RETO UAM – UNIVERSO MAVEL - EPISODIO 1 - 1ª PARTE

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

---

Lo primero descargamos el fichero del reto, y vemos que es un pcap que contiene trazas de red. Se aprecian las tramas el protocolo 802.11, que es el protocolo de comunicación inalámbrica (WIFI), y se aprecia que está encriptado. Esto es el tráfico que se captura sobre una Wifi protegida para la cual no tenemos la password de conexión.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000		Motorola_ee:04:d1 (…	802.11	10	Acknowledgement, Flags=........
2	-0.000022		LgElectr_f6:fd:84 (…	802.11	10	Acknowledgement, Flags=........
3	0.018448	XaviTech_45:ea:dd	Broadcast	802.11	115	Beacon frame, SN=1167, FN=0, Flags=........,
4	0.097262		D-LinkIn_b7:62:88 (…	802.11	10	Acknowledgement, Flags=........
5	0.137712		LgElectr_f6:fd:84 (…	802.11	10	Acknowledgement, Flags=........
6	0.184810		LgElectr_f6:fd:84 (…	802.11	10	Acknowledgement, Flags=........
7	0.185856		Motorola_ee:04:d1 (…	802.11	10	Acknowledgement, Flags=........
8	0.186356		Ubiquiti_8d:3e:f1 (…	802.11	10	Acknowledgement, Flags=........
9	0.229378		Motorola_ee:04:d1 (…	802.11	10	Acknowledgement, Flags=........
10	0.366570		LgElectr_f6:fd:84 (…	802.11	10	Acknowledgement, Flags=........
11	0.432612		Tp-LinkT_29:9d:82 (…	802.11	10	Acknowledgement, Flags=........
12	0.434656		Ubiquiti_8d:3e:f1 (…	802.11	10	Acknowledgement, Flags=........

Por tanto, tenemos que intentar, por fuerza bruta, averiguar dicha password para poder descifrar el trafico de red. Usaremos la herramienta de Kali “aircrack-ng”, la cual si la ejecutamos con el fichero pcap ya nos indica el ESSID de la red, así como el tipo de encriptación de la Wifi (WPA):

```
nacho@kali:~/Forensic$ aircrack-ng capture-01.cap
Opening capture-01.cape wait...
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.cape wait...
Read 5786 packets.
```

En este caso, lanzamos el ataque directamente usando el diccionario rockyou:

```
