session xmlns="urn:ietf:params:xml:ns:xmpp-session"/></iq>
<iq xmlns='jabber:client' id='jcl_19' type='result'/>

En esta transacción, el cliente pide la información de la Nómina del usuario al servidor. El servidor responde con la información de los JIDs que hacen parte de la Nómina.

<iq id="jcl_20" type="get"><query xmlns="jabber:iq:roster"/></iq>
<iq id="jcl_21" type="get"><query xmlns="jabber:iq:private"><storage xmlns="storage:bookmarks"/></query></iq>
<iq id="jcl_22" to="lelajels.homelinux.org" type="get"><query xmlns="http://jabber.org/protocol/disco#info"/></iq>
<iq xmlns='jabber:client' type='result' id='jcl_20'><query xmlns='jabber:iq:roster'><item name='Clarita' subscription='both' jid='inesamu@lelajels.homelinux.org'><group>Friends</group></item><item name='yacano' subscription='both' jid='yacano@lelajels.homelinux.org'><group>Friends</group></item><item name='Maria' subscription='both' jid='marialozanos@lelajels.homelinux.org'><group>Friends</group></item><item name='isalcedo' subscription='both' jid='isalcedo@lelajels.homelinux.org'><group>Friends</group></item></query></iq>

En esta transacción, el cliente solicita información de presencia de los usuarios al servidor. El servidor posteriormente responde con la información de los usuarios que se encuentran disponibles.

<presence><c node="http://exodus.jabberstudio.org/caps" ver="0.9.1.0" xmlns="http://jabber.org/protocol/caps"/><priority>1</priority></presence>
...
<presence xmlns='jabber:client' to='lelajels@lelajels.homelinux.org/Exodus' from='yacano@lelajels.homelinux.org/Exodus'><c xmlns='http://jabber.org/protocol/caps' ver='0.9.1.0' node='http://exodus.jabberstudio.org/caps'/><status>Available</status><priority>1</priority></presence>

En este bloque se observan los mensajes que se intercambian entre usuarios del servicio, y claramente puede observarse el contenido de estos entre etiquetas del tipo <body>.

<message xmlns='jabber:client' from='yacano@lelajels.homelinux.org/Exodus' type='chat' to='lelajels@lelajels.homelinux.org' id='jcl_8'><thread>40b0674fe4d5402a658bebbc12167f5077452683</thread><body>Inge. Javier, esperemos que entre el ing Ivan</body><x xmlns='jabber:x:event'><composing/></x></message>
<message to="yacano@lelajels.homelinux.org/Exodus" type="chat"><x xmlns="jabber:x:event"><composing/><id>jcl_8</id></x></message>
<message to="yacano@lelajels.homelinux.org/Exodus" type="chat"><x xmlns="jabber:x:event"><composing/><id/></x></message>
<message to="yacano@lelajels.homelinux.org/Exodus" type="chat"><x xmlns="jabber:x:event"><composing/><id>jcl_8</id></x></message>
<message to="yacano@lelajels.homelinux.org/Exodus" type="chat"><x xmlns="jabber:x:event"><composing/><id/></x></message>
<message to="yacano@lelajels.homelinux.org/Exodus" type="chat"><x xmlns="jabber:x:event"><composing/><id>jcl_8</id></x></message>
<message id="jcl_25" to="yacano@lelajels.homelinux.org/Exodus" type="chat"><thread>dc00bb8db43edc4e9400dd250bccbda3c85f2ed9</thread><body>Perfecto, asi procedemos con el orden del dia</body><x xmlns="jabber:x:event"><composing/></x></message>

Claramente puede observarse que por defecto el servicio Jabber no es seguro, hecho que justifica la necesidad de utilizar encripción SSL en las transacciones para garantizar la privacidad de las conversaciones empresariales.

4.7.2. Sesión encriptada mediante SSL. Cuando el servidor se configura para establecer conexiones mediante SSL, únicamente puede observarse con claridad la negociación de conexión segura. En este caso, el servidor le informa al cliente que las transacciones deben realizarse en forma segura. Una vez el cliente acepta la oferta, el servidor le informa sobre el certificado de seguridad con el cual se codificarán todos los mensajes.

