PAI. 2

AUDITORES DE INTEGRIDAD DE TRANSMISIÓN PUNTO A PUNTO PARA INSTITUCIONES FINANCIERAS

El documento se elabora a petición del profesor Ángel Jesús Varela Vaca de la Facultad SSII de la Universidad de Sevilla con el fin de proceder a la presentación del informe final del proyecto titulado "Verificadores de Integridad en la Transmisión Punto a Punto para Entidades Financieras".

Dirigida por:

François Barberot

Gustavo Antonio Souza De Barros

Pepe Ortiz Roldan

Fecha:

22/10/23

LISTA DE VERIFICACIÓN DE DOCUMENTOS

Documento:	**PAI2-ST_6**
Denominación:	**VERIFICADORES DE INTEGRIDAD EN LA TRANSMISIÓN PUNTO-PUNTO PARA ENTIDAD FINANCIERA**
Edición:	**01**
Fecha:	**22 octubre 2023**
Dirigida por:	**François Barberot, Gustavo Antonio Souza De Barros, Pepe Ortiz Roldan**
Aprobado por:	Ángel Jesús Varela Vaca (Universidad de Sevilla)

CONTROL DE CAMBIOS

Pasos	Fecha	Retroalimentación
01	08/10	Introducción
02	10/10	Planificación de todo el proyecto (diagrama de Gantt)
03	11/10	Diseño e implementación de sistemas
05	18/10	Plan de pruebas para el sistema
07	20/10	Conclusión y Bibliografía / Webgrafía
08	20/10	Corrección y finalización del informe

ÍNDICE

1 Introducción

Este documento es el informe detallado del proyecto enfocado al desarrollo de un sistema de verificación de integridad de transmisión punto a punto para una entidad financiera. Este proyecto es de vital importancia en el contexto actual del sector financiero, donde la seguridad de los datos y la privacidad de las transacciones están en el centro de las preocupaciones.

El objetivo principal de este proyecto es reforzar la seguridad de las comunicaciones entre las entidades financieras y sus clientes garantizando la integridad de los datos intercambiados. Las transmisiones punto a punto, que a menudo se utilizan en transacciones financieras, están expuestas a una variedad de amenazas, incluidos los ataques de intermediarios y los intentos de reproducir datos. Estas amenazas potenciales requieren una respuesta proactiva y sólida.

Nos centraremos en desarrollar un sistema de verificación de integridad que garantice que los datos intercambiados entre las entidades financieras y sus clientes permanezcan confidenciales, auténticos e inalterados. Este proyecto representa un paso importante en el fortalecimiento de la confianza de los clientes en la integridad de sus transacciones financieras.

Como parte de nuestro enfoque, exploraremos en detalle cada elemento del diseño del sistema, desde el establecimiento de una conexión cliente-servidor segura hasta la generación automatizada de códigos de autenticación de mensajes (MAC) para garantizar la integridad de los datos. Cada uno de estos elementos se explicará en profundidad para una comprensión completa del sistema establecido.

La elección de este enfoque se basa en la necesidad de prevenir y detectar de forma proactiva las amenazas potenciales. Si bien existen otros sistemas de seguridad, diseñamos este proyecto con un enfoque en las especificidades de la industria financiera, donde la privacidad y la seguridad son primordiales.

También reconocemos los desafíos y las compensaciones inherentes a esta elección, incluida la posibilidad de detección a posteriori en caso de un ataque importante, pero consideramos que los beneficios de este sistema, como la detección de amenazas internas y el impacto limitado en el rendimiento del sistema, son de vital importancia.

En este informe, detallaremos los pasos de desarrollo, las pruebas realizadas, los resultados de la simulación "Man in the Middle" con indicadores clave de rendimiento (KPI) y la prevención de repeticiones de datos. Nuestro objetivo final es hacer una contribución significativa a la seguridad del sector financiero y garantizar la protección de los datos sensibles de nuestros clientes.

2 Planificación

Como parte de nuestro proyecto, utilizamos una herramienta de gestión del tiempo: el diagrama de Gantt. Este diagrama ofrece una visión clara de la planificación de nuestros proyectos, identificando las tareas y sus plazos. Desempeña un papel esencial en la organización, coordinación y seguimiento de nuestras actividades.

