

ANEXO IV

ESPECIFICACIONES TÉCNICAS PARA LA ELABORACIÓN DE UN CLIENTE QUE PUEDA SOLICITAR Y OBTENER ACCESO A LOS SERVICIOS WEB PUBLICADOS POR LA DIRECCIÓN NACIONAL DE ADUANAS, MEDIANTE LA UTILIZACIÓN DEL WEB SERVICE DE AUTENTICACIÓN Y AUTORIZACIÓN (WSAA).

ESPECIFICACIONES TÉCNICAS PARA LA ELABORACIÓN DE UN CLIENTE QUE PUEDA SOLICITAR Y OBTENER ACCESO A LOS SERVICIOS WEB PUBLICADOS POR LA DIRECCIÓN NACIONAL DE ADUANAS, MEDIANTE LA UTILIZACIÓN DEL WEB SERVICE DE AUTENTICACIÓN Y AUTORIZACIÓN (WSAA).

Introducción

Propósito

Describir las especificaciones técnicas para la elaboración de un cliente que pueda solicitar y obtener acceso a los Servicios Web publicados por la DNA mediante la utilización del Web Service de Autenticación y Autorización (WSAA).

Especificación

Descripción General del Servicio

El Web Service de Autenticación y Autorización (WSAA) es un servicio Business to Business (B2B) que permite a dos computadores (cliente / servidor) en una red insegura demostrar su identidad mutuamente en forma segura. El servidor brinda dos servicios al cliente:

- Autenticación mediante la firma de mensajes utilizando criptografía de clave pública (PKI).
- Autorización para la invocación de un Servicio Web mediante la expedición de Ticket de Acceso.
- En dicha tarea intervienen los siguientes componentes:
- Un cliente que solicita acceso a un Servicio Web.
- El Servidor WSAA, publicado por la DNA (Dirección Nacional de Aduanas), que implementa la autenticación y autorización de los computadores.

Al usar especificaciones y protocolos estándares (PKI, XML, CMS, WSDL y SOAP) el cliente puede ser desarrollado con cualquier lenguaje de programación moderno.

Para que un cliente pueda utilizar efectivamente un Servicio Web, deberá solicitar al WSAA un "Ticket de Acceso" (TA). Dicha solicitud se realiza mediante el envío de un "Ticket de Requerimiento de Acceso" (TRA), mediante mensajería SOAP.

El WSAA realiza la verificación del TRA y si la solicitud es correcta, devuelve un mensaje que contiene el TA que habilita al Cliente a utilizar el Web Service solicitado. Una vez que Cliente obtiene el TA, el mismo debe utilizarlo para acceder al Web Service en cada solicitud que realice.

En la actualidad, los Web Services, no están incluidos en un UDDI (Universal DescriptionDiscoveryIntegration) de acceso externo, por lo tanto para acceder a los servicios ofrecidos, es necesario utilizar el WSDL publicado en una URL

definida por la DNA. A partir del WSDL el desarrollador puede construir un Cliente para poder consumir el Servicio Web correspondiente.

Toda esta comunicación se realiza utilizando el protocolo HTTP sobre SSL (HTTPS). La DNA proporciona un certificado digital a cada cliente que desea invocar el servicio del WSAA.

Para el correcto funcionamiento de WSAA, todos los computadores intervinientes en una solicitud deben tener sus relojes sincronizados con algún servidor de hora de Internet.

Flujo Principal

1. El cliente crea un TRA con los siguiente datos:
 - a. source: El DN del certificado X509 del Cliente.
 - b. destination: El DN del servidor WSAA.
 - c. uniqueId: Un entero de 32 bits generado en forma aleatoria.
 - d. generationTime: La fecha y hora de generación del Ticket de Requerimiento.
 - e. expirationTime: La fecha y hora de expiración del Ticket de Requerimiento.
 - f.service: El nombre del servicio para el cual se solicita un Ticket de Acceso.
2. Firma el TRA creado con su clave privada. Se utiliza RSA+SHA1.
3. Incluye el mensaje y su firma en un mensaje CMS, junto con su certificado X509 que le fue proporcionado por la DNA.
4. Obtiene la notación ASN1 del mensaje CMS creado.
5. Codifica el mensaje ASN1 a Base 64.
6. Invoca el WebServiceLoginCms del WSAA pasándole como parámetro en mensaje ASN1 codificado a Base 64.
7. Obtiene los siguientes campos del Ticket de Acceso (TA)
 - a. source: El DN del certificado X509 del Servidor WSAA.
 - b. destination: El DN del certificado del Cliente que invocó el Web Service.
 - c. uniqueId: Un entero de 32 bits generado en forma aleatoria.
 - d. generationTime: La fecha y hora de generación del Ticket de Acceso.
 - e. expirationTime: La fecha y hora de expiración del Ticket de Acceso.
 - f.token: El token de acceso al servicio solicitado.
 - g. sign: La firma digital del token de acceso. El token fue firmado con la clave privada del Servidor WSAA.
8. Verifica que el TA recibido cumpla con el esquema XSD especificado en este documento.
9. Verifica que el token haya sido firmado por el Servidor WSAA, para ello se utiliza la clave pública del servidor contenida en su certificado X509.

Obs. Los formatos de Fecha y Hora deben cumplir con la especificación de la clase `com.sun.org.apache.xerces.internal.jaxp.datatype.XMLGregorianCalendarImpl`.

Al convertir a cadena la Fecha y Hora adopta el siguiente formato:
yyyy-mm-ddThh:mm:ss.sss-GTM time zone (-03:00)
2007-10-29T13:04:35.975-03:00

Obs. Al obtener el TA, el cliente tiene acceso al servicio para el cual el TRA fue generado. Cada vez que el cliente desea invocar dicho servicio, debe incluir en la invocación los datos token y sign. Antes de invocar un servicio, el Cliente debe verificar que el ticket no esté vencido.

