Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
2014-20

Caso de Estudio (Caso 2) – Canales Seguros
Sistema de Apoyo a las Aplicaciones Misionales de una Entidad Oficial

Objetivos

- Identificar los requerimientos de seguridad de los canales usados para transmisión de la información en el sistema de apoyo a las aplicaciones de la entidad administradora de pensiones.
- Construir un prototipo a escala del sistema que permita satisfacer algunos de los requerimientos de seguridad identificados. Entendiendo las garantías de seguridad y las limitaciones de la implementación propuesta.

Descripción General:

En esta segunda parte, su tarea es determinar los requerimientos de seguridad para la construcción de algunos componentes del sistema de tecnologías de información que soporta las operaciones de la entidad administradora de pensiones.

Problemática:

Como se indicó en el enunciado del caso, las principales tareas del sistema son el manejo de afiliaciones, recaudos, AFE, historia laboral, reconocimiento, nómina de pensionados, tutelas y portal web.

En este contexto, surgen diversos problemas de seguridad para algunas de las transacciones que el sistema soporta, tanto a nivel de transmisión como a nivel de almacenaje de datos. Como consecuencia, es necesario evaluar riesgos y vulnerabilidades y determinar medidas para mitigar los riesgos detectados.

Su tarea en este caso es actuar como consultor y analizar, desde el punto de vista de la seguridad, las tareas relacionadas con el proceso de afiliación.

Tareas:

Suponga que la arquitectura del sistema incluye un cliente en la empresa que maneja las afiliaciones y un servidor en la oficina principal de la entidad de manejo de pensiones. En este servidor se corre el front-end de la aplicación que se encarga de procesar las afiliaciones. El cliente se comunica con el servidor vía internet.

Los desktops de los empleados que trabajan en la oficina central de la entidad de manejo de pensiones están conectados a una subred privada y aislada de la subred en la que correo el servidor. El servidor maneja dos interfaces de red, una para conexiones con los clientes externos, entre los que se cuenta el servidor de la empresa que maneja las afiliaciones, y una para conexiones con los computadores de la red privada interna.

A. [20%] Análisis y Entendimiento del Problema.

Considerando el contexto entregado con el caso 1 y el sistema descrito en el párrafo anterior:

1. Identifique y describa cinco amenazas del sistema. Explique su respuesta en cada caso(*) y responda la pregunta ¿Si la amenaza se consolida, cómo afectaría al sistema?.
2. Identifique cinco vulnerabilidades del sistema, teniendo en cuenta únicamente aspectos técnicos (no organizacionales o de procesos). Identifique vulnerabilidades no solo en lo relacionado con la comunicación sino también con el almacenamiento de los datos. Explique su respuesta en cada caso (*).

(*) *Sus explicaciones DEBEN estar ligadas al contexto del problema planteado e indicar cómo. NO se aceptarán respuestas para contextos genéricos.*

B. [10%] Propuesta de Soluciones.

Para cada una de las amenazas que usted identificó en el punto anterior, proponga mecanismos de resolución/mitigación.

- Los mecanismos propuestos deben ser detallados, por ejemplo, si se habla de cifrado sobre un canal de comunicaciones, debe identificar los participantes en la comunicación, y si es cifrado simétrico o asimétrico (y mencionar las ventajas del tipo de cifrado seleccionado).
- Además, debe justificar los mecanismos propuestos. Es decir, identifique explícitamente cuál riesgo mitiga y cómo.

