Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
2017-10

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

Objetivos:

- Identificar los requerimientos de seguridad 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.

Problemática:

Como se indicó en el enunciado del caso, las principales tareas del sistema son: Afiliación, Recaudo, AFE, Historia laboral, 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 determinar medidas para mitigar los problemas 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 manejo de historia laboral.

Tareas:

Suponga que la arquitectura del sistema incluye un servidor en la oficina principal de la entidad, para manejo de historias laborales. Los clientes se pueden conectar a este servidor por medio de una aplicación cliente que corre en las sucursales de la entidad. Los empleados de las sucursales deben autenticarse con el servidor para tener acceso a la información. Las comunicaciones se dan vía internet.

Los computadores de escritorio 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 principal. El servidor maneja dos interfaces de red, una para conexiones con los clientes externos, entre los que se cuentan las sucursales, 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 a la aplicación de manejo de historias laborales. 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 explicados, 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 justificar la decisión).
- Además, debe justificar los mecanismos propuestos. Es decir, diga explícitamente cómo contribuye el mecanismo a la solución del problema.

C. [70%] Implementación del Prototipo

En esta parte del proyecto nos concentraremos únicamente en el sistema que maneja la historia laboral de los empleados. Usted debe construir la aplicación cliente para enviar la información al servidor que maneja la información. 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 común para asegurar las comunicaciones.

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 o no.
3. El cliente envía su certificado digital (CD). El CD debe seguir el estándar X509.
4. El servidor envía su certificado digital (CD). El CD debe seguir el estándar X509.
5. Generación, envío y verificación de retos para autenticación. Un reto es un número aleatorio grande (por facilidad usaremos java.util.Random, un generador seudoaleatorio que permita establecer una semilla. Cada grupo debe escoger un evento para generar la semilla). Tanto el cliente como el servidor deben verificar que el reto enviado y el recibido corresponden antes de continuar con la comunicación.
6. El servidor genera una llave simétrica (LS) y la envía al cliente protegida (cifrada).
7. El cliente descifra y usa la llave simétrica para enviar la consulta de forma protegida. La consulta es un número de cédula CC.
8. El servidor responde.
9. El cliente recibe la respuesta y los terminan la comunicación.
10.

