CTF EUSKALHACK III
WRITE UP

Solución a algunas pruebas realizadas durante el concurso.
Precuela CTF Euskalhack III

Delmer

ikerburguera@gmail.com

INTRODUCCIÓN

En este documento se proporciona una solución a algunas pruebas realizadas en la precuela CTF de la Euskalhack III, la cual comenzó el Viernes 8 de Junio a las 12:00 y terminó el Domingo 10 de Junio a las 20:00 del 2018.

La temática de las pruebas abordaban disciplinas de tipo web, crypto, forensic, exploiting, reversing, etc.

AGRADECIMIENTOS

Agradecer al equipo de Euskalhack y a todos los patrocinadores del evento el esfuerzo y sacrificio que hacen todos los años para que tengamos un fin de semana de lujo.
Dar las Gracias también al equipo de IHackLabs por la organización del concurso CTF

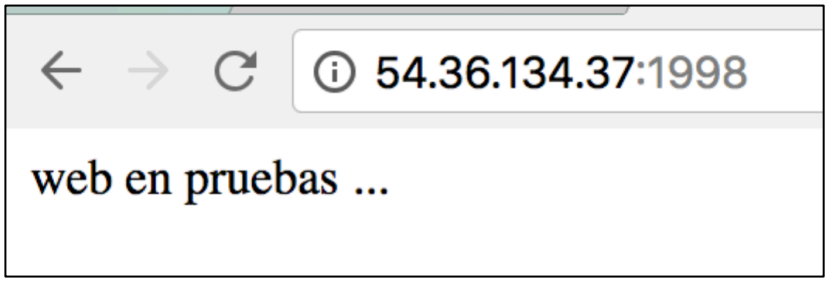
¡Zorionak a todos!.

LA VIDA ES UN RECON-CHINEO

HINT

Conectate a http://54.36.134.37:1998/ pd: ni fuerza bruta ni nada. gugle. Hint: XORitANDitNORit esta en github.

Activamos la prueba junto con la VPN y nos conectamos a la web http://54.36.134.37:1998/ y nos indica que es una web que está en pruebas.



Miramos el código fuente de la página y nos da una pista:

web en pruebas ...

```
<!-- XORitANDitNORit/puertrasera --><!-- gugle -->
```

Buscamos en Google la referencia pero no encontramos nada.

Accedemos a mi Github (https//github.com:iburguera:CTF) para rescatar información y alguna herramientas de CTF que nos puedan ayudar y probamos suerte a poner los textos de arriba XORitANDitNORit y puertrasera.

Resulta que **XORitANDitNORit** es un usuario de github y tiene código en su repositorio!

(Según su descripción pone que se unió hace 5 días a Github, quizás Google no lo haya indexado todavía y no nos salía ninguna referencia. Además de github también se ha probado a buscar en Linkedin, Facebook, Twitter, etc)

Entramos en el repositorio puertrasera del usuario XORitANDitNORit y vemos que hay un fichero index.php. Procedemos a leer su código fuente.

```
<?php
if(isset($_COOKIE['secreto'])) {
$x="Dcq7CsMgFADQX7lD4OpSKJQuwalkCIU6OIYQrF6tEB+oWVr67+2ZT0hmPywxND
eval(gzinflate(base64_decode($x)));
}
?>
```

Leemos el código. Parece que está ofuscado. Para que el código de abajo se ejecute debe estar creada la variable Cookie[‘secreto’]. Si está activada entra en la función y procede a ejecutar el código.

Buscamos información sobre **gzinflate** para saber que es lo que hace.

“

There are many ways to encode and decode PHP code. From the perspective of site security, there are three PHP functions — `str_rot13()`, `base64_encode()`, and `gzinflate` — that are frequently used to obfuscate malicious strings of PHP code.

Parece que se php tiene funciones para ofuscación de código y en código de arriba hace uso de ello.

