Informe de prácticas de ISDCM

Entrega 05

Alumno/a: Cristian Matas

Alumno/a: Pavel Khralovich

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

FIB

Contenido

Introducción 1

Ejercicio 1 1

Ejercicio 2 ..... **Error! Bookmark not defined.**

Ejercicio 5 1

Decisiones de diseño 2

Repositorios de código consultados 4

Bibliografía consultada 4

Introducción

El objetivo de esta práctica es entrar en detalle en el funcionamiento del protocolo de autorización XACML a través de dos herramientas distintas, así como poder ver el funcionamiento del sistema de firma de documentos XML Signature. Para ello, hemos planteado un proceso que consta de tres partes:

1. Analizar las políticas y las peticiones de muestra que se nos han proporcionado para poder hacer los ejercicios 1 y 2.
2. Plantear una aplicación que funciona por comandos para poder trabajar sobre los ejercicios 3, 4 y 6.
3. Comparar las herramientas Sun y Balana para poder completar el ejercicio número 5.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

FIB

Ejercicio 1

Policy/Preguntas	Pregunta_1	Pregunta_2	Pregunta_3	Pregunta_4
Policy_1	Cualquier usuario	urn:mvideo:Mavericksv.mp4 (línea 14)	play (línea 23)	Que la fecha sea menos o igual a 2015-01-01 (de línea 30 a 36)
Policy_2	Usuarios Premium	urn:mvideo:Baztan.mp4 (línea 20)	copy (línea 30)	Que el número de copias sea menor que 4 (línea 39 a 47)
Policy_3	Cualquier usuario	urn:bbc:mdocum:Planets.mp4 (línea 14)	playDocumental (línea 23)	Que la localización del usuario sea el Reino Unido(línea 31 a 36)

Ejercicio 2

Request/Preguntas	Pregunta_1	Pregunta_2	Pregunta_3	Pregunta_4
Request_1	bobRay	urn:mvideo:Mavericksv.mp4	playDocumental	21/09/2013
Request_2	bobRay -> noPremium	urn:mvideo:Baztan.mp4	Copy	Countcopy == 1
Request_3	bobRay	urn:bbc:mdocum:Planets.mp4	playDocumental	Reino Unido
Request_4	leoVergara	urn:bbc:mdocum:Planets.mp4	playDocumental	Italia
Request_5	leoVergara -> premium	urn:mvideo:Baztan.mp4	Copy	Countcopy == 1

Ejercicio 5

En lo que respecta al uso de las dos herramientas de autorización, consideramos que en lo que respecta a la complejidad, Balana es mucho más apropiada para el desarrollo. Hacemos esta afirmación por los siguientes motivos:

- Hemos podido añadir fácilmente la referencia a este entorno con Maven, cosa que no ha sido posible son Sun.
- El número de líneas de código es bastante menor con Balana, con prácticamente la mitad de líneas. Además, creemos que en general toda la parte de configuración es mucho más sencilla de entender gracias al menor número de líneas y de clases que hay que utilizar.

Para poder medir la velocidad, hemos creado un proceso que aplica la misma política y petición un total de 1000 veces, midiendo el tiempo total de ejecución en nanosegundos. El resultado que hemos obtenido de este análisis es que la ejecución de Sun es aproximadamente un 20% más rápida que la de Balana para la ejecución del mismo trabajo.

Todo esto se ha podido comprobar mediante la creación de un Script de Bash donde se comprueba el resultado para las dos políticas diferentes de las 5 request y 3 policies dadas por el enunciado. Así pues, se ha comprobado manualmente que todos los resultado son los esperados, así como la creación de la firma y el test de velocidad. Todo esto se puede encontrar dentro del directorio: “ISDCM_5/test/”.

Decisiones de diseño (Ejercicio 3 y 4)

Para el ejercicio 3 y 4 hemos decidido crear una única aplicación Java que se debe ejecutar por consola de comandos. Esta aplicación recibe un conjunto de parámetros y en función de ellos realiza las acciones pertinentes a partir de las especificaciones del enunciado. Remarcar también que esta aplicación también incluye el ejercicio número 6, que se encarga de realizar una firma de un archivo XML con el estándar XML Signature.

En todos los casos, la aplicación saca por consola el resultado de la validación, ya sea el resultado de ejecutar una petición en caso de que se indique una acción relacionada con el protocolo XACML, el resultado de la firma en caso de XML Signature o el tiempo de ejecución en el caso de ejecutar un test.

En concreto, los posibles parámetros que se pueden especificar se recogen en la Tabla 1 que podemos ver a continuación.

Parámetro	**Valores**	**Comentarios**
-m	Motor o acción por realizar. Permite los valores sun y balana para la aplicación de XACML o xmldsig si se quiere firmar un archivo.	Valor obligatorio.
-p	Path a la política o al archivo XML a firmar.	Valor obligatorio.
-r	Path de la petición a ejecutar.	Aplicable solo en caso de que -m sea sun o balana.
-t	Indica si se debe ejecutar el modo test. Ejecuta 1000 iteraciones con el motor, política y petición especificados para poder valorar la velocidad del algoritmo.	Aplicable solo en caso de que -m sea sun o balana.