<stream:stream to="lelajels.homelinux.org" xmlns="jabber:client" xmlns:stream="http://etherx.jabber.org/streams" xml:lang="es" version="1.0" >
<?xml version='1.0'?><stream:stream xmlns:stream='http://etherx.jabber.org/streams' xmlns='jabber:client' from='lelajels.homelinux.org' version='1.0' id='7xz751wieoesg1yxytcgvc5yuhb6hfuodtwius19'>
<stream:features xmlns:stream='http://etherx.jabber.org/streams'><starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'><required/></starttls></stream:features>
<starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls"/>

<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
....U...Q..E.X.F
|..1j....~.~..x.'.......+...*.....
.f.............e.d.c.b.a.`.............
....J...F..E.Z...H..hR.... sK...:N@..`.)(.. ?..B../...-./N.....4..\&..;%K...
.................0..{0.............3..v.QV0 ..U.
..STS1.0...U...
..U.
..STS1.0...U...
...2n.....a...px(.....t1.[.......V{7..}.....B...nb.....T..e....n.\...u.L..u.........0..0...U......A. K.....;..
...A.m|0....U.#...0....A. K.....;..
...A.m|......0..1.0...U....CO1.0...U....Santander1.0...U....Bucaramanga1.0
..U.
..STS1.0...U...
.Y...s'.a..SR.h-q.d(.."B....o.(:t...|N.T.RI.P.,.....

Anteriormente se observó la firma del certificado. A partir de este momento todos los mensajes que se intercambian entre el cliente y el servidor son encriptados y por lo tanto no son claramente visibles. Debido a la extensión de la captura realizada, solo se incluirá un fragmento de las conversaciones encriptadas.

.............x{.. .V-..`u...t.m>.cK..U.;.....4K0u2(,....*;n....e.9..2.-.,....d. ......B.O....B.........le.j...*.I.$[...z
..Z'y..K.+@.c.
..5.............(.&y.9]{.......d.;n..............h.......
..........(.
........o.....6......{...
........82..^A....?!......c.............Q..di.o..r!.....5X..6=U.4#...g.{..\..>.W.F.....=.Ej;(M.O.2.O...$.I6R.......8...w.@....\I....
1G...?..NP.|H...K.4..&#..L.8...D;.AO......H/.K]X..^.%.>......v.v.p!.....=+N-.$?.[G3.x..C.`..$N..j~(.(.d..s....X..sh^....?.{Ra.
......L.g....n|....Q.|....&zt........A....6...n.<._<w...
.U.g...@g...{.^|y..1K..I.Yzm9...9~..7...(mV.T.C.I.G...JNp..F.j}.R.....
.......J.....U0 PY$..%"`j...........lzh ......s.:..9.._....p..s3"...I^..eu...m.m......P.*m6....?:...<..);S.......I...%.m.-
^7.h.ie*c....I..w/..o.7......7.........{AlN...rIJ.....u.aMII..w..*A.9b=..|?.|~G.
......{..-...bH=
.....4)p.%.*."...;.I7,.Y...2D.......M.S*6m.E...`.^......3!..q.m.... [..'1\G......R...Z.....qbH.._.V......T.7.............#...W..}0..U....Zf.n.....

De esta forma, hemos garantizado conexiones seguras entre los clientes y el servidor.

CONCLUSIONES

Después de consultar las experiencias y opiniones de diferentes autores, todos coinciden que un servicio de Mensajería Instantánea al menos debe garantizar confidencialidad y seguridad en la Información. Es difícil crear unas políticas generales para todas las organizaciones, pues cada una de ellas tiene necesidades particulares; por el contrario, nuestro trabajo no pretende ser un decálogo de la Mensajería Instantánea Empresarial, pero si puede tomarse como un buen punto de partida para que cualquier administrador de red pueda implementar un servicio propio para su empresa.

