Caso de Estudio 2

Integrantes:

Margarita Gómez Ballén	201423591
Fabio López Corredor	201423782
Jose Gabriel Tamura	201424484

Análisis y entendendimiento del problema

Amenazas de fuente humana

Web spoofing (Confidencialidad)

Esta amenaza se daría en caso de que un tercero cree una página web, tal como la de Novasoft financiero en línea, para así engañar a uno de los usuarios. En dado caso en que esta amenaza se consolide, el tercero podría robar las credenciales y la información que los usuarios intenten ingresar en la página falsa. Al tener esto en su poder, podría suplantar a los usuarios de Novasoft financiero en línea y solicitar o introducir información, haciéndose pasar por ellos.

Tampering (Integridad)

Esta amenaza consiste en la adulteración de datos con los que trabaja la aplicación, lo cual podría darse en caso de que un tercero logre acceder al sistema o en caso de que altere información que pase por los medios de transmisión. En caso de que logre acceder al sistema y alterar datos, la aplicación perdería su propósito, pues estaría operando con datos falsos que no reflejan información que la empresa requiere. Asimismo, la aplicación no cuenta con una manera de identificar eso, pues no se menciona que se realice algún tipo de verificación de la integridad de los datos que fluyen en el software.

Espionaje (Confidencialidad)

Se da en caso de que un tercero logre acceder a información confidencial de la empresa como movimientos financieros o información sensible de los usuarios. Este acceso podría darse si los algoritmos de cifrado son descubiertos y revertidos para poder descifrar la información, lo cual ocurriría debido a que se menciona que la empresa utiliza algoritmos propios, que no es una táctica recomendada a la hora de asegurar los datos. Si se consolida esta amenaza, la información personal de los usuarios podría caer en manos de ladrones de identidad y la información financiera podría verse comprometida por su confidencialidad e integridad.

Elevación de privilegios (Confidencialidad, autenticación, integridad)

Esto ocurriría en caso de que un usuario de la aplicación logre obtener más permisos de los que cuenta normalmente. Esto podría darse si un usuario decide averiguar la contraseña de un administrador a fuerza bruta, logrando obtener el acceso al archivo de control de acceso y cambiar los permisos que tiene como usuario o que otros usuarios tengan. También por el hecho de que no se ve explícita una política de control de accesos en la red, dejando que desde el servicio web alguien no autorizado pueda llegar a acceder información importante. Si la amenaza se llega a dar, el impacto de esto varía dependiendo de los privilegios a los cuales logre acceder. Podría obtener información, alterarla o crear información errónea.

Denegación de servicio (Disponibilidad)

Esta amenaza se podría llevar a cabo si sobrecargan al sistema con peticiones, pues no se evidencia un servidor preparado para un vasto número de éstas dado que no se espera un gran número de usuarios. En caso de que un tercero decida realizar un ataque DDoS, impediría el correcto funcionamiento de esta herramienta. Además, esta sobrecarga del sistema ocasionaría la indisponibilidad de la aplicación, la cual es vital para sus usuarios, pues es de uso permanente.

Vulnerabilidades

Permisión de claves débiles

El sistema no menciona restricción alguna en cuanto a la autenticación con el uso de claves. Permitiendo que todo tipo de usuarios utilicen como contraseña “123” o “abc”, haciendo que sea mucho más fácil que un atacante logre obtener acceso a las cuentas de los usuarios.

Poco espacio de almacenamiento

Según el primer enunciado, la aplicación cuenta con 60 usuarios potenciales y un espacio actual de almacenamiento de 8GB con un crecimiento anual esperado de 500 MB. Es importante que el sistema esté preparado para algo más masivo en caso de que tanto los usuarios como sus peticiones crezcan de manera inesperada. No se menciona algo que haga del sistema uno con alta escalabilidad y es un atributo de calidad que se debería tener en cuenta para no tener problemas en cuanto al almacenamiento de datos.