Tabla 1. Parámetros del programa

En lo que respecta al conjunto de clases desarrolladas para conseguir un correcto funcionamiento de esta aplicación por consola, hemos decidido la siguiente arquitectura que describimos a continuación. También podemos ver un diagrama de clases en la Figura 1. El conjunto de clases diseñadas es:

- **Interfaz BaseXACML:** contiene un método que permite realizar la verificación de una petición respecto al estándar XACML. El método de verificación devuelve el resultado en forma de cadena de text.

- **Clase BalanaXACML:** implementa la interfaz BaseXACML y contiene los métodos necesários para poder realizar la validación de una petición respecto a una política que se debe indicar en el constructor.
- **Clase SunXACML:** implementa la interfaz BaseXACML y contiene los métodos necesarios para poder realizar la validación de una petición respecto a una política que se debe indicar en el constructor.
- **XmlDsig:** clase que implementa la firma según el estándar XML Signature. Para su utilización, se debe crear una instancia y posteriormente llamar al método sign indicando el path del archivo que se quiere firmar.
- **Main:** clase principal que se encarga de procesar los argumentos introducidos por consola, así como de coordinar el funcionamiento del resto de clases especificadas en los puntos anteriores.

En lo que respecta al código, hemos consultado ejemplos para conocer que clases se deben usar en cada caso. Posteriormente, hemos construido nuestro programa según la arquitectura que hemos creído más conveniente.

En el caso de Balana, hemos podido añadir la referencia a la librería a través del entorno Maven, facilitando así su integración. En el caso de Sun, hemos tenido que añadir directamente el código fuente al proyecto, ya que no hemos conseguido que funcione ni con Maven ni añadiendo el Jar directamente al proyecto. Además, para conseguir que compile correctamente hemos tenido que añadir otra referencia que contenía clases para trabajar con XPath. Para la firma de XML no hemos necesitado añadir ningún tipo de referencia externa.

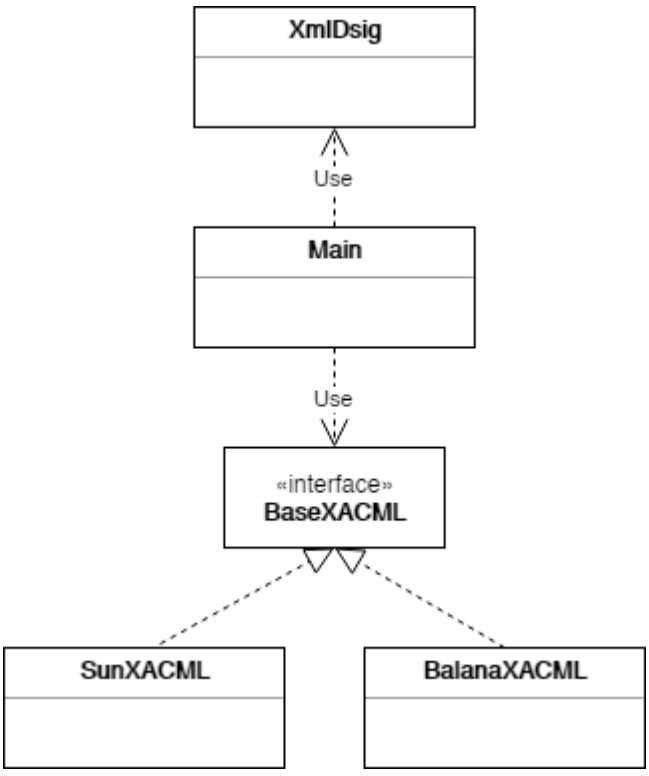

Figura 1. Diagrama de clases

Por último, para poder realizar el testeo fácilmente del código desarrollado, hemos creado un archivo de Bash para Windows, que se encarga de probar todas las combinaciones posibles a partir de las políticas y de las peticiones proporcionadas con el enunciado.

Repositorios de código consultados

Como ya hemos dicho, el código que hemos desarrollado es totalmente propio a partir de la información de las clases que hemos encontrado. Los ejemplos de código que hemos usado son:

- **XML Signature:** https://docs.oracle.com/javase/8/docs/technotes/guides/security/xmldsig/XMLDigitalSignature.html

- **Balana:** https://docs.oracle.com/javase/8/docs/technotes/guides/security/xmldsig/XMLDigitalSignature.html

- **Sun:** en este caso hemos usado el código de ejemplo que podemos encontrar en el código fuente proporcionado para la práctica. En concreto, hemos usado el que se encuentra en la ruta sunxacml-1.2\sunxacml-1.2\sample\src\SimplePDP

Bibliografía consultada

Hemos consultado el material de las clases de teoría y también las referencias indicadas en el enunciado de la práctica, por lo que no adjuntamos los enlaces a la bibliografía.