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

Caso de Estudio – Canales Seguros
Sistema de Gestión Empresarial y Operativa de una Compañía Transportadora
Caso 2 - Canales seguros

Objetivos

- Identificar los requerimientos de seguridad de los canales usados para transmisión de la información en el sistema de gestión empresarial y operativa de una compañía transportadora.
- 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 la recepción de órdenes de recogida, gestión de rutas, rastreo de unidades de distribución y paquetes, y gestión administrativa contable de recursos y de clientes.

En este contexto, surgen diferentes problemas de seguridad para algunas de las transacciones que el sistema soporta, tanto a nivel de transmisión, como en procesamiento y 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 de seguridad y analizar, considerando solo aspectos de seguridad, las tareas relacionadas con el manejo de órdenes de recogida.

Tareas:

Suponga que la arquitectura del sistema incluye tres servidores: uno se encarga del manejo y rastreo de unidades de distribución y paquetes, el segundo del manejo de órdenes de recogida, y el último se encarga del manejo administrativo y contable de recursos y clientes.

- Los puntos de atención al cliente se comunican periódicamente con el servidor de manejo de órdenes para informar su estado.
- Para el rastreo de unidades de distribución y paquetes y optimización de rutas, las unidades se comunican cada 60 segundos con el servidor para informar su estado. El servidor recibe la información y la almacena para procesamiento. Por otro lado, el servidor de manejo de unidades de distribución calcula diariamente a la 1 a.m. las rutas del día. En condiciones excepcionales, el servidor puede cambiar las rutas de las unidades de distribución durante el día.
- El servidor de manejo de órdenes se comunica con el de rastreo y rutas: las rutas se calculan con base en los puntos de atención que han recibido paquetes.
- El servidor de manejo administrativo contable no atiende consultas de clientes vía web; solamente responde a consultas iniciadas en la intranet de la compañía.

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

Suponiendo que el sistema descrito en el párrafo anterior implementa control de acceso a nivel de sistema operativo en todos los servidores, y cuenta con un firewall que filtra paquetes a la entrada de la red:

1. Identifique y describa los datos que deben ser protegidos en el sistema de manejo de órdenes. Explique su respuesta en cada caso(*) y responda la pregunta ¿Si un actor no autorizado consigue acceso al dato mencionado, ya sea en modo lectura o escritura, cómo podría afectar la empresa?
2. Identifique cuatro vulnerabilidades del sistema de manejo de órdenes, 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 y procesamiento de los datos. Explique su respuesta en cada caso (*).

(*) *Sus explicaciones DEBEN corresponder al contexto planteado (de forma explícita). NO se aceptarán respuestas para contextos genéricos*.

B. [10%] Propuesta de Soluciones.

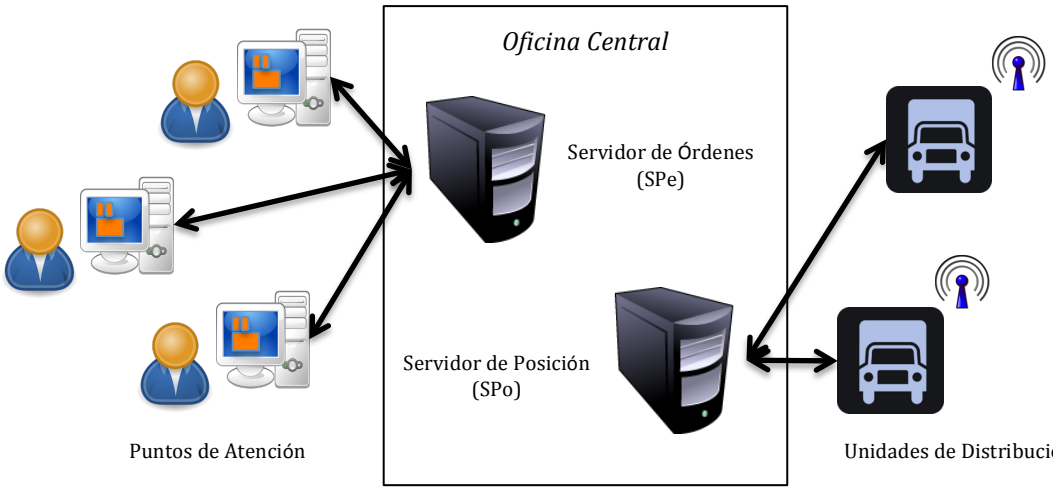
Para cada una de las vulnerabilidades que usted identificó en el punto anterior, proponga mecanismos de resolución.