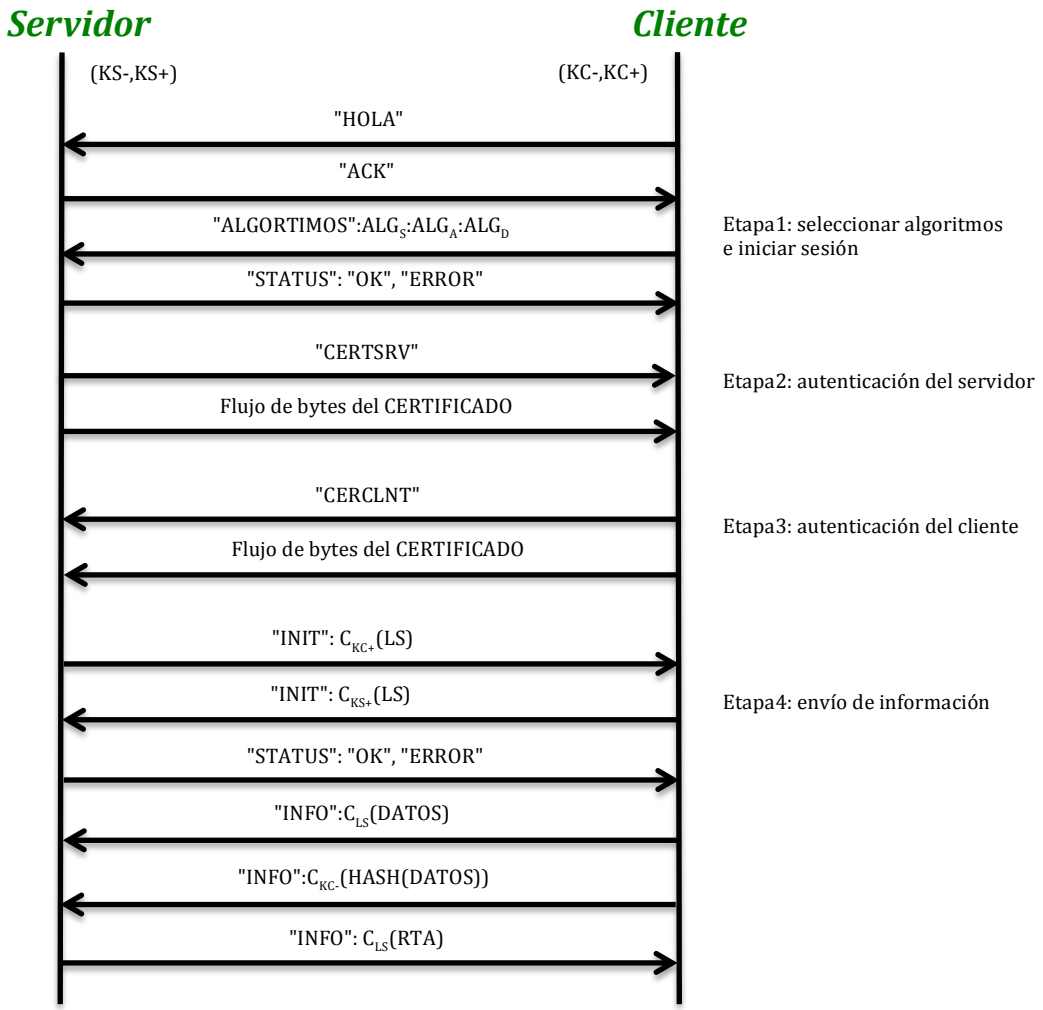
En sus justificaciones tenga en cuenta aspectos relacionados con eficacia, costo, eficiencia, flexibilidad, aspectos de implementación, y otros aspectos técnicos que considere convenientes.

C. [70%] Implementación del Prototipo.

En esta parte del proyecto nos concentraremos en el sistema que recibe y responde a la entidad afiliadora. Usted debe construir la aplicación cliente que la entidad afiliadora correrá para enviar la información al servidor web que maneja la información de las afiliaciones. Como queremos concentrarnos en el protocolo de comunicaciones y sus requerimientos de seguridad, construiremos una aplicación cliente/servidor simplificada en Java, sin usar el protocolo https, que es el más común para correr una aplicación web.