Creamos un fichero php modificado para que muestre el código sin ofuscar y sin pedirnos la cookie. Lo ponemos en un servidor local y lo ejecutamos

```
<?php
$x="Dcq7CsMgFADQX7lD4OpSKJQuwalkCIU6OIYQrF6tEB+oWVr67+2ZT0hmPy
echo eval(gzinflate(base64_decode($x)));
¿>
```

Nos devuelve el siguiente código

```
include('config.php');
if(gzinflate(base64_decode($_COOKIE['secreto'])) === 'megustaprogramar')
{
  $_SESSION['admin'] = 1;
}
```

el código de arriba nos indica que el resultado de

gzinflate(base64_decode($_COOKIE['secreto']) debe ser igual a *"megustaprogramar"* para que la sesión sea admin. Así que debemos hacer el proceso inverso para que el valor de la cookie secreto sea un valor válido.

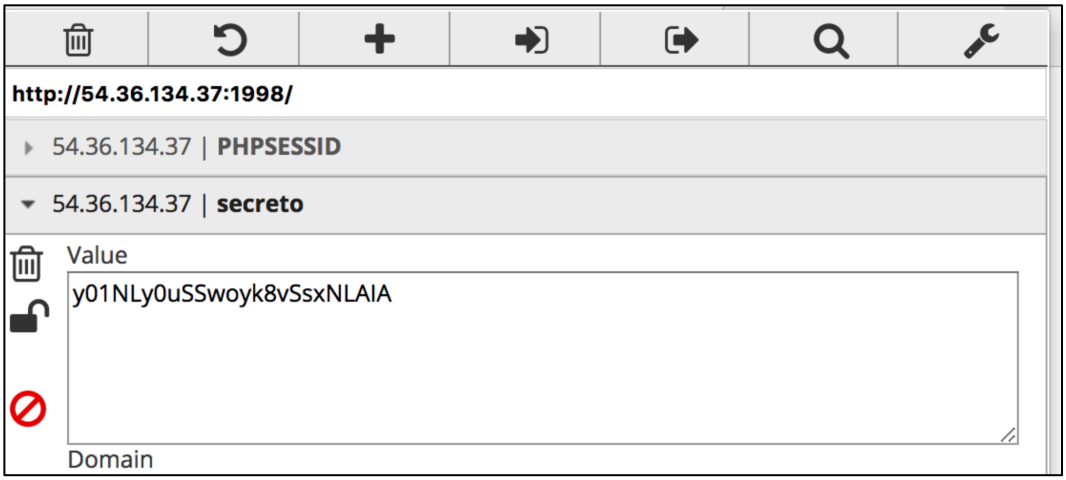
Creamos el siguiente código en PHP.

```
<?php
$cookie = base64_encode(gzdeflate('megustaprogramar',9));
print "Texto cookie: $cookie";
?>
```

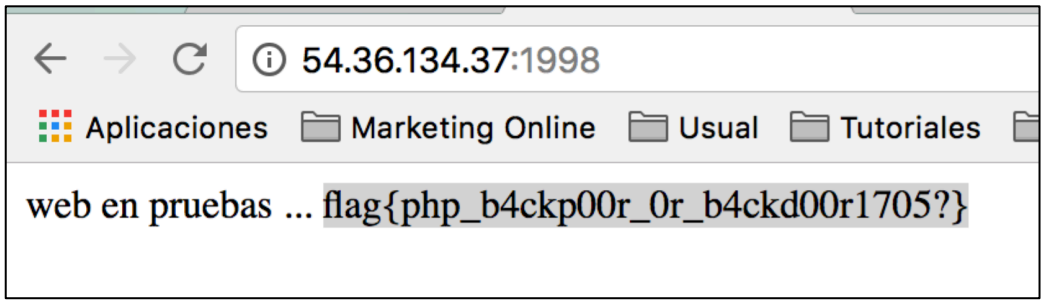
Ejecutamos el código en nuestro servidor local y obtenemos el siguiente resultado.

```
Texto cookie: y01NLy0uSSwoyk8vSsxNLAIA
```

Utilizamos una extensión de Chrome llamada ***EditthisCookie*** para crear la cookie secreto e introducimos el valor que nos ha dado el programa que hemos creado anteriormente.