Almacenamiento de información sin cifrar

Otro aspecto que no se menciona, es el estado en el cual los datos son almacenados en la base de datos con la que cuenta la aplicación. Esto da a entender que esta es guardada sin ningún tipo de protección en cuanto a espionaje, por lo que si alguien tiene acceso a esta base de datos, no debería poder ver la información delicada de los usuarios o de la empresa. Esto se presta a que en caso de que alguien logre acceder a los mismos permisos de quien maneja la base de datos, podría ver esta información y darle un mal uso.

Utilización de algoritmos propios y privados

Este es un punto débil en cuanto al método de cifrado, pues la fortaleza de un algoritmo no debería depender de su clandestinidad. De esta manera, el hecho de que sean propios y privados, da a entender que pueden ser débiles. Esto significa que atacantes podrían descifrar la información, robarla y cometer fraude con estos datos. También, un integrante de la misma empresa podría divulgar este algoritmo, comprometiendo la integridad y la confidencialidad de la información.

Carencia de niveles de defensa

Un punto importante del sistema es que no se menciona la existencia de firewalls, anti-malware, IDS (Intrusion detection system), ni de otros sistemas de seguridad de esta índole. Esto permite que hayan ataques al sistema y accesos no autorizados que comprometen la integridad y la confidencialidad de los datos.

Conexión a internet

El hecho de que una aplicación cuente con conexión a internet siempre representa una vulnerabilidad, pues este es un medio por el cual pueden ingresar terceros al sistema, aumentando las posibilidades de que hayan ataques, comprometiendo la seguridad en general del sistema.

Propuesta de soluciones

Web spoofing

Para mitigar esta amenaza, se recomienda que el sistema cuente con firewalls y anti-malware que evite que este tipo de intrusiones ocurran. Así, el firewall evitaría el acceso a personas o aplicaciones que representen un peligro para la seguridad de la información. Asimismo, esto se podría mitigar aislando la aplicación de internet, sin embargo como esta lo requiere para asegurar la comunicación entre las distintas sedes, no sería la mejor solución.

Tampering

Utilizar métodos de cifrado que aseguren la integridad entre la comunicaciones con otros usuarios y el servidor sería la mejor opción para detectar que la información ha sido adulterada. Esto se podría garantizar con un método donde el mensaje llegue concatenado con su digest, para que al recibir el mensaje se aplique la función hash al mensaje y se compare el digest obtenido con en el que viene concatenado.

Espionaje en los medios de transmisión

Esta amenaza se podría mitigar haciendo uso de algoritmos públicos y open source. Esto se debe a que si se hace uso de algoritmos ya reconocidos por su fortaleza ante intentos de descifrado fortalece la seguridad en general. Asimismo, un método de cifrado seguro entre el servidor y los usuarios sería con un sobre digital. Donde la llave de sesión se cifran con la llave pública del receptor, haciendo del mensaje y futuras peticiones confidenciales para terceros que puedan estar interesados en ver estos datos.

Elevación de privilegios

Para evitar que un usuario encuentre la contraseña de otro a fuerza bruta, se debería crear un esquema de manejo que no permita contraseñas débiles. Esto podría funcionar evitando que

se tengan contraseñas cortas, de solo letras o solo números y que tenga la misma contraseña por más de cierto periodo de tiempo. Así, tratar todas las combinaciones posibles no sería una solución viable para lograr acceder a una cuenta con más privilegios.

Denegación de servicios

Para evitar este tipo de ataques al sistema, se puede instalar software que se dedique al monitoreo de actividades ejecutadas, de tráfico de red y mantener logs de quien modifica o solicita cosas. En caso de que una misma IP esté realizando peticiones de manera exagerada, se debería poder bloquear. De la misma manera, se debería mejorar la escalabilidad del programa para que soporte una mayor cantidad de usuarios y de memoria, mejorando los recursos con los que se dispone.