Final Cripto 10-07-23

Ejercicio 1

Una empresa tiene un sistema donde hay varios sensores que comunican a un dispositivo maestro sus diferentes niveles de sensado. Intercambian claves con Diffie Helman en su primera comunicación, y a partir de eso en todos los mensajes que los sensores envían (encriptados con AES-CBC) tienen la siguiente estructura:

$$\#sensor||e_k(\#sensor||valor\_sensado||datos\_adicionales)$$

1. Hay un atacante que hace de alguna forma que los mensajes se alteren y tengan datos adicionales no validos, generando que el dispositivo maestro tire una señal de alerta. Cómo es posible? Encontrar una manera de solucionar este error en el protocolo sin nuevas claves. (1,5 puntos)
2. Hay otro atacante que está enviando repetidamente la señal de sensado de un sensor, haciendo que el dispositivo maestro registre niveles de sensado que no son correctos. Cómo es posible? Encontrar una forma de modificar el mensaje para que esto no ocurra sin usar nuevas claves. (1,5 puntos)

Ejercicio 2

Se quiere enviar un conjunto de datos muy grande. Si bien se puede mandar la información junto con un MAC de la información entera, en caso de que una parte del mensaje tenga un error, es necesario mandar el mensaje entero de vuelta. Se buscan alternativas para MAC, donde se parte la data en bloques más pequeños, de forma que permitan no tener que enviar la información entera en caso de una falla en uno de los bloques. Las alternativas son las siguientes:

1. En vez de calcular el MAC de cada bloque, se calcula el MAC’ que es: el contenido del bloque anterior concatenado al MAC del bloque actual. Excepto en el bloque cero, que como no tiene anterior, se ponen 0s. (1 punto)
2. Se calcula el MAC de la concatenación de los MAC de cada bloque. (1 punto)
3. Se calcula el MAC usando el árbol de Merkle (buscar ejemplo de arbol de Merkle). (1 punto)

Para cada caso evaluar si tiene integridad y en caso de que sí, cómo harían cuando un bloque se envía con un dato erróneo (si es que hay una alternativa a no enviar todo el mensaje de vuelta)

Ejercicio 3

Se tiene una página de noticias donde hay usuarios que pueden ratear una noticia positiva o negativamente. Los publicadores de noticias pueden subir noticias pegándola a una API REST. Los usuarios interactúan con la página a través de una página web que se comunica con la API. Los ingresos de las publicidades de las noticias se pagan a los creadores según la cantidad de visitas que tiene una noticia.

Se encontraron 4 vulnerabilidades:

1. Para evaluar la positividad de una noticia, los usuarios tocan un botón que hace un GET a `/{news_id}/positive` o `/{news_id}/positive/negative`, pero se encontró que se puede hacer SSRF a través de esto.
2. El contenido de una noticia se postea tal cual como se recibe, sin hacer validaciones.
3. Las cookies de sesión no son http_only, y se pueden acceder a través de javascript.
4. Un publicador de noticia hace un llamado al backend con una API KEY que se introduce en el pedido de la siguiente forma: `/api/{API_KEY}/publish`

a. Explicar como podría un atacante sumarse dinero de forma maliciosa con los puntos 2 y 3. (1,5 puntos)

b. Explicar cómo tomaría ventaja un atacante de la vulnerabilidad 1. (1,5 puntos)

c. Explicar por qué es inseguro que se envíen las API KEYS como lo indica el punto 4 e indicar cómo lo corregirían (1 punto)