Ejercicio 1

Enunciado

Una empresa está desarrollando un chip que puede insertarse bajo la piel para exponer información de cualquier tipo. Sus usos iniciales están asociados a información de contacto ante emergencias e información médica, pero el chip puede ser utilizado para almacenar cualquier tipo de información.
Para ello, el chip cuenta con 32 compartimentos de 1024 bytes. Cada compartimiento está asociado a una clave pública de un par de claves generados aleatoriamente (o sea, hay 32 claves públicas, junto con sus 32 claves privadas – el chip tiene en su memoria las 32 públicas, cada una asociada a un único compartimento). Todos los chips tienen las mismas claves.
Cuando alguien “compra” uno de los 32 espacios del chip, obtiene el par de claves asociados a ese compartimiento.

Para escribir el compartimiento, el chip debe recibir utilizando una transmisión por proximidad (NFC) una trama con el siguiente formato: Header (4 Bytes) || Size (2 Bytes) || Data (Variable) || Signature (20 Bytes)
Donde:
Header son 4 bytes con valor fijo compuesto por los valores ASCII de las letras “MTRX”
Size es un número entre 0 y 1024 indicando la cantidad de información a guardar
Data es la información a guardar.
Signature es una firma digital utilizando DSS con la clave privada correspondiente de todo el mensaje hasta el byte anterior a la firma inclusive.

Para leer un compartimiento, el chip debe recibir utilizando una transmisión por proximidad (NFC) una trama con el siguiente formato: Header (4 Bytes) || Offset (2 Bytes) || Size (2 Bytes) || Signature (20 Bytes)
Donde:
Header son 4 bytes con valor fijo compuesto por los valores ASCII de las letras “RCAL”
Offset es un número entre 0 y 1023 indicando desde qué posición leer en el compartimiento
Size es un número entre 1 y 1024 indicando cuánto leer
Signature es una firma digital utilizando DSS con la clave privada correspondiente de todo el mensaje hasta el byte anterior a la firma inclusive.

a. Indicar paso a paso que debe hacer el programa dentro del chip para validar un pedido de lectura y ejecutarlo si solo debe permitirse leer el contenido de un compartimento a su dueño (1 pto.)
b. Si suponemos que el dato a escribir en una operación de escritura no viaja cifrado, proponer una mejora en el protocolo que permita garantizar la confidencialidad de dicho campo. (1 pto.)
c. Si suponemos que los datos leídos vuelven sin ningún tipo de cifrado, proponer un cambio al protocolo en la operación de lectura que permita garantizar la confidencialidad de los datos, y que no requiera que el chip tenga que conocer de antemano más claves que las que ya tiene en el diseño original. (1 pto.)

Nota: considerar que se trata de un protocolo ad-hoc para un sistema embebido, así que soluciones complejas como TLS están fuera del alcance

Ejercicio 2

Enunciado

Considerar el problema conocido como Private Information Retrieval (PIR), donde se busca que una parte consulte información de un servidor (imaginar un registro en una base de datos), sin que este último pueda saber cual es la información consultada.
Está demostrado que la única implementación con garantías absolutas de garantizar esa propiedad consiste en que el servidor envía en cada requerimiento la base de datos completa y que el cliente tome el registro que le interesa. Claramente esto no es muy práctico.
Una implementación de PIR con algunas restricciones consiste en contar con N servidores independientes, donde cada entrada de la base de datos se cifra utilizando un criptosistema de umbral con parámetros (k, N). Para guardar una entrada nueva, cada servidor almacena una sombra del secreto compartido.
Para recuperar una entrada, el cliente elige k servidores al azar y les solicita la sombra asociada a la entrada que quiere recuperar, reconstruyendo localmente el secreto.

a. ¿Por qué desde el punto de vista de un servidor no es posible saber qué información fue solicitada? (1 pto.)
b. ¿Cuál debe ser la relación entre k y N para garantizar que el compromiso de la un cuarto de los servidores no afecte la privacidad de los requerimientos y la caída hasta un tercio de los servidores no afecte la disponibilidad de la información en el modelo teórico? Enumerar todos los pares (k, N) que satisfacen esa restricción para un cluster de 12 nodos. (1 pto.)
c. ¿Cómo puede un cliente, luego de haber consultado a todos los servidores necesarios, estar seguro que está reconstruyendo el valor correcto? Explicar siguiendo el escenario donde uno de los servidores contesta con un valor que no es el correcto. (1 pto.)

Ejercicio 3

Enunciado

ElBuenVestir S.A., una pequeña empresa dedicada a la confección de piezas de indumentaria de alta costura, decide ofrecer sus servicios de forma virtual.
Para ello contrata el desarrollo de una aplicación móvil que permita que un futuro cliente elija el tipo de prenda deseada, en función de la prenda se le vaya guiando en que medidas debe tomar, y luego de ingresarlas pueda ver un estimado del tiempo de realización (que se calcula en función del resto de los trabajos aceptados) y pueda pagar una seña para confirmar la compra.