aircrack-ng capture-01.cap -w /usr/share/wordlists/rockyou.txt
```

Cuando lleva aproximadamente la mitad del tiempo consumido, encuentra la password “hydra54321”:

```
[01:17:07] 4868824/9822768 keys tested (1334.29 k/s)

Time left: 1 hour, 1 minute, 53 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  : 09 3A DF 40 98 96 26 1C EB 58 75 99 B8 F1 29 D1
                 CF C7 7A EC 56 A8 DF D0 1D E6 F0 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

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

Ahora, ya con el password, usamos otra herramienta de la suite “airdecap-ng” para descifrar todo el fichero, y transformarlo a un pcap en claro:

```
nacho@kali:~/Forensic$ airdecap-ng capture-01.cap -e "Hydra Corp" -p hydra54321
Total number of stations seen            5
Total number of packets read          5786
Total number of WEP data packets         0
Total number of WPA data packets      1814
Number of plaintext data packets         0
Number of decrypted WEP  packets         0
Number of corrupted WEP  packets         0
Number of decrypted WPA  packets      1521
Number of bad TKIP (WPA) packets         0
Number of bad CCMP (WPA) packets         0
```

Que si lo abrimos con WireShark, ya se puede analizar su tráfico:

	Time	Source	Destination	Protocol	Length	Info
123	4.286724	192.168.105.190	34.247.69.86	TCP	74	41572 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TS
124	4.354812	34.247.69.86	192.168.105.190	TCP	74	80 → 41572 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1452 SAC
125	4.354820	192.168.105.190	34.247.69.86	TCP	66	41572 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1998507938
126	4.354820	192.168.105.190	34.247.69.86	HTTP	504	GET /universomarvel/episodio1 HTTP/1.1
127	4.423420	34.247.69.86	192.168.105.190	TCP	66	80 → 41572 [ACK] Seq=1 Ack=439 Win=28032 Len=0 TSval=151717902
128	4.423420	34.247.69.86	192.168.105.190	HTTP	679	HTTP/1.1 301 Moved Permanently (text/html)
129	4.423940	192.168.105.190	34.247.69.86	TCP	66	41572 → 80 [ACK] Seq=439 Ack=614 Win=30464 Len=0 TSval=1998508
