SAD - U3 Seguridad Lógica

Descargar estas actividades

Índice

Actividades Seguridad Lógica

Actividad 1. Ataques a contraseñas

Una buena opción para acortar los tiempos de los ataques a las contraseñas es tener algún indicio del formato de las mismas.

Partiendo del archivo **hashes_practicas.txt** que contiene los hashes de las contraseñas de los usuarios `user1.x`, `user2.x` y `user3.x`, intenta obtener sus contraseñas.

Realiza un ataque a las contraseñas, teniendo en cuenta que las contraseñas tienes las siguientes propiedades:

- **user1.x** se obtiene con un ataque de diccionario
- **user2.x** se obtiene con un ataque por fuerza bruta con máscaras, sabiendo que están formadas por 3 minúsculas seguidas de 3 dígitos, por ejemplo "abc123"
- **user3.x** se obtiene con un ataque por diccionario con máscaras, sabiendo que las contraseñas están formadas por una palabra del diccionario (500_passwords.txt) seguidas de un año comprendido entre 2000 y 2019.

Documéntalo todo con capturas de pantalla en las que se vea el comando y los resultados producidos.

Actividad 2. Contraseñas de Sistemas Operativos

Hemos conseguido sacar esta información del archivo SAM de Windows

100:AE4D4025B89026B533A46849C79CEE3D:7FFB9A84B18B17F66DA382F2C2FEC342:::

y esta contraseña de diccionario sacada de un sistema Linux

600.zqUv8$nAOCHqjtXJ8QjPIFlXdZes604kCXGlPqypNh5ON/McDRxHn7Mip3dx3gaSLIaE9ieRJaPvjUpq9KD5bmUkRue/

¿De qué formas podemos romper estas contraseñas?

La esteganografía es la práctica de ocultar un mensaje secreto dentro (o incluso encima) de algo que no es secreto. Ese algo puede ser casi cualquier cosa que desees. Por ejemplo; ocultar un mensaje secreto o un script dentro de un documento de Word o Excel.

Hay tres aspectos principales para la ocultación de la información:

- **Capacidad**: cantidad de información que se puede ocultar.
- **Seguridad**: Dificultad para detectar información oculta.
- **Solidez**: modificaciones que el medio de cobertura puede resistir antes de que la información oculta se corrompa.

Los principales tipos de esteganografía son:

- **Pura**: no requiere el intercambio de un cifrado como un stego-key. Se asume que ninguna otra parte tiene conocimiento de la comunicación.
- De **clave secreta**: la clave secreta (stego) se intercambia antes de la comunicación. Esto es más susceptible a la intercepción. Solo las partes que conocen la clave secreta pueden revertir el proceso y leer el mensaje secreto.
- De **clave pública**: se utiliza una clave pública y una clave privada para una comunicación segura. El remitente utilizará la clave pública durante el proceso de codificación y solo la clave privada, que tiene una relación matemática directa con la clave pública, puede descifrar el mensaje secreto.

En StegOnline puedes encontrar una guía de las diferentes técnicas esteganográficas y ejemplos de cómo se pueden utilizar.

- a) Descubre los secretos ocultos en las técnicas que tienen una "Custom Image para probar"

Actividad 2. Criptografía híbrida

La criptografia simètrica és més insegura ja que el fet de passar la clau és una gran vulnerabilitat, però es pot xifrar i desxifrar en menor temps del qual tarda la criptografia asimètrica, que és el principal inconvenient i és la raó per la qual existeix la criptografia híbrida.

La criptografía simétrica es más insegura ya que el hecho de pasar la clave es una gran vulnerabilidad, pero se puede cifrar y descifrar muy rápidamente grandes cantidades de datos. Ese es el principal inconveniente de la criptografía asimétrica. Para aprovechar lo mejor de las dos técnicas, existe la criptografía híbrida.

Para ello, la comunicación se divide en dos partes: **intercambio de claves** y **intercambio de mensajes**.

El proceso para usar un sistema criptográfico híbrido es el siguiente (para enviar un archivo cifrado a un compañero):

Intercambio de claves

1. Generar una clave pública y otra privada (en el receptor): `privHibrida_tunombre.pem` y `pubHibrida_tunombre.pem`.
2. Guardar una clave simétrica en un archivo: `claveSimetrica.txt`.
3. Cifrar el archivo claveSimétrica.txt con la clave pública de un compañero: `claveSimetrica.aes`.
4. Enviar el archivo cifrado `claveSimetrica.aes` al compañero al que pertenece la clave pública que hemos usado para cifrar.