Actualizamos la página y nos da la Flag



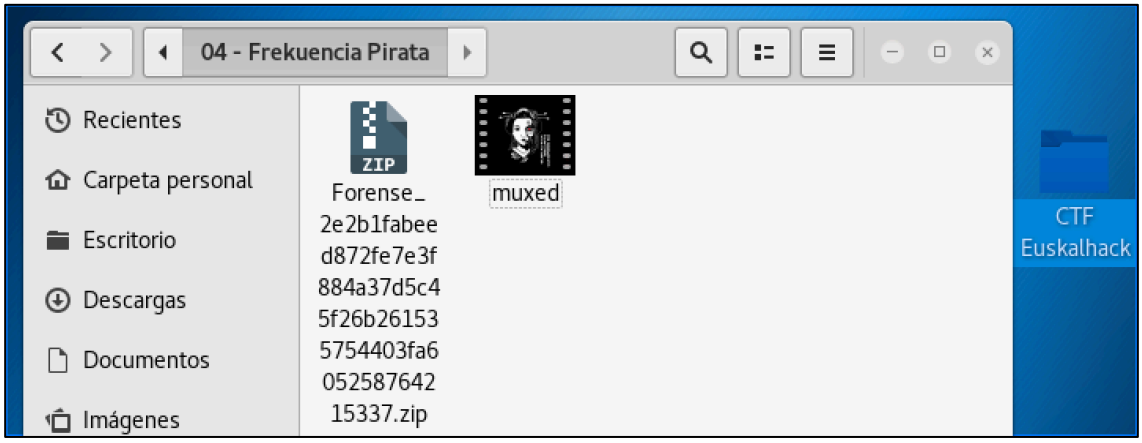

```
flag{php_b4ckp00r_0r_b4ckd00r1705?}
```