130	4.425476	192.168.105.190	34.247.69.86	HTTP	505	GET /universomarvel/episodio1/ HTTP/1.1
134	4.596994	192.168.105.190	34.247.69.86	HTTP	551	[TCP ACKed unseen segment] [TCP Previous segment not captured]
159	4.668158	34.247.69.86	192.168.105.190	TCP	1506	[TCP ACKed unseen segment] [TCP Previous segment not captured]

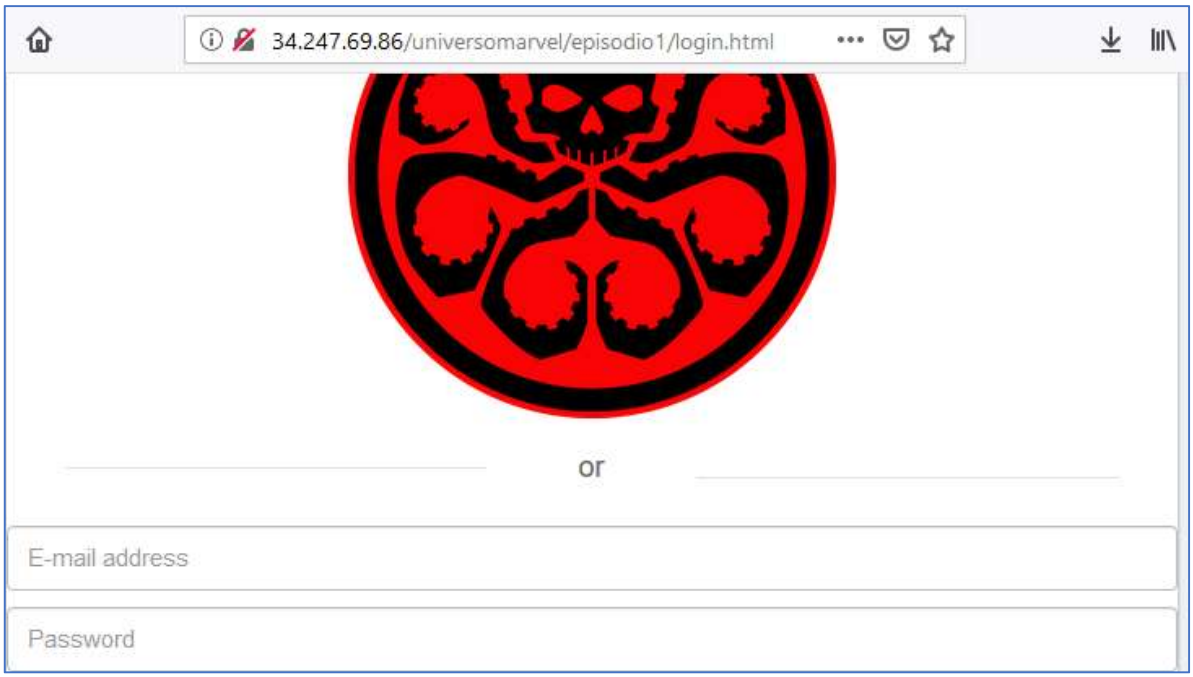
Ahora analizamos el fichero. Vemos principalmente trafico HTTP, así que vamos a filtrar por este protocolo:

Source	Destination	Protocol	Length	Info
192.168.105.190	34.247.69.86	HTTP	504	GET /universomarvel/episodio1 HTTP/1.1
34.247.69.86	192.168.105.190	HTTP	679	HTTP/1.1 301 Moved Permanently (text/html)
192.168.105.190	34.247.69.86	HTTP	505	GET /universomarvel/episodio1/ HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	551	[TCP ACKed unseen segment] [TCP Previous segment not captured] GET /universomarvel…
192.168.105.190	34.247.69.86	HTTP	534	GET /universomarvel/episodio1/css/styles.css HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	534	GET /universomarvel/episodio1/bootstrap/js/bootstrap.min.js HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	517	GET /universomarvel/episodio1/js/custom.js HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	563	GET /universomarvel/episodio1/images/logo_shield.png HTTP/1.1
34.247.69.86	192.168.105.190	HTTP	407	HTTP/1.1 200 OK (text/css)
34.247.69.86	192.168.105.190	HTTP	501	HTTP/1.1 200 OK (application/javascript)
34.247.69.86	192.168.105.190	HTTP	648	HTTP/1.1 200 OK (application/javascript)
192.168.105.190	34.247.69.86	HTTP	686	GET /universomarvel/episodio1/authenticate.php?username=gward%40hydra.com&password…
34.247.69.86	192.168.105.190	HTTP	409	HTTP/1.1 302 Found (text/html)
192.168.105.190	34.247.69.86	HTTP	626	GET /universomarvel/episodio1/panel.php HTTP/1.1
34.247.69.86	192.168.105.190	HTTP	351	HTTP/1.1 200 OK (text/html)
