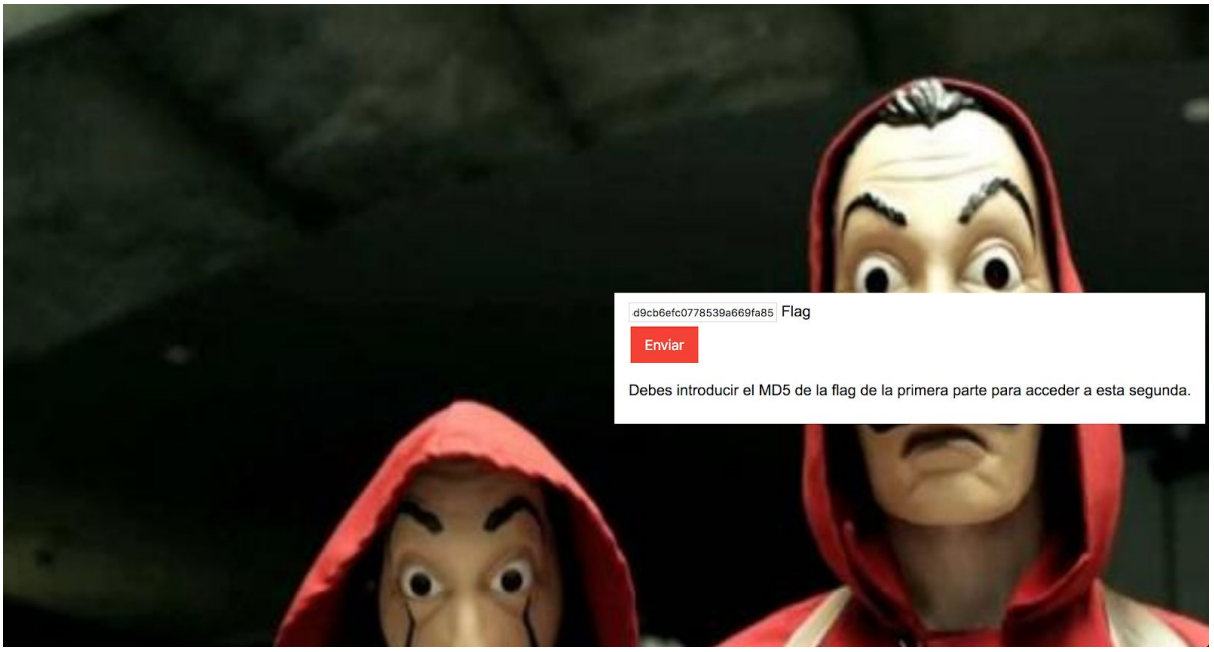
# Episodio 1 - parte 2

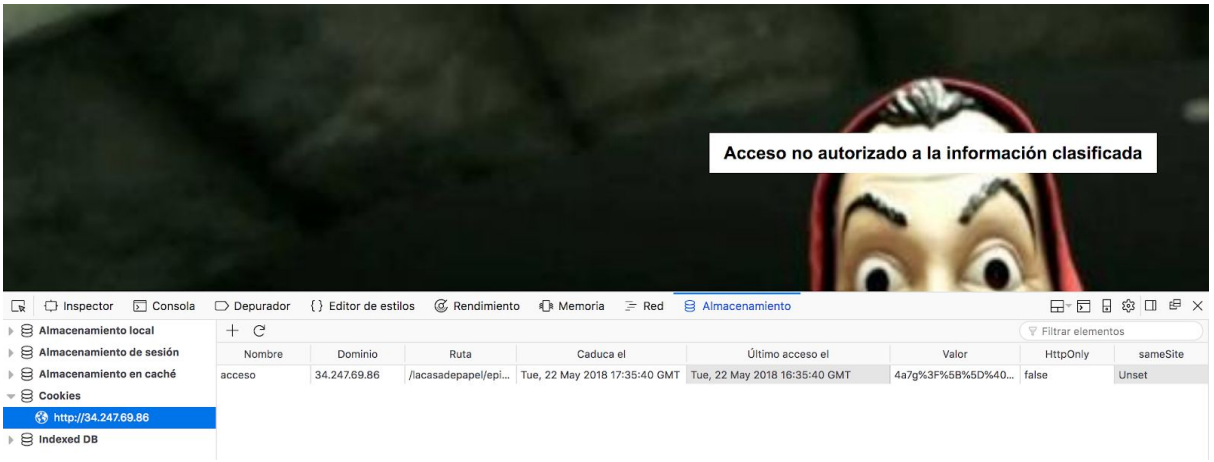
Accedemos a la plataforma:
http://unaalmes.hispasec.com/challenges#EPISODIO%201%20-%202%C2%AA%20PARTE


