



Documentación técnica sobre seguridad

Versión 5.0
Septiembre 2009

© KnowGate 2009

© KnowGate 2009 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>
Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

Tabla de Contenidos

<u>SEGURIDAD EN EL SOFTWARE BASE.....</u>	<u>3</u>
ARQUITECTURA DE HIPERGATE	3
SEGURIDAD DE LA INSTALACIÓN BASE	4
ACCESO TELNET/SSH.....	4
PUERTOS	4
BASE DE DATOS POSTGRESQL	4
APACHE TOMCAT	4
FICHEROS EXTERNOS A LA BASE DE DATOS.....	5
<u>AUTENTIFICACIÓN INTERNA DE HIPERGATE.....</u>	<u>5</u>
ESTABLECIMIENTO Y MANTENIMIENTO DE SESIÓN.....	6
AUDITORÍA DE LOGINS.....	7
DOMINIOS Y ÁREAS DE TRABAJO POR DEFECTO	7
<u>SISTEMA DE PAGOS ONLINE DE EOI.....</u>	<u>¡ERROR! MARCADOR NO DEFINIDO.</u>

Seguridad en el software base

Arquitectura de hipergate

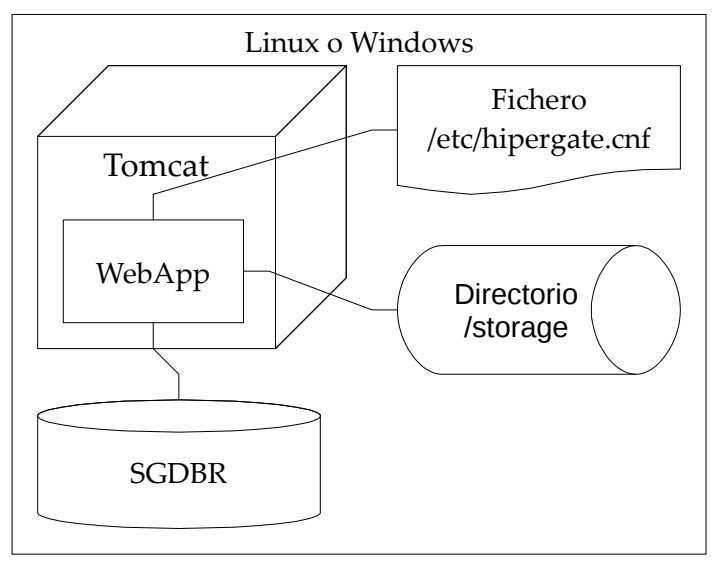
hipergate es una aplicación JSP 100% Java puro que funciona sobre Tomcat u otro contenedor de servlets en Linux o Windows.

Los datos se almacenan principalmente en una base de datos relaciona que puede ser MySQL, PostgreSQL, Oracle o Microsoft SQL Server.

Los documentos adjuntos de gran tamaño se almacenan fuera de la base de datos en un directorio aislado de la webapp principal de Tomcat.

Los parámetros de conexión al SGBDR se almacenan por defecto en el fichero /etc/hipergate.cnf (en Linux) o C:\Windows\hipergate.cnf

No se requiere una cuenta con privilegios especiales para ejecutar hipergate.



Seguridad de la instalación base

Acceso Telnet/SSH

Para prevenir ataques por fuerza bruta se recomienda deshabilitar el acceso a la máquina mediante usuario y contraseña y restringirlo a la identificación mediante claves SSH.

También se recomienda restringir las direcciones IP desde las cuales se tiene acceso por SSH a la máquina.

Puertos

Para que el producto funcione es preciso que la máquina tenga abiertos los puertos 80, 8080 (Tomcat) 110 (POP3) 25 (SMTP) 465 y 995 (Google).

La máquina debe ser capaz de acceder a URLs externas de Internet mediante http.

Base de datos PostgreSQL

La configuración de PostgreSQL 8.4 es la que hace por defecto el autoinstalador del producto estándar. El acceso a la base de datos se realiza mediante el usuario `postgres` y una contraseña de alta fortaleza.

La contraseña de acceso a la base de datos se almacena por defecto en el fichero `/etc/hipergate.cnf`

Apache Tomcat

La configuración de Tomcat 6.0 es la que hace por defecto el autoinstalador del producto estándar.

En el fichero `web.xml` se ha cambiado la contraseña de shutdown para el puerto 8005.

Además de la seguridad estándar de hipergate, es posible y recomendable restringir mediante autenticación básica del servidor web el acceso a determinados directorios.

En particular **es imprescindible limitar el acceso al subdirectorio `/admin`** de la webapp de Tomcat (o incluso eliminarlo por completo) ya que este

subdirectorío contiene una página llamada `sql.htm` que es un interfaz SQL sin restricciones contra la base de datos.

Ficheros externos a la base de datos

Los ficheros externos a la base de datos se almacenan en el subdirectorío apuntado por la propiedad `storage` del fichero `hipergate.cnf`.

El subdirectorío `/storage` debe estar siempre fuera de la webapp de Tomcat.

Los ficheros de `/storage` pueden ser descargados mediante el servlet `com.knowgate.http.HttpBinaryServlet` que verifica el e-mail y contraseña del usuario final de hipergate que está solicitando el acceso al fichero.

