SEGURIDAD

OCTUBRE 2015

Versión 1

1 INDICE

2 INTRODUCCIÓN

2.1 Requisitos

Antes de seguir esta guía se recomienda leer la guía **SOFIA2-Conceptos SOFIA2.doc**

2.2 Objetivos y alcance del presente documento

Esta guía describe los mecanismos de seguridad que aplican a Sofia2.

La plataforma Sofia2 es un entorno de interoperabilidad controlado, que implementa varios mecanismos de seguridad que operan a lo largo de las diferentes capas ofreciendo autorización, autenticación, consistencia y en su conjunto protección integral de los datos.

La seguridad en la plataforma Sofia2 es implementada a través del mecanismo de plugins que permiten realizar una personalización total de la Autenticación y la Autorización, permitiendo implementar mecanismo basados en diferentes estándares (LDAP, BD, Oauth…).

La Seguridad cubre las diferentes capas del aplicativo:

- **Las Comunicaciones**: Todos los Gateway implementados en la plataforma permiten la securización del canal a través del uso de Certificados SSL.
- **Los Clientes:** La plataforma solo permite la conexión de aquellos clientes que han sido autorizados en la plataforma.
- **Las Operaciones**: La plataforma solo permite realizar las operaciones para las que se tiene permiso.
- **La Información**: La plataforma solo te permite interactuar con la información para la que has sido autorizado, y valida que la información que se persiste en la plataforma cumple con el modelo de la información esperado.

3 Seguridad en las Comunicaciones

Sofia2 utiliza el canal seguro de comunicaciones https para todas las comunicaciones con el exterior. Este canal utiliza un protocolo SSL/TSL con un cifrado estándar RSA que necesita de un certificad de seguridad X.509. Es el estándar de cifrado más extendido y aceptado en el mundo del www.

La clave privada se instala en el SIB Sofia2 y su parte pública (conocida como certificado) deberá instalarse en los navegadores o ser incorporado en las apps que necesiten acceder a la plataforma desde la extranet. La seguridad mediante certificados prevendrá filtrado o manipulación de datos entre cliente y servidor mediante ataques del tipo man-in-the-middle.

Hay que destacar que el sistema de autenticación será mediante certificado de servidor, no de cliente-servidor (como podrían ser los certificados de la fnmt). Esta tipo de encriptado garantiza que el cliente se está conectando con el servidor requerido y no a un posible servicio impostor, pero no autentica al cliente.

4 Los Clientes

Un usuario deberá registrar en la plataforma sus KPs (Cliente), de lo contrario, la plataforma rechazará la conexión de los mismos.

Para ello necesitará de un Token de Autenticación que el usuario podrá crear asociado al KP (según las restricciones de su rol). Los usuarios con rol Colaborador o rol Usuario únicamente podrán dar de alta tokens para KP's de los que sean propietarios.

Los KP's tienen también una clave de cifrado que sólo es necesaria si quiero usar XXTEA como protocolo de encriptación. Se utiliza en dispositivos que no soportan HTTPs, como por ejemplo Arduinos.

Un KP podrá hacer uso de una o varias ontologías, siendo esta la información que producirá o consumirá de la plataforma.

Una vez registrado en la plataforma, el KP ya podrá establecer conexiones con la misma.

Tras la creación de un KP podemos dejar definida una instancia KP a través de la Consola Web.

Una instancia KP identifica al cliente que se va a conectar a la plataforma Sofia2.

La conexión de un KP con la plataforma debe ser vista como dos tipos de conexión

- **Conexión Física**: Establecida por el protocolo de trasporte utilizado para la conexión por un KP (TCP/IP, MQTT, WebSockets, REST, WS, JMS, Ajax-Push…). La manera de realizar esta tipo de conexión depende en gran medida del API de KP utilizado (Java, Javascript, Arduino, C++...).
- **Conexión Lógica**: Establecida por el protocolo SSAP (Smart Space Access Protocol) de mensajería definido en SOFIA. Es común a todos los APIs de KP.

Nos vamos a centrar en la seguridad a nivel de conexión Lógica que debe mantener un KP con la plataforma, pues la seguridad a nivel Física es cubierta por la seguridad en la capa de Comunicaciones.

Para que un KP pueda conectarse a la plataforma y producir/consumir datos e interoperar con otros KP, es necesario que abra una sesión con un SIB de la plataforma.

El protocolo SSAP proporciona dos operaciones en este sentido:

- **JOIN**: Donde un KP informa a la plataforma el usuario y password de su propietario así sus datos de instancia, de manera que si todo es correcto, la plataforma

autentica al KP y abre una sesión con el mismo (devolviendo una session key que podemos usar durante un tiempo preestablecido y configurable).

- **LEAVE**: Donde un KP informa a la plataforma que va a cerrar la sesión.

Mientras exista una sesión entre el KP y la plataforma, el KP podrá utilizar el resto de operaciones del protocolo SSAP para producir/consumir información.