El sistema resultante consiste en un servidor que corre en una cuenta a cargo de la empresa que desarrolló el software, un par de aplicaciones móviles para las plataformas más utilizadas (Android/iOS), y una aplicación web de backoffice, utilizada por los empleados de ElBuenVestir para administrar el sistema, tomar los pedidos y actualizar el estado de cada trabajo.
Siguiendo la metodología de hipótesis de falla, establecer 4 hipótesis de vulnerabilidades que tengan alta probabilidad de existir en el escenario descripto. Por cada una describir solamente la hipótesis y la prueba que realizaría para confirmarla o refutarla.

Incluir al menos 1 hipótesis sobre cada componente.

Ejercicio 1

a. Primero, el programa debe verificar que que compartimento se esta intentando leer. Entonces, el programa hace $Vrfy_{PK}(Signature)$ para todos los compartimentos. Luego, el compartimento que se quiere leer es aquel para el cual $Vrfy_{PK}(Signature) = 1$. Una vez encontrado el compartimento, se verifica que el header contenga los 4 bytes correspondientes a los valores ASCII de las letras "RCAL" Finalmente, nos movemos x bytes del offset y devolvemos la cantidad de bytes a leer.

b. Lo que se puede hacer es modificar el comportamiento de los chips. Cada chip tendra dos claves, una publica y una privada. Luego, cuando alguien compra uno de los compartimentos, obtiene dos pares de claves, uno para autenticacion y el otro para confidencialidad. Al mandar un mensaje para escribir, el dueño encripta con la clave publica y el chip desencripta con la privada.

c. Como no se debe utilizar la misma clave para autenticacion y confidencialidad, el dueño puede agregar una clave publica cuando solicite una lectura. Entonces, el programa primero valida que la solicitua proviene del dueño mediante la signature. Luego, encripta el mensaje con la clave publica y el dueño la desencripta con su clave privada.

Ejercicio 2

a. Un servidor no podría reconstruir el secreto (a menos que $k = 1$ pero no cumpliría PIR). Esto se debe a que se necesitan k servidores para construir el secreto.

b. Por un lado, k debe ser mayor estricto al numero de servidores que pueden estar comprometidos para que no puedan reconstruir el secreto $\Rightarrow k > N/4$. Por otro lado, en caso de que se caigan $N/3$ nodos, k debe ser menor estricto a $N - N/3 = 2N/3$ i.e N menos la cantidad de nodos caídos.

Si $N = 12$: $(4, 12)$, $(5, 12)$, $(6, 12)$ y $(7, 12)$

c. El cliente podría pedir a 3 combinaciones distintas de servidores y comparar los secretos. Aquellos dos secretos que sean iguales están devolviendo el correcto.

Obs: se asume que N es al menos 2 unidades mayor que k

Ejercicio 3

(1) Improper Authentication

Amenazas: tampering, information disclosure y denial of service

Esta vulnerabilidad ocurre cuando las funciones de autenticacion no estan implementadas correctamente. En este caso, un atacante podria acceder como empleado y administrar el sistema. Tambien puede ocurrir que las aplicaciones moviles no manejen bien la autenticacion y un atacante tome control de la cuenta de otro usuario.

Pruebas: tanto en la aplicacion web como en las aplicaciones moviles

- Registrarse con contraseñas simples (Ej. 12345678) y ver si la aplicacion las acepta
- Probar usuarios y contraseñas comunes
- Intentar acceder a la aplicacion web sin autenticarse

(2) 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 contenerlos. En este caso, si el servidor no contempla estas situaciones, un atacante podria pasar en alguna entrada un texto extremadamente grande y romper el sistema.

Pruebas: en un ambiente de pruebas para no romper el sistema

Hacemos un posible POST enviando valores muy grandes como parametros y observar que nos devuelve el sistema

(3) SQL Injection

Amenazas: information disclosure, tampering y spoofing

Esto se ocasiona cuando no se hacen los chequeos necesarios a los inputs de los usuarios. Un atacante podria acceder a la cuenta de otro usuario u obtener informacion sensible mediante un SQL injection.

Pruebas:

- Ingresar algo como a' OR '1 = 1 en la contraseña y ver si el sistema nos deja acceder
- En todos los inputs accesibles por los usuarios, probar algun SQL Injection que nos permita obtener informacion (intentar no borrar datos)

(4) Improper Authorization

Amenazas: tampering e information disclosure

Esta vulnerabilidad se da cuando el sistema verifica mal los accesos a los recursos en base a los privilegios y permisos de los usuarios. En este caso, si el servidor no hace los controles necesarios, un atacante podria pegarle directamente al servidor y hacer cosas que no tendria que poder hacer (Ej. tener acceso a informacion de otros usuarios).

Pruebas:

- Hacer accesos directos a todos los endpoints que deben estar restringidos
- Intentar acceder a recursos no permitidos por los distintos roles