Tareas	9 oct.	10 oct.	11 oct.	12 oct.	13 oct.	14 oct.	15 oct.	16 oct.	17 oct.	18 oct.	19 oct.	20 oct.	21 oct.	22 oct.
Fase 1: Desarrollo del proyecto														
1.1 Conexión cliente-servidor														
1.2 Nonce														
1.3 Base de datos de nonces														
1.5 MAC														
1.6 Automatización														
1.7 Claves														
Fase 2: Automatización y pruebas														
2.1 Man in the Middle														
2.2 Replay														
Fase 3: Documentación y informe														
3.1 Redacción del informe														

3 Desarrollo de proyectos

3.1 Diseño del sistema

El diseño del sistema se basa en una arquitectura robusta para garantizar la integridad de las transmisiones punto a punto para las entidades financieras. Estos son los detalles de cada etapa clave del desarrollo:

Conexión cliente-servidor: La conexión cliente-servidor es la piedra angular de nuestro sistema. Para garantizar la integridad de las transmisiones, hemos establecido una conexión segura utilizando protocolos de encriptación robustos. Los datos intercambiados entre el cliente y el servidor se cifran mediante algoritmos de cifrado. Esto garantiza que toda la información esté protegida de miradas indiscretas. Además, la autenticación se realiza mediante certificados digitales para asegurar la identidad de cada parte.

Nonce: Los nonces, generados aleatoriamente con cada transacción, son un componente esencial. Cada nonce es único y está asociado a una transacción en particular. Evitan eficazmente los ataques de repetición, ya que un nonce no se puede reutilizar.

Base de datos de nonces: Los nonces generados se almacenan en una base de datos dedicada. Esta base de datos asegura una gestión eficiente de los nonces garantizando su unicidad, lo que evita la duplicación involuntaria. También proporciona protección contra intentos de acceso no autorizados, mejorando la seguridad de todo el sistema.

La automatización del servidor de base de datos es esencial para garantizar una gestión fluida de nonce. Permite la creación, lectura, actualización y eliminación eficientes de expresiones, lo que minimiza el riesgo de error humano. Además, existen mecanismos de registro para realizar un seguimiento de todas las operaciones realizadas en los nonces, lo que mejora la trazabilidad.

MAC: El cálculo del código de autenticación de mensajes (MAC) está automatizado para cada transacción. Esta automatización garantiza que cada mensaje esté protegido contra la manipulación. Las claves MAC se almacenan de forma segura y se supervisa su uso. La automatización del proceso MAC garantiza la coherencia y la integridad de los datos intercambiados.

Claves: Aunque la implementación detallada de las claves no se incluye en este proyecto, es esencial tener en cuenta su papel central en la seguridad del sistema. Las claves de cifrado y autenticación deben generarse, almacenarse y gestionarse de forma segura para garantizar la confidencialidad e integridad de los datos. La gestión adecuada de las claves es crucial para la seguridad general del sistema.

3.2 Simulación "Man in the Middle" y Replay Attack con KPIs

Para evaluar la robustez del sistema y su capacidad para resistir posibles ataques, se llevó a cabo una simulación de "Man in the Middle". Esta simulación creó un entorno controlado en el que se simularon ataques "Man in the Middle".

Se utilizaron indicadores clave de rendimiento (KPI) para evaluar el rendimiento del sistema. Se analizaron los resultados de los KPIs para evaluar la efectividad del sistema bajo condiciones de ataque simuladas.

La simulación también tuvo en cuenta la posibilidad de repeticiones. El sistema ha sido evaluado por su capacidad para detectar y prevenir repeticiones de datos, lo cual es crucial para la seguridad de las comunicaciones punto a punto.

En la siguiente sección del informe se analizarán los resultados detallados de esta simulación, incluidas las conclusiones y las lecciones aprendidas de todo el proceso de desarrollo.

3.3 Implementación

El proyecto se basa en dos partes distintas: el Cliente y el Servidor, y se ha desarrollado utilizando las tecnologías Python y SQLite.

En la implementación, el Cliente y el Servidor se comunican y trabajan juntos para lograr los objetivos del proyecto. Python se utiliza como el lenguaje de programación principal para escribir el código tanto en el lado del Cliente como en el Servidor. SQLite se emplea como la base de datos para almacenar y gestionar los datos necesarios para el proyecto.

3.3.1 Cliente

Se ha desarrollado un software que se conecta a un servidor a través de un socket y envía mensajes automáticamente. La cantidad de mensajes a enviar está configurada de manera fija pero puede ser modificada. Los mensajes siguen un formato predefinido, que es del tipo {origen: id_cuenta_origen,

destino: id_cuenta_destino, cantidad: dinero_transferido}, y son generados de manera aleatoria por el software.

El proceso de envío de mensajes se divide en varias etapas:

1. Solicitud de Nonce: En primer lugar, el cliente envía una solicitud al servidor para obtener un nonce. El servidor genera un valor único y aleatorio conocido como nonce, que se utilizará en el mensaje a enviar.

2. Creación del Mensaje: Una vez que se recibe el nonce, se añade al mensaje original. Además, se calcula y agrega un código de autenticación de mensaje (MAC) utilizando el algoritmo HMAC con SHA-256. Este MAC garantiza la integridad y autenticidad del mensaje.