5 Seguridad de Acceso

La plataforma está dividida en dos áreas con independencia en su modelo de seguridad.

5.1 La Administración.

En la plataforma se han definido tres tipos de Roles, que definen las funcionalidades que dispondrán los usuarios a nivel administrativo:

- Rol **Administrador**.
- Rol **Colaborador**: este rol permite operar con la información de la plataforma, volcando y consumiento información y crear nuevas estructuras de información además de realizar tareas de procesado de información (Reglas, Script, Informes).
- Rol **Usuario**: este rol permite operar con la información de la plataforma, volcando y consumiento información de estructuras de información existentes en las que ha sido autorizado.

5.2 La Operación.

Los permisos, para los que también existen 3 tipos, definen las funcionalidades de los usuarios a nivel operativo sobre la información.

Estos permisos se aplican a nivel de Ontología – Usuario.

Los **administradores** tienen Permiso Total sobre todas las Ontologías.

Los **Colaboradores** tienen Permiso Total sobre las ontologías de las que son Propietarios:

- **Permiso de Lectura**: Permite a un usuario o colaborador realizar únicamente operaciones de tipo Query sobre las ontologías para las que se le ha asignado este permiso.
- **Permiso de Inserción**: Permite a un usuario o colaborador realizar únicamente operaciones de tipo Insert sobre las ontologías para las que se le ha asignado este permiso.
- **Permiso Total**: Permite a un usuario o colaborador realizar todas las operaciones sobre las ontologías para las que se le ha asignado este permiso.

6 Seguridad en los Datos

Todas las operaciones son validadas a nivel de Autenticación, para lo que la plataforma comprueba si el Cliente se ha autenticado con la plataforma.

Una vez que se ha comprobado la Autenticación del Cliente se comprueba su autorización en dos niveles:

- Primero, se valida que el usuario puede operar con la Ontología para la que quiere realizar la operación.
- Segundo, se valida que la operación (Query, Insert, Delete, Update) que quiere realizar el usuario, la puede realizar sobre esa ontología (Tiene los permisos adecuados).

Si todos los pasos anteriores han sido correctamente comprobados y la operación es Insert o Update todavía se ha de realizar una tercera validación, que consisten en comprobar que la información que se inserta cumple escrupulosamente con el Esquema que se ha definido, a través de la validación del JSON Schema, casando la información que está insertando con la estructura de la Ontología.

7 Implementación de Referencia

La implementación de referencia de la Seguridad está basada en tres plugins:

7.1 plugin-sofia-user

Este plugin (Usado únicamente a nivel de Administración) es el encargado de recuperar la información de los usuarios. En la implementación de referencia la recupera de la base de datos de configuración.

Tiene la capacidad de trabajar con Password encriptada o en claro, permitiendo configurar el Algoritmo de encriptación.

```
public void persist(Usuario user) throws NotImplementedException;

public void remove(Usuario user) throws NotImplementedException;

public void merge(Usuario user) throws NotImplementedException;

public long countUser() throws NotImplementedException;

public List<Usuario> findAllUser() throws NotImplementedException;

public Usuario findUser(String idUsuario) throws NotImplementedException;

public List<Usuario> findUsers(String qlString, List<Object> parametros)
throws NotImplementedException;

public Usuario findLoginUser(String identificador, String credential, String
sourceInfo) throws EmptyResultDataAccessException;

public List<Usuario> findUserByCriteria(Usuario usuario) throws
NotImplementedException;

public List<Usuario> findUserByIdentificacion(String identificacion) throws
EmptyResultDataAccessException;
```

7.2 plugin-console-security

Este plugin es el encargado de gestionar la Autenticación y Autorización en la consola de Administración y se basa en Spring Security. En la implementación de referencia hace uso de la base de datos de configuración.

Hace uso de plugin-sofia-user para recuperar la información de los usuarios.

7.3 plugin-sib-security

Este plugin es el encargado de gestionar la Autenticación y Autorización a las operaciones del SIB y está basado en un mecanismo de Token – SessionKey.

Hace uso de plugin-sofia-user para recuperar la información de los usuarios.

Este plugin debe cumplir con el siguiente interface:

```
String authenticate(SSAPMessage message) throws AuthenticationException;

void checkSessionKeyActive(String sessionKey) throws AuthenticationException;

void closeSession(String sessionKey) throws AuthenticationException;

public void removeAuthenticationContextSessionkey(String sessionKey) throws
AuthenticationException;

void checkAuthorization(SSAPMessageTypes operationType, String ontologyName,
String sessionKey) throws AuthorizationServiceException;

void checkAuthorizationConfig(SSAPMessageTypes operationType, String
tableName, String sessionKey) throws AuthorizationServiceException;

void checkAuthorization(SSAPMessageTypes operationType, String kpName, String
instanceKpName, String token) throws AuthorizationServiceException;
```