FREKUENCIA PIRATA

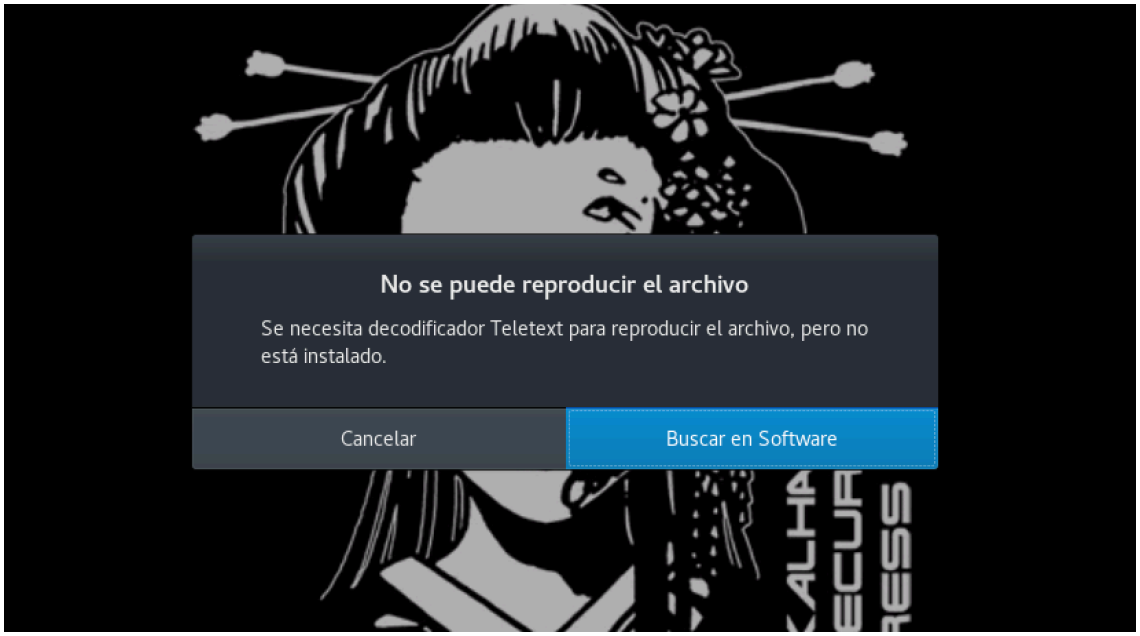
HINT

Encuentra la información que se oculta en este fichero. HINT: Transport Stream es un protocolo de comunicación para audio, vídeo y "datos".

Entramos en la prueba y nos descargamos el fichero para la prueba.



Descomprimimos el fichero ZIP y nos extrae un fichero llamado **muxed**. Abrimos el fichero y nos aparece el siguiente mensaje junto a un video y la música épica de Terminator .

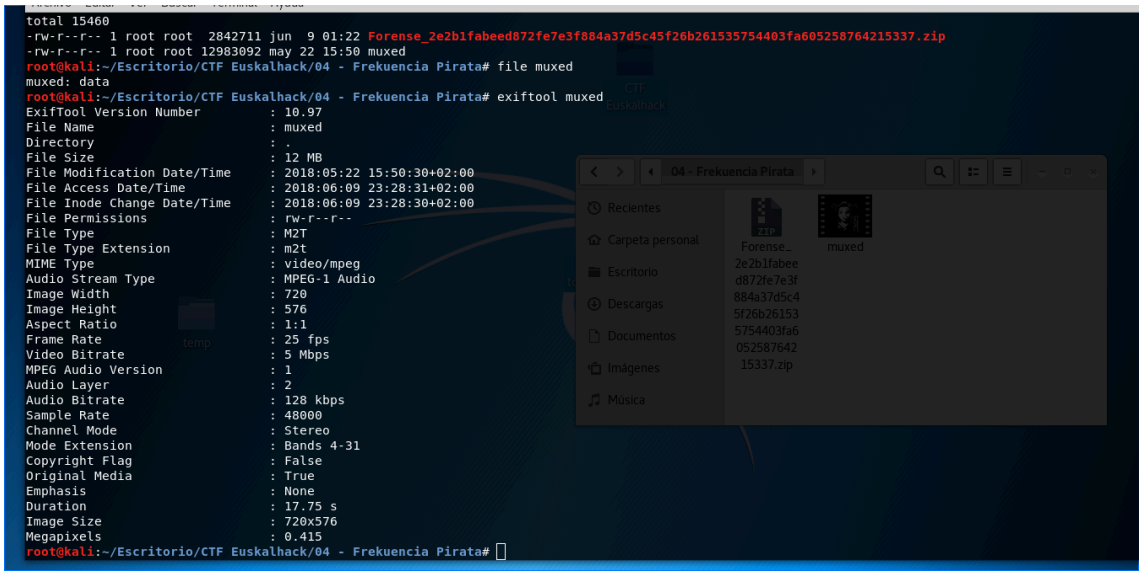


¿Un video con Teletexto? ¿El Teletexto no era lo de la TV?

Abrimos un terminal para ver que tipo de fichero es y ver si podemos encontrar alguna otra pista.

Ejecutamos los comandos para obtener más información:

file muxed
exiftool muxed
strings muxed



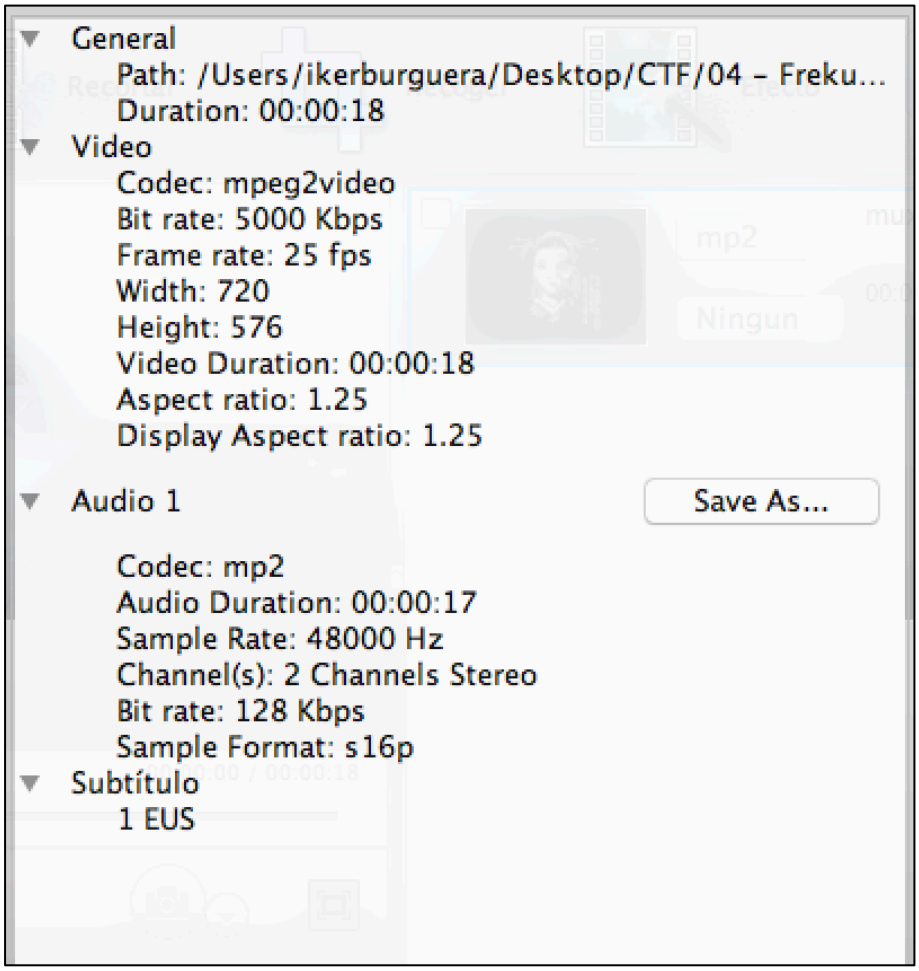
¿EuskalhackTV One? Puede que al final si tenga que ver con la TV después de todo.
Revisamos la prueba y han subido un HINT:

Transport Stream es un protocolo de comunicación para audio, vídeo y "datos".

Revisamos el protocolo Transport Stream y verificamos que MPEG puede almacenar VIDEO/AUDIO/DATOS

"MPEG transport stream (transport stream, MPEG-TS, MTS or TS) is a standard digital container format for transmission and storage of audio, video, and Program and System Information Protocol (PSIP) data."

Tenemos el video y el audio pero nos falta el dato.
Recopilamos datos y hacemos balance de la situación actual para seguir adelante.
Con las pistas de que nos falta el dato, que MPEG puede llevar datos, la pista de EuskalHackTV One y el decodificador Teletexto, deducimos que los datos están en el Teletexto.