Autenticación interna de hipergate

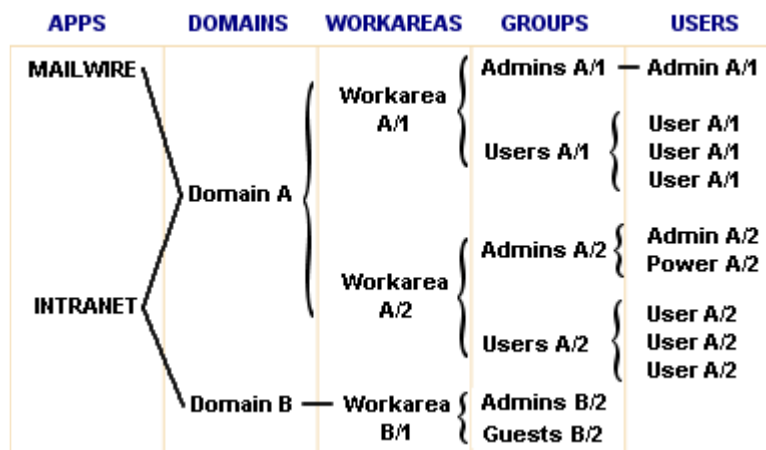
Los usuarios y contraseñas de hipergate se almacenan en la tabla `k_users` de la base de datos. En esta tabla se encuentran las contraseñas de acceso en texto claro para todos los usuarios de la aplicación.

El modelo de datos de hipergate es internamente *multitenant*. Los datos se separan en Dominios que a su vez contienen Áreas de Trabajo. Conceptualmente, se puede visualizar el Dominio como una empresa aislada en un compartimento estanco dentro de la base de datos compartida con otras empresas y el Área de Trabajo es como un departamento de dicha empresa.

Cada usuario de la aplicación pertenece a un único de Dominio pero puede tener acceso a varias Áreas de Trabajo.

Para cada Área de Trabajo existen 4 roles (o Grupos) de usuarios: Administradores, Superusuarios, Usuarios e Invitados. Lo que puede hacer cada uno de estos roles depende de la implementación a medida de cada parte de la aplicación en un proyecto dado.

Independientemente de los Dominios, existe en otra dimensión una lista de Aplicaciones. El sistema de acceso otorga autorización a un Grupo de Usuarios para ver (o no) una determinada aplicación de un Área de Trabajo.



Establecimiento y mantenimiento de sesión

La sesión de hipergate se mantiene mediante cookies. Las credenciales en estas cookies se verifican en cada petición HHTP.

Hay 5 cookies implicadas en el proceso de autenticación.

domainid: es el identificador numérico del Dominio actual.

workarea: es el GUID (Global Unique IDentifier) del Área de Trabajo actual.

userid: es el GUID (Global Unique IDentifier) del Usuario actual.

authstr: es la contraseña de acceso de el usuario. La contraseña está encriptada mediante un algoritmo RC4 implementado en la clase java com.knowgate.acl.RC4 La clave de encriptación para este algoritmo se encuentra en el fichero /WEB-INF/classes/com/knowgate/acl/acl.cnf perteneciente a las librerías de la webapp

appmask: máscara de 32 bits acceso a las aplicaciones, cada aplicación está representada por 1 bit.

Cuando se recibe una petición de página JSP, el método authenticateCookie() del include /methods/authusr.jspf hace lo siguiente:

1. Lee userid y authstr, desencripta la contraseña, verifica que el usuario existe, la cuenta está activada, la contraseña no ha caducado y coincide con la almacenada en el campo tx_pwd de la tabla k_users. La contraseña puede estar cacheada en memoria

© KnowGate 2009 Esta documentación se distribuye bajo la licencia Creative Commons Attribution-NoDerivs-NonCommercial. <http://creativecommons.org/licenses/by-nd-nc/1.0/>

Se permite copiar, redistribuir y modificar el documento sólo según los siguientes términos: 1º) Debe aparecer la atribución original a KnowGate. 2º) No se permite el uso del original ni ninguna modificación con fines comerciales. 3º) No se permiten trabajos derivados basados en esta documentación. 4º) Cualquier redistribución debe contener estos términos.

RAM del servidor web para evitar constantes accesos a la tabla k_users.

2. Verifica que el usuario pertenece al dominio especificado.
3. Verifica que el usuario está incluido en algún Grupo de Usuarios que tiene acceso a el Área de Trabajo actual.
4. Verifica que la máscara de acceso a las aplicaciones para el usuario coincide con la almacenada (o cacheada) en el servidor.

Si el proceso de autenticación falla, el sistema redirige al usuario a la página /common/errmsg.jsp

Auditoría de logins

Cada vez que un usuario se intenta conectar a la aplicación, tanto si tiene éxito como si no, se deja un registro en la tabla k_audit_logins con la fecha+hora, dirección IP de acceso, y, si falló el intento, la contraseña errónea que se especificó.

El sistema puede configurarse para bloquearse automáticamente tras N intentos pero la instalación por defecto no lo hace.

Dominios y áreas de trabajo por defecto

hipergate trae 5 dominios por defecto: SYSTEM, MODEL, TEST, DEMO y REAL.

Hay que cambiar siempre las contraseñas por defecto de todos los usuarios de los dominios SYSTEM y MODEL.

Se recomienda también borrar o deshabilitar los dominios de trabajo por defecto TEST, DEMO y REAL, y usar otros nuevos, aunque esto no es imprescindible.