Esquemas

Esquema TRA

La construcción del documento XML TRA deberá ajustarse al siguiente esquema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
<xsd:documentation xml:lang="es">
Esquema de Ticket de pedido de acceso a un Web Service.
</xsd:documentation>
</xsd:annotation>
<xsd:element name="loginTicketRequest" type="loginTicketRequest" />
<xsd:complexType name="loginTicketRequest">
<xsd:sequence>
<xsd:element name="header" type="headerType" minOccurs="1" maxOccurs="1"/>
<xsd:element name="service" type="serviceType" minOccurs="1" maxOccurs="1"/>
</xsd:sequence>
<xsd:attribute name="version" type="xsd:decimal" use="optional" default="1.0" />
</xsd:complexType>
<xsd:complexType name="headerType">
<xsd:sequence>
<xsd:element name="source" type="xsd:string" minOccurs="1" maxOccurs="1"/>
<xsd:element name="destination" type="xsd:string" minOccurs="1" maxOccurs="1"/>
<xsd:element name="uniqueId" type="xsd:unsignedInt" minOccurs="1"
maxOccurs="1"/>
<xsd:element name="generationTime" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
<xsd:element name="expirationTime" type="xsd:dateTime" minOccurs="1"
maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="serviceType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="[a-z][a-z,\-,\_,0-9]*/>
<xsd:minLength value='3'/>
<xsd:maxLength value='32'/>
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

Esquema TA

La construcción del documento XML TA deberá ajustarse al siguiente esquema:

```
<?xml version="1.0" encoding="UTF8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:annotation>
<xsd:documentation xml:lang="es">
```

Esquema de Ticket de respuesta al pedido de acceso a un Servicio Web.

```
</xsd:documentation>
</xsd:annotation>
<xsd:element name="loginTicketResponse" type="loginTicketResponse" />
<xsd:complexType name="loginTicketResponse">
<xsd:sequence>
<xsd:element name="header" type="headerType" minOccurs="1" maxOccurs="1" />
<xsd:element name="credentials" type="credentialsType" minOccurs="1"
maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="version" type="xsd:decimal" use="optional" default="1.0" />
</xsd:complexType>
<xsd:complexType name="headerType">
<xsd:sequence>
<xsd:element name="source" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="destination" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="uniqueId" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"
/>
<xsd:element name="generationTime" type="xsd:dateTime" minOccurs="1"
maxOccurs="1" />
<xsd:element name="expirationTime" type="xsd:dateTime" minOccurs="1"
maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="credentialsType">
<xsd:sequence>
<xsd:element name="token" type="xsd:string" minOccurs="1" maxOccurs="1" />
<xsd:element name="sign" type="xsd:string" minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Ejemplos

Ejemplo de mensaje TRA.

```
<?xml version="1.0" encoding="UTF-8"?>
<loginTicketRequest version="1.0">
<header>
<source>CN=malvarez, O=dna, C=py</source>
<destination>SERIALNUMBER=PY 800292227, CN=WSAA, O=DNAPY,
C=PY</destination>
<uniqueId>1193670228</uniqueId>
<generationTime>2007-10-29T12:03:48.890-03:00</generationTime>
<expirationTime>2007-10-29T13:03:48.875-03:00</expirationTime>
</header>
<service>test</service>
</loginTicketRequest>
```

Ejemplo de mensaje TA.

```
<?xmlversion="1.0" encoding="UTF-8"?>
<loginTicketResponseversion="1.0">
<header>
<source>SERIALNUMBER=PY 800292227, CN=WSAA, O=DNAPY, C=PY</source>
<destination>CN=malvarez, O=dna, C=py</destination>
<uniqueId>1193670275</uniqueId>
<generationTime>2007-10-29T12:04:35.975-03:00</generationTime>
<expirationTime>2007-10-29T13:04:35.975-03:00</expirationTime>
</header>
<credentials>
<token>VG9rZW4gZGUgcHJ1ZWJhC0gVG9rZW4gZGUgcHJ1ZWJh</token>
<sign>p4ozVJFqLOqhNyyMPNPOM3GAJh/OYIER8d8IEAH0octwKjAazwhC1npQrN
RJKKMjux2mbZhaPOVLJksPuN8GPRqkCTY/qYog1ShD3iS7dw9ENBcHbq+z
K0Da1HlyjQ5TxI6R/9XNULzp7kJM0DaSI+jgtZjcx1FNHkA0ejK2ckU=
</sign>
</credentials>
</loginTicketResponse>
```

Obtener Claves

- 1- Genere su propia clave privada ejecutando el siguiente comando:
 - a. opensslgenrsa 2048 > pkey.pem
- 2- Genere su certificaterequest (ATENCIÓN: Ingrese solo los campos: País, Compañía y ComonName)
 - a. openssl req -new -key privada -out myreq.pem
- 3- Emita el archivo myreq.pem al departamento de seguridad informática de la DNA.
- 4- La DNA le retorna el archivo newcert.pem. Su nuevo certificado firmado por una CA de confianza.
- 5- Exporte su nuevo certificado y su clave privada a un archivo pkcs12.
 - a. openssl pkcs12 -export -in newcert.pem -inkey pkey.pem -name unalias -out clientkstore.p12
- 6- Copie el archivo clientkstore.p12 a un lugar accesible por su cliente.
- 7- Utilice el certificado y la clave privada contenidos en el archivo clientkstore.p12 para generar una Solicitud de Ticket de Acceso.