PREGUNTA 1

El concejo deliberante del partido de Tecnociudad solicitó la construcción de un sistema distribuido que permita realizar licitaciones de forma transparente y segura. Cuando el partido necesita contratar cualquier tipo de servicios, genera un pliego con la descripción de lo solicitado y espera que distintos contratistas presenten ofertas. Durante esta fase, los costos son secretos para evitar que un oferente tenga preferencia. Una vez vencido el plazo, se publica la propuesta de cada uno, junto con el precio, y se selecciona el que cumple con las condiciones con el menor costo.

La implementación del sistema asegura que estas propiedades ocurran utilizando criptografía. La descripción a alto nivel es la siguiente:

- Se publica la licitación con un ID único asociado.

- Por cada licitación, se reciben archivos cifrados con AES-GCM de parte de los oferentes interesados. Los archivos incluyen un documento con el ID de la licitación, la descripción técnica, el CUIT de la empresa y el precio.

- Al terminar el periodo de la licitación los oferentes comparten la clave utilizada, y con ello se revisan todas las ofertas, y se adjudica la ganadora.

- Si un oferente no provee la clave, queda automáticamente descalificado.

- Para transparencia, se publican todas las ofertas recibidas ya descifradas y la adjudicada en un portal público que puede ser consultado libremente.

a) Explicar por qué ningún oferente puede alterar su propuesta sin ser detectado. (1 pto.)

b) Qué medidas adicionales serán necesarias para evitar que algún administrador del sistema ignore a un proveedor diciendo que nunca entregó la clave (1 pto.)

c) Considerar el caso de un oferente que envía un archivo a nombre de un competidor, con un precio irrisoriamente bajo que lo perjudica. ¿Cómo evitar ésto usando funciones criptográficas? (1 pto.)

PREGUNTA 2

El siguiente protocolo fue propuesto para que N nodos de un sistema puedan elegir un supervisor, de tal manera que ningún nodo pueda ganar o perder sistemáticamente manipulando el intercambio:

1) Cada nodo genera un valor ri de 64 bits y envía al resto H(ri)

2) Luego de recibir todos los valores, cada nodo publica ri

3) Cada nodo calcular rx =r1 xor r2 xor .... xor rn

El nodo a menor distancia de rx es el nuevo supervisor, utilizando la distancia de Hamming como métrica (se comparan bit a bit los dígitos binarios de los dos valores, y cada discrepancia suma 1 de distancia - por ejemplo 0010 y 1110 están a distancia 2). En caso de empate, se reinicia

a) Explicar por qué el paso 1 es necesario y no se puede comenzar el intercambio en el paso 2 directamente (1 pto.)

b) Considerar el caso de solo tres nodos. ¿Qué pasaría si el nodo 2, por ejemplo, decide abusar del sistema, y repite en el paso 1.el valor de H(r1) como si fuese el suyo? Proponer una modificación simple que evite el problema (1,5 ptos.)

c) Proponer una mejora al algoritmo para que no sea necesario repetir todo el intercambio en caso de empate y siempre se llegue a un resultado que no pueda ser manipulado. (0.5 ptos).

PREGUNTA 3

Agrolin S.A. Se dedica a la construcción de soluciones tecnológicas para el campo.

Recientemente se publicó un informe que menciona múltiples vulnerabilidades en uno de sus productos y provocó muchas pérdidas de ventas a punto de concretarse. Para corregir la situación, la empresa decidió contratar un servicio de pen testing, realizar una evaluación más exhaustiva y corregir el producto para empezar a recuperar el daño a su imagen.

El producto en cuestión permite hacer un seguimiento del rendimiento de las cosechas a lo largo del tiempo y determinar las rotaciones óptimas para mejorar la productividad. El sistema está compuesto por un frontend web que es un single page web app, donde el usuario puede registrarse, autenticarse, dar de alta lotes marcando sus límites en un mapa, actualizarlos o eliminarlos. El sistema muestra según datos históricos de todos los clientes de la zona, el rinde esperado para la superficie en cuestión, sugiere el tipo de cultivo con mayor rentabilidad en función de los valores de rinde y precios de mercados futuros estimados, y permite que el usuario ingrese los datos del tipo de cultivo y luego de la cosecha el rinde obtenido para comparar y poder mejorar en las próximas sugerencias.