Como era de esperar, el fichero flag.zip está protegido con contraseña

Accedemos a la url e insertamos el md5 del flag de la primera parte tal y como nos solicita el formulario: e30f35ad8d9cb6efc0778539a669fa85

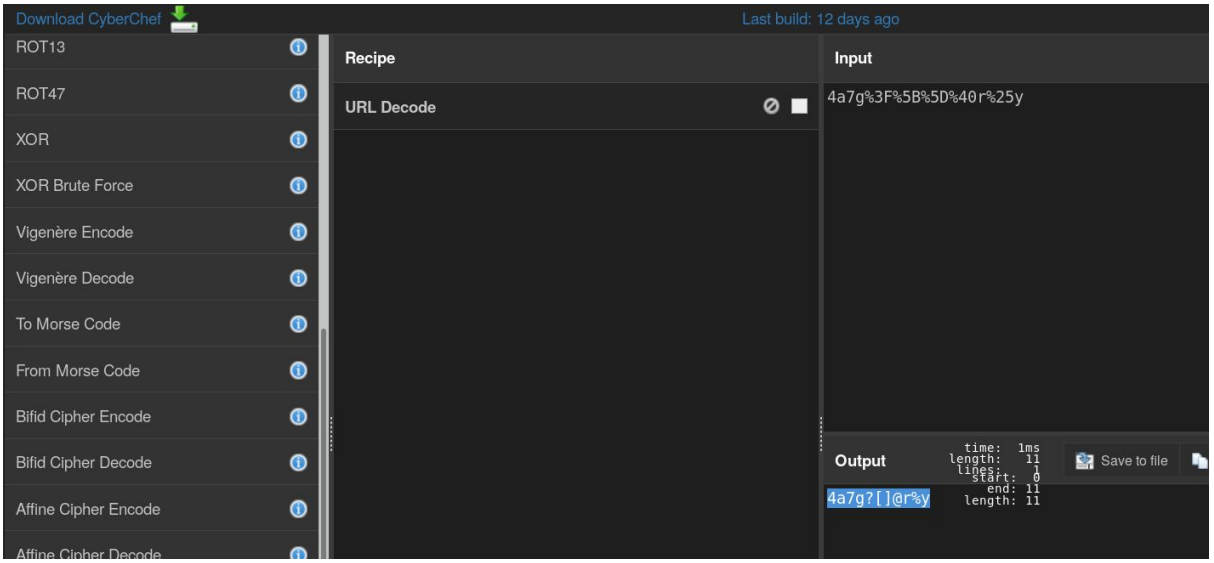



Enviar y…
vemos esto:




¡Una cookie! un poco fea, la verdad: 4a7g%3F%5B%5D%40r%25y

Nada que un URL decode no arregle en [cyberchef]




```
acceso=4a7g?[]@r%y
```

Estará codificada la cookie? Pruebo a jugar con las típicas herramientas de decoding online y algunas que encontré en github…
https://github.com/UltimateHackers/Decodify
https://github.com/psypanda/hashID
https://gchq.github.io/CyberChef
https://md5hashing.net/hash_type_checker
https://cryptii.com/
https://www.tools4noobs.com/online_tools/
http://dcode.fr/