3. Base de Datos de Nonces: Cada cliente mantiene una base de datos propia para llevar un registro de los nonces utilizados. Esto ayuda a evitar la reutilización de nonces, lo que es esencial para la seguridad. Los nonces están almacenados en "client_nonces.db".

Para simular posibles ataques, se ha introducido un componente de aleatoriedad en el proceso de envío de mensajes:

- Se genera un número aleatorio entre 1 y 6.
- Si el número es 1, el mensaje se envía utilizando un nonce previamente utilizado, lo cual es un comportamiento no permitido.
- Si el número es 2, se altera el HMAC del mensaje, comprometiendo la integridad de la información.
- Si el número es 3, se envía un nonce previamente utilizado y se altera el HMAC, lo que representa un escenario de ataque doble.
- Si el número está entre 4 y 6, el mensaje se envía de manera normal y adecuada.

Además, se lleva un contador en la base de datos de nonces que se incrementa cada vez que se utiliza un nonce generado por el servidor, asegurando que no se reutilicen.

Este diseño proporciona un nivel de seguridad en la comunicación entre el cliente y el servidor al garantizar la integridad de los mensajes y evitar la reutilización de nonces. Las simulaciones de ataques permiten evaluar la capacidad del sistema para resistir posibles amenazas y proporcionan una capa adicional de seguridad.

3.3.2 Servidor

Se ha desarrollado un software para el servidor que espera mensajes de los clientes. Estos mensajes pueden contener datos de transferencia o solicitudes de un nonce.

Cuando se recibe una solicitud de nonce, el servidor genera una cadena hexadecimal de 16 bytes de forma segura utilizando una biblioteca que produce números aleatorios criptográficamente seguros. El servidor también mantiene una base de datos llamada "nonces.db" donde se almacenan los nonces utilizados. El nonce recién generado se almacena en esta base de datos con un contador inicializado en 0, indicando que aún no se ha utilizado.

En el caso de que el mensaje contenga datos de transferencia, el servidor realiza varias verificaciones de integridad:

Validación del HMAC: Se verifica si el HMAC (Código de Autenticación de Mensaje) es válido, lo que garantiza la integridad y autenticidad del mensaje. Esto se hace utilizando el algoritmo HMAC con SHA-256.

Comprobación de Nonce: Se verifica si el nonce incluido en el mensaje ya ha sido utilizado previamente. El servidor consulta la base de datos de nonces para asegurarse de que el nonce no se haya empleado en una transferencia anterior.

Si ambas verificaciones de integridad pasan, el servidor actualiza el contador del nonce en la base de datos a 1 para indicar que se ha utilizado.

Además, se registran los resultados de cada mensaje en una base de datos de KPI (Indicadores Clave de Rendimiento) donde se almacena:

- El mensaje en sí.
- El HMAC asociado al mensaje.
- El nonce utilizado en el mensaje.
- Un indicador de integridad (0 o 1) para indicar si la integridad del mensaje se ha mantenido.
- En caso de que la integridad se haya visto comprometida, se registra una razón o explicación correspondiente.

Este diseño garantiza la seguridad y la integridad de las transacciones y permite llevar un registro de los resultados y cualquier posible compromiso de la integridad en el servidor.

3.3.3 Claves

En el software desarrollado, tanto el cliente como el servidor utilizan una clave compartida para la encriptación HMAC. Esta clave está incrustada (hard-coded) en el archivo Python. Sin embargo, para fines de estudio, aquí se presenta una explicación detallada de cómo funciona el algoritmo de Diffie-Hellman:

El algoritmo de Diffie-Hellman es un método criptográfico que permite a dos partes acordar una clave secreta compartida a través de una comunicación insegura. A continuación, se detallan las etapas del algoritmo:

1. Selección de Parámetros Comunes: Ambas personas acuerdan de antemano dos números: un número primo grande (p) y un número entero positivo (g) llamado generador. Estos valores son públicos.

2. Generación de Claves Públicas: Cada persona selecciona un número secreto privado (a para la primera persona y b para la segunda) y mantiene estos números en secreto.

3. Cálculo de Claves Públicas: Cada persona calcula su clave pública compartiendo: A = g^a (para la primera persona) y B = g^b (para la segunda persona).

4. Intercambio de Claves Públicas: Ambas personas se envían sus claves públicas (A y B) a través de un canal inseguro.

5. Cálculo de la Clave Compartida: Cada persona utiliza la clave pública del otro y su clave privada para calcular la misma clave secreta: K = (clave pública del otro)^clave privada propia.

6. Resultado: Ahora ambas personas tienen una clave secreta compartida (K) que pueden usar para cifrar y descifrar mensajes de forma segura.