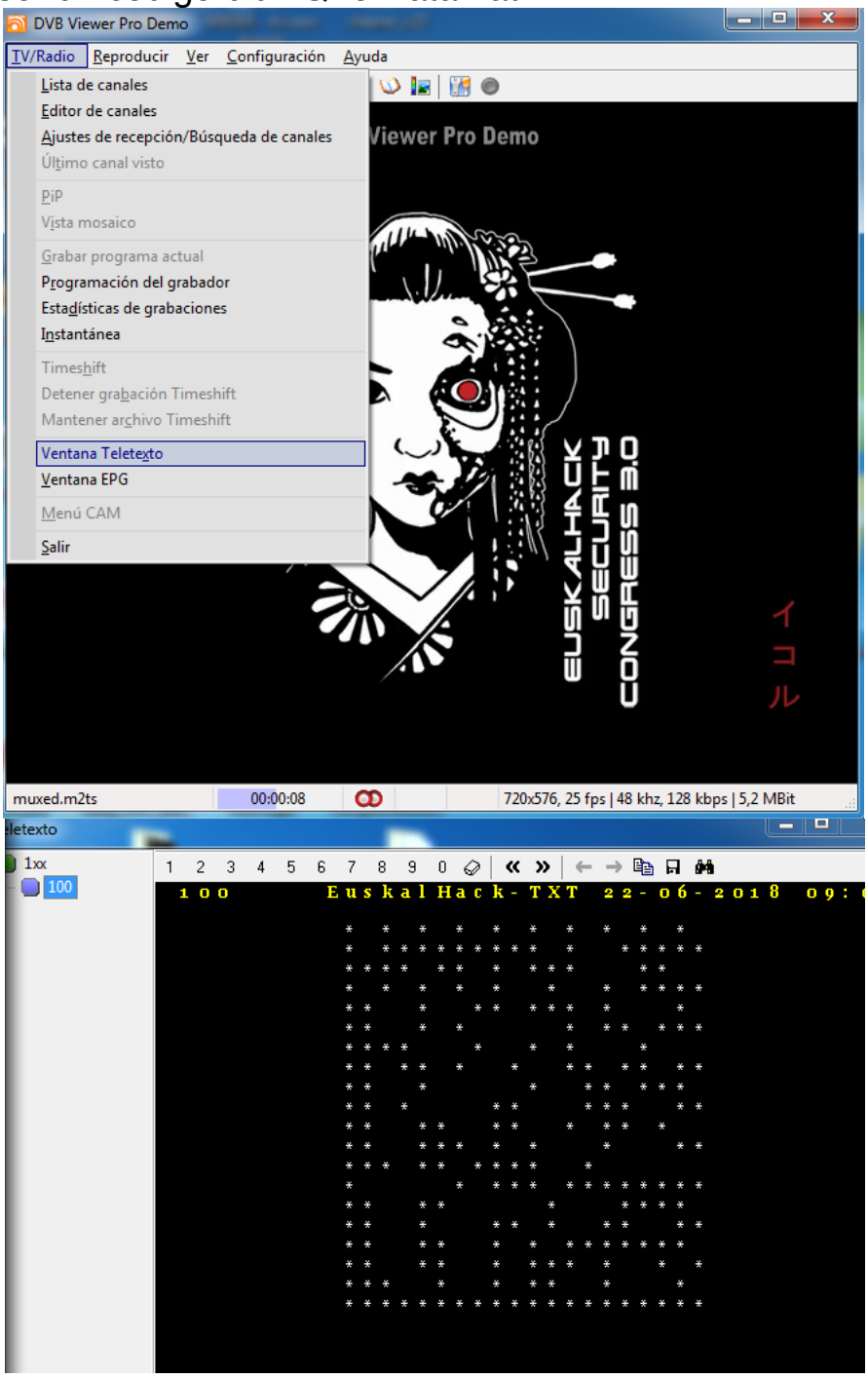
Comprobamos que el video lleva un fichero de subtítulos con VLC.

Buscamos un programa en Google capaz de reproducir video y mensajes de Teletexto.

Resulta que hay uno llamado **DVBViewer Pro** el cual tiene una demo y nos la podemos descargar gratuitamente.