El cliente y el servidor seguirán el siguiente protocolo para comunicarse (la figura ilustra el protocolo):

1. El cliente inicia la comunicación enviando una solicitud de inicio de sesión, a continuación espera un mensaje de confirmación de inicio del servidor.
2. El cliente envía la lista de algoritmos de cifrado que usará durante la sesión y espera un segundo mensaje del servidor confirmando si soporta los algoritmos seleccionados (si no, el servidor corta la comunicación).
3. El servidor envía su certificado digital (CD) para autenticarse con el cliente. El CD debe seguir el estándar X509.
4. El cliente envía su certificado digital (CD) para autenticarse con el servidor. El CD debe seguir el estándar X509.
5. El servidor genera una llave simétrica (LS) y la envía al cliente protegida (cifrada).
6. El cliente descifra el mensaje y obtiene la llave simétrica que el servidor envía. Como mecanismo de confirmación el cliente envía la misma llave al servidor, protegida, y espera la confirmación del servidor (OK o ERROR).
7. El cliente usa la llave simétrica para enviar la información protegida.
8. Además, el cliente envía el código *hash* correspondiente, cifrado con su llave privada (de tal forma que el servidor podrá comprobar el origen con la llave pública correspondiente).
9. El servidor responde (RTA), RTA: OK o ERROR, anunciando el resultado de la transacción y la terminación de la comunicación.



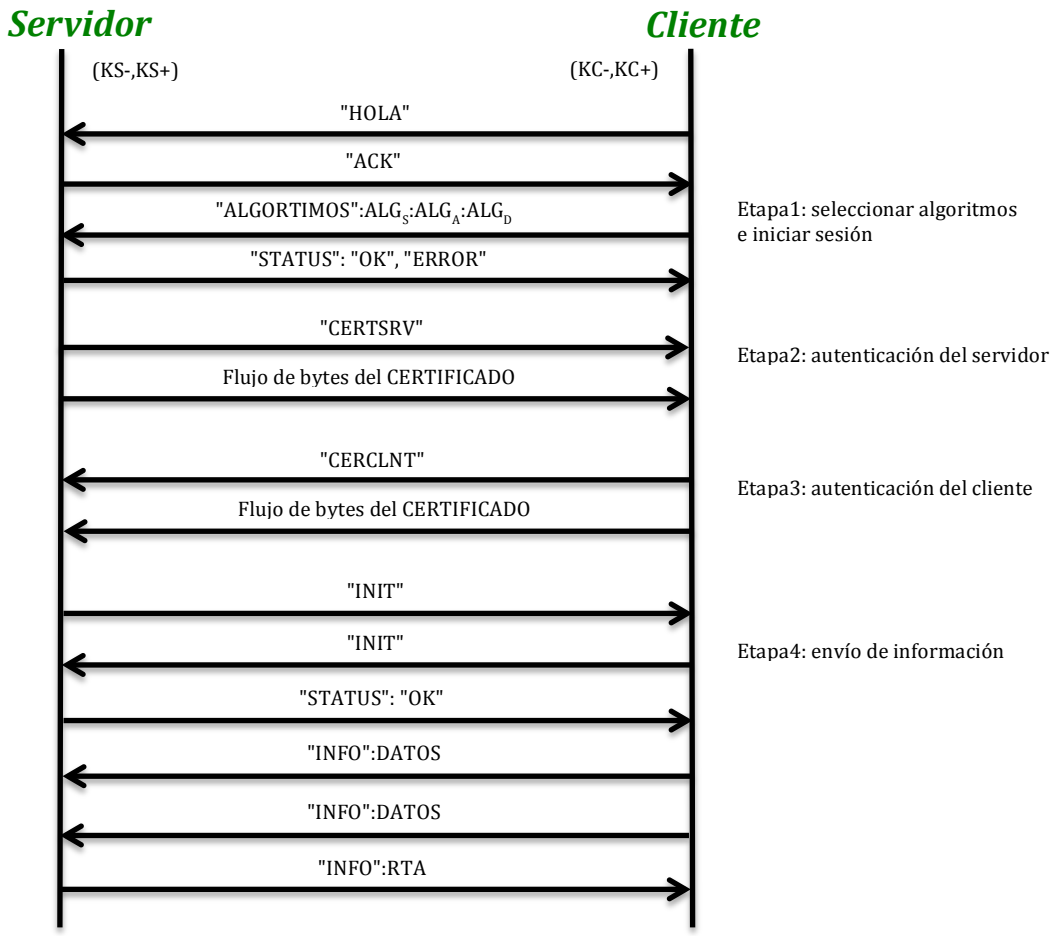
PARA TENER EN CUENTA:

- El protocolo de comunicación maneja la siguiente convención:
  - Cadenas de Control: "HOLA", "ACK", "ALGORITMOS", "STATUS", "OK", "ERROR", "CERTSRV", "CERTCLNT", "INIT", "INFO".
  - Separador Principal: " : "
- A continuación se presentan los algoritmos disponibles en el servidor para manejo de las tareas de cifrado. Es decir, los algoritmos que deben reemplazar las cadenas ALG_S, ALG_A y ALG_H en el protocolo. Para implementar el cliente usted debe seleccionar un algoritmo en cada caso.
  - Simétricos (ALG_S):
    - DES. Modo ECB, esquema de relleno PKCS5, llave de 64 bits.
    - AES. Modo ECB, esquema de relleno PKCS5, llave de 128 bits.
    - Blowfish. Cifrado por bloques, llave de 128 bits.
    - RC4. Cifrado por flujo, llave de 128 bits.
  - Asimétricos (ALG_A):
    - RSA. Cifrado por bloques, llave de 1024 bits.
  - HMAC (ALG_H):
    - HmacMD5
    - HmacSHA1
    - HmacSHA256

  Las cadenas que identifican cada uno de los algoritmos son: "DES", "AES", "Blowfish", "RC4", "RSA", "HMACMD5", "HMACSHA1", "HMACSHA256".
- Utilizaremos la versión 3 del estándar X509 para el CD. La idea es que el cliente puede comprobar la identidad del servidor a partir de un CD (en un caso real este debería ser expedido por una entidad certificadora pero aquí se va a generar localmente). El CD debe seguir el estándar X509, en particular, debe contener la llave pública para usarla en el proceso de comunicación (se recomienda revisar la librería Bouncycastle para la generación del certificado). El cliente se autentica con el servidor por medio de un usuario y una clave.
- La comunicación se realiza a través de sockets de acuerdo con el protocolo de comunicación definido.
- El código de envío del certificado debe lucir como se indica abajo. Es decir, primero se indica que se enviará el certificado y luego se envía el contenido (en bytes).

```
writer.println( CERTIFICADO );
java.security.cert.X509Certificate cert = certificado( );
byte[] mybyte = cert.getEncoded( );
socket.getOutputStream( ).write( mybyte );
socket.getOutputStream( ).flush( );
```

- Dado que existen problemas en la transmisión de los bytes cifrados, se manejará encapsulamiento con cadenas hexadecimales para transmisión de enteros.
- Después de que el cliente recibe la llave simétrica, puede continuar con el protocolo. En adelante los datos que se envían y reciben deben estar cifrados con dicha llave.
- La dirección del servidor es infracomp.virtual.uniandes.edu.co (puerto 443).
- Habrá una versión sin seguridad corriendo en otro puerto para hacer pruebas del protocolo (puerto 80). La figura siguiente ilustra el protocolo sin seguridad.



Entrega:

Cada grupo debe entregar un archivo zip que incluya el informe (con las respuestas a las tareas A y B) y un proyecto Java con la implementación correspondiente al cliente (descrito en la parte C). El informe vale 30% y la implementación 70% de la calificación del caso 2.

Referencias:

- *Cryptography and network security*, W. Stallings, Ed. Prentice Hall, 2003.
- *Computer Networks*. Andrew S. Tanenbaum. Cuarta edición. Prentice Hall 2003, Caps 7, 8.
- *Blowfish*. Página oficial es: http://www.schneier.com/blowfish.html
- *RSA*. Puede encontrar más información en: http://www.rsa.com/rsalabs/node.asp?id=2125
- *CD X509*. Puede encontrar la especificación en: http://tools.ietf.org/rfc/rfc5280.txt
- *MD5*. Puede encontrar la especificación en : http://www.ietf.org/rfc/rfc1321.txt