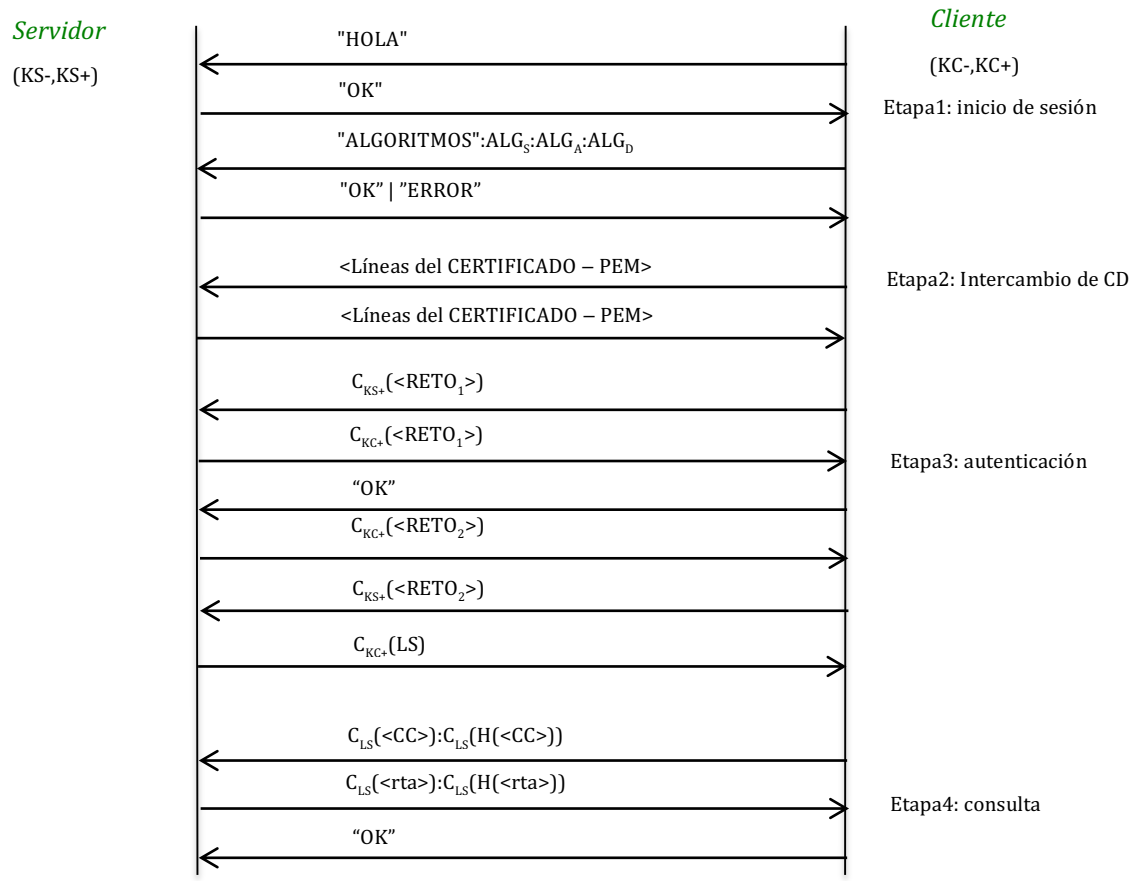

Fig.1 Protocolo de comunicación

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.
- Tenga en cuenta que el envío del certificado es línea por línea.
- Dado que existen problemas en la transmisión de los bytes cifrados, se manejará encapsulamiento con cadenas hexadecimales para transmisión de enteros.
- Se entregará una versión del servidor para que puedan realizar pruebas.
- Habrá una versión sin seguridad para hacer pruebas del protocolo. La figura 2 ilustra el protocolo sin seguridad.

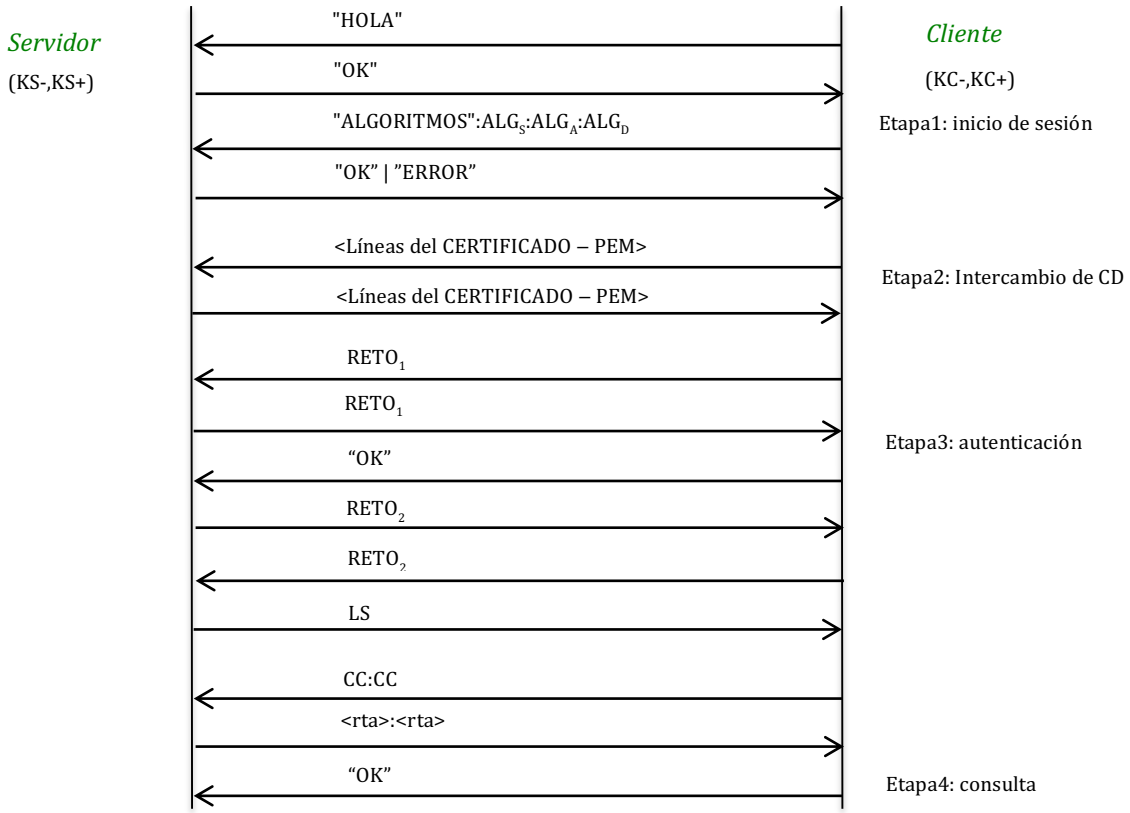

Fig. 2 Protocolo de comunicación sin manejo de cifrado

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