WRITEUP: El despistado del Juancho
Categoría: Crypto

Enunciado

Nos hemos encontrado un mensaje cifrado de un viejo amigo, el cual tiene mala memoria y es un poco descuidado a veces. Juancho fue criptógrafo antes que marinero. Antes de cruzar el Mar Angosto, situado entre el continente de Poniente y el continente oriental de Essos, trabajó para una gran empresa de ciberseguridad informática. En la empresa de Juancho empleaban los algoritmos simétricos DES (con clave de 8 bytes) y AES (con clave de 16 bytes).

No sabe cual de los dos algoritmos utilizó para cifrar el mensaje, y por tanto no se acuerda de si la clave era de 8 bytes o 16 bytes. Nos contó que como clave para el cifrado se basó en una palabra que le gustó mucho y que obtuvo del contenido en `https://sugusuva.es/`. Recuerda también que dicha palabra tenía al menos 7 caracteres. Esa palabra no sólo la utilizó como clave (key) para el algoritmo de cifrado, sino que también la utilizó como *passphrase* para comprimir un fichero ZIP (el cual contiene el mensaje secreto **cifrado**). ¡Qué tío! ¡Cómo le gusta reutilizar! ¡Este es de Greenpeace seguro!

Para el cifrado DES dicha palabra le cuadraba perfectamente: eran justo 8 bytes. Lo que nos contó también es que como la clave para el cifrado DES era de 8 bytes en Latin-1, si hubiese utilizado AES, simplemente hubiera duplicado la palabra para generar la clave. De esta forma, la clave para el algoritmo AES sería: "palabrapalabra". Finalmente nos confesó que era un poco descuidado y que como vector de inicialización (IV) siempre ponía tantos ceros como requiriese el algoritmo.

Información adicional sobre los 'ingredientes' del algoritmo de cifrado:

Key: codificación Latin-1

IV: codificación Latin-1 y son 8 o 16 ceros, en función del algoritmo que se haya usado

Modo de operación: cifrado por bloques (CBC)

Juancho es muy campechano y dice que siempre que quiere cifrar o descifrar un mensaje, tanto el propio mensaje en texto plano como el texto cifrado van en crudo (raw). No se anda con bobadas.

Resolución

El primer paso para resolver este reto es organizar y anotar toda la información que nos proporciona el enunciado. Nos desvela cuál es el algoritmo correcto (DES o AES) con el que se encriptó el mensaje, así como información relativa a varios componentes del mismo.

Una de las primeras cosas que podemos intentar hacer para descomprimir el ZIP, sabiendo que la *passphare* está en la web de SUGUS y que tiene una longitud mínima de 7 caracteres, es lanzar el siguiente comando:

```
cewl https://sugusuva.es -w diccionario.txt
```

Observaríamos que nuestro diccionario estaría vacío, a pesar de que la web muestre varias palabras en su contenido. Revisando el código fuente de `https://sugusuva.es` nos damos cuenta de

la redirección de contenido que existe.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="front\src\img\sugus_logo_mini.jpg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>SUGUS UVa</title>
    <script type="module" crossorigin src="/assets/index-fe61ba6e.js"></script>
    <link rel="stylesheet" href="/assets/index-4e5c0698.css">
  </head>
  <body>
    <div id="root"></div>

  </body>
</html>

```

Figura 1: Código fuente

Concretamente, indagando nos percatamos de que nos interesa la redirección que hay a "/assets/index-fe61ba6e.js".

Ahí podemos encontrar efectivamente todas las palabras que aparecían en la web (junto a otras muchas más). Otro método hubiera sido copiarlas y pegarlas a mano para crear nuestro propio diccionario, ¡no son tantas! ¡A la vieja usanza! Por tanto, ahora modificaremos ligeramente la orden anterior:

```
cewl https://sugusuva.es/assets/index-fe61ba6e.js -w diccionario.txt
```

Ahora sí, lograremos el diccionario que (como mínimo) queremos y necesitamos.

Para poder romper la *passphrase* del fichero ZIP, podemos generar un hash del mismo con:

```
zip2john Encrypted.zip > hash
```

Para después, mediante un ataque por diccionario, intentar averiguar cuál es la contraseña que nos impide visualizar su contenido:

```
john --wordlist=diccionario.txt hash
```

```
┌──(kali㉿kali)-[~/Documents/CTF_SUGUS_2024/PASSPHRASE]
└─$ john --wordlist=diccionario.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
aprender         (Encrypted.zip/secret.enc)
1g 0:00:00:00 DONE (2024-02-21 21:31) 20.00g/s 36920p/s 36920c/s 36920C/s return..UVa
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figura 2: Voilà!