- 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, identifique explícitamente qué vulnerabilidad resuelve y justifique.

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 centraremos únicamente en el sistema de manejo de órdenes.



Su tarea consiste en construir el cliente de un punto de atención que se comunique con el servidor de manejo de órdenes para reportar su estado (número de órdenes recibidas).

El punto de atención (PA) y el servidor seguirán el protocolo descrito a continuación para su comunicación:

1. El PA se comunica con el servidor para iniciar una sesión de actualización de estado, y espera un mensaje de confirmación.
2. El PA envía la lista de algoritmos de cifrado que usará durante la sesión y espera un mensaje del servidor confirmando que soporta los algoritmos seleccionados (si no, el servidor envía un mensaje de terminación).
3. El PA envía un número aleatorio y su certificado digital (CD) para autenticarse con el servidor, y espera confirmación del servidor.
4. El servidor verifica el certificado digital del PA, responde con otro número aleatorio y su CD, y espera confirmación del PA.
5. El servidor responde cifrando con su llave privada el número aleatorio del PA y espera confirmación.
6. El PA responde cifrando con su llave privada el número aleatorio del servidor y espera confirmación.
7. El PA usa la llave pública del servidor y su llave privada para enviar la llave para generación del HMAC.
8. El PA usa la llave pública del servidor para enviar el número de órdenes acumuladas hasta el momento, y el código HMAC correspondiente.
9. El servidor recibe la información y chequea integridad. Si no hay problemas la almacena. Después envía respuesta al cliente, OK o ERROR, anunciado el resultado de la comunicación y la terminación de la comunicación.

La figura 1 ilustra el protocolo.

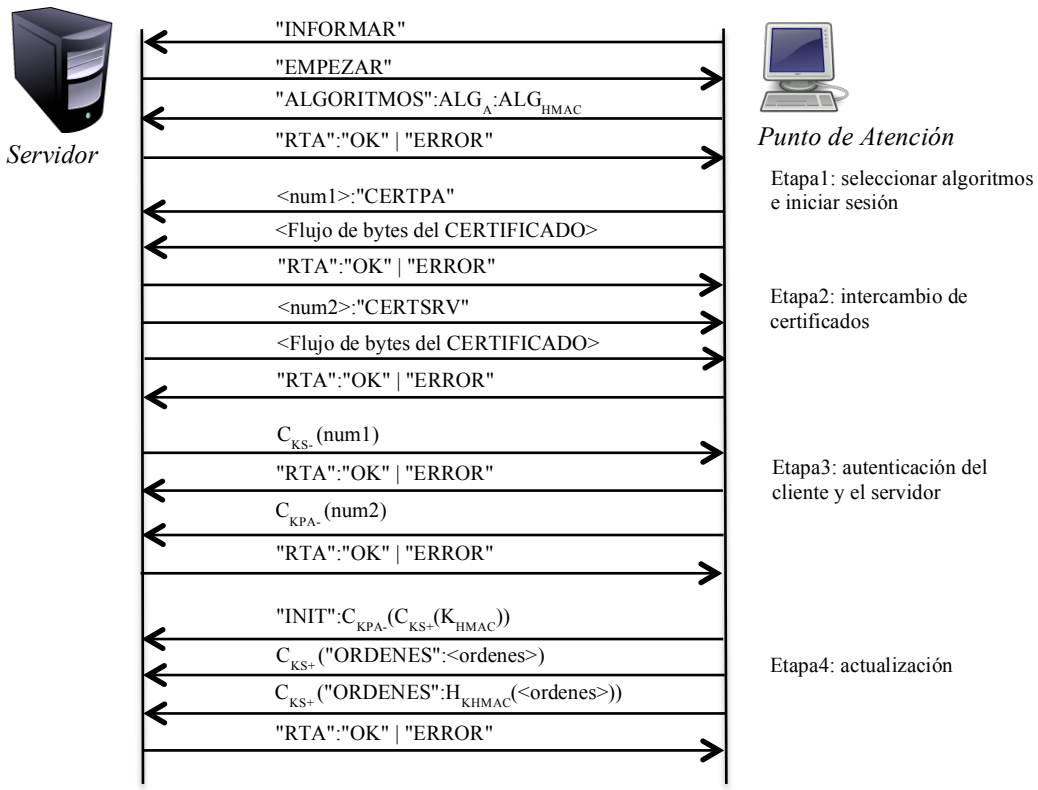


Figura 1. Protocolo de comunicación entre el punto de atención y el servidor de órdenes.