192.168.105.190	34.247.69.86	HTTP	555	GET /universomarvel/episodio1/images/ward.png HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	604	GET /universomarvel/episodio1/bootstrap/fonts/glyphicons-halflings-regular.woff HT…
192.168.105.190	34.247.69.86	HTTP	540	GET /universomarvel/episodio1/vendors/fullcalendar/fullcalendar.js HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	532	GET /universomarvel/episodio1/vendors/fullcalendar/gcal.js HTTP/1.1
192.168.105.190	34.247.69.86	HTTP	518	GET /universomarvel/episodio1/js/calendar.js HTTP/1.1

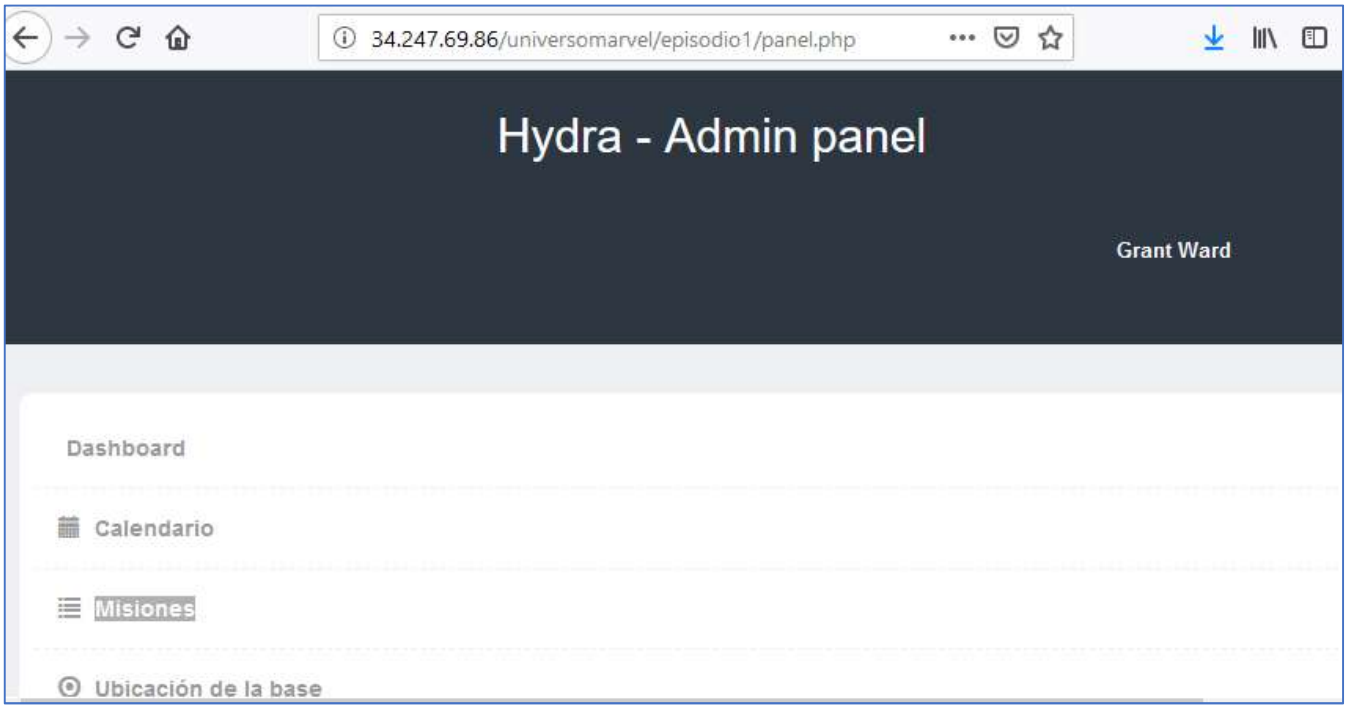
Vemos llamadas a una Web llamada “/universomarvel/episodio1”, con la descarga típica también de sus ficheros css, js, etc. Una petición nos llama la atención, es la página de autenticación:

```
HTTP      686 GET /universomarvel/episodio1/authenticate.php?username=gward%40hydra.com&password=rUHp6e7FVds2nRPZ HTTP/1.1
```

Así que abrimos el navegador y reproducimos las llamadas. Vemos que nos carga un portal y nos pide un usuario y password:



Del fichero pcap extraemos el usuario y el password del usuario, así que lo introducimos y accedemos al Admin Panel:



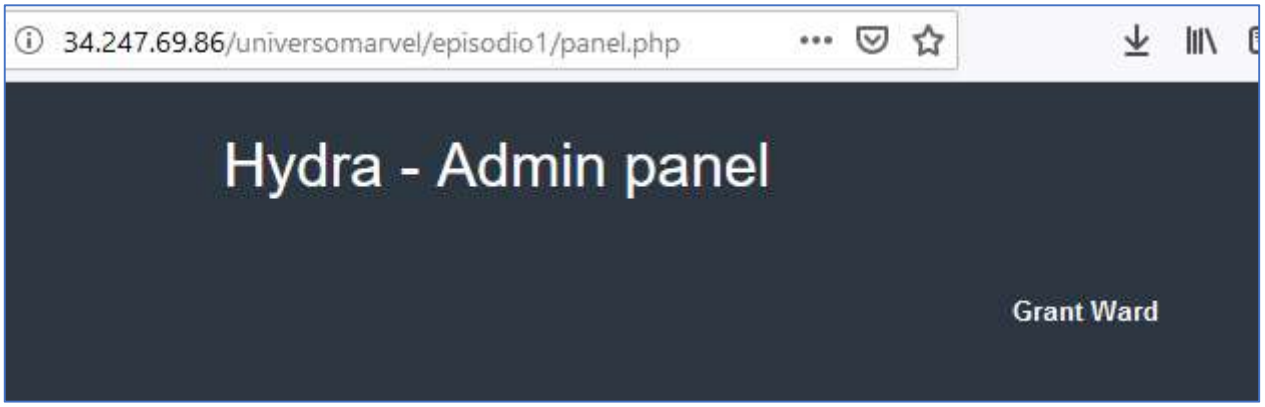
Vemos un enlace a “Ubicación de la base”, que parece que es el objetivo del reto. Pero si pinchamos en él, nos da error de permisos:

El resto de enlaces (Calendario, Misiones, etc.) si que funcionan, por tanto, es como si este usuario no tuviera permisos para ejecutar ese link, y por tanto parece que tenemos que escalar privilegios buscando mayor nivel de permisos.

Por tanto, vamos a volver a logarnos, a ver si encontramos algún usuario que tenga mayores permisos. Volvemos a la pantalla de login, pero vemos que no nos vuelve a mostrar la página de usuario y password, sino que sigue en el portal. Le damos a la opción de "logout" del usuario, pero igual, no nos muestra el usuario y password:

Al final decido borrar las cookies, y entonces si que me vuelve a pedir usuario y password. Es un comportamiento sospechoso, y que puede estar relacionado con las cookies de sesión.