Partiendo del supuesto de que Jabber v.2.0 podría servir para implementar un servicio ligero, seguro y de bajo costo de Mensajería Instantánea empresarial, de acuerdo con nuestra experiencia pudimos determinar que este protocolo/servicio se ajusta muy bien a los requerimientos siempre y cuando la necesidad de la empresa sea estrictamente las comunicaciones entre sus miembros. Ningún servicio de Mensajería Instantánea es seguro si personas ajenas a la organización pueden tener acceso a él. El costo de implementación de Jabber v.2.0 es relativamente bajo si se tiene en cuenta que su distribución es libre, ya que únicamente podría presentarse el caso en que deba pagarse a un tercero por su instalación y configuración.

Para ofrecer sus características, Jabber v.2.0 se apoya de otras aplicaciones de distribución libre. En nuestra experiencia, se recomienda la utilización de MySQL como paquete de almacenamiento de datos, pues cuenta con el mejor desempeño y soporte. De igual manera, es una buena práctica delegar la autenticación de usuarios a PAM, ya que se integra fácilmente a otros mecanismos e impiden la

creación de nuevas cuentas o cambiar claves desde los usuarios. Para garantizar la seguridad en las conexiones entre los clientes y el servidor, de manera que la autenticación de los usuarios y el intercambio de mensajes no sean claramente visibles al hacer una captura y posterior decodificación de paquetes, Jabber v.2.0 requiere de los servicios de OpenSSL. Es necesario especificar un certificado digital en la configuración del servicio Jabber v.2.0, previamente creado mediante OpenSSL, para codificar la toda la información que intercambian los usuarios con el servicio.

Los beneficios que comprende la Mensajería Instantánea para las empresas incluyen la capacidad de dotar a los empleados de herramientas tecnológicas con el propósito de mejorar la comunicación interna y mantener un diálogo en tiempo real que reduzca el tiempo de resolución de problemas. La Mensajería Instantánea empresarial dejará de ser una herramienta aislada para venir cada vez más integrada en las aplicaciones de las organizaciones, con el fin de conseguir que la mensajería de datos interpersonal entre empresas y consumidores sea en tiempo real; el software de Mensajería Instantánea empresarial, debe incluir características de seguridad, auditoria e integración con los diferentes canales de comunicación.

BIBLIOGRAFÍA

- FAJARDO C., Daniel. Mensajería instantánea: El nuevo modelo de comunicación personal. 22 de Enero de 2007. <url:http://www.edicionesespeciales.elmercurio.com/destacadas/detalle/index.asp?>.

- RODRÍGUEZ, Diana. La Mensajería Instantánea sirve para trabajar y estudiar? 4 de enero de 2007. <url:http://biblioticando.blogspot.com>.

- El Observatorio de Internet Móvil –OIM- Mensajería Instantánea: el fin de los mensajes cortos? 25 de octubre de 2006. <url:http://jlarienza.blogspot.com/2006/10/mensajeria-instantanea-el-fin-de-los.html>.

- VELÁSQUEZ, Eugenio. Políticas para el uso de Mensajería Instantánea Empresarial. 11 de Agosto de 2006. <url:www.ehui.com>

- <url:http://foros.monografias.com/archive/index.php>

- CECCONI, Paola. Especial para 3 puntos.com. La mensajería instantánea busca su perfil empresarial, correspondencia en tiempo real. <url:http://www.3puntos.com/seccion.php3>

- SERRANO M., GIOVANNI E. REVISTA ENTER, Mensajería instantánea: que no sea un problema... Septiembre 5 de 2005. <url:http://www.symantec.com/es/es/about/news/release/article.jsp>

- J.A.Llorente & O.Cuenca, Déborah Rosano. Las tecnologías de colaboración serán decisivas para el desarrollo de la industria en los próximos tres años. Reuters Latam. Angeles Patrón Costas. <url:http://about.reuters.com/latam/prensa/prensa_gartner.asp>

- Flynn, Nancy. Instant Messaging Rules: A Business Guide to Managing Policies, Security, and Legal Issues for Safe IM Communication.. AMACOM. 2004 ISBN: 0-8144-7253-2**.**

- Adams, DJ. Programming Jabber: Extending XML Messaging.. O'Reilly Publishers. January 2002. ISBN: 0-596-00202-5.

- Extensible Message and Presence Protocol: Core. P. Saint-André, Jabber Software Foundation. October 2004.

- W. Kalmishlian, R. Norris. Jabberd v.2.0 Administration Guide.. Creative Commons, 2003.