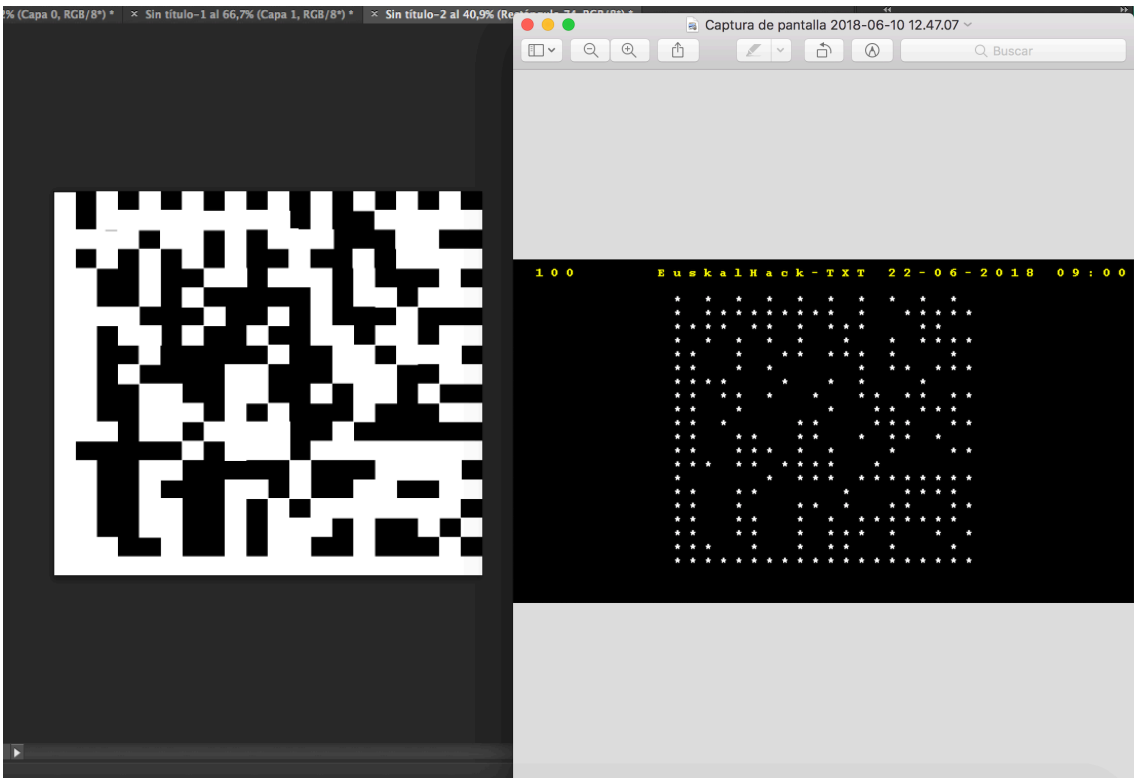
Reproducimos el video y vamos a la opción de ver Ventana de Teletexto.
Se abre otra ventana con contenido en la página 100.

Después de pasar tiempo pensando que son esos asteriscos y espacios en blanco (Código morse, Braille, etc) caemos en que podría ser un código bidi. Qr o Datamatrix.



Generamos una imagen nueva cogiendo como referencia la imagen que nos ha mostrado el Teletexto.

Obtenemos el siguiente resultado



Nos bajamos un lector BIDI desde la playStore, apuntamos con la cámara a nuestro código y nos muestra un mensaje en el móvil con la flag!


```
flag{OldStyleHacking}
```

CTF FINAL EUSKALHACK

Conseguimos clasificarnos para la final en el puesto Nº5.
No hemos podido dedicarle mucho tiempo salvo el fin de semana y sin mucho éxito.

Clasificación

#	Equipo	Puntos
1	DANITORWS	686
2	RAMANDI	650
3	IKASTEN.IO	430
4	THEREARWINDOW	382
5	DELMER	360
6	ODEI	163
7	MRPNKT	160
8	HACKITURIA	160
9	GR4M3N4W3R	160
10	BULW4RK	160
11	NOT TODAY	160
12	SCOOBYCOOKIES	150
13	-HACK&+BEERS	150
14	P3GM4N	150
15	RUN3R	10

Adjunto al correo, he incluido todas las notas que fui apuntando sobre los servidores y las pistas que iba encontrando durante el tiempo que le dediqué al CTF.