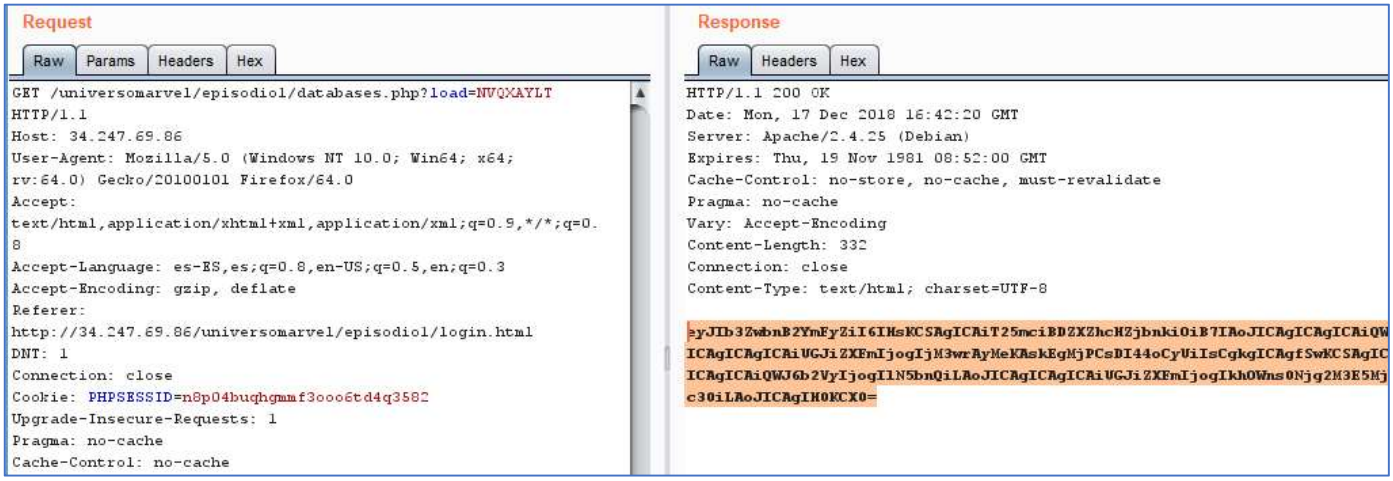
Ahora pruebo otros usuarios, la idea que tengo es hacer otro ataque por fuerza bruta buscando algún usuario admin, pero vemos que probando con cualquier usuario (en ese caso "pepe", password "pepe"), el acceso es correcto, y carga los mismos datos que con el usuario que sacamos del fichero pcap:



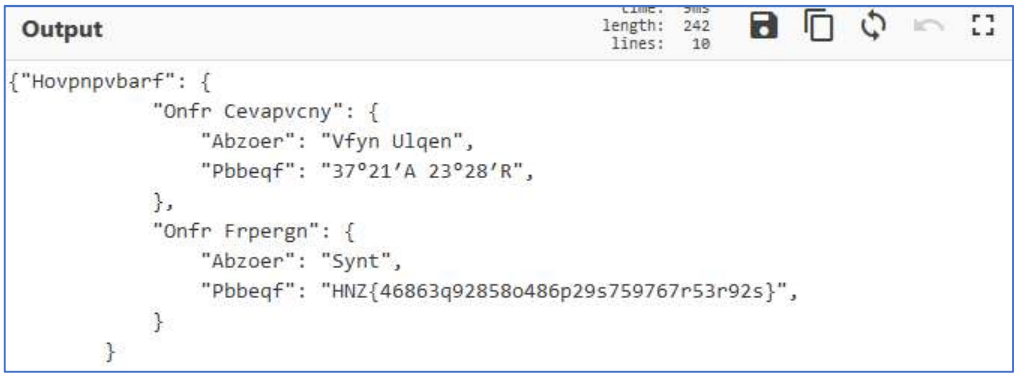
Por tanto, descarto la fuerza bruta. Es muy raro, todos los usuarios que pruebo me cargan los mismos datos. Por tanto, la autenticación podría no estar haciéndose a través del usuario y el password, y en este caso podría hacerse a través de la cookie de sesión PHPSESSID:

En este caso, donde vimos que la opción de “logout” no hacía nada realmente, pruebo a hacer la llamada a la página inicial cambiando el valor de la cookie, y si que me vuelve a pedir el usuario y password, así que la cookie es fundamental en este tema.

Se me ocurre probar el mismo valor de la cookie que venía en el fichero pcap, por si estaba hardcodeado a dicho valor. Abrimos BURP para hacer la petición, sin embargo, pruebo primero la llamada a la página del “Mapa de la base” simplemente cambiando el valor del PHPSESSID y me sale este contenido:



Esa respuesta parece base64, si la decodifico con Cyberchef:



Esto ya tiene un formato sospechoso, así que solo hace falta agregarle un ROT13 posterior, y…

Por tanto, la flag es: UAM{46863d92858b486c29f759767e53e92f}

José Ignacio de Miguel González

User UAM: nachinho3

Telegram @jignaciodemiguel