Desafíos criptográficos

Este sitio contiene una serie de desafíos criptográficos que corresponden a la asignatura Criptografía de la Diplomatura en Ciberseguridad de la UNC.

Los desafíos son los siguientes:

Desafío Inicial

`MD5Crypt` es un algoritmo utilizado en la biblioteca ADOdb de PHP. Es muy fácil de quebrar. El desafío consiste en descifrar un mensaje cifrado con dicho algoritmo.

Instrucciones para el desafío

Aleatoriedad

Generador de números pseudoaleatorios de Java

La clase `java.util.Random` utiliza un *generador congruencial lineal*, que no tiene finalidades criptográficas (para ello existe `java.security.SecureRandom`). El desafío consiste en adivinar el siguiente número a producir por el generador.

Instrucciones para el desafío

Clave generada a partir de la fecha y hora

Existen numerosos ejemplos de sistemas quebrados porque utilizan alguna información temporal provista por la máquina como forma de inicializar el generador de números pseudoaleatorios. El desafío consiste en descifrar un mensaje cifrado con una clave generada de esta forma.

Instrucciones para el desafío

Cifrado de flujo

Textos cifrados con la misma secuencia cifrante

Si en un cifrado de flujo se reutiliza la secuencia cifrante, es posible eliminarla realizando el o-exclusivo de los textos cifrados. Este desafío consiste en obtener la secuencia cifrante utilizada para cifrar varios mensajes.

Instrucciones para el desafío

Cambio de bits en el texto cifrado (*bit flipping*)

Si en un texto cifrado producido por un cifrador de flujo se cambia un bit, el texto claro obtenido al descifrar estará alterado en ese mismo bit. Esto permite producir alteraciones predecibles del texto claro si uno conoce su estructura. El desafío consiste en cambiar el texto cifrado de manera de aumentar los privilegios de un usuario.

Instrucciones para el desafío

Cifrado de bloques

Falsificación en modo ECB

El modo ECB es maleable, en cuanto es posible intercambiar o intercalar bloques. Esto permite producir alteraciones predecibles del texto claro si uno conoce su estructura. El desafío consiste en cambiar el texto cifrado de

manera de aumentar los privilegios de un usuario.

Instrucciones para el desafío

Cambio de bits en modo CBC (*bit flipping*)

Si en un texto cifrado en modo CBC se cambia un bit, el bloque correspondiente del texto claro obtenido al descifrar estará completamente cambiado, pero el bloque siguiente estará alterado en ese mismo bit. Esto permite producir alteraciones predecibles del texto claro si uno conoce su estructura. El desafío consiste en cambiar el texto cifrado de manera de aumentar los privilegios de un usuario.

Instrucciones para el desafío

Descifrado en modo ECB

El desafío consiste en descifrar, mediante un ataque de texto claro elegido, un mensaje secreto.

Instrucciones para el desafío

Funciones de hash

Búsqueda de una segunda preimagen

El desafío consiste en encontrar una segunda preimagen en una función de hash de 24 bits.

Instrucciones para el desafío

Búsqueda de una colisión

El desafío consiste en encontrar una colisión en una función de hash de 48 bits.

Instrucciones para el desafío

Ataque de extensión de longitud

El desafío consiste en encontrar una falsificación de un MAC basado en la aplicación de una función de hash sobre un mensaje concatenado con un prefijo secreto.

Instrucciones para el desafío

CBC-MAC

El desafío consiste en encontrar una falsificación para un mensaje autenticado con CBC-MAC.

Instrucciones para el desafío

Padding Oracle

El desafío consiste en descifrar un mensaje cifrado con AES en modo CBC, utilizando un servidor que funciona como *padding oracle*.

Instrucciones para el desafío

RSA Broadcast

El desafío consiste en descifrar un mensaje cifrado con RSA en modo *libro de texto* y con distintas claves públicas.

Instrucciones para el desafío

RSA con clave pequeña

El desafío consiste en descifrar un mensaje cifrado con RSA con una clave con módulo pequeño, y por lo tanto factorizable.

Instrucciones para el desafío

DSA con reutilización de `k`

El desafío consiste en encontrar la clave privada utilizada para firmar mensajes en una implementación defectuosa que reutiliza `k`.

Instrucciones para el desafío