PARA TENER EN CUENTA:

- El protocolo de comunicación maneja la siguiente convención:
  - Cadenas de Control: `"INFORMAR", "EMPEZAR", "ALGORITMOS", "RTA", "OK", "ERROR", "CERTPA", "CERTSRV", "ORDENES"`, etc.
  - Separador Principal: " : "
- A continuación se presentan los algoritmos disponibles en el servidor para manejo de integridad y confidencialidad. Es decir, los algoritmos que deben reemplazar las cadenas ALG_A y ALG_{HMAC} en el protocolo. Para implementar el cliente usted debe seleccionar un algoritmo de generación de código criptográfico de hash.
  - Asimétricos (ALG_A):
    - RSA. Cifrado por bloques, llave de 1024 bits.
  - HMAC (ALG_D):
    - HmacMD5
    - HmacSHA1
    - HmacSHA256

  Las cadenas que identifican cada uno de los algoritmos son: `"RSA", "HMACMD5", "HMACSHA1", "HMACSHA256"`.
- Utilizaremos la versión 3 del estándar X509 para los certificados digitales (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.
- Recuerde que el algoritmo RSA, con una llave de 1024 bits, solo es capaz de cifrar bloques de 117 bytes. De requerir cifrar datos más grandes, realice el procedimiento por bloques.
- Para facilidad de la transmisión, los números <num1> y <num2> deben ser manejados como cadenas de caracteres (String).
- El .jar del servidor será publicado en SICUA+. Además, se publicará otra versión sin seguridad (figura 2).

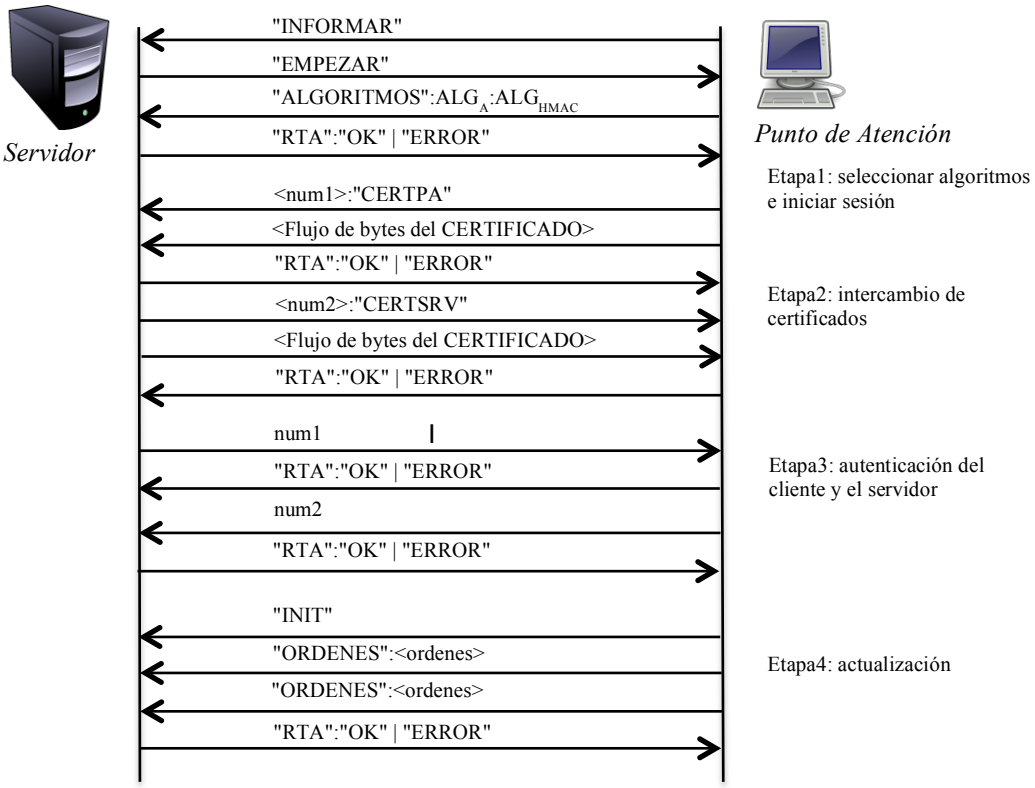


Figura 2. Protocolo sin seguridad.

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.
- *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