La belleza de este proceso radica en que, aunque las claves públicas se compartan a través de un canal inseguro, un atacante no puede determinar la clave secreta compartida sin conocer los valores privados (a y b) debido a la complejidad del cálculo exponencial modular.

4 Probando el sistema desarrollado

Se presentan cinco mensajes generados de forma aleatoria. En el lado derecho, se muestra la pantalla del cliente, y en el lado izquierdo, se muestra la pantalla del servidor con las correspondientes interacciones del terminal:

En estas capturas de pantalla, se pueden observar las interacciones entre el cliente y el servidor, donde se muestran los mensajes generados en tiempo real a medida que se comunican.

Como se puede observar, en las capturas de pantalla, se presentan diferentes escenarios que muestran casos de HMAC modificado, comportamiento normal y nonce ya utilizado. Es importante destacar que el software ha demostrado ser capaz de detectar y gestionar estos casos, lo que contribuye significativamente a la identificación de ataques de repetición (replay attacks) y ataques de intermediario (man-in-the-middle).

Además, en la base de datos de Indicadores Clave de Rendimiento (KPI), se han registrado estas ocurrencias. Esto proporciona un registro detallado de las actividades y eventos relacionados con la seguridad en el sistema.

79c1c387a8284323257a770708dd01938a2	d9fa84e3e1541b2385a493e34a86aa58	0	Possible Repeat Attack
	1d3f3bd97a843d1fe28bfda67f05bca7	0	Invalid HMAC!; Possible Repeat Attack
284f618e1bbac37ba583ae2135ae3399517	9e3986bac3ec487ab1802cbe8cc7988e	1	
	27edd3bf7a0195d89904562d0bc8b547	0	Invalid HMAC!
144fee570ca8a8862ccdd9adaf36e7d29b5	5ea391fd7fc39a181aa4a0ce2e961c45	0	Possible Repeat Attack
54d4c33a69b0b84c832df8a8abae79e8151	3cebedaab73b5859f354c901e7e1cb8b	1	

5 Sugerencias de mejora

En nuestra implementación, no implementamos el almacenamiento e intercambio de claves de manera segura. Para eso, algunas opciones fueron estudiadas:

- Utilizar algún algoritmo de claves asimétricas, como RSA. De esta forma, el cliente tenía una clave pública y el servidor una privada.
- Autenticación por certificados digitales.
- Uso de un protocolo de intercambio de claves, como por ejemplo, Diffie-Helman.

Otra posible mejora, es hacer un registro más completo de los problemas de integridad, añadiendo al base de datos KPI cosas como la fecha de la ocurrencia y el endereço IP del atacante.

6 Conclusiones

El proyecto representa un hito importante en nuestra trayectoria como estudiantes de Seguridad de Sistemas Informáticos e Internet. Nuestro principal objetivo era crear un sistema de verificación de integridad para transmisiones punto a punto, específicamente adaptado al sector financiero.

El desarrollo de este sistema ha sido a la vez desafiante y gratificante. Nos enfrentamos a complejos desafíos técnicos durante todo el proceso. La implementación de una conexión cliente-servidor segura requería un conocimiento profundo de los protocolos de cifrado, los certificados digitales y los mecanismos de autenticación. Garantizar la confidencialidad y autenticidad de los datos intercambiados ha demostrado ser un trabajo exigente, pero esencial para satisfacer las necesidades de seguridad del sector financiero.

La gestión de los nonces y la base de datos asociada también fue una tarea complicada. Generar nonces únicos para cada transacción y almacenarlos de forma segura en una base de datos requirió una planificación cuidadosa. La sincronización entre el cliente y el servidor para mantener la unicidad de los nonces añadió una importante dimensión técnica al proyecto.

La generación automatizada de códigos de autenticación de mensajes (MAC) para cada transacción fue otro paso complejo. Esto implicó la coordinación entre las partes y la puesta en marcha de mecanismos para garantizar la integridad de los datos.

Más allá de los desafíos técnicos, este proyecto ha fortalecido nuestra comprensión de la seguridad de los sistemas de información. Nos permitió poner en práctica conceptos avanzados de seguridad, al tiempo que desarrollábamos habilidades esenciales para proteger los datos sensibles. Hemos adquirido una valiosa experiencia en la seguridad de las transacciones financieras, un área en la que la confianza del cliente es clave.

En última instancia, este proyecto demostró la importancia de la seguridad en un mundo digital en constante cambio, especialmente en el sector financiero.

7 Bibliografía y webgrafía

- Contenido visto en clase.
- Diffie-Hellman. https://pt.wikipedia.org/wiki/Troca_de_chaves_de_Diffie%E2%80%93Hellman