Intercambio de mensajes

1. Descifrar el archivo recibido `claveSimetrica.aes` con nuestra clave privada.
2. Obtener la clave simétrica `claveSimetrica.txt`. **Ahora los dos sabemos la clave simétrica**.
3. Cifrar el archivo `claveSimetrica.txt` con un algoritmo de clave simétrica, usando la clave recibida: `claveSimetrica2.aes`.

4. Enviar el archivo cifrado `claveSimetrica2.aes` al compañero que generó la clave.
5. El compañero descifra `claveSimetrica2.aes` con la clave que creó al principio: `claveSimetrica2.txt`.
6. Compara la clave obtenida con la que generó inicialmente.
7. Si son iguales, ya puede realizar un intercambio de archivos, sin importar el tamaño de los mismos, usando criptografía simétrica, con la clave intercambiada.

a) Realiza y documenta gráficamente este proceso con un compañero de clase. Incluye capturas de pantalla de los archivos generados y de los pasos realizados.

Las capturas de pantalla deben incluir elementos que permitan identificar que tú has realizado el proceso, como el nombre de la máquina, un nombre de directorio, etc.

Actividad 3. Funciones resumen

Crea una tabla con las funciones resumen más utilizadas en criptografía.

La tabla debe tener al menos la siguiente información:

- Nombre del algoritmo
- Fecha de creación
- Tamaños de salida
- Uso más común
- Segura: Si se ha encontrado alguna vulnerabilidad en el algoritmo y cuándo se descubrió
- Ejemplo de uso con un texto de ejemplo (el mismo texto para todas las funciones) con OpenSSL
- Otros datos que consideres relevantes

Actividad 4. Algoritmos simétricos

Crea una tabla con los algoritmos simétricos más utilizados en criptografía.

La tabla debe tener al menos la siguiente información:

- Nombre del algoritmo
- Fecha de creación
- Tamaños de clave
- Uso más común
- Seguro: Si se ha encontrado alguna vulnerabilidad en el algoritmo y cuándo se descubrió
- Ejemplo de uso con un texto de ejemplo (el mismo texto para todos los algoritmos) con OpenSSL
- Otros datos que consideres relevantes

Actividad 5. Algoritmos asimétricos

Crea una tabla con los algoritmos asimétricos más utilizados en criptografía.

La tabla debe tener al menos la siguiente información:

- Nombre del algoritmo
- Fecha de creación
- Tamaños de clave
- Uso más común
- Seguro: Si se ha encontrado alguna vulnerabilidad en el algoritmo y cuándo se descubrió
- Ejemplo de uso con un texto de ejemplo (el mismo texto para todos los algoritmos) con OpenSSL

- Otros datos que consideres relevantes

Actividad 6. Algoritmos

Completa la siguiente tabla marcando una X en la columna correspondiente a las funcionalidades que proporciona cada algoritmo.

Algoritmo	C. Simétrico	C. Asimétrico	F. Resumen	Passwd Linux	Passwd Windows	Firma digital
RSA						
AES						
SHA-256						
MD5						
3DES						
DSA						
Blowfish						
El-Gamal						
ECC						

Casos prácticos

Caso práctico 1. Cifrado de unidades de almacenamiento

Bitlocker es una herramienta de cifrado de unidades de almacenamiento que se utiliza en sistemas Windows.

En sistemas Linux, se puede utilizar **LUKS** (Linux Unified Key Setup) para cifrar unidades de almacenamiento o bien **Dislocker** para acceder a unidades cifradas con Bitlocker en sistemas Linux.

Otra herramienta interesante es **VeraCrypt**, una herramienta de cifrado de código abierto que se puede utilizar en Windows, Linux y macOS.

a) Investiga cómo usar Dislocker para cifrar una unidad USB con Bitlocker y cómo montarla en un sistema Linux para acceder a los datos de la unidad cifrada. Este es un mecanismo muy útil para compartir datos entre sistemas Windows y Linux de forma segura.

b) Documenta los pasos necesarios para crear un contendedor cifrado con VeraCrypt en Windows y montarlo en una unidad del sistema.