La aplicación web se ejecuta en los navegadores de cada cliente, y hace llamadas a un servidor backend a través de un api REST.

Siguiendo la metodología de hipótesis de falla, establecer 4 hipótesis de vulnerabilidades que tengan alta probabilidad de existir en el escenario descrito. Para cada una, describir únicamente la hipótesis y la prueba que realizan para confirmar o refutar (1 pto. por hipótesis).

Aclaraciones: No se considerarán válidas hipótesis genéricas. Tampoco se consideran dos hipótesis que sean variantes del mismo tipo de falla (por ejemplo, el mismo problema en dos pantallas distintas).

Ejercicio 1

a. El cifrado autenticado AES-GCM devuelve el cifrado junto con el tag. Si un atacante altera su propuesta, se altera tanto el cifrado como el tag y se podría constatar que los tags no coinciden.

b. Una vez que el administrador recibe la clave del proveedor, le envía una Sign(clave oferente) con su clave privada al proveedor. Así, el oferente tiene prueba de que envió su clave y que el administrador la recibió.

c. Se debe pedir al proveedor que firme el archivo bajo su clave privada. Así, el proveedor no podrá impersonar a otro proveedor dado que no tiene su clave privada.

Ejercicio 2

a. El paso 1 es necesario para que los nodos no puedan cambiar su r_i en el paso siguiente. En caso de que algun nodo lo haga, se chequea el hash y se puede comprobar que hubo alteracion.

b. Si el nodo 2 envia $h(r_1)$, entonces $r_x = r_1 \text{ xor } r_1 \text{ xor } r_3 = r_3$. El nodo 2 puede manipular la eleccion del nodo supervisor. Una modificacion seria impedir que los hashes sean iguales. Dado que las funciones de hash son deterministicas, si dos nodos publican el mismo hash implica que estan enviando el mismo r_i. En caso que dos nodos envien el mismo hash, se puede pedir a los nodos que reenvien sus r_i.

c. Suponiendo que se cumple el punto anterior y que $r_i \neq r_j \ \forall i \neq j$, se puede hacer un nuevo xor entre los nodos que empataron (con los mismos valores) y volver a evaluar r_x.

Ejercicio 3

(1) Improper Authentication

Amenazas: tampering, information disclosure y denial of service

Esto ocurre cuando la implementacion de funciones de autenticacion no es correcta. En este caso, si no se maneja bien la autenticacion, un atacante puede hackear la cuenta de otro usuario.

Pruebas: Probar tanto del client side como del server side

- Intentar registrarse con contraseñas simples y ver si el sistema las acepta
- Probar usuarios y contraseñas comunes
- Probar contraseñas por fuerza bruta
- Intentar acceder a recursos que requieren autenticacion pero sin estar autenticado

(2) SQL Injection

Amenazas: tampering, spoofing e information disclosure

Esto se da cuando el sistema no sanitiza el input de los usuarios y los pega directamente a instrucciones SQL.

Si este es el caso, un atacante puede modificar la informacion en la base de datos o incluso acceder a informacion sensible.

Pruebas: Probar tanto del client side como del server side

- Ingresar a' OR '1=1 en la contraseña de una cuenta previamente registrada
- Probar obtener informacion de la base de datos en todos los inputs accesibles a usuarios

(3) Cross Site Scripting

Amenazas: information disclosure, denial of service y tampering

El xss es una vulnerabilidad que ocurre cuando no se sanitizan los campos de entrada, aceptando el envio de scripts completos.

Pruebas: en un ambiente de prueba para no crashear el sistema

Probar <script> alert("Error") </script> en los inputs accesibles por los usuarios y ver si se rompe la aplicacion.

(4) Buffer Overflow

Amenazas: tampering, information disclosure y denial of service

Esto ocurre cuando el sistema copia una cantidad de datos sobre un espacio que no es lo suficientemente grande para almacenarlos. Un atacante podria ingresar, en los campos de entrada, datos muy grandes y provocar fallas en el sistema o incluso obtener informacion sensible.

Pruebas: en un ambiente de pruebas

Probar ingresar datos muy grandes o datos de tipo incorrecto (Ej. texto en el rinde obtenido) y ver si el sistema falla.