Observamos que la passphare es **aprender**. Posteriormente, podremos descomprimir el ZIP introduciendo la contraseña, obteniendo así el mensaje cifrado (secret.enc).

```
unzip Encrypted.zip
```

Sabemos que la passphrase fue utilizada con key o clave para el algoritmo de cifrado, y que se trata de un algoritmo de cifrado simétrico (DES o AES en este caso). Es por ello, que con la información proporcionada en el enunciado y una correcta comprensión de esta, podemos lograr descifrar el mensaje con herramientas como CyberChef.

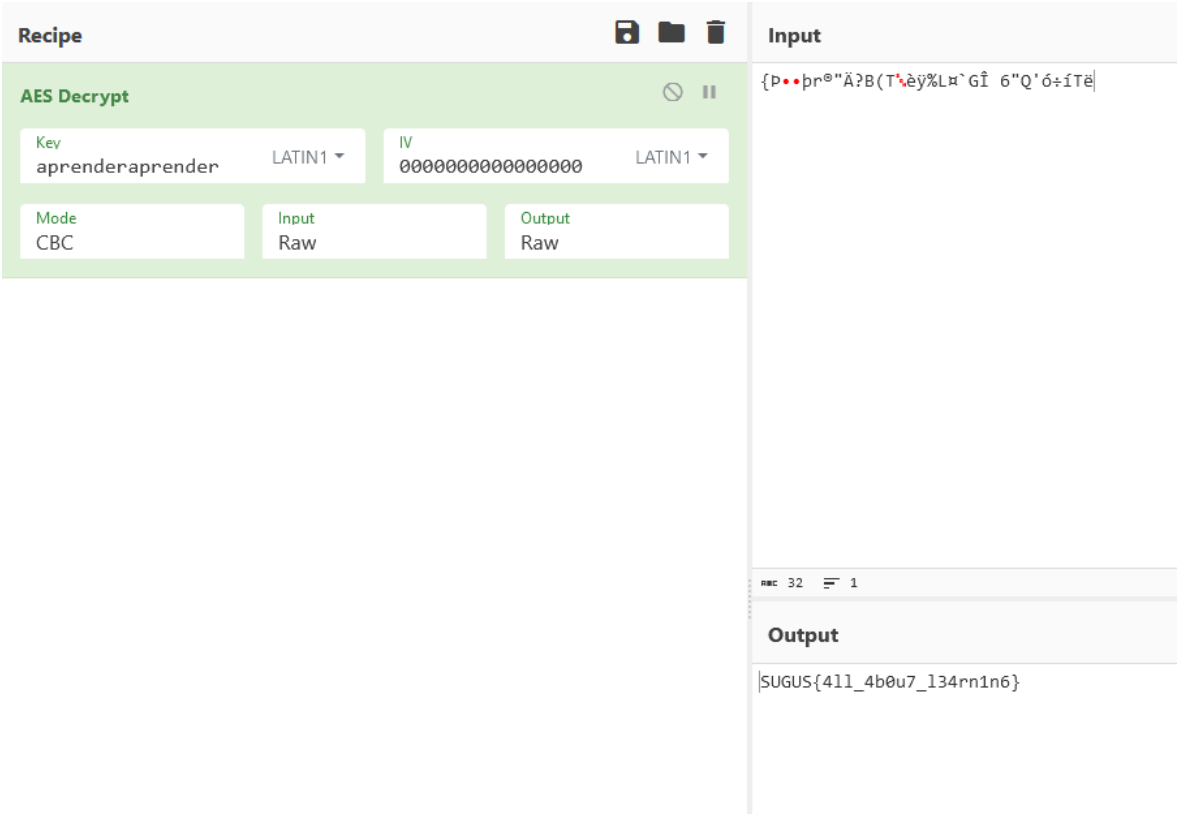


Figura 3: Descifrando el mensaje

La flag resultante es:

```
SUGUS{4ll_4b0u7_l34rn1n6}
```