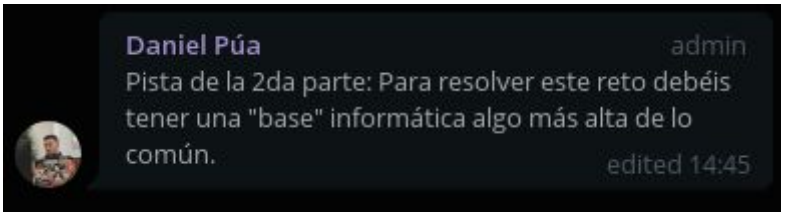
Como no doy con nada después de un rato, voy a repasar el código fuente de la página…

```html
<html>
    <head>
    <title>2da parte</title>
    <style>
    body {
        background-image: url("images/background2_1.jpg");
        background-size: 1920px 1080px;
    }
    form {
        position: absolute;
        top: 45%;
        left: 650px;
        background-color: #FFFFFF;
        padding: 10px 15px;
        font-family: "Arial";
        }
    h3 {
        position: absolute;
        top: 45%;
        left: 750px;
        background-color: #FFFFFF;
        padding: 10px 15px;
        font-family: "Arial";
        }
    .text {
        position: absolute;
        left: 200px;
        background-color: #FFFFFF;
        padding: 10px 15px;
        font-family: "Arial";
        }
    .nice {
        position: absolute;
        top: 35%;
        left: 600px;
        background-color: #FFFFFF;
        padding: 10px 15px;
        font-family: "Arial";
        background-image: url("images/background2_2.jpg");
        background-size: 1920px 1080px;
        }
```

La clase .nice tiene otra imagen distinta a la que estamos viendo.
¿Qué tiene background2_2.jpg?


Me bajo la imagen, a ver si esconde algo usando steghide pero no encuentro nada. Supongo que se verá de fondo cuando consiga resolver la cookie.

Vuelvo a probar (MUCHOS) encoders y encuentro esto gracias a la pista que ‘pinearon’ los admins en el canal público de Una al mes de Telegram:



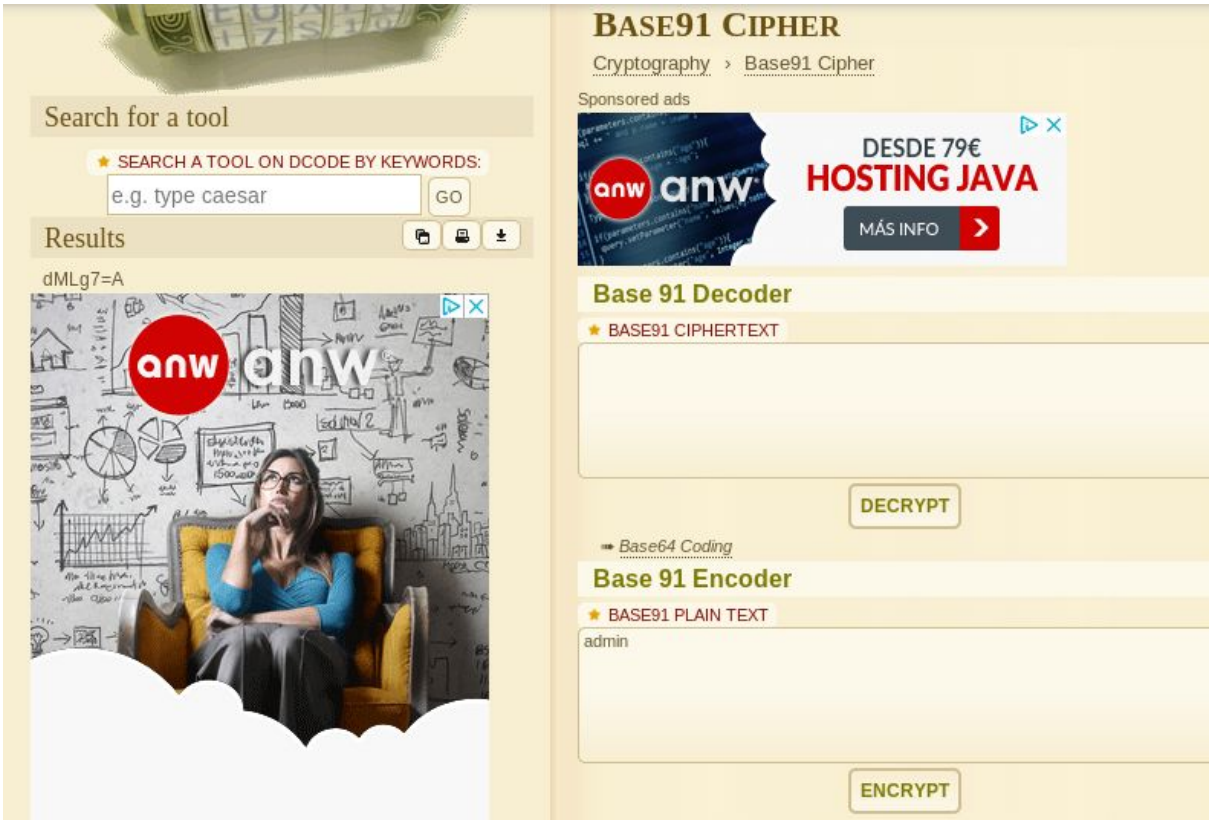

y en la plataforma...

> Hint: Para resolver este reto debéis tener una "base" informática algo más alta de lo común.

¡Hay que estar atento a los canales! Este tipo de pistas son cruciales, que ya me estaba atascando probando y probando. Así pues, ¿“base“ más alta? Creo que en dcode.fr no probé el rot91...


visitante

Parece que la cookie está codificada en base91 pero ahora no sé qué debería usar como nombre o palabra para la cookie. Probaré a generar algunas cookies para nombres de usuario típicos: root, administrador, admin… y tras generar unos cuantos hashes con rot91 y usando curl consigo una respuesta distinta usando la cookie codificada para admin: dMLg7=A




```
curl --header "Cookie:acceso=dMLg7=A"
http://34.247.69.86/lacasadepapel/episodio1/2da_parte.php
```

```
        }
    .button {
            font-size: 15px;
            background-color: #f44336;
            border: none;
            color: white;
            padding: 10px 15px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            margin: 4px 2px;
            cursor: pointer;
    }
    .advice{
            background-color: #FFFFFF;
            text-align: center;
            font-size: 18px;
            font-weight: bold;
    }
    </style>
</head>
    <body>
            </body>
    <body class="nice">
    <h3 class="text">El codigo para descomprimir el zip está claro... ApdnioimcuFqoftnpSBLLeugbu</h3>
    </body>
    <body>
            </body>
```

Sustituyo el valor de la cookie en el navegador y refresco para verlo en todo su esplendor


A ver qué es ApdnioimcuFqoftnpSBLLeugbu

parece que el password no está tan claro y habrá que decodificarlo también.

Buscando en algunas herramientas online de decoding encontré este texto en un multisolver [https://geocaching.dennistreysa.de/multisolver/](https://geocaching.dennistreysa.de/multisolver/)


Four Square pinta bien pero… ELCOIOGOESFQLISONUAMPARKER?

Parece que pone algo así como ELCODIGOESALISONUAMPARKER pero debe ser alguna otra cosa porque con eso como password el zip no se abre.
No encuentro manera de hacer funcionar ninguna otra herramienta con mejor resultado salvo

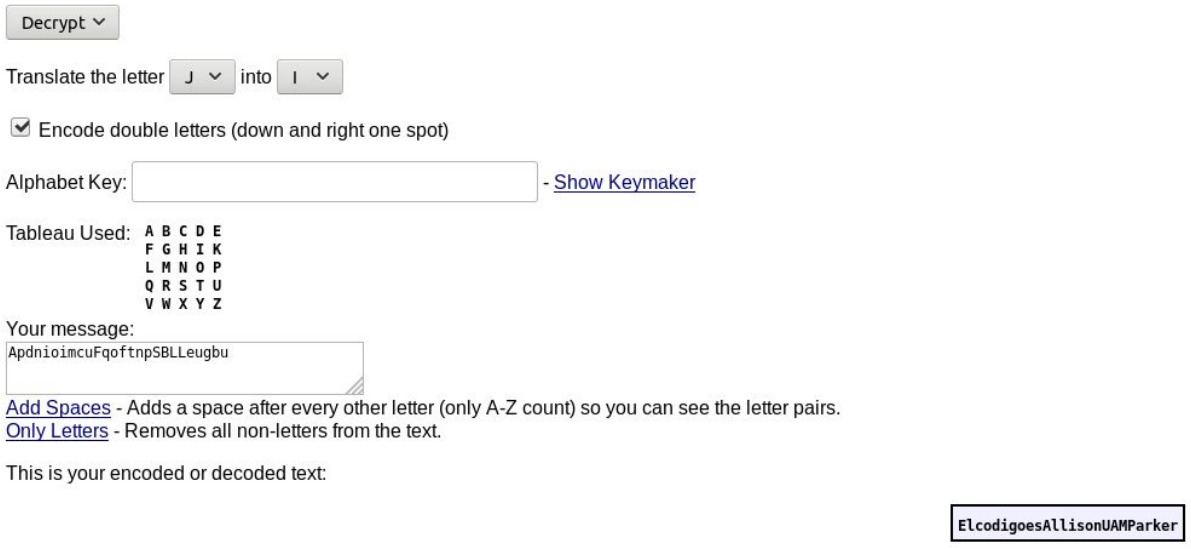
https://www.geocachingtoolbox.com/index.php?page=fourSquareCipher

Parece que las mayúsculas y minúsculas importan.

pruebo a descomprimir el zip que nos dieron al principio de la prueba con distintas versiones del texto. Mayúsculas, minúsculas, parte del texto... está claro que es el texto que buscamos pero esos problemas al decodificarlo complican el asunto.

Tras varias pruebas más di con la clave correcta en
http://rumkin.com/tools/cipher/playfair.php

No era four square, era playfair y lo podría haber visto antes si en el multisolver de dennistreysa.de hubiera bajado con el scroll sólo un poco más… :/




**ElcodigoesAllisonUAMParker**

7za x flag.zip -pAllisonUAMParker -y


Vamos a ver qué ha salido del zip.

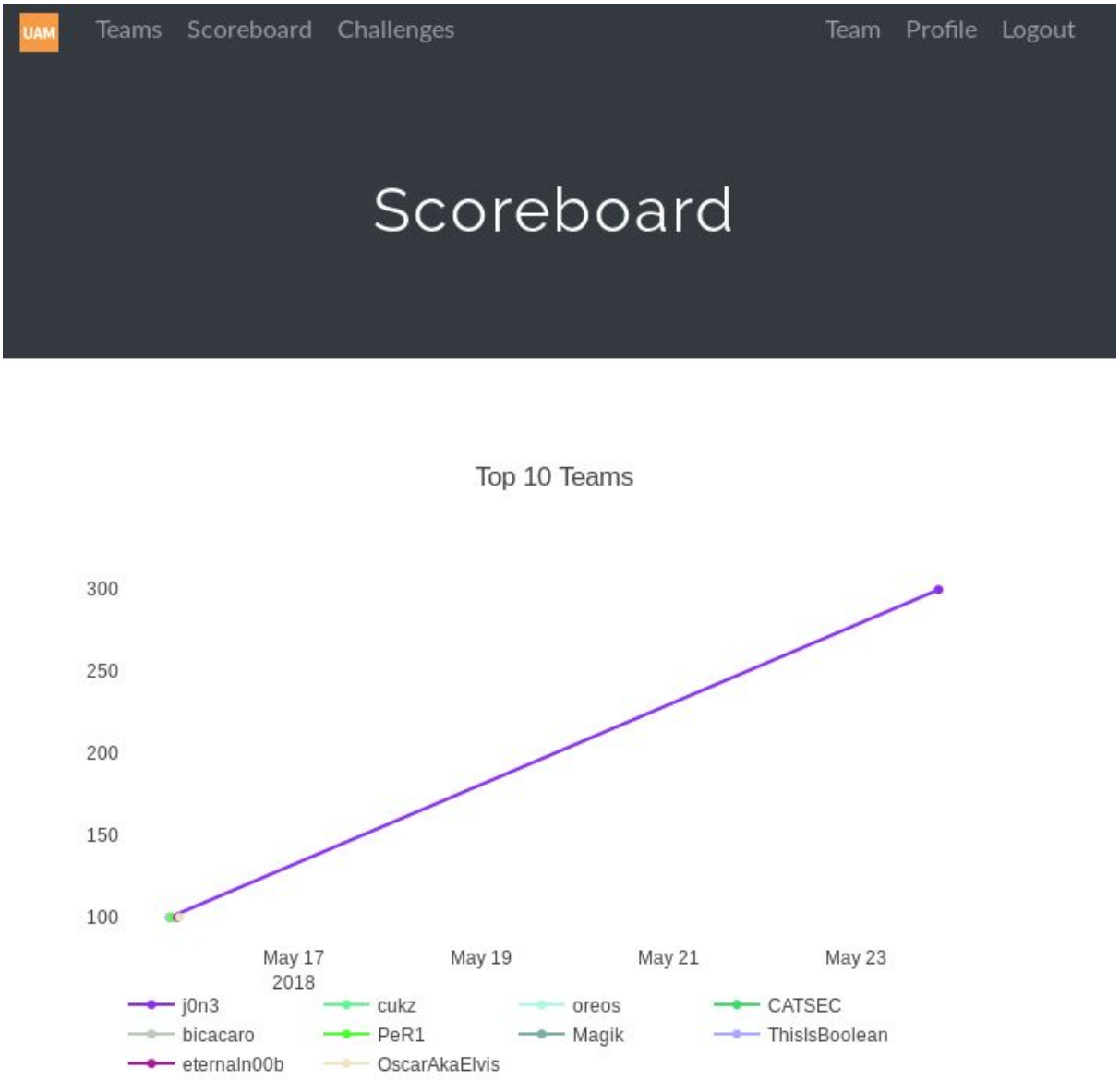
flag.txt

hago un cat de flag.txt pensando que ahora habrá alguna otra put… ¡anda, la flag en claro!

UAM{c9beec67d71c56a0f9b683fe5232e76e}

Lo meto en la plataforma y…

¡Yeah! ¡Segundo podio!




| Place | Team | Score |
|---|---|---|
| 1 | j0n3 | 300 |
| 2 | cukz | 100 |
| 3 | oreos | 100 |

Ha sido una prueba sencilla técnicamente pero fue a base de probar y probar. Buscar y buscar. Y volver a probar con codificaciones distintas y nombres de usuario, y herramientas y probar más… hasta dar con ello. Ha sido más la perseverancia para encontrar las herramientas que el conocimiento, aunque siempre es divertido resolver estos enigmas. Por el camino siempre encuentro muchas cosas interesantes y nuevas herramientas para el

cajón. Además son ejemplos sencillos de lo que debemos evitar si queremos mantener nuestros entornos y desarrollos seguros.

¡Muchas gracias a todos los creadores, admins y participantes! Ánimo y a seguir buscando y probando.

Herramientas usadas:
Firefox
Curl
https://gchq.github.io/CyberChef/
https://www.dcode.fr/base-91-encoding
https://geocaching.dennistreysa.de/multisolver/
http://rumkin.com/tools/cipher/playfair.php
7zip
cat :D

José Ángel